

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN  
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



**Proyecto Fin de Carrera**

## **Desarrollo de un Prototipo para Interconectar una Red de Sensores vía el Sistema GSM**



AUTOR: Javier Belando Gracia  
DIRECTOR(ES): Leandro Juan Llácer  
Francisco Javier Garrigós Guerrero  
Septiembre / 2008



<b>Autor</b>	Javier Belando Gracia
<b>E-mail del Autor</b>	<a href="mailto:Javibelando@gmail.com">Javibelando@gmail.com</a>
<b>Director(es)</b>	Leandro Juan Llácer, Francisco Javier Garrigós Guerrero
<b>E-mail del Director(es)</b>	<a href="mailto:Leandro.Juan@upct.es">Leandro.Juan@upct.es</a> , <a href="mailto:Javier.Garrigos@upct.es">Javier.Garrigos@upct.es</a>
<b>Título del PFC</b>	Desarrollo de un prototipo para interconectar una red de sensores vía el sistema GSM
<b>Resumen</b>	
<p>El objetivo de este proyecto es interconectar remotamente una red de sensores a un destino conectado a Internet, de manera que se puedan almacenar los datos sensorizados, para su explotación y análisis, y actuar en consecuencia en el sistema captador de información ejerciendo ciertas acciones. Este proyecto se centrará en el diseño de los medios para el intercambio de información entre la red de sensores/actuadores, y el sistema remoto controlador, a través de las redes de telecomunicación, mediante el desarrollo de un prototipo real, quedando fuera de este proyecto la explotación de dichos datos, y aplicación final derivada del uso de esa información, de forma que el sistema pueda ser particularizado o utilizado según las necesidades de la aplicación que necesite un posible usuario.</p>	
<b>Titulación</b>	Ingeniería de Telecomunicación
<b>Intensificación</b>	Sistemas y Redes de Telecomunicación
<b>Departamento</b>	Tecnologías de la información y las comunicaciones
<b>Fecha de Presentación</b>	Septiembre- 2008

# Índice

---

<b>1. Introducción y Objetivos .....</b>	<b>5</b>
1.1 Introducción .....	5
1.2 Objetivos.....	5
1.3 Herramientas utilizadas .....	6
1.3.1. <i>Hardware</i> .....	6
1.3.2. <i>Software</i> .....	6
1.4 Contenido .....	7
<b>2. Redes de sensores .....</b>	<b>8</b>
Generalidades sobre redes de sensores.....	8
2.1 Evolución Histórica .....	9
2.2 Sensores .....	11
2.2.1. <i>Unidad sensora</i> .....	11
2.2.2. <i>Unidad de procesamiento</i> .....	12
2.2.2.1. <i>Procesador</i> .....	12
2.2.2.2. <i>Unidad de almacenamiento</i> .....	13
2.2.3. <i>Módulo radio</i> .....	13
2.2.4. <i>Suministro de energía</i> .....	14
2.3 Clasificación de Redes de Sensores .....	15
2.4 Arquitectura en redes de sensores inalámbricos .....	16
2.5. Redes WSN a nivel funcional.....	19
2.7. Protocolos en Redes de Sensores Inalámbricas .....	20
2.7.1. <i>Protocolos del Nivel de Aplicación</i> .....	21
2.7.2. <i>Protocolos del Nivel de Transporte</i> .....	21
2.7.3. <i>Protocolos del Nivel de Red</i> .....	22
2.7.4. <i>Protocolos del nivel de Enlace de Datos</i> .....	23
2.8. Mote Kit 410CB Multichannel .....	24
2.8.1. <i>Gateway MIB510</i> .....	24

2.8.2. <i>Módulo inalámbrico MPR410</i> .....	26
2.8.3. <i>Placa sensora</i> .....	28
2.9 Mensajes enviados por los sensores .....	29
<b>3. El sistema GSM/GPRS .....</b>	<b>33</b>
<b>Historia .....</b>	<b>33</b>
<b>3.1. ¿Qué es GSM?.....</b>	<b>34</b>
<b>Introducción técnica .....</b>	<b>35</b>
<b>3.2. Arquitectura de red GSM.....</b>	<b>36</b>
<b>3.2.1. <i>Mobile Station</i> .....</b>	<b>37</b>
<b>3.2.2. Base Station Sub-System: .....</b>	<b>39</b>
<b>3.2.3. <i>Network Sub-System</i>: .....</b>	<b>40</b>
<b>3.2.4. <i>Network Management Center</i>: .....</b>	<b>44</b>
<b>Introducción al servicio de transmisión de datos: .....</b>	<b>45</b>
<b>3.3 ¿Qué es GPRS? .....</b>	<b>46</b>
<b>3.4. Arquitectura GPRS:.....</b>	<b>48</b>
<b>3.5. Tipología del servicio GPRS. ....</b>	<b>50</b>
<b>3.5.1. <i>Tipología Punto a Punto (PTP)</i>: .....</b>	<b>50</b>
<b>3.5.2. <i>Tipología Punto Multipunto (PTM)</i>: .....</b>	<b>57</b>
<b>3.6. Protocolo GPRS:.....</b>	<b>63</b>
<b>3.7. Calidad del servicio GPRS: .....</b>	<b>65</b>
<b>3.8. Módem utilizado (Nokia N12) .....</b>	<b>66</b>
<b>4. Breve descripción del sistema GPS.....</b>	<b>69</b>
<b>4.1 Historia .....</b>	<b>69</b>
<b>4.2. Características técnicas y prestaciones.....</b>	<b>70</b>
<b>4.2.1. <i>Segmento espacial</i>.....</b>	<b>71</b>

4.3. Evolución del sistema GPS .....	72
4.4. Funcionamiento .....	73
4.5. Fiabilidad de los datos.....	74
4.5.1. Fuentes de error.....	75
4.6. Aplicaciones.....	76
<b>5. La tecnología Java J2ME .....</b>	<b>78</b>
5.1 La arquitectura J2ME .....	79
5.2 Especificaciones .....	80
5.2.1 Configuraciones.....	80
5.2.2 Perfiles .....	81
<i>Paquetes opcionales</i> .....	<b>82</b>
5.3 El midlet. ....	82
5.4 Consideraciones a tener en cuenta en el desarrollo de una aplicación J2ME. .	83
5.5 Elección de las tecnologías/productos .....	84
<b>6. Prototipo desarrollado .....</b>	<b>87</b>
6.1 Configuración de la red de sensores y el Gateway .....	90
6.2 Placa FPGA. Funcionalidad. Reuso del anterior prototipo.....	91
6.3 Dispositivo Teltónica.....	92
6.4. Programa desarrollado en tecnología J2ME.....	101
<b>7. Conclusiones y futuras líneas de trabajo .....</b>	<b>119</b>
<b>Bibliografía.....</b>	<b>121</b>

## **1. Introducción y Objetivos**

### **1.1 Introducción**

- La interconexión de sistemas y redes para el acceso a la información está en auge hoy en día en la sociedad de la información. En particular, el acceso a información en tiempo real es uno de los campos a los que se dedica gran parte de la investigación e inversión, y también uno de los cuales genera multitud de aplicaciones y oportunidades de negocio. Además, la posibilidad de conocer la localización geográfica exacta donde se está produciendo la información, a través del sistema de posicionamiento global por satélite (GPS), abre un abanico de aplicaciones amplísimo, en el que el límite lo pone la imaginación. El ahorro de costes que supone poder administrar sistemas de forma remota, así como poder controlar distintos lugares simultáneamente, así como el imparable desarrollo de Internet, presente hoy en día en todas partes, han fomentado que cambie la orientación de la forma de trabajar de muchas empresas, de manera que las distancias ya no sean importantes, y disponer de información en el momento en que ésta se produce, así como poder incorporarla a procesos productivos, y usarla para agilizar las actividades, ha revolucionado la productividad.

### **1.2 Objetivos**

El objetivo de este proyecto es interconectar remotamente una red de sensores a un destino conectado a Internet, de manera que se puedan almacenar los datos sensorizados, para su explotación y análisis, y actuar en consecuencia en el sistema captador de información ejerciendo ciertas acciones. Este proyecto se centrará en el diseño de los medios para el intercambio de información entre la red de sensores/actuadores, y el sistema remoto controlador, a través de las redes de telecomunicación, mediante el desarrollo de un prototipo real, quedando fuera de este proyecto la explotación de dichos datos, y aplicación final derivada del uso de esa información, de forma que el sistema pueda ser particularizado o utilizado según las necesidades de la aplicación que necesite un posible usuario.

Se recogerá la información producida por una red de sensores medidora de parámetros ambientales, como pueden ser temperatura, luz ambiental, intensidad de sonido, magnetización, movimiento, entre otros. Además, incorporaremos un receptor GPS que proporcionará las coordenadas geográficas del lugar donde se está colocada la red de sensores. Una estación base, recogerá tanto los datos de los sensores, como las coordenadas geográficas, es decir, contendrá dos receptores: un nodo central (Gateway) de los sensores, y un receptor GPS. Esta estación base, transmitirá los datos de forma inalámbrica a través de un transmisor GSM/GPRS, que hará que la red de telefonía móvil haga de puente para llegar a un destino final que puede estar en cualquier parte del mundo, conectado a la red global Internet.

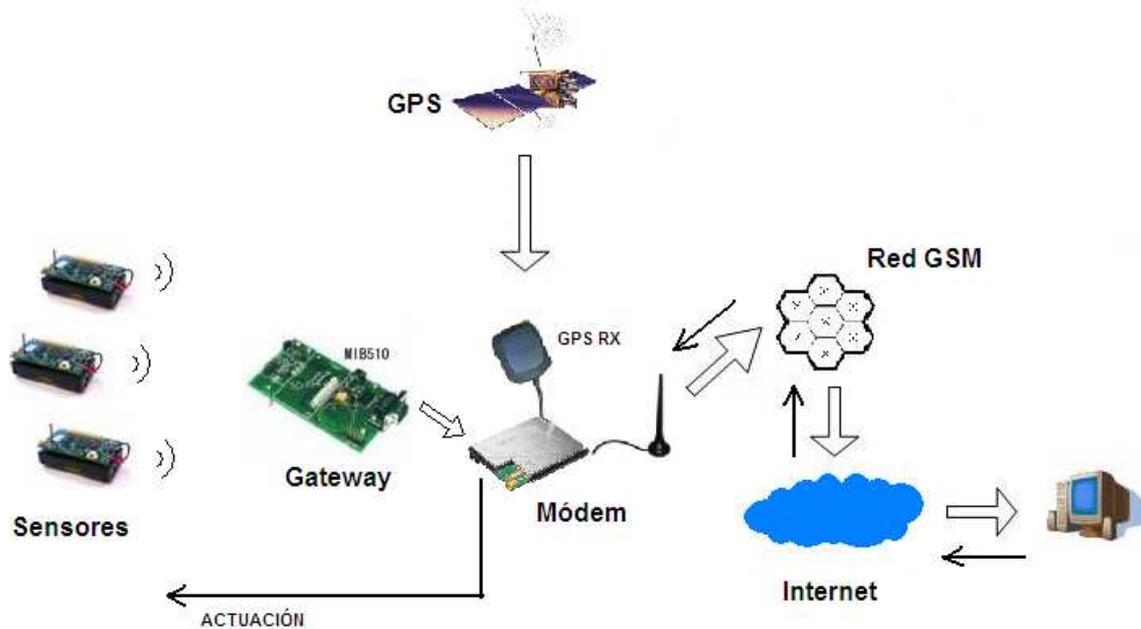


Figura 1. Esquema de funcionamiento

### 1.3 Herramientas utilizadas

A continuación se muestra un resumen de las herramientas usadas para el desarrollo del prototipo, tanto hardware como software.

#### 1.3.1. Hardware

- Red de sensores MOTE-KIT 400 de la empresa Crossbow.
- FPGA Spartan 3E de 400.000 puertas lógicas
- Dispositivo de comunicaciones FM3101 de la marca Teltónica. Este módulo se compone de:
  - Módulo GSM Nokia N12i
  - Modulo receptor GPS
  - Interfaz de comunicación RS-232
- Fuente de alimentación

#### 1.3.2. Software

- Herramienta Moteworks y compilador Cygwin, proporcionados por Crossbow.

Este software incluye:

- Aplicación Surge-View para poder visualizar los datos y la topología de la red.
- Código de diversas aplicaciones para los motes.
- Sistema Operativo Tynios para programar los sensores.

- Consola Cygwin para ejecutar aplicaciones.

- Herramienta de diseño Xilinx ISE necesaria para programación de la FPGA.
- Entorno de programación Eclipse
- Nokia Configurator Software

#### 1.4 Contenido

Esta memoria se compone de 8 capítulos.

El **capítulo segundo** hace un completo análisis sobre las redes de sensores inalámbricos, comenzando con una evolución histórica, componentes de un sensor, clasificación de redes, protocolos en redes de sensores, y red utilizada en este proyecto.

El **capítulo tercero** hace una amplia descripción del sistema GSM y GPRS. Arquitectura de red, elementos que la componen, protocolos y funcionamiento. También se hace una descripción del módem GSM/GPRS usado en este proyecto.

En el **capítulo cuarto** se realiza una breve descripción del sistema de posicionamiento global GPS.

En el **capítulo quinto** se profundiza en la tecnología J2ME. Posibilidades que ofrece este lenguaje de programación, ventajas, limitaciones, modo de uso.

En el **capítulo sexto**, se expone explica el prototipo desarrollado en este proyecto. Se explican las partes de que está compuesto, así como la función que realiza cada una de ellas.

El **capítulo séptimo** hace un resumen de las conclusiones extraíbles de este proyecto, así como plantea posibles futuras líneas de trabajo que se derivan del mismo.

En el **capítulo octavo** se recogen las principales referencias bibliográficas consultadas para el desarrollo de este proyecto.

## 2. Redes de sensores

### Generalidades sobre redes de sensores

Un sistema WSN (Wireless Sensor Network) de sensores inalámbricos es una red con numerosos dispositivos distribuidos espacialmente, que utilizan sensores para controlar diversas condiciones en distintos puntos, entre ellas la temperatura, el sonido, la vibración, la presión y movimiento o los contaminantes. Los dispositivos son unidades autónomas que constan de un microcontrolador, una fuente de energía (casi siempre una batería), un radiotransceptor y un elemento sensor.[1]

Los nodos autoorganizan sus redes en una forma ad hoc, en lugar de tener una topología de red previamente programada. Además, WSN tiene capacidad de autorrestauración, es decir, si se avería un nodo, la red encontrará nuevas vías para encaminar los paquetes de datos. De esta forma, la red sobrevivirá en su conjunto, aunque haya nodos individuales que pierdan potencia o se destruyan.

La clave para el funcionamiento de estos sensores autónomos radica en que deben gastar la menor cantidad de energía posible para asegurar que sus baterías perduren lo máximo posible.

Las restricciones más importantes que imponen las nuevas redes de sensores son:

- El suministro de energía es limitado, por lo que se requieren protocolos eficientes en el consumo de energía.
- Las capacidades de cómputo son limitadas (debido al tamaño de los sensores), por lo que no se pueden ejecutar protocolos sofisticados de red.
- El ancho de banda en enlaces inalámbricos que interconectan los sensores es limitado.

Las redes de sensores pueden utilizar distintas tecnologías inalámbricas, incluyendo IEEE 802.11, LANS sin cable, Bluetooth e identificación de la frecuencia de radio.

Las últimas investigaciones apuntan hacia una eventual proliferación de redes de sensores inteligentes, redes que recogerán enormes cantidades de información hasta ahora no registrada que contribuirá de forma favorable al buen funcionamiento de fábricas, al cuidado de cultivos, a tareas domésticas, a la organización del trabajo y a la predicción de desastres naturales como los terremotos. Este tipo de redes, cada vez más baratas, se están convirtiendo en el objetivo de más de una compañía informática. Empresas de microprocesadores como Intel están ampliando sus productos investigando en esta nueva área, mientras otras empresas como Crossbow, Moteiv o Dust Networks centran sus productos en estas nuevas redes inteligentes.

En los últimos años, varios laboratorios de investigación, y especialmente multinacionales como Intel, han apostado fuertemente por esta tecnología. En diversos informes se augura que este tipo de redes conllevarán una revolución tecnológica similar a la que tuvo la aparición de Internet. De hecho, DARPA (Defense Advanced Research Projects Agency), institución dependiente del Departamento de Defensa estadounidense, también se ha involucrado en el desarrollo de este tipo de redes. Ya se habla de redes de vigilancia global del planeta, capaces de registrar los hábitos de la

gente, realizar un seguimiento de personas y mercancías concretas, monitorizar el tráfico, etc. Aunque para ello habrá que esperar todavía unos años, sí que han surgido múltiples iniciativas y proyectos de investigación de enorme interés y aplicabilidad práctica.

## 2.1 Evolución Histórica

Las redes inalámbricas de microsensores se han identificado como una de las tecnologías más importantes del siglo XXI.

El desarrollo de las redes de sensores requiere tecnologías que provienen de tres áreas diferentes:

- Sensorización.
- Comunicación.
- Cómputo (incluyendo hardware, software, y algoritmos).

Como muchas tecnologías, aplicaciones en defensa han conducido a investigaciones y desarrollos en las redes de sensores. Durante la Guerra Fría, el SOSUS (Sound Surveillance System), un sistema acústico de sensores (hidrophones) en el fondo del océano, fue desplegado en localizaciones estratégicas para detectar y realizar un seguimiento secreto de los submarinos soviéticos. Años más tarde, otras redes acústicas más sofisticadas se han desarrollado para vigilancia submarina. SOSUS es utilizado en la actualidad por la Administración Atmosférica y Oceanográfica Nacional (NOAA) para monitorizar eventos en el océano.

También durante la Guerra Fría, redes de radares defensa aérea fueron desarrollados y desplegados para defender a Estados Unidos y Canadá.

Estas redes de sensores normalmente adoptan una estructura de proceso jerárquico donde el proceso sucede en niveles consecutivos hasta llegar a la información sobre los eventos de interés que requiere el usuario.

Las modernas investigaciones las redes de sensores empezaron alrededor de 1980 con la *Red de Sensores distribuida* (DSN) programada en DARPA (Defense Advanced Research Projects Agency). Los componentes de una DSN fueron identificados en un taller de redes de sensores distribuidas en 1978.

Éstas incluían sensores (acústicos), comunicación (protocolos de alto nivel que enlazan procesos trabajando en aplicaciones comunes en una red de compartición de recursos), técnicas de procesado y algoritmos (incluyendo algoritmos de autolocalización para sensores), y software distribuido (sistemas distribuidos dinámicamente modificables y diseño de lenguaje) [3].

Las redes de trabajo de sensores esponsorizados por DARPA (Defense Advanced Research Projects Agency) en la universidad de Carnegie-Mellon University en Pittsburgh, Pennsylvania fueron de las primeras en aparecer. El interés debido a

sistemas militares de vigilancia lleva a trabajar en la comunicación y en el cálculo de compensaciones de las redes de sensores, incluyendo su uso en un entorno medioambiental. Las investigaciones se enfocaron en proporcionar un sistema de red operativo que permitiera acceso transparente y flexible para distribuir recursos necesarios para una WSN tolerante a fallos. Ellos desarrollaron un sistema de comunicación orientada operativa denominada *Accent*. *Accent* evolucionó en el *sistema operativo Mach* que encontró aceptación comercial. Este interés se elevó con los microsensores integrados inalámbricos de baja potencia de DARPA (LWIM) proyecto de mediados de 1990 y que continuó con el lanzamiento del proyecto SensIT en 1998, que se centraba en redes inalámbricas para el caso de sistemas de sensores militares de gran distribución.

Las redes SensIT tienen nuevas capacidades. Las redes son interactivas y programables con tareas dinámicas. Una característica multitarea en el sistema permite múltiples y simultáneos usuarios.

La Universidad de California en Los Ángeles, que suele trabajar en colaboración con el Centro de ciencia RockWell, ha tenido un proyecto de Red de sensores inalámbrico integrados (*WINS*) desde 1993. Se ha comercializado con la financiación de Sensoria Corporation en 1998. Este programa cubre casi todos los aspectos del diseño de las redes de sensores inalámbricos, desde el sistema microelectromecánico (MEMS) de sensores y transmisores de integración a nivel de circuito, arquitecturas de procesamiento de señal, y diseño de protocolos de red, para estudiar los principios fundamentales de la teoría de detección y sensorización. El grupo prevé que *WINS* proveerá una red distribuida y acceso a Internet para sensores, controles y procesadores profundamente incrustados en equipamiento, servicios, y el entorno.

Jan M. Rabaey de la Universidad de California en Berkeley empezaron el *programa Picoradio* en 1999. El protocolo MAC propuesto combina lo mejor de las técnicas de espectro ensanchado y la técnica CSMA. Un nodo seleccionaría aleatoriamente un canal y chequear para ver si hay actividad. Si el canal estuviera activo, el nodo seleccionaría otro canal de los que quedan disponibles, hasta que un canal sin actividad se encuentre y el mensaje sea enviado. Si no se encuentra ningún canal sin actividad, el nodo se quedará esperando un tiempo aleatorio para cada canal. Entonces se utilizaría el canal que primero terminara y se limpiarían el resto de tiempos que ha estado esperando para el resto de canales.

Hay que destacar que el programa Picoradio de Berkeley es distinto del programa más conocido quizás que es el *Smart Dust*, en el cual los MOTES (pequeños y autónomos ordenadores con sensores para monitorizar el entorno y radios para comunicarse con otros iguales) basados en MEMS pueden ser lo suficientemente pequeños para permanecer suspendidos en el aire, flotando debido a corrientes de aire, sensorizando y comunicando durante horas o días enteros. Este programa es el hasta ahora más exitoso y el que da nombre coloquial a las nuevas redes de sensores inalámbricas.

## 2.2 Sensores

Los componentes de las redes de sensores inalámbricas no son exactamente sensores, sino que son una especie de microcomputadores con diversas funciones denominadas *motes*.

Un mote es un ordenador autónomo pequeño, con sensores para monitorizar el entorno y con radios para comunicarse con otros motes.

Los componentes básicos de un mote son [5]:

- Unidad sensora, con una serie de sensores que se encargan de adquirir los datos. Estos sensores pueden ser de temperatura, humedad, luz, etc....
- Unidad de procesamiento (CPU), con el que se realiza el procesamiento de los datos.
- Equipamiento Radio, para la comunicación con otros motes.
- Suministro de energía.

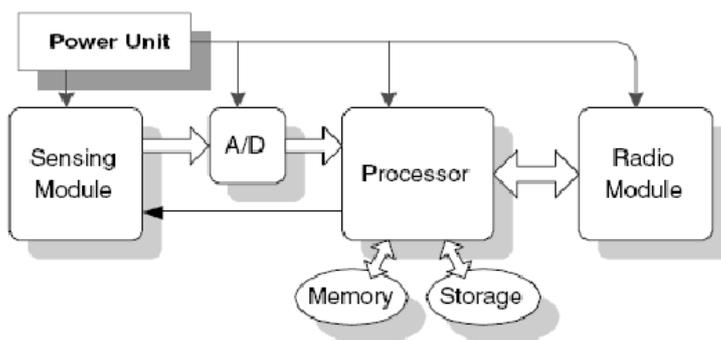


Figura 2.1. Arquitectura de un sensor [5].

### 2.2.1. Unidad sensora

El propósito de los nodos de una red de sensores no es computar o comunicar, sino detectar. El componente sensor de los motes es la actual tecnología de cuello de botella, estas tecnologías actualmente no están progresando con la rapidez de los semiconductores. Las limitaciones conceptuales son significativamente más estrictas para los sensores que para los procesadores o el almacenamiento. Por ejemplo, la interfaz de los sensores con el mundo físico, mientras que las unidades de cómputo y comunicación se relacionan con un gran entorno controlado de un solo chip. Los transductores son unos componentes en un mote que se usan para transformar una forma de energía en otra. El diseño de los transductores se considera fuera del alcance de la arquitectura del sistema.

Además, los sensores pueden tener otros cuatro componentes: convertidos analógico-digital, un convertidor digital-analógico y un microcontrolador. El diseño más simple puede incluir un solo transductor.

Uno de los retos principales de las redes de sensores es seleccionar el tipo y la cantidad de sensores, así como determinar su emplazamiento. Esta tarea es difícil ya que hay numerosos tipos de sensores con diferentes propiedades como la resolución, el coste, la precisión, el tamaño y el consumo de potencia.

Además, se suele necesitar más de un tipo de sensor para asegurar la exactitud de las operaciones y de los datos que provienen de diferentes sensores que se pueden combinar.

Los sensores utilizados en las distintas plataformas pueden ser de diferentes clases:

- Modalidad de baja potencia:
- Fotodetector (sensor de luz)
- Sensor de temperatura
- Acelerómetro de dos dimensiones
- Modalidad de media potencia:
- Micrófono (detector acústico de umbral)
- Magnetómetro
- Modalidad de alta potencia:
- Sensor de imagen
- Sensor de video
- Sensor de forma de rayo



Figura 2.2. Placa sensora de Crossbow [8].

### ***2.2.2. Unidad de procesamiento***

Esta unidad se encuentra dividida a su vez en dos subunidades: el procesador y la unidad de almacenamiento.

#### ***2.2.2.1. Procesador***

Los microcontroladores son la primera opción para el procesamiento dentro del nodo. Los requerimientos de memoria dependen de la aplicación que se vaya a implementar.

El modelo más incluido en los sensores es el Atmel ATmega128L que se muestra en la figura 2.3.



Figura 2.3. Atmel ATmega128L [9].

Entre las características más destacadas de este modelo posee una arquitectura RIS de 133 instrucciones y dos multiplicadores en el propio chip. En cuanto a memoria posee una memoria EEPROM, una memoria Flash reprogramable y una memoria interna SRAM. En cuanto a energía se refiere, posee seis modos de funcionamiento diferentes para ahorrar energía.

### **2.2.2.2. Unidad de almacenamiento**

Dependiendo de la estructura de toda una red de sensores, los requerimientos de almacenamiento en términos de rapidez y no-volatilidad de memoria en cada nodo pueden ser diferentes. Por ejemplo, si uno sigue el modelo arquitectónico en el cual la información se envía instantáneamente al nodo central, se necesita muy poco almacenaje local en los nodos individuales. Sin embargo, en un escenario donde la meta es minimizar la cantidad de comunicación y conduce a una parte importante en cada nodo individual, será necesario un requerimiento significativo de almacenaje local. Existen al menos dos alternativas de almacenamiento de datos en un nodo local:

- La primera opción es usar una memoria flash, que es una solución muy atractiva en términos de coste y capacidad de almacenamiento. Sin embargo, tiene relativamente serias limitaciones en cuestión de las veces que puede ser utilizada para el almacenamiento de diferentes tipos de datos en las mismas posiciones físicas.
- La segunda opción es utilizar memorias MRAM basadas en elementos nanoeléctricos. Se espera que las MRAM se puedan utilizar pronto para apoyar un número elevado de aplicaciones en número de áreas.

### **2.2.3. Módulo radio**

Las radios de corto alcance como los componentes de comunicación son excepcionalmente importantes porque la parte del presupuesto de energía dedicado a enviar y recibir mensajes normalmente domina el conjunto del presupuesto de energía. Durante el diseño y la selección de las radios, se debe considerar al menos tres capas diferentes de abstracción: capa física, capa MAC y la capa de red.

La capa física es la responsable de establecer los enlaces físicos entre el transmisor y uno o más receptores. Las principales tareas a este nivel incluyen la modulación de señal y el encriptamiento de los datos para mantener la comunicación en presencia de un canal de ruido y señales interferentes. Para usar de manera eficiente el ancho de banda y

reducir el coste de desarrollo, se utilizan varias radios que comparten el mismo medio de interconexión. La participación media es facilitada por la capa MAC.

Finalmente, la capa de red es la responsable de establecer el camino que un mensaje debe realizar a través de la red para que se transfiera de la fuente a su destino.

El diseño de potencia y la eficiencia de ancho de banda de las radios es una de las principales investigaciones y tareas de desarrollo. Es importante darse cuenta de que la arquitectura radio es una función en la que se emplea arquitecturas de red y protocolos.

Hay que llegar a un compromiso entre el coste relativo de energía en transmisión y recepción ya que escuchar el canal es muy caro en términos energéticos. Por lo tanto, es necesario desarrollar esquemas que combinen periodos de actividad con periodos de standby o inactividad en los sensores.

Las radios se encuentran en chips comercialmente. Las bandas de frecuencias en las que suelen operar son: 433MHz, 916MHz, 2.4MHz en las bandas ISM.

La potencia típica que transmite es 0dBm y la sensibilidad en recepción es tan baja como -110dBm. La comunicación se realiza en banda estrecha con la modulación FSK o espectro ensanchado. Generalmente tiene unas tasas de transmisión relativamente bajas (menores a 100kbps), lo que permite ahorrar potencia [6].

#### **2.2.4. Suministro de energía**

Un gran consenso es que la energía será una de las restricciones principales tecnológicas para los nodos de las redes de sensores.

La energía puede ser suministrada de dos formas principalmente:

- Equipando cada nodo sensor con una (recargable) fuente de energía. En la gran mayoría de las plataformas existentes, se suministra la potencia mediante las baterías AA. Estas baterías son las que dan el mayor o menor tamaño del nodo. Pero aunque las baterías alcalinas ofrecen una alta densidad de energía a un precio bajo, la curva de descarga está lejos de ser plana. Otra opción son las baterías de botón que son más compactas y poseen de una curva plana de descarga.



Figura 2.4. Sensor alimentado con una pila de botón [10].

- La segunda alternativa es recoger la energía disponible en el entorno como pueden ser las células solares. Actualmente ya son ampliamente utilizadas para aplicaciones móviles como las calculadoras y se pueden utilizar para algunas aplicaciones.

También existen otras formas alternativas todavía en investigación como:

- Las células de combustible.

- Suministro de potencia mediante un sistema inalámbrico de baja batería que recoge el calor del ambiente. El principal componente de este sistema es un convertidor DC-DC con condensador conmutado, un módulo microtermoeléctrico hace el sistema posible. [5]

### 2.3 Clasificación de Redes de Sensores

Como veremos más adelante la configuración y arquitectura de las WSN varían en función de la tarea a desempeñar, pues están muy orientadas a la aplicación. Sin embargo, todas las WSN que se crean tienen propiedades comunes desde un punto de vista general. Atendiendo a estas propiedades comunes haremos la siguiente clasificación [5]:

- En función de la distancia de los nodos a la estación base:
- *El nodo se conecta directamente con la estación base (red single-hop).*
- *El nodo se conecta a otro nodo y éste último a la estación base (red multihop),* por lo cual el primer nodo envía la información a la estación base a través de este segundo nodo.

Es obvio que las redes multihop son mucho más versátiles y funcionales que las single-hop, sin embargo se debe pagar el precio de la complejidad de su implementación.

#### - Según la densidad de la red y la dependencia de los datos:

- *Redes agrupadas (aggregating),* se da en redes densas, donde todos los datos de cada nodo individual son enviadas al destino tal cual. Cuando la carga computacional de los nodos intermedios es relativamente pequeña y el sistema alcanza gran precisión. Sin embargo este tipo de redes tienen la contraprestación de que el tráfico en la red completa puede incrementar rápidamente con el tamaño de la red, consumiendo mayor energía y produciéndose un mayor número de colisiones.

En este tipo de red los usuarios finales sólo están interesados en la información global del sistema por lo que los nodos intermedios analizan la información sin tratar del resto de nodos y eliminan información redundante. Esto implica que los nodos intermedios han de poseer amplias memorias para almacenar información.

- *Redes no agrupadas (nonaggregating)*, es un esquema aplicado a redes poco densas, donde el usuario final demanda una extrema precisión. Los datos se tratan antes de enviarlos.

- **En función de la distribución de los sensores:**

- *Determinista*, donde la posición de los sensores es fija, conocida y se ha planificado con antelación. La ventaja de esta configuración es su sencillez y facilidad. Sin embargo este tipo de redes se implementa en pocas ocasiones debido a que la posición de los sensores rara vez se conoce con antelación.

- *Dinámica*, donde la posición de los sensores se desconoce a priori. Es la forma más usual dado que la posición de los sensores rara vez es dada previamente. La configuración dinámica es más flexible pero requiere algoritmos de control más complejos.

- **En función del esquema de control:**

- *No autoorganizadas*, los sensores no son capaces de organizarse por sí mismos pero depende de un controlador central que realiza la función de recopilar información por ellos así como ordenarles. Este tipo de redes sólo pueden ser utilizadas en pequeñas redes.

- *Autoorganizadas*, son capaces por sí mismas de establecer y mantener conectividad y cumplir conjuntamente con las tareas de control y sensorización. Este esquema es más adecuado para sistemas a gran escala para implementar tareas complicadas de monitorización y recolección de información. Estos sistemas son los que más se adaptan a las necesidades actuales.

## **2.4 Arquitectura en redes de sensores inalámbricos**

La arquitectura de una red de sensores genérica viene representada en la siguiente figura. Las redes de sensores se diseñan para transmitir los datos desde un array de sensores a un nodo central o colector. No usan necesariamente un único camino para enviar los datos a través de la red. Los elementos del sistema tomarán decisiones sobre qué datos pasar para minimizar la energía consumida y maximizar el contenido de la información [7].

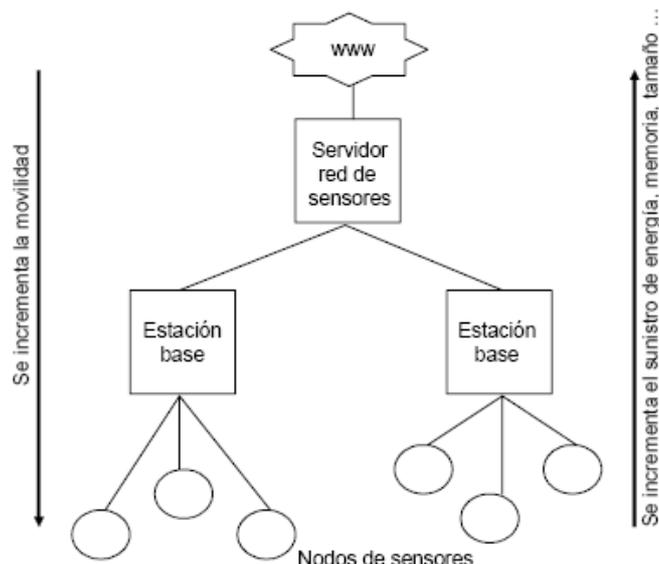


Figura 2.5. Esquema de una arquitectura genérica para una WSN [7].

Antes de profundizar en los tipos de arquitecturas, se ha de tener en cuenta que las redes de sensores inalámbricas pueden ser *homogéneas* o *heterogéneas*. Las WSN *homogéneas* son aquellas en que todos los sensores poseen la misma funcionalidad, es decir, realizan la misma función. Por el contrario, las redes *heterogéneas* son aquellas en que no todos los sensores realizan la misma función, es decir, hay grupos de nodos (o nodos aislados) que realizan funciones diferentes: unos transmiten los datos, otros se dedican solamente a tomar los datos de las medidas a sensorizar, etc. Como puede suponerse, una red homogénea es más simple y fácil de desplegar, mientras que una red heterogénea es más compleja y su despliegue es más complicado debido a que se dispone de diferentes tipos de nodos que deben proveer diferentes funcionalidades. Sin embargo, éstas últimas son más versátiles y funcionales por lo que como veremos a continuación, son las predominantes [5]. La figura 2.5 sólo representaba un esquema genérico de una red de sensores, pero las arquitecturas que se implementan en la configuración de la red son diversas y dependen fuertemente con la aplicación que se desee desarrollar. Aun con todo esto podemos hacer una pequeña clasificación de las arquitecturas más comunes a implementar:

- *Direct Connected* (directamente conectado), donde todos los nodos de la red (con independencia de la función que desempeñan), se conectan directamente con el colector o centro de procesamiento central. Sin embargo, debido a las limitaciones que impone el suministro de energía, esta implementación no es muy viable en casos de una red de grandes dimensiones.

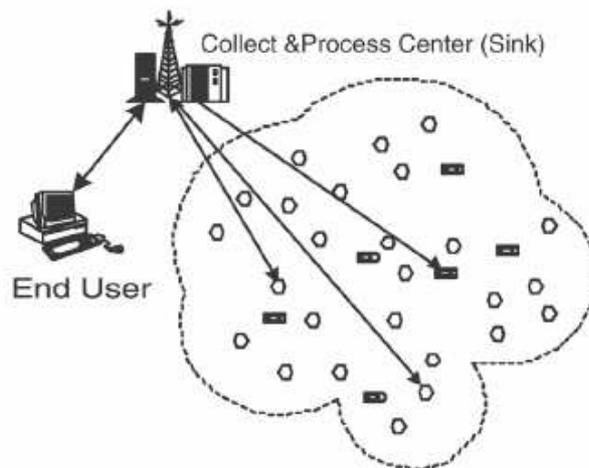


Figura 2.6. Arquitectura *Direct Connected* [5].

Una vez se ha comprobado cómo las estructuras single-hop no resultan eficientes en cuanto al consumo de energía para redes de un tamaño considerable, pasaremos a ver los tipos de arquitecturas multihop. Éstas últimas presentan la ventaja del ahorro de energía ya que la distancia entre la que se comunican es menor.

- *Flat ad hoc*, en donde algunos nodos de la red implementan funciones de routing e interconexión, permitiendo que el resto de nodos de la red envíen la información a través de ellos. Aunque este modo es flexible y utiliza eficientemente la energía, la escalabilidad de la red sigue siendo un problema.

Los nodos que estén próximos al colector serán principalmente utilizados para enviar los paquetes procedentes de otros nodos al centro de procesado. Si el tamaño de la red es grande, estos nodos enviarán una gran cantidad de información de un lado a otro, provocando el agotamiento de la energía de que disponen, provocando la caída de la red.

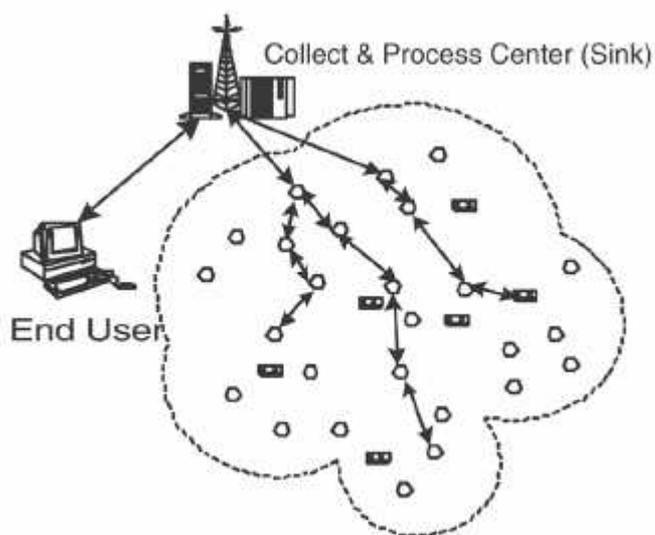


Figura 2.7. Arquitectura *Flat ad hoc multihop* [5].

- *Cluster-Based multihop* (multihop basado en clústeres), donde los sensores forman clústeres y en cada uno de estos clústeres se selecciona un nodo (sensor) central al cual envían la información el resto de nodos pertenecientes al clúster. Aunque en la figura siguiente se muestra una arquitectura en clústeres de un solo nivel, es usual que este tipo de arquitecturas sea jerárquica, donde varios clústeres forman uno más grande en el cual hay un nodo central al que le llega información del resto de nodos centrales pertenecientes a los clústeres de nivel inferior. La desventaja que presenta este sistema es, como en el caso anterior, que los nodos centrales de los clústeres agotan su energía antes que el resto.

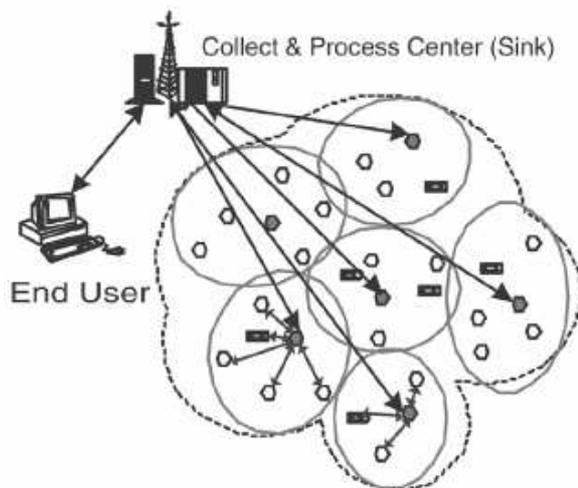


Figura 2.8. Arquitectura *Clustered-based multihop* [5].

Los casos que hasta ahora se han descrito han descrito, suponen que el colector o centro de procesamiento, está inmóvil. Sin embargo, también se pueden implementar arquitecturas en donde el colector se mueve por la red, según muestra la figura 2.9.

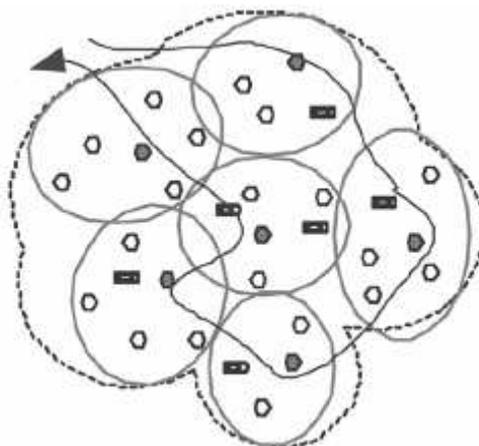


Figura 2.9. Arquitectura con colector móvil [5].

## 2.5. Redes WSN a nivel funcional

Como ya se dijo anteriormente, mientras que las redes tradicionales se diseñan para implementar varias aplicaciones, en las redes de sensores la arquitectura depende mucho

con la aplicación. La organización y la arquitectura de una WSN debe ser diseñada o adaptada para una tarea especial, tal como optimizar la función para la cual fue diseñado, maximizar el tiempo de vida, etc. La figura 3.10 muestra los diferentes niveles de funciones de una red de sensores distribuida [5]:

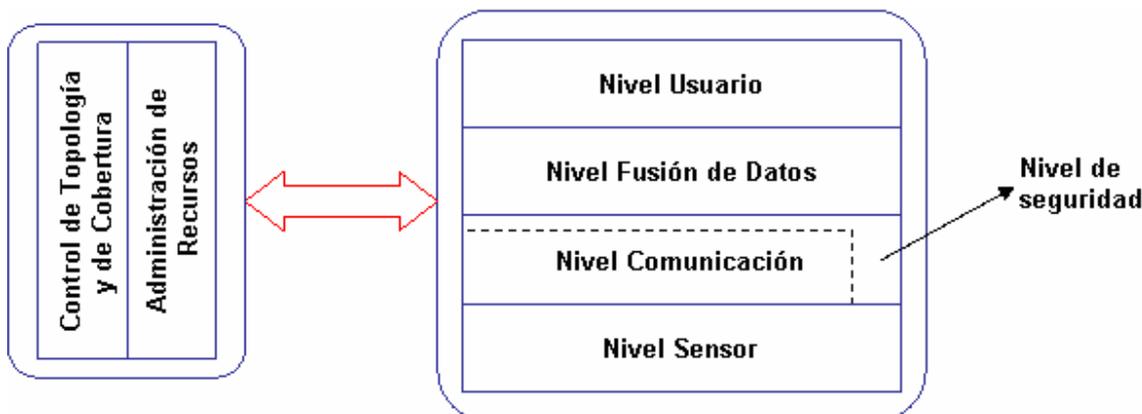


Figura 2.10. Niveles funcionales de una red de sensores inalámbrica.

- *Nivel Sensor*: se encarga de adquirir los datos o parámetros de las propiedades o fenómenos a sensorizar.

- *Nivel Comunicación*: realiza las funciones de correlación y compresión de datos, y routing. La función de este nivel es enviar al nodo central los resultados de los análisis estadísticos realizados. Debido a las limitaciones de energía que imponen las WSN, el protocolo de control de acceso al medio (MAC) debe ahorrar energía. El mecanismo de fusión de datos determina qué parte o qué tipo de información debe ser transmitida, mientras que el mecanismo de ruteo (routing) toma la decisión sobre cómo transmitir los datos y qué rutas se deben seguir. Estos mecanismos también deben tener en cuenta el ahorro de energía. El nivel de seguridad como se muestra en la figura, se incluye dentro del nivel de comunicación y se encarga de los problemas relacionados con la seguridad y la autenticación.

- *Nivel Fusión de Datos*: este nivel procesa los datos recibidos del nivel de comunicación y los combina usando procesado de datos, inteligencia artificial, fusión de datos, y otras técnicas de procesado de datos. Después de este tratamiento de señal, el nivel de fusión de datos, produce los resultados finales de una red de sensores.

- *Nivel Usuario*: este nivel sólo se encarga de proveer una interfaz final al usuario donde presentarle los resultados finales en el ordenador. También implementa funciones para la interacción usuario-WSN. También proporcionan funciones adicionales sobre el control de la topología, la cobertura, etc.

## 2.7. Protocolos en Redes de Sensores Inalámbricas

En una WSN, los protocolos implementados deben de llevar a cabo diferentes misiones tales como fusionar, extraer y agregar datos procedentes de los diferentes fenómenos

sensorizados; auto organizarse en clústeres para llevar a cabo las tareas para que ha sido diseñada la red; etc. Es decir, los protocolos han de proporcionar a la red una inteligencia artificial, que no es característica en redes ad hoc [5].

Por otro lado, aunque las redes de sensores inalámbricas se comunican de forma inalámbrica, es obvio que los protocolos de las redes inalámbricas comunes, debido a la propia naturaleza de la red. Es por esta razón que en este capítulo se intenta dar una idea de los protocolos que existen para redes de sensores inalámbricas (en función de los diferentes niveles de la arquitectura).

### ***2.7.1. Protocolos del Nivel de Aplicación***

Entre los protocolos del nivel de aplicación se encuentran: SMP, TADAP y SQDDP. SMP (Sensor Management Protocol), es un protocolo que gestiona el acceso a la red de sensores a través de otras redes y sobre todo a través de Internet. Además, también realiza tareas administrativas tales como administración temporal, apagado y encendido de los sensores, etc.

Por su parte, TADAP (Task Assignment and Data Advertisement Protocol) se encarga de gestionar eventos en los que el nodo avisa al usuario que tiene datos que entregar o también para que usuario pregunte al sensor si tiene datos disponibles para proporcionarle.

Por último, SQDDP (Sensor Query and Data Dissemination Protocol) es un protocolo que gestiona la interacción entre el usuario y los nodos. Por ejemplo, si el usuario quisiera conocer qué nodos detectan una temperatura mayor a 30 °C sería SQDDP el que se encargaría de enviar la pregunta al sensor y de recolectar las respuestas para presentárselas al usuario.

### ***2.7.2. Protocolos del Nivel de Transporte***

La siguiente capa de comunicación según OSI, es la de transporte. Esta capa ha de cumplir una serie de objetivos tales como: comunicar los niveles de red y de aplicación; proporcionar un servicio de entrega fiable entre la fuente y el colector (sink) con un mecanismo de control de errores; y regular la cantidad de tráfico que se inyecta a la red a través de mecanismos de control de congestión y de flujo. En una red de sensores, la comunicación puede establecerse en sentido desde los nodos hacia el colector que es la más usual, pero también desde el colector hacia los nodos. Los protocolos de nivel de transporte se diferencian según estos dos tipos de comunicación.

Centrándonos sólo en la *comunicación desde el nodo sensor hacia el colector*, existe el ESRT (Eventto- Sink Reliable Transport Protocol) que es un novedoso protocolo que ha sido desarrollado para lograr comunicaciones extremo a extremo eficientes pero empleando la mínima energía.

### 2.7.3. Protocolos del Nivel de Red

En cuanto al nivel de red, se han de tener en cuenta algunos criterios de diseño:

- Ahorro de energía, al igual que para todos los niveles y aspectos de la red.
- Una red de sensores ideal posee direccionamiento basado en atributos (los que cumplen una cierta condición en función de los atributos o fenómenos que sensorizan).
- Las redes de sensores están, la mayoría de ellas, basadas en los datos (data-centric) y no en los nodos de la red.
- El protocolo de nivel de red ha de ser fácilmente compatible con otro tipo de redes como por ejemplo, Internet.

Un protocolo basado en los datos, que a partir de ahora se denominará como data-centric, ha de implementar el concepto de agregación de datos (redes *aggregating* o agrupadas) que ya se introdujo al principio del apartado 4 y que se muestra la figura 2.11.

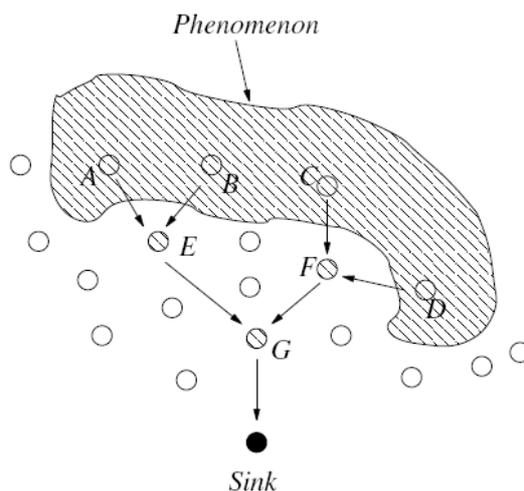


Figura 2.11. Concepto de agregación de datos [5].

Ampliando el concepto de agregación de datos, consiste en que los sensores que están en contacto con el área en que han de sensorizar, envían sus datos a nodos intermedios, que toman los datos de varios sensores y han de fusionarlos (agregarlos). Son estos nodos intermedios los que envían los datos fusionados procedentes de los sensores, hacia el colector (sink).

Otro de los principios de diseño del nivel de red es permitir la integración con otras redes tales como Internet o la red de satélites, tal y como se muestra en la siguiente figura. El colector (sink) o nodo central es el gateway entre la red de sensores y el resto de redes.

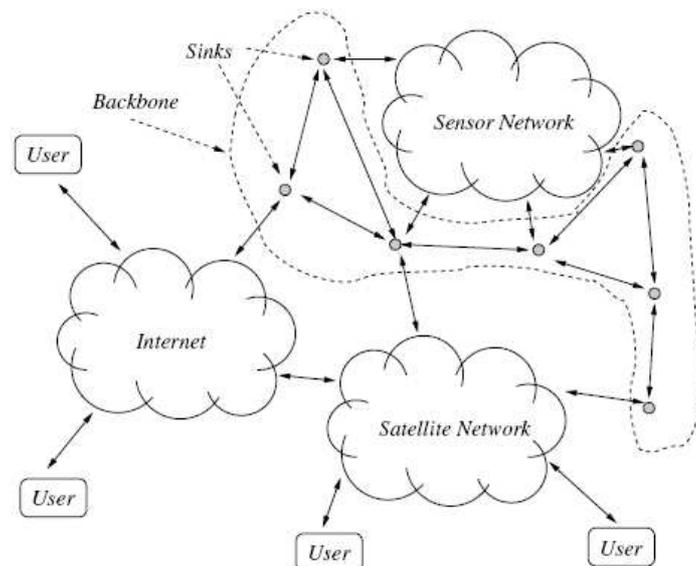


Figura 2.12. Posible interconexión de una red de sensores con otras redes [5].

Todas estas consideraciones a tener en cuenta hacen complejo el diseño del nivel de red. Protocolos como LEACH que forma clústeres para ahorrar la máxima energía posible o GOSSIPING que envía los datos a un vecino aleatorio están todavía en fase de implementación.

#### 2.7.4. Protocolos del nivel de Enlace de Datos

En general, el nivel de enlace de datos se encarga, entre otras cosas, de detectar las tramas recibidas, de multiplexar el flujo de bits, o el control de errores, pero es sin duda, el control de acceso al medio, su misión más importante. Los protocolos MAC que hasta ahora se han propuesto para implementar soluciones de de acceso al medio, particulares para solventar las necesidades de una WSN son:

- *TDMA*: es quizá el método que más energía conserve ya que los nodos pueden estar ‘dormidos’ mientras que no llegue su turno de transmisión. SMACS (Selforganizing Medium Access Control for Sensor Networks) es la variante de TDMA para WSN, basado en transmisión en slots de tiempo. El inconveniente de este esquema de acceso al medio es el coste de sincronización temporal ya que los nodos han de ‘despertarse’ y volverse a sincronizar antes de poder transmitir de nuevo.
- *TDMA/FDMA Híbrido*: combina la multiplexación en el tiempo (en slots) y en la frecuencia (diferentes portadoras o canales) mezclando las características que poseen ambas técnicas. El número óptimo de canales será aquel que proporcione el mínimo consumo de energía al sistema. Este número depende de la relación entre el consumo del transmisor y el consumo del receptor. Si el transmisor consume más potencia, se tenderá más hacia un esquema TDMA mientras que si es el receptor el que consume más potencia, se tiende hacia FDMA.
- *CSMA*: es un refinamiento de CSMA tradicional basado en detección de portadora y mecanismo de backoff. Recientemente se ha propuesto un modelo de CSMA para redes

de sensores y de las simulaciones realizadas se extrae que los periodos de ‘escucha’ y la introducción un retardo aleatorio previo a cualquier transmisión además del mecanismo de backoff hacen un uso eficiente de la energía.

## 2.8. Mote Kit 410CB Multichannel

La red de sensores empleada en el desarrollo de este proyecto es el kit de desarrollo *Mote Kit 410CB*. Este kit se muestra en la figura 2.13 y consta de:

- Un gateway o nodo central de la red, el MIB510.
- Tres módulos inalámbricos o módulos radio (motes) MPR410.
- Dos placas sensoras MTS300.
- Tres antenas para cada uno de los MPR410.



Figura 2.13. Red de sensores empleada en el desarrollo de este proyecto [8].

### 2.8.1. Gateway MIB510

Este dispositivo posee un puerto serie RS232 por el que envía las tramas recibidas y procesadas y por el que se le dan ciertas ordenes de configuración y/o de programación tanto de la propia estación base como de los propios sensores. En la figura 2.14 se muestran los principales componentes de este dispositivo.

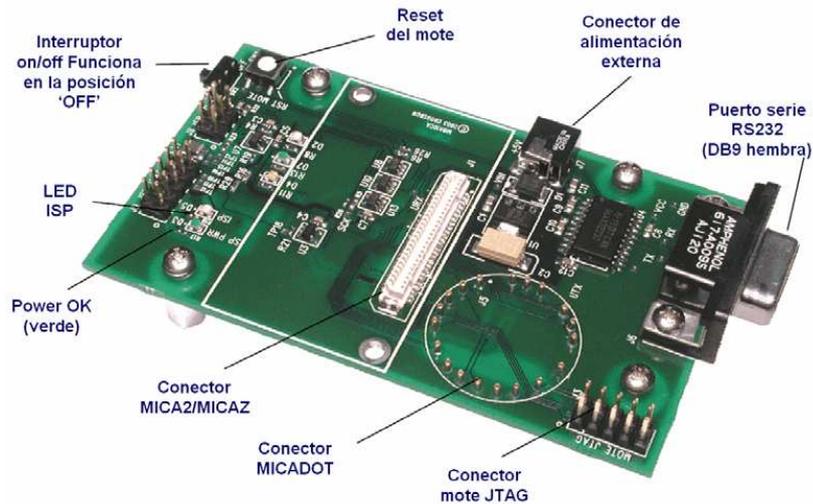


Figura 2.14. Gateway MIB510 y sus diferentes elementos.

Principalmente el MIB510 se emplea para recibir y transmitir a otro dispositivo los datos sensorizados de los motes. Sin embargo, el MIB510 no posee capacidad para transmitir y recibir tramas por sí mismo. Para comunicarse con el resto de sensores de la red, ha de insertarse un mote en el conector de 51 pines superior que se mostraba en la figura 2.14. De esta manera, estamos dotando al gateway de un módulo radio. Si además, en otro conector de 51 pines inferior conectamos una placa sensora, estamos convirtiendo el MIB510 en otro sensor de la red además de ser la estación base o nodo central (lo que antes de denominó como colector o sink) de la red. Esta configuración se muestra en la figura 2.15.



Figura 2.15. Gateway MIB510 con mote y placa sensora [11].

Como se observa, al situar el módulo radio, se da la posibilidad al MIB510 de funcionar sin corriente externa, sólo con la energía que proporcionan las dos pilas AA.

El MIB510 también se emplea para programar los motes, es decir, para cargar los programas de funcionamiento en los módulos radio que son los que tienen capacidad de programación (ya que las placas sensoras no poseen capacidad alguna de procesamiento de datos). Para programar cada mote, hay que conectar cada uno de ellos al MIB510 a través del conector de 51 pines y poner el interruptor del gateway en la posición on. La programación de los motes se realiza a través de un microprocesador ATMega16L integrado en la placa, que es lo que se denota como *ISP*, on-board in-system processor. El código se descarga al ISP a través del puerto serie. A continuación, el ISP descarga el

programa en el mote. Cuando el mote se está programando el LED ISP se enciende en rojo. En la figura 2.16 se muestra la comunicación entre estos dispositivos.

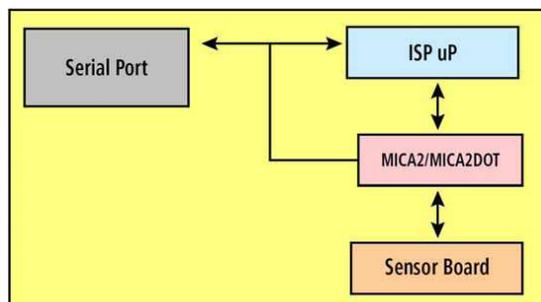


Figura 2.16. Diagrama de bloques del gateway MIB510 [11].

En el desarrollo de este proyecto no ha sido necesario programar los motes con un código propio ya que por un lado, no era el objetivo del proyecto y por otro, el programa que viene por defecto instalado en el mote es totalmente adecuado para cubrir las necesidades requeridas. Sin embargo, para una mayor información acerca de la programación de los motes, rogamos al lector se dirija al *Getting Started Guide* [11], capítulo tercero.

### 2.8.2. Módulo inalámbrico MPR410

El módulo inalámbrico o módulo radio es quizá la parte más importante de un nodo sensor puesto que brinda al dispositivo la capacidad para comunicarse con otros nodos, procesar los datos sensorizados, y en muchos casos, la capacidad de poner en marcha procesos en función de los datos capturados del entorno.

El mote empleado en el desarrollo de este proyecto es el MPR410, que a partir de ahora denominaremos como Mica2 y que se muestra en la figura 2.17. Con este nombre el fabricante ha querido distinguir este tipo de sensores de los MicaZ que se emplean para redes ZigBee en la banda de 2,4 GHz, pero cuyo aspecto exterior es muy similar. Sin embargo, los Mica2 trabajan en otro rango de frecuencias, existiendo tres modelos según la frecuencia a la que deseamos trabajar. El MPR410 trabaja en el rango de los 433 a 434,8 MHz pudiendo seleccionar la frecuencia exacta de funcionamiento dentro de este rango mediante programación como se comentó en el apartado 5.2 de este capítulo.

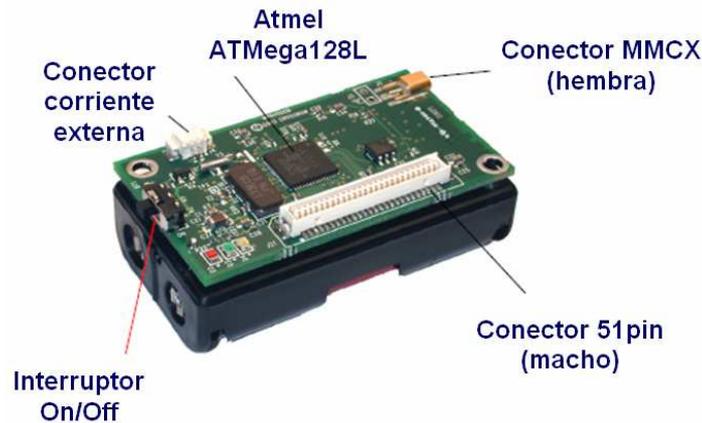


Figura 2.17. Mica2 y sus principales componentes.

En la figura 2.17 se muestra el aspecto físico del MPR410 y sus principales componentes, mientras que en la figura 2.18 se muestran los componentes relacionados de manera funcional mediante un diagrama de bloques.

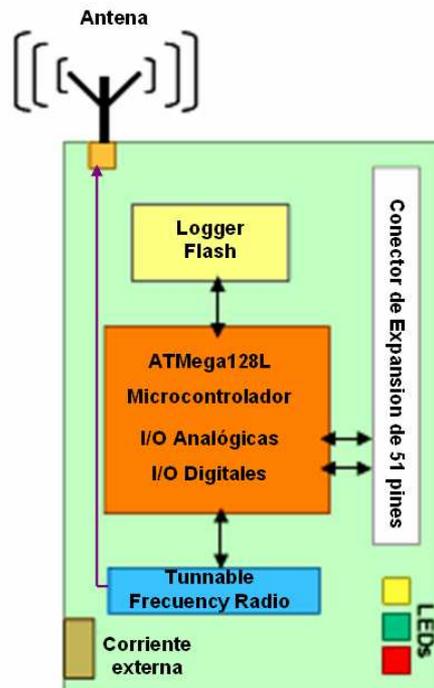


Figura 2.18. Diagrama de bloques de Mica2.

El mote Mica2 posee una memoria Flash para almacenar datos y medidas (en la figura 2.18 aparece como logger flash) que es una Atmel AT45DB041 y puede almacenar hasta cien mil lecturas. El microcontrolador es el Atmel ATmega128L es el microcontrolador que se encarga de tomar decisiones acerca de los datos sensorizados, comunicarse con el conector de 51 pines para la entrada/salida de señales analógicas y digitales, etc.

Otro módulo importante es el que se encarga de seleccionar la frecuencia que el usuario asignó cuando el mote fue programado.

Éste módulo se comunica con el microcontrolador para conocer la frecuencia a la que tiene que trabajar y envía la señal, modulada en FSK, al conector de la antena (conector MMCX). La antena es un monopolo de cuarto de onda (diagrama omnidireccional) cuya potencia de salida es también programable desde -20 a 10 dBm aunque por defecto viene configurada para máxima potencia de transmisión (10 dBm). La sensibilidad cuando el mote está recibiendo es de -101 dBm [14]. Otra consideración a tener en cuenta con respecto a los parámetros radio es que en el MPR410 pueden configurarse hasta cuatro canales para transmisión/recepción. Sin embargo, en [12] se recomienda al menos 500kHz de separación entre las frecuencias centrales de cada uno de estos canales para así evitar interferencia cocanal.

Finalmente, con respecto a la batería se ha de señalar que como se observa en la figura 2.17 el MPR410 funciona con dos pilas alcalinas AA de capacidad típica 200 mA-hr cuyo rango de tensiones operativo se encuentra entre los 3,6 y los 2,7 V. También se puede conectar una fuente de energía externa como se muestra en la figura 2.19. En este caso se ha conectado un compartimento para pilas AA pero puede ser cualquier otro tipo de fuente externa de energía.

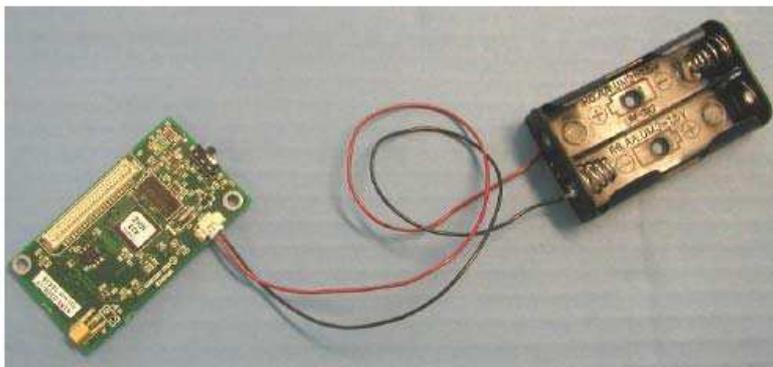


Figura 2.19. Conexión de energía externa al MPR410 [11].

### 2.8.3. Placa sensora

El objetivo primordial de una red de sensores es, obviamente, sensorizar los parámetros del entorno. Para este objetivo se añade al mote una placa sensora que se comunica con el mote a través del conector de expansión de 51 pines. La placa que se ha empleado en el desarrollo de este proyecto es la MTS310 que contiene:

- Sensor de luz
- Sensor de temperatura
- Acelerómetro de dos ejes (x e y).
- Magnetómetro de dos ejes (x e y).
- Circuito de detección de tono
- Micrófono
- Altavoz

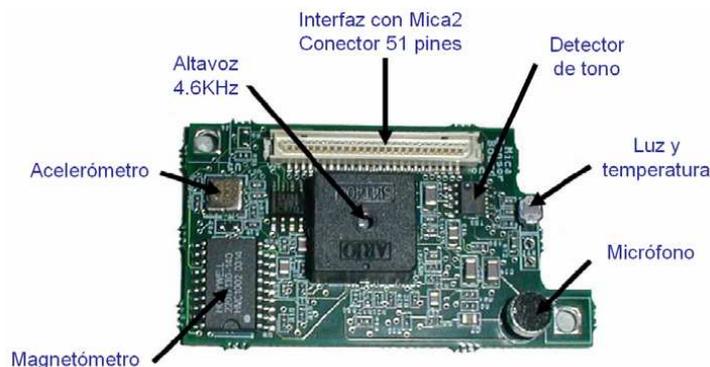


Figura 2.20. MTS310 y sus principales componentes.

En el desarrollo del proyecto y según el programa que ejecutaban los sensores, se monitorizan las siguientes variables:

- Luz.
- Temperatura.
- Batería del mote.
- Aceleración en eje x e y.
- Magnetismo en eje x e y.

Este tipo de sensores están orientados a monitorización ambiental, detección de vehículos, detecciones sísmicas de baja precisión, rango acústica, detección de movimiento, robótica y otras aplicaciones.

Para especificaciones acerca de las características técnicas de cada sensor, el lector puede dirigirse al manual de usuario de las placas MTS.

## 2.9 Mensajes enviados por los sensores

Los motes están continuamente ejecutando el programa por defecto instalado, el *Surge\_Reliable* desarrollado por Crossbow. Este programa hace que los nodos sensores envíen sus datos al nodo central empleando para ello el camino más corto [11]. Esto hace que los propios motes puedan decidir a quién envían sus datos, esto es, si es más conveniente enviarlos directamente al nodo central o si en cambio es preferible enviarlo a otro mote y que éste reenvíe el mensaje. Esta situación da a cada mote una funcionalidad de routing que hasta ahora se había comentado además de capacidad de reconfiguración a la topología de red. Cada nodo puede decidir a dónde es más conveniente enviar tanto su propia información como la que le envían a él otros sensores. Esto es lo que antes se había denominado en el apartado 2.3.2 como redes multisalto o *multi-hop*. Una vez comentado el funcionamiento del programa y aunque ya se ha comentado qué programas nos proporcionan una información de los datos sensorizados con cada sensor, se ha de conocer cual es la estructura exacta de las tramas que envían los motes para poder realizar aplicaciones propias. En general, la estructura de una trama sigue el siguiente patrón [16]:

SYNC_BYTE	Packet Type	Payload Data	SYNC_BYTE
0	1	2...n-1	n

Figura 2.22. Paquete de datos [16].

En donde *SYNC\_BYTE* es el byte 0x7E que indica que comienza o termina una nueva trama. Se ha de señalar que ya que este byte se reserva a este cometido, ningún byte en la trama puede tener ese valor. Si algún campo de la trama toma ese valor, se sustituirá por los bytes 0x7D5E.

El campo *Packet Type* es el tipo de paquete que en este caso será el byte 0x42 que indica que son paquetes que no necesitan de paquetes de reconocimiento (ACKs). En *Payload Data* se insertan la trama de datos que suele ser un mensaje Tinyos de longitud variable de hasta 255 bytes. Éstos siguen la siguiente estructura:

Address		Message Type	Group ID	Data Length	Data	CRC	
0	1	2	3	4	5...n-2	n-1	n

Figura 2.23. Mensaje Tinyos [16].

El campo *Address* indica la dirección a quién va enviado el mensaje. Si es un mensaje dirigido a todos los nodos, broadcast, la dirección será 0xFFFF. En el caso de ser un mensaje de un nodo, al puerto serie del gateway, como ocurrirá la mayor parte de las ocasiones, los bytes recibidos serán 0x007E. Se ha de tener en cuenta que, en equipos con sistema operativo Windows XP®, los bytes que llevan delante el byte 0x00 pueden estar desordenados y aparecer detrás. En el caso de la dirección 0x007E aparecería como 0x7E00.

El campo *Message Type* indica el tipo de mensaje que va insertado en el campo de datos. Para el caso de la aplicación instalada por defecto en los motes, Surge-Reliable, el byte será 0x11.

*Group ID* indica el identificador que posee el grupo de sensores que participan en la red. Sólo los motes con el mismo ID podrán comunicarse entre ellos.

Obviamente *Data* es el campo donde va insertado el mensaje en sí, que depende de la aplicación. Por su parte, *Data Length* indica la longitud en bytes del campo de datos sin tener en cuenta el CRC.

El programa instalado en los motes, como se dijo anteriormente, implementa una red multisalto. Algunas redes multihop poseen un protocolo propio de comunicación, como es el caso que aquí ocupa. El mensaje que se envía siguiendo este protocolo se encapsula en el campo de datos de la trama que se indica en la figura 2.23 y su estructura es la que se muestra en la figura siguiente.

Source Address		Origin Address		Sequence Number		Hop Count	Application Data
0	1	2	3	4	5	6	7...n

Figura 2.24. Mensaje Multihop [16].

El campo *Source Adres* indica la dirección que envía el mensaje mientras que *Origin Address* es la dirección del nodo que originó dicho mensaje (en una red multihop no tiene por qué ser el mismo).

El número de secuencia o *Secuence Number* se emplea para controlar el correcto envío de los mensajes. *Hop count* se emplea para calcular la ruta seguida por el mensaje ya que indica el número de nodos atravesados. Finalmente, en el campo *Application Data* sí se incluye el mensaje que se genera con la ejecución de la aplicación embebida en los notes. Este mensaje se muestra en la figura 2.25.

Type	Reading		Parent Addr		Sequence Number				Light	Temp	Mag X	Mag Y	Accel X	Accel Y
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Figura 2.25. Mensaje Surge-Reliable [16].

El byte *Type* indica la acción que se realiza con el envío del mensaje. Para un mensaje proveniente de las lecturas de las variables sensorizadas el byte type será 0x00. El campo *Reading* no se usa mientras que *Parent Addr* indica la dirección del nodo padre, esto es, la ID del nodo al que el nodo origen envía sus mensajes. Los siguientes seis bytes indican cada uno de ellos las lecturas de la luz, la temperatura, el campo magnético en x e y y la aceleración en x e y.

A modo de resumen de lo anteriormente descrito, se presenta un ejemplo de un paquete recibido:

**7E 42 7D 5E 00 11 7D 5D 16 00 00 02 00 9D 00 00 00 00 00 00 6F 00 80 DB F9  
7D 5E FF FF 01 C8  
EC D9 7E**

El mensaje Tinyos correspondiente, sustituyendo los 0x7D5E por su valor correspondiente es el siguiente:

**7E 00 11 7D 16 00 00 02 00 9D 00 00 00 00 00 00 00 6F 00 80 DB F9 7E FF FF 01  
C8 EC D9 7E**

En la tabla siguiente se muestra la interpretación de cada uno de estos bytes, aunque para más información puede dirigirse a [16].

<b>Bytes</b>	<b>Description</b>
7E 00	The UART Serial address – 126 (0x00 7E)
11	The message type. Surge message – 17 (0x11)
7D	Group ID. Crossbow mote default is 125 (0x7D)
16	Data length – 22 (0x16)
00 00	The address of the last forwarding node. (node 0)
02 00	The origin address. (node 2)
9D 00	The sequence number – 157 (0x00 9D)
00	The hop count
00	Surge message type (00 – sensor reading)
00 00	Reading – not used
00 00	Parent Node address
6F 00 80 DB	Battery voltage (<< 9 bits), surge sequence no (>>23 bits)
F9	Raw light value
7E	Raw temp value
FF	Raw mag x value
FF	Raw mag y value
01	Raw accel x value
C8	Raw accel y value
EC D9	CRC

Figura 2.25. Interpretación de la trama [16].

### 3. El sistema GSM/GPRS

#### Historia

A principios de los '80, los sistemas telefónicos móviles analógicos han tenido un rápido desarrollo en Europa, y así cada nación desarrolló un sistema propio, pero que era incompatible con los sistemas de otros países. Esta situación no era agradable, porque no sólo los sistemas móviles debían limitar su operatividad dentro de los confines nacionales, (que en una Europa unida se estaban convirtiendo cada vez en más numerosos, menos importantes), sino que también creaba un mercado muy limitado para los varios tipos de preparaciones necesarias a la implantación y al desarrollo de las redes, de tal forma que no podían realizarse economías de gran escala con los consiguientes ahorros tanto a favor del usuario como de los operadores de red.

En 1982, un gestor público de servicios de telefonía móvil de los países nórdicos (Nordic PTT) envió una propuesta al CEPT (Conference Europeenne de Postal et Télécommunications) para la implantación de un servicio común de telefonía móvil europeo en la frecuencia de los 900 MHz. El CEPT decidió entonces formar el Groupe Speciale Mobile (del que proviene el nombre GSM) con el fin de desarrollar un Standard pan-europeo para las comunicaciones celulares. Actualmente el acrónimo GSM está para Global System for Mobile Communication, donde se ha querido utilizar el término global a causa de la adopción de este Standard en cada continente del planeta.

Entre 1982 y 1985 se planteó qué tipo de sistema construir: digital o analógico. Pero en 1985, tras numerosas discusiones, el grupo decidió implantar un sistema basado en tecnología digital. El paso siguiente fue el de elegir entre la solución de banda ancha (broadband) o banda estrecha (narrowband). Por esta razón, en 1987 se efectuaron en París pruebas de campo, en las que diferentes fabricantes propusieron soluciones diversas (broadband y narrowband). En mayo de 1987 se eligió la solución **narrowband TDMA** (Time División Múltiple Access).

En el mismo periodo las 13 primeras naciones (en el Reino Unido dos operadores) firmaron el **MoU** (Memorandum of Understanding), comprometiéndose a respetar las normativas y prometiendo tener el mismo sistema basado en el Standard GSM operativo a partir de primeros de Julio de 1991. El cuerpo del Standard estaba formado inicialmente por poco más de cien normativas -a cuya redacción colaboraron PTT, centros de búsqueda y empresas manufactureras de toda Europa- y representa uno de los proyectos más ambiciosos de los últimos diez años de la European Telecommunications Standard Institute (**ETSI**), al cual la CEE le ha mandado unificar la normativa europea en el sector de las telecomunicaciones y que en 1990 publicó la Fase I de las normativas del sistema GSM. Los primeros servicios comerciales fueron lanzados a mediados de 1991, y en 1993 estaban ya operativos 36 redes GSM en 22 países.

Las normativas se ampliaron enseguida para incluir una interfaz aérea también para la banda de los 1800-1900 MHz (**DCS1800- PCS1900**). En particular a USA se le ha concedido la banda de los 1900 MHz y a Europa y a los otros países extranjeros la de los 1800 MHz.

A pesar de que se haya estandarizado ya en Europa, el sistema GSM no es sólo un standard europeo, de hecho hay redes GSM operativas o planificadas en 1996 en otros 100 países de todo el mundo. El aumento de los abonados ha sido vertiginoso.

El standard GSM reúne una serie de mejoras e innovaciones respecto a las redes celulares existentes, destinadas a un uso eficiente del espectro de las radio-frecuencias (RF), a la seguridad de la transmisión, a la mejora de la calidad de conversación, a la reducción del coste de los terminales, de las infraestructuras y de la gestión, a la capacidad de soportar nuevos servicios y a la plena compatibilidad con la red **ISDN** (Integrated Services Digital Network) y con otras redes de transmisión de datos.

Además, la red radio móvil GSM constituye el primer sistema estandarizado para usar una técnica de transmisión numérica por el canal radio: este punto representa una característica peculiar de la red, en tanto en cuanto todos los sistemas radio celulares anteriores, utilizaban técnicas de transmisión analógicas. Otra característica de base del sistema es el *roaming* (movilidad), es decir la posibilidad ofrecida al usuario de móvil, de acceder a los servicios GSM también cuando se encuentra físicamente fuera del área de cobertura de la propia red de suscripción, registrándose como usuario visitante. El Roaming es completamente automático dentro de todas las naciones con cobertura del sistema GSM.

Además de la posibilidad de efectuar Roaming, el GSM ofrece nuevos servicios para el usuario, como la transmisión de datos, el servicio fax y el servicio de transmisión breve de mensajes de texto.

El sistema GSM reúne un cierto número de “interface open” (**OSI = Open System Interconnection**) ofreciendo funciones de servicio y de capacidad, permitiendo a las industrias una flexibilidad de implantación de la red y a los operadores más facilidad en instalación y manutención de los equipamientos.

Resumiendo, las principales características de este nuevo proyecto, se dirigen a alcanzar los siguientes **objetivos**:

- Posibilidad de usar el mismo terminal radio en todos los Países del área CEE, y en aquellos Países no pertenecientes a la Comunidad pero que utilizan el mismo standard (Roaming internacional).
- Mejora de la eficiencia espectral respecto a las actuales redes radiomóviles celulares de tipo analógico.
- Seguridad de la transmisión radio (para impedir interceptar las conversaciones y los datos identificativos de los usuarios).
- Empleo de la técnica numérica, para permitir mejorar la calidad fónica, la transmisión de datos y la compatibilidad con los standards internacionales a nivel OSI (Open System Interconnection) e ISDN (Integrated Services Digital Network).

### 3.1. ¿Qué es GSM?

Global System for Mobile Communication, gran protagonista de la Comunicación de los años'90, supera las barreras nacionales y llega al extranjero. Por tanto si vuestros compromisos de trabajo abarcan a menudo los límites nacionales, el móvil GSM es Vuestro compañero de viaje ideal.

Hasta ahora podía uno disfrutar de la libertad de comunicación únicamente dentro del territorio donde se había efectuado la suscripción. Los **standards técnicos**, diferentes de nación a nación, incompatibles entre sí, impedían recibir llamadas o hacerlas fuera de la propia nación. Hoy las barreras de comunicación han sido abatidas: además del notable Servicio Radiomóvil Etacs (Extended Total Access Cellular System), se pone a vuestra disposición el Servicio Radiomóvil Internacional **GSM**, basado en una tecnología digital de absoluta vanguardia.

El sistema de comunicación GSM, introducido en Italia a principios de octubre de 1992, permite, en particular, efectuar **Roaming** Internacional o bien hacer o recibir, en el extranjero, llamadas como si se hicieran desde Italia. De hecho, se han firmado acuerdos bilaterales que permiten a los Clientes de Telefonía Móvil GSM trasladarse con el propio móvil y continuar disfrutando del servicio en diferentes países europeos y no europeos. En un primer momento el servicio GSM se concibió como el standard europeo para las comunicaciones móviles digitales, pero se está convirtiendo, de hecho, en standard mundial. Otras 100 redes GSM se hicieron operativos en junio de 1996; cada día en todo el mundo se realizan cerca de 30.000 contratos y se efectúan cerca de 40 millones de llamadas. En el 2000 los expertos prevén que estarán en circulación por el mundo alrededor de 100 millones de abonados. El aspecto "más espectacular" del sistema es que gracias a una tecnología de absoluta vanguardia, con el GSM no es el móvil el que contiene los datos del abonado, sino más bien "una tarjeta inteligente" denominada SIM Card (Subscriber Identity Module), para insertar en el aparato desde el que se desea llamar: la suscripción está en la tarjeta, no en el móvil. En otras palabras, se puede llamar también, aunque no se tenga a mano el propio aparato GSM, o bien en aquellos países que adoptan el standard GSM con frecuencias diferentes (DCS 1800-PCS 1900): es suficiente con tener uno predispuesto para recibir la tarjeta compatible, es decir, con el standard GSM. Existen dos tipos de SIM Card, una de las dimensiones de una tarjeta de crédito (**ISO**), preparada principalmente para los radioteléfonos vehiculares; otra pequeña como un sello preparada para los teléfonos palmarios (**Plug In**). Existe en todo caso un adaptador que permite transformar una tarjeta SIM desde el formato Plug In al formato ISO. En la tarjeta SIM se pueden memorizar números telefónicos asociados a nombres, además de aquellos que se pueden memorizar en la memoria del móvil. Aquella dispone, además, de dos códigos de seguridad el **PIN** y el **PUK**. abonado ni descifrar las conversaciones vía radio.

### **Introducción técnica**

A diferencia de lo que sucede en la red telefónica fija, en la que el terminal de cada usuario está conectado a la red mediante un punto de acceso unívoco, en una red radio-móvil, el abonado puede desplazarse por cualquier punto de la misma. Por tanto, los datos relativos al abonado deben ser memorizados en una base de datos que se pueda consultar y actualizar desde cualquier punto de la red.

La característica de base de un sistema radio-móvil puede resumirse en términos de enlaces entre los aparatos radio, los nodos radio-móviles, la base de datos y la red PSTN/ISDN, con el fin de identificar los terminales móviles, para estabilizar, controlar y terminar las conexiones y actualizar los datos de gestión.

En todos los sistemas radio-móviles el factor que tiene mayor importancia en el proyecto del sistema, es el espectro de frecuencia disponible (ancho de banda), de hecho el número de frecuencias radio asignado a estos servicios es limitado.

Para aprovechar al máximo el ancho de la banda disponible, con el fin de servir a más usuarios a la vez en un mismo sector, el sistema se estructura subdividiendo el área de servicio (Service area) en zonas delimitadas llamadas **celdas**. Cada celda tiene una Estación Radio Base (**BTS**) que opera en un set de canales radio, diferentes a los utilizados en las celdas adyacentes, para evitar interferencias. Este tipo de subdivisión permite la reutilización de las mismas frecuencias en celdas no adyacentes. La unión de las celdas, que en su conjunto utilizan todo el espectro radio disponible, se llama **cluster**.

Generalmente se utilizan formas regulares de celdas y por tanto de clusters para cubrir un área de servicio. Teóricamente las celdas se pueden imaginar con forma hexagonal, aunque en realidad su forma es irregular a causa de la no homogénea propagación de la señal de radio, debido principalmente a la presencia de obstáculos.

Reduciendo el diámetro de las celdas la capacidad del sistema aumenta, aunque el uso de esta elección supone la disminución de la distancia de reutilización de las frecuencias, es decir de la distancia entre dos celdas co-canal, que conlleva el aumento de la interferencia co-canal. Parece evidente que la capacidad del sistema, amén del número de canales disponibles, está ligada a este tipo de interferencia y, por ello, el sistema GSM utiliza técnicas, que se describirán a continuación, destinadas a minimizarlas.

El standard GSM utiliza la tecnología de acceso a división de frecuencia (**FDMA**) combinada con la de acceso a división de tiempo (**TDMA**): 8 canales vocales (*Full rate*) o bien 16 (*Half rate* "multiplexadas" en un único canal radio, junto a las informaciones de control de error, necesarias para disminuir la interferencia debida al ruido, y a las informaciones de sincronización y señalización.

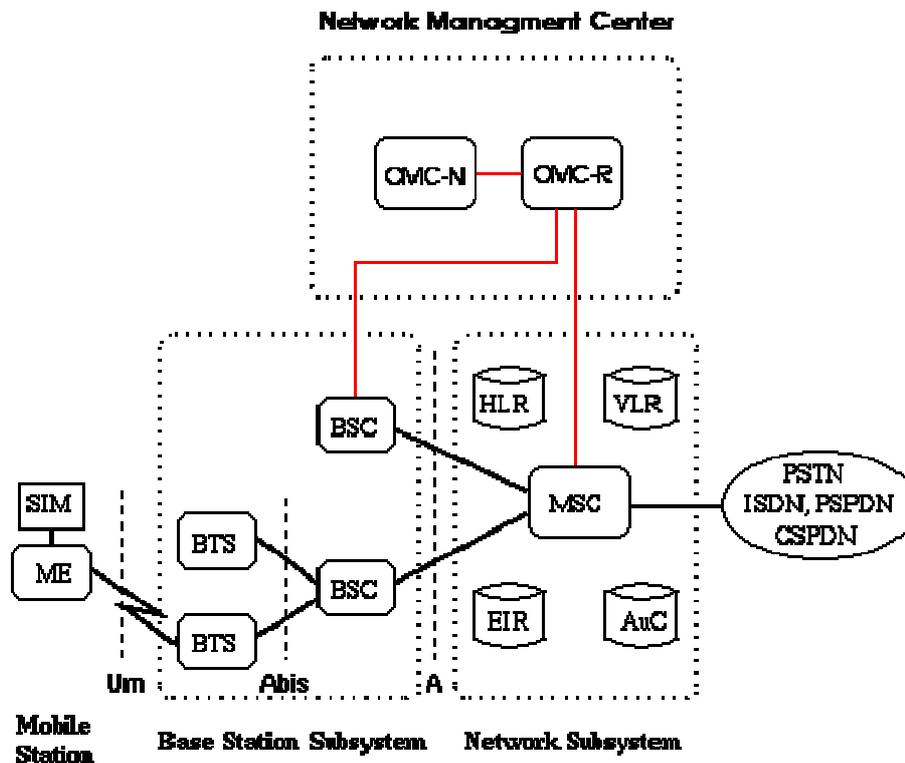
### 3.2. Arquitectura de red GSM

La arquitectura de base del sistema GSM prevé cuatro subsistemas principales cada uno de los cuales contiene un cierto número de unidades funcionales y está interconectados con el otro mediante interfaces standard que se describirán a continuación. Los subsistemas principales del network y los elementos que lo componen son:

- MS (Mobile Station).
  - ME (Mobile Equipment).
  - SIM (Subscriber Identity Module).
  
- BSS (Base Station Sub-System).
  - BSC (Base Station Controller).
  - BTS (Base Transceiver Station).
  
- NSS (Network Sub-System).

- MSC (Mobile Switching Center)
  - (Home Location Register).
  - HLR VLR (Visitor Location Register).
  - AUC (Autentication Center).
  - EIR (Equipment Identity Register).
- NMC (Network Management Center).
    - OMC (Operation and Maintenance Center).

La Mobile Station es el terminal radiomóvil transportado por el abonado. El Base Station Sub-System se ocupa del control de la conexión radio con el MS. El Network Sub-System realiza la conmutación de las llamadas entre redes móviles y la red fija o hacia otras redes radiomóviles y se ocupa además de la supervisión de la movilidad de los abonados. Desde el Network Management Center se pueden controlar todas las operaciones en curso, además de efectuar la configuración de la red. A continuación sigue la descripción de las mencionadas entidades.



SIM Subscriber Identity Module	BSC Base Station Controller	MSC Mobile services Switching Center
ME Mobile Equipment	HLR Home Location Register	EIR Equipment Identity Register
BTS Base Transceiver Station	VLR Visitor Location Register	AuC Authentication Center
OMC-R Operation Management Center Regionale	OMC-N Operation Management Center Nazionale	

Figura 3.1. Esquema de la red GSM

### 3.2.1. Mobile Station

La Mobile Station está formada por el Mobile Equipment (el terminal GSM) y por el Subscriber Identity Module (**SIM**), una pequeña tarjeta dotada de memoria y microprocesador, que permite identificar al abonado independientemente del terminal usado; y por tanto la posibilidad de continuar recibiendo y efectuando llamadas y utilizar todos los servicios suscritos insertando la tarjeta SIM también en un terminal que no sea el propio.

- **Mobile Equipment**

El Mobile Equipment está inequívocamente identificado dentro de cualquier red GSM por el International Mobile Equipment Identity (**IMEI**). EL IMEI es un número de 15 cifras y tiene la siguiente estructura:

$$\text{IMEI} = \text{TAC} / \text{SNR} / \text{Sp}$$

Donde:

**TAC** = Type Approval Code, determinado por el cuerpo central del GSM (6 cifras)

**FAC** = Final Assembly Code, identifica al fabricante (2 cifras).

**SNR** = Serial Number (6 cifras).

**Sp** = Cifra suplementaria de reserva (1 cifra).

Los terminales GSM están subdivididos en cinco clases basándose en la misma potencia con la que pueden transmitir sobre el canal radio, que varía desde un máximo de 20 Watt a un mínimo de 0.8 Watt. La siguiente tabla resume las características de estas cinco clases.

CLASE	POTENCIA MÁXIMA	TIPO
1	20	VEHICULAR
2	8	PORTÁTIL
3	5	PALMARIO
4	2	PALMARIO
5	0.8	PALMARIO

La potencia del MS determina la capacidad de ésta última para alejarse de la estación transmisora/receptora (**BTS**) de la red y poder seguir disfrutando del servicio.

Una peculiaridad de los **MS** está formada por la capacidad de variar la potencia de emisión de la señal sobre el canal radio de forma dinámica en 18 niveles, con el fin de poder mantener en cada momento la potencia de transmisión óptima, limitando así las interferencias co-canal inducidas sobre las celdas adyacentes y por tanto reduciendo los consumos del terminal. Estos dos últimos aspectos están mejorados por el Discontinuos Transmit (**DTX**) que inhibe la transmisión cuando el usuario no habla, gracias a la función Voice Activity Detection (**VAD**) que verifica la presencia o no de actividad vocal. El aumento o la disminución de la potencia de la señal transmitida le llega a la MS desde **BSS** que monotoriza constantemente la calidad de comunicación.

- **SIM**

La tarjeta SIM contiene la International Mobile Subscriber Identity (**ISMI**), usada para identificar al abonado en cualquier sistema GSM, los procedimientos de criptografía que garantizan la confidencialidad de la información del usuario, otros datos como por ejemplo memorias alfanuméricas del teléfono y memorias para mensajes de texto (**SMS**) y finalmente una contraseña para impedir el uso no autorizado de dicha tarjeta y para el acceso a posteriores funciones.

La IMSI tiene la siguiente estructura:

**MCC / MNC / MSIN**

donde:

**MCC** = *Mobile Country Code* (2 o 3 cifras, para Italia 39)

**MNC** = *Mobile Network Code* (2 cifras, en Italia 01 para TIM y 10 para Omnitel)

**MSIN** = *Mobile Station Identification Number* (max 13 cifras)

### 3.2.2. Base Station Sub-System:

El Base Station Sub-System controla la interfaz radio. Está compuesto por una o más Base Transceiver Station (**BTS**) y por un Base Station Controller (**BSC**). Estos elementos se comunican entre sí a través de una interfaz estandarizada tipo A-bis, con el fin de permitir operaciones incluso entre componentes construidos por fabricantes diferentes, además el BSC está conectado al Mobile Switching Center (**MSC**) mediante una interfaz tipo A.

#### □ **Base Transceiver Station:**

El Base Transceiver Station aloja todos los receptores transmisores que sirven una celda y que se interesan por recibir y enviar información al canal radio, abasteciendo una interfaz física entre la Mobile Station y el BSC. EL BTS ejerce una serie de funciones descritas a continuación:

- Capacidad de gestionar canales Full Rate y Half Rate.
- La gestión de la *Antenna Diversity*, es decir la utilización de dos antenas de recepción para mejorar la calidad de la señal recibida; las dos antenas reciben de forma independiente la misma señal y están afectadas de distinto modo por el *fading*; la posibilidad de que ambas sean afectadas es muy pequeña.
- Supervisión del *Relación Ondas Estacionarias (ROS)* en antena.
- *Frequency Hopping (FH)*: cambio de la frecuencia usada en un canal radio a intervalos regulares, con el fin de mejorar la calidad del servicio a través de las distintas frecuencias.
- *Discontinuos Transmission (TDX)* ya sea en el up-link como en el down-link.

- El *Control Dinámico de la Potencia (DPC)* del MS y de la BTS: el BSC determina la potencia óptima con la que del MS y la BTS deben transmitir sobre el canal radio (explotando las mediciones realizadas por MS y BTS), para mejorar la eficiencia espectral.
- La gestión de los algoritmos de clave: la información de los usuarios criptografía para garantizar al abonado una cierta discreción sobre el canal de tráfico y el de señalización. El proceso de criptografía de los datos debe ser realizado por la BTS sobre las informaciones transmitidas el canal radio; el algoritmo de criptografía que debe utilizarse es comunicado a la BTS por la BSC en base a las indicaciones recibidas por la MSC y la clave de criptografaciones única para casa usuario. Actualmente el estándar GSM Fase II admite 8 algoritmos de clave.
- Monitorización de la conexión radio realizando medidas significativas sobre señales RF, medidas que luego se envían a la BSC para la elaboración con la finalidad de asegurar un elevado nivel de calidad de la conexión.

### ***Base Station Controller***

El Base Station Controller gobierna los recursos radio para una o más BTS, controlando la conexión entre las BTS y las MSC (centrales de conmutación que proporcionan la conexión a la red física y a otras redes), y además gestionando los canales radio, la señal, el frequency hopping y los handover.

En particular permite:

La gestión y configuración del canal radio: para cada llamada tiene que elegir la celda correcta y una vez en su interior seleccionar el canal radio más apto para efectuar la conexión.

La gestión de los handover: sobre la base de las medidas recibidas por el BTS, decide cuando efectuar el handover, es decir el cambio de celda cuando el usuario se desplaza durante una conversación dentro del área de cobertura de su competencia.

Funciones de transcodificación de los canales radio Full Rate( 16 kbps ) o Half rate ( 8 kbps ) en canales a 64 kbps.

### ***3.2.3. Network Sub-System:***

El Network Sub-System explica las funciones de conmutación para la conexión con otros abonados de la red fija o móvil mediante la MSC y las funciones de database, distribuidas en 4 nudos inteligentes (**HLR, VLR, AUC, EIR**) para la identificación de los terminales y de los usuarios, la actualización de su posición, la autenticación y conducción de las llamadas a un abonado en roaming.

- ***Mobile Switching Centre***

El Mobile Switching Centre (**SMC**) es el elemento central del NSS. Se ocupa, basándose en las informaciones recibidas desde el NLR y desde el **VLR**, de la conducción (routing) y gestión de la señal de todas las llamadas directas y provenientes desde varios tipos de redes, como **PSTN**, **ISDN**, **PLMN** y **PDN**. Implementa además las funciones de gateway con los otros componentes del sistema y de gestión de los procesos de handover, conmutando las llamadas en curso entre BSC diferentes o hacia otro MSC.

Dentro del servicio pueden estar presentes más MSC y cada una es responsable de la gestión del tráfico de una o más BSS y desde el momento en que los usuarios se trasladan por toda el área de cobertura, para garantizar a cada uno un nivel de servicio constante, los MSC tienen que encontrarse en situación de gestionar números de usuarios variables en tipología además de en calidad.

Otras funciones fundamentales de los MSC se describen a continuación:

1. Autenticación del que llama; la identificación de la MS que ha efectuado la llamada

es necesaria para determinar si el usuario está habilitado para disfrutar del servicio.

2. Confidencialidad acerca de la identidad del usuario: para garantizar la confidencialidad acerca de la identidad de un usuario en el canal radio, aún estando ya todas las informaciones criptografiadas, el sistema no transmite nunca el **IMSI** asignado cuando el usuario suscribe el abono; sin embargo se le asigna el Temporary Mobile Subscriber Identity (**TMSI**), que se asigna en el momento de la llamada y tiene un significado temporal: crear la correspondencia entre TMSI e IMSI es tarea del MSC y cuando el móvil se desplaza a la location area controlada por otro MSC, se le tiene que asignar un nuevo TMSI.

3. Proceso de *handover*: en la red GSM un usuario puede continuar utilizando el servicio aunque atraviere durante la conversación los límites de la celda en la que se encuentra. Se pueden verificar dos casos:

- La MS se traslada a una celda controlada siempre por el mismo MSC; en este caso el proceso de *handover* es gestionado por el mismo MSC.

- La nueva celda a la que se traslada la MS está controlada por otro MSC; en este caso el proceso de *handover* se produce desde dos MSC basándose en las medidas de señal monitorizadas por la BTS que reciben la MS.

### ***Home Location Register***

Cuando un usuario suscribe un nuevo abono a la red GSM, todas las informaciones para su identificación se memorizan en la **HLR**. Tiene la función de comunicar al VLR, que posteriormente veremos, algunos datos relativos a los abonados, en el momento en que estos se desplazan desde una *Location Area* a otra. Dentro del HLR los abonados son identificados por el número:

$$\text{MSISDN} = \text{CC} / \text{NDC} / \text{SN}$$

donde:

**CC** = *Country Code*, prefijo internacional (el CC italiano es 39).  
**NDC** = *National Destination Code*, prefijo nacional del abonado sin el

Cero.

**SN** = *Subscriber Number*, número que identifica al usuario móvil.

La *Home Location Register (HLR)* es un database (archivo) que puede ser único para todo el network o bien distribuido en el sistema; se pueden por tanto tener MSC sin los HLR, pero conectadas al de otras MSC. Cuando existen más HLR, a cada uno de ellos se les asigna un área de numeración, es decir un set de *Mobile Station ISDN Number (MSISDN)*. El MSISDN identifica unívocamente una suscripción de teléfono móvil en el plano de numeración de la red telefónica conmutada pública internacional.

El **HLR**, como todos los demás database que después veremos, está implementado en una workstation cuyas prestaciones (memoria, procesadores, capacidad de los discos) son actualizables cuando crece el número de abonados. Aquél contiene todos los datos relativos a los abonados y en particular las informaciones que están contenidas en él son:

#### **Informaciones de tipo permanente:**

1. La *International Mobile subscriber Identity (IMSI)*, que es la información que identifica al abonado dentro de una cualquiera de la red GSM y que está contenido también en el interior de la SIM.
2. El *Mobile Station ISDN Number (MSISDN)*.
3. Los tipos de servicio suscritos por el abonado a los cuales tiene derecho a acceder (voz, servicio datos, **SMS**, eventuales bloqueos para llamadas internacionales, otros servicios auxiliares).

#### **Informaciones de tipo dinámico:**

1. Posición corriente del **MS**, es decir la dirección del **VLR** en la que está registrada.
2. El estado de eventuales servicios auxiliares.

#### **Visitor Location Register**

El Visitor Location Register(**VLR**) es un database que memoriza de modo temporal los datos de todos los abonados que se encuentran en un área geográfica bajo su control. Estos datos se piden al **HLR** perteneciente al abonado. En general para simplificar las señalizaciones requeridas y la estructura del sistema, los fabricantes implementan el **VLR** y el **MSC** juntos, de modo que el área geográfica controlada por el **MSC** corresponde a la controlada por el **VLR**.

En particular las informaciones que contiene son:

- *Temporary Mobile Subscriber Identity (TMSI)*, usado para garantizar la seguridad del **IMSI**, se asigna cada vez que se cambia LA.
- Estado de la **MS** (standby, ocupado, apagado).
- El estado de los servicios suplementarios como *Call Waiting, Call Divert, Call Barring*, etc.
- Los tipos de servicios subscriptos por el abonado al que se le permite acceder (voz, servicio datos, **SMS**, otros servicios auxiliares).
- La *Location Area Identity (LAI)* en la que se encuentra la **MS** dentro de aquéllas bajo el control del **MSC/VLR**.

### ***Authentication Center***

La Authentication Center es una función del sistema que se ocupa de verificar si el servicio ha sido solicitado por un abonado legítimo, proporcionando ya sea los códigos para la autenticación como la clave, para proteger tanto al abonado como al operador de red, de intrusiones del sistema por parte de terceros.

El mecanismo de **autenticación** verifica la legitimidad de la SIM sin transmitir sobre el canal radio las informaciones personales del abonado, como IMSI y llaves de clave, a fin de verificar que el abonado que está intentando el acceso sea el verdadero y no un clon; la **clave** sin embargo genera algunos códigos secretos que se usarán para encriptar toda la comunicación cambiada por error sobre el canal radio. Los códigos de autenticación y clave están generados casualmente por cada abonado en particular por algunos sets de algoritmos definidos por el estándar y que son memorizados además de en la **AUC** también en la SIM.

La autenticación se produce cada vez que la MS se conecta a la red y más precisamente en los siguientes casos:

- Cada vez que la MS recibe o efectúa una llamada.
- Cada vez que se efectúa la actualización de la posición de la MS.
- Cada vez que se solicita la activación, desactivación o información .

sobre los servicios suplementarios.

### ***Equipment Identity Register***

El Equipment Identity Register es un data-base que verifica si un Mobile Equipment (ME) está autorizado o no para acceder al sistema. El data-base está dividido en tres secciones:

- **White List:**

Contiene todos los IMEI designados a todos los operadores de las varias naciones con las que se tienen acuerdos de roaming internacional.

- **Black List:**

Contiene todos los IMEI que se consideran bloqueados (por ejemplo los robados).

- **Grey List:**

Contiene todos los IMEI marcados como *faulty* o también los relativos a aparatos no homologados. Los terminales introducidos en la *Grey List* les son señalados a los operadores de sistema a través de una alarma cuando solicitan el acceso, permitiendo la identificación del abonado que utiliza el terminal y del área de llamada en donde se encuentra.

A cada tentativa de conexión de la **MS** con el network, la **MSC** mediante la **EIR** verifica la existencia de uno de los siguientes casos, para permitir o no el acceso:

- 1.- El terminal está homologado para la conexión con un network GSM.
- 2.- El terminal no ha sido robado o utilizado fraudulentamente.
- 3.- El terminal no está marcado como faulty.

El **EIR** puede ser único para todo el sistema o bien puede estar implementado en una configuración distribuida. Puede encontrarse en la misma workstation en que se encuentran **HLR** y **AUC**, pero generalmente es preferible tenerlo en una máquina a parte por razones de seguridad. Se puede acceder también por control remoto para permitir la actualización de las diferentes listas contenidas en él desde cada punto de la red. En el futuro está prevista la interconexión con todos los EIR de los diferentes operadores GSM, para evitar el uso de aparatos robados en países distintos de aquellos en los que ocurrió el robo.

### ***3.2.4. Network Management Center:***

#### ***Operation and Maintenance Center***

La Operation and Maintenance Center tiene las siguientes funciones:

- Acceso remoto a todos los elementos que componen el network GSM (BSS, MSC, VLR, HLR, EIR y AUC).
- Gestión de las alarmas y del estado del sistema con posibilidad de efectuar varios tipos de test para analizar las prestaciones y verificar el correcto funcionamiento del mismo.
- Recogida de todos los datos relativos al tráfico de los abonados necesarios para la facturación.

- Supervisión del flujo de tráfico a través de las centrales e introducción de eventuales cambiantes del flujo mismo.
- Visualización de la configuración del network con posibilidad de cambiarla por control remoto.
- Administración de los abonados y posibilidad de poder conocer su posición dentro del área de cobertura.
- En algunos sistemas de grandes dimensiones, pueden existir más **OMC**. En este caso existirá un **OMC** general desde el que es posible controlarlo todo (**OMC-N**) y otros **OMC** limitados al control de algunas zonas (**OMC-R**).

### **Introducción al servicio de transmisión de datos:**

Si bien en 1992 el servicio comercial GSM empezó exclusivamente con los servicios de voz, éste fue predispuesto desde el principio para responder a la creciente demanda de servicios de transmisión de datos, que permiten la integración de un PC portátil y de un teléfono GSM a través de una tarjeta PCMCIA o, en algunos casos, simplemente a través de un cable serial.

Para enviar datos a través de una línea telefónica analógica terrestre PSTN (Public Switched Telephone Network), es necesario utilizar los dispositivos módem (modulator-demodulator), que convierten los datos en señales variables en el tiempo (modulación), capaces de atravesar la red telefónica como si se tratase de una conversación normal; por tanto, la señal variable es reconvertida por otro módem en una señal numérica (desmodulación).

Considerando que la red GSM es una red totalmente digital, no es necesaria la utilización del módem (que realizan una conversión analógico-digital), sino que se necesita sólo un adaptador especial que adapta el flujo de datos proveniente del PC, al flujo de datos utilizado en el enlace digital entre el teléfono y la red GSM. Si la llamada de datos es directa hacia un ordenador no conectado a la red GSM, es necesario reconvertir la señal digital en analógica en el último tramo de la conexión, entre la red GSM y el ordenador al que se ha llamado. Los adaptadores disponibles en el mercado son fundamentalmente de tres tipos:

- PCMCIA.
- Serial.
- Serial integrado en el teléfono

El tipo de adaptador más utilizado es el PCMCIA, ya sea por sus reducidas dimensiones como porque toma la energía requerida para su funcionamiento directamente del ordenador, sin necesidad de otras fuentes de alimentación externas, necesarias en el caso del adaptador serial. Recientemente se han presentado algunos terminales GSM, que tienen el adaptador integrado en el teléfono y que se conectan con

el PC a través del puerto serie con un sencillo cable. En este caso la alimentación del adaptador es la propia batería del teléfono.

### 3.3 ¿Qué es GPRS?

La red GSM prevé unos servicios de transmisión de datos desde la fase inicial. Sin embargo, se trata de servicios con modalidad de transferencia por conmutación del circuito, es decir, donde la red, una vez establecida la conexión física de cabo a rabo entre dos usuarios, dedica los recursos propios hasta que no es solicitado expresamente el establecimiento de la conexión, independientemente del hecho de que los dos usuarios se intercambien datos durante todo el tiempo de conexión.

Esta modalidad de transferencia es óptima sólo en el caso en que los dos usuarios tengan que intercambiarse una cantidad significativa de datos (transferencia de ficheros o archivos); resulta ineficiente en cuanto los datos a intercambiarse son de pequeña entidad o bien, en el caso más frecuente, el tráfico de datos es de tipo interactivo o transitorio, es decir, el tiempo de uso efectivo de los recursos de la red supone sólo una parte con respecto al tiempo total de conexión (como, por ejemplo, la navegación en Internet a través de la World Wide Web).

Es decir, se crea el mismo problema para el **GSM** que para la PSTN (Public Switshed Telephone Network) hace unos años: prever una modalidad de transferencia por paquetes de datos, en la que los datos de los usuarios, contenidos en entidades de protocolo autosuficientes con indicación del remitente y del destinatario, pueden ser transportados por la propia red sin necesidad de una estrecha asociación con un circuito físico. Ya se ha dado un paso intermedio en esa dirección con el GSM de fase 2, previendo servicios con acceso a las puertas pertinentes de la red PSPDN (Public Switched Packet Data Network). Sin embargo, siempre es necesario establecer una conexión física (por conmutación del circuito) en la red de radio, incluso cuando se accede a un canal virtual de la red de paquetes. El resultado de ello es que el recurso de radio es igualmente infrautilizado y el usuario ocupa un canal de tráfico (por cuyo uso tendrá que pagar presumiblemente por el tiempo empleado), para conectarse a otra red en la cual, sin embargo, la información no viaja a un rendimiento fijo (y el transporte relativo se suele pagar en base al volumen de datos transportados).

Con el sistema **GPRS** (General Packet Radio Service), introducido por ETSI (European Telecommunication Standard Institute) para la fase 2+ del sistema GSM, el acceso a la red de paquetes se lleva al nivel del usuario del móvil a través de protocolos como los TCP/IP (Transmission Control Protocol), **X.25**, y **CLNP** (Connectionless Network Protocol), sin ninguna otra necesidad de utilizar conexiones intermedias por conmutación del circuito.

Al contrario que el servicio de transferencia de datos con modalidad de conmutación de circuito, el servicio GPRS permite la trasmisión de paquetes en modalidad link by link, es decir, los paquetes de información se encaminan en fases separadas a través de los diversos nodos de soporte del servicio, denominados GSN.

En los servicios GSM los recursos son gestionados según la modalidad resource reservation, o sea, se emplean hasta el mismo momento en que la petición de servicio no se ha llevado a término. En el GPRS, sin embargo, se adopta la técnica del context reservation, es decir, se tiende a preservar las informaciones necesarias para soportar ya sea las peticiones de servicio de forma activa o las que se encuentran momentáneamente en espera. Por tanto, los recursos de radio se ocupan, en efecto, sólo cuando hay necesidad de enviar o recibir datos. Los mismos recursos de radio de una celda se dividen así entre todas las estaciones móviles (MS), aumentando notablemente la eficacia del sistema.

El servicio GPRS, por tanto, está dirigido a aplicaciones que tienen las siguientes características:

- Transmisión poco frecuente de pequeñas o grandes cantidades de datos (por ejemplo, aplicaciones interactivas)
- Transmisión intermitente de tráfico de datos bursty (por ejemplo, aplicaciones en las que el tiempo medio entre dos transacciones consecutivas es de duración superior a la duración media de una única transacción.

Como por ejemplo:

- **RTI** (Road Traffic Informatics)
- Telemetría
- Tele alarma
- Control del tráfico ferroviario
- Acceso a internet usando la WWW (World Wide Web)

Desde el punto de vista físico los recursos pueden ser reutilizados y existen algunos puntos comunes en la señalización, así en el mismo portador radio pueden coexistir simultáneamente tanto los time slots reservados a la conmutación del circuito, como los time slots reservados al uso del GPRS. La optimización en el empleo de los recursos se obtiene a través de la repartición dinámica de los canales reservados a la conmutación del circuito y de aquellos reservados al GPRS. Cuando se presenta una llamada de voz hay tiempo suficiente para liberar los recursos usados por el GPRS, de tal forma que la llamada por conmutación de circuito a mayor prioridad, pueda ser efectuada sin problemas.

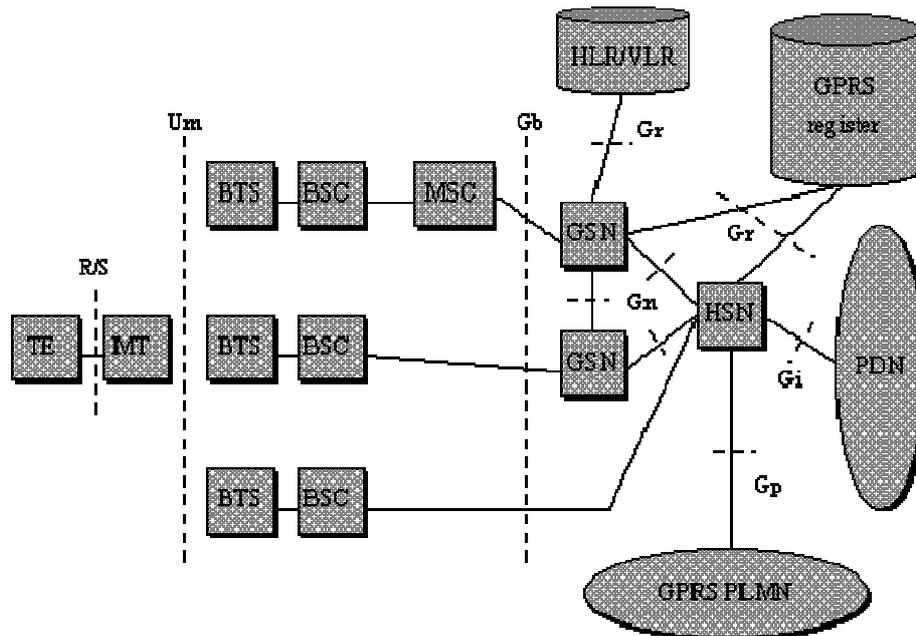


Figura 3.2. Arquitectura GPRS

El nodo de soporte **GSN** (Gateway Support Node) del GPRS es el elemento principal de la infraestructura. Este router puede proporcionar la conexión y el inter-trabajo con otras redes de datos, de administrar la movilidad de los usuarios a través de los registros del GPRS y es capaz de entregar los paquetes de datos a las estaciones móviles, independientemente de su posición. Físicamente el GSN puede estar integrado en el **MSC** (Mobile Switching Center) o puede ser un elemento separado de la red, basando en la arquitectura de los routers de las redes de datos. Los paquetes de datos del usuario pasan directamente entre el GSN y el **BSS** (Base Station Subsystem), gracias a la señalización que acontece entre GSN y el MSC.

### 3.4. Arquitectura GPRS:

Para la realización de un servicio de datos por paquetes en la red celular GSM se pueden seguir dos inicializaciones diferentes:

- Inicialización de sistema separado
- Inicialización de sistema integrado

La primera inicialización prevé que toda la infraestructura necesaria para el soporte del servicio sea añadida a la de la red GSM, mientras que la segunda prevé el añadido de la funcionalidad necesaria para el soporte del GPRS a las entidades que componen la infraestructura de la red GSM. En realidad, también la inicialización de sistema integrado requiere la introducción de nuevas entidades, garantizando de todos modos, desde el punto de vista económico, un impacto menos vistoso sobre los costes necesarios para la implementación del servicio.

Las entidades que tienen que ser añadidas, desde el punto de vista de la integración del servicio GPRS en la red GSM, son:

- **GSN** (*Gateway Support Node*), que constituyen los nodos de soporte del servicio GPRS.
- **GPRS register**

Los nodos GSN pueden verse como entidades en las que está localizada gran parte de las funciones necesarias para soportar el GPRS. En el **GPRS PLMN** (*Public Land Mobile Network*), generalmente hay más nodos GSN y la infraestructura que los conecta, denominada backbone network (ruta de enlace), permite el routing de los paquetes transmitidos por los usuarios de la red o dirigidos a éstos. En relación con la localización de la estación móvil genérica GPRS, se usan los **HSN** (*Home Support Node*) y el **VSN** (*Visited Support Node*). El HSN es el nodo de la backbone network al que llegan los paquetes dirigidos al móvil en base al valor de su dirección de la red; además, cuando el móvil es localizado en el área gestionada por otro nodo de la ruta de enlace, el HSN vuelve a mandar hacia ese nodo los paquetes destinados al móvil.

EL **VSN** es el nodo de la backbone network en cuya área se encuentra normalmente el móvil.

La backbone network puede ser una red pública de datos de paquetes, lo que permite limitar los costes de realización, o bien una red de datos de paquetes dedicada y contruida ad hoc y, por lo tanto, optimizada para el soporte del servicio. La primera solución determina, con respecto a la segunda, mayores retrasos de transmisión cuando los paquetes se intercambian entre usuarios de la GPRS PLMN y usuarios de otra red, mientras que la segunda presenta unos costes de realización más elevados.

A la backbone network también están conectadas las entidades de inter-trabajo, que garantizan la interconexión de la GPRS PLMN a otras redes de datos como, por ejemplo, la red Internet, las redes PSPDN (*Public Switched Packet Data Network*), las redes privadas de paquetes y otras.

Las principales funciones desempeñadas por estas entidades son: la conversión de los protocolos y el mapeo de las direcciones de red de las entidades envueltas en la comunicación de datos. Otra nueva entidad necesaria para el soporte del servicio es el GPRS register, que no tiene que verse necesariamente como una nueva entidad física, en cuanto que se puede pensar en ampliar el conjunto de las funciones de los VLR/HLR de la red GSM.

Las funciones llevadas a cabo por un GPRS register son esencialmente las de memorizar informaciones relativas al servicio GPRS; en particular cada GPRS register contiene:

- Información necesaria para el routing de los paquetes dirigidos a un móvil GPRS; por ejemplo, la dirección de red del móvil para un determinado protocolo de red y el tipo de protocolo de red a cuya dirección se refiere.
- Información relativa al perfil de suscripción del abonado; por ejemplo, informaciones características de la calidad del servicio solicitada por el usuario (**QOS=Quality Of Service**).

La llave de acceso a estas informaciones relativas al abonado genérico GPRS es el **IMSI** (*International Mobile Subscriber Identity*). La introducción de nuevas entidades a la red GSM lleva a la definición de nuevos interfaces; entre éstas, la Gr soporta sólo señalización, mientras que todas las demás soportan tanto señalización como datos.

### 3.5. Tipología del servicio GPRS.

#### 3.5.1. Tipología Punto a Punto (PTP):

la estación móvil, en el marco de una comunicación con la tipología PTP, puede adoptar uno de las siguientes modalidades operativas:

- **Reposo:** esta modalidad coincide con la modalidad Reposo (Idle) definida por el GSM y ve el móvil sintonizado con los canales comunes de control.
- **Espera:** es una modalidad intermedia entre la Reposo y la modalidad activa y se caracteriza por el hecho de que el móvil no utiliza recursos físicos.
- **Activa:** en esta modalidad el móvil envía y/o recibe paquetes de datos de la red.

Las transiciones de una modalidad a otra son dirigidas por la modalidad en uso y por las funciones, que en tal estado, se utilizan. En el siguiente gráfico se muestra un diagrama de estado que resume las posibles transiciones en función de cuáles son las funciones necesarias para gestionar el **servicio PTP**.

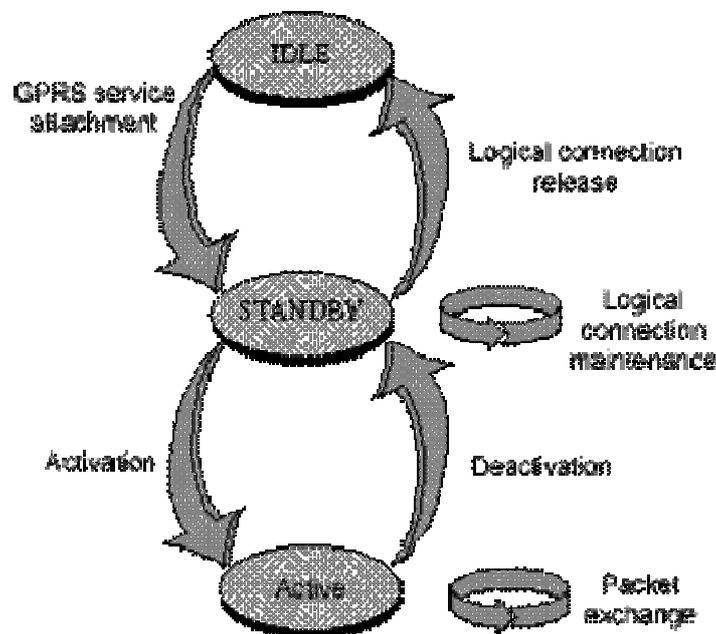


Figura 3.3. Diagrama de estado de las transiciones

La transición de modalidad Reposo a modalidad de Espera prevé el uso de las siguientes funciones:

- La **MS GPRS** inicia el acceso al servicio (para solicitar una conexión o bien para responder a un paging (solicitud) por parte de la red)
- Verificación de la **MS GPRS**
- Establecimiento de una conexión virtual punto a punto entre llamante y llamado (esta función se realiza sólo cuando se solicita un servicio **PTP CONS**)

La transición inversa, de Espera a Reposo, prevé que, tanto la MS como el nodo GSN a la que está conectada cancelen las informaciones que caracterizan la conexión lógica MS/GSN y que se denominarán sintéticamente en el curso del contexto de la conexión.

En **Espera** se realizan las siguientes funciones:

- Actualización del contexto memorizado del **HSN** (Home GPRS Support Node)
- Gestión de la conexión lógica entre móvil y **VSN** (Visited GPRS Support Node)
- Gestión de la movilidad del usuario **GPRS**

La transición entre modalidad de **Espera** y modalidad **Activa** prevé que se actualice, en el contexto de la conexión, la variable que describe el estado de la estación móvil; esta operación da lugar, como se verá luego, al inicio de un oportuno reloj que tiene como misión desactivar automáticamente la conexión nada más que finaliza un período de tiempo preestablecido, cuya duración, de todos modos, puede ser establecida entre el usuario y la red.

En la modalidad Activa se realizan las siguientes funciones:

- Recolecta de informaciones necesarias para la tarificación de la llamada
- Encapsulamiento de los **PDU** (Protocol Data Unit) en frame GPRS (función G-Relay)
- Routing de los paquetes de información (función **PRT**, Packet Routing and Transfer)
- Transmisión de los paquetes por el interfaz radio

A continuación se ofrece una descripción de las principales funciones cuyo uso determina las transición de modalidad especificadas antes.

### **Transición Reposo/Espera**

Esta transición, como ya se ha señalado, puede ser originada por una solicitud de servicio por parte del móvil, o bien por efecto de una llamada de datos dirigida al móvil.

En el segundo caso, cuando la **HSN** recibe un **PDU**, se requiere al GPRS register en modo tal que de la dirección de destino del **PDU** se pueda retornar al IMSI (International Mobile Subscriber Identity) correspondiente; a continuación el **HSN** envía a un Gateway MSC un mensaje MAP, conteniendo como parámetro el IMSI, con

el que solicita la interrogación del HLR (Home Location Register) para la recuperación del número de itinerancia del móvil.

Determinado el VMSC (Visited Mobile Switching Center) en cuya área se encuentra el móvil en modalidad Idle, el Gateway MSC envía a ese VMSC un mensaje MAP con el IMSI del móvil; recibido el mensaje MAP, el VMSC inicia el procedimiento de pagina (búsqueda) del móvil. El móvil accede al RACCH (Random Access Channel) para responder al pagina y, recibida en el canal AGCH (Access Grant Channel) la información relativa al canal SDCCCH (Stand alone Dedicated Control Channel) asignado por la BSS (Base Station Sub-System), envía una solicitud de acceso al servicio GPRS.

La verificación del usuario se realiza basándose en los parámetros memorizados por el VLR, así como la ejecución del procedimiento de encriptación de las informaciones de usuario.

La función de encriptación es una función de nivel uno para los servicios de conmutación de circuito en el GSM, mientras que para el GPRS tal función pertenece a un nivel más elevado. Como a continuación se verá, para transferir los **PDU** del nodo GSN al móvil se recurre al encapsulamiento de los paquetes en tramos GPRS, cada uno de los cuales está constituido por una cabecera o etiqueta (header) y por un SDU (Service Data Unit) que no es más que el **PDU** oportunamente cifrado, si ha sido seleccionado el modo de transmisión cifrado.

El modo de transmisión cifrada, en la perspectiva GPRS, considera por tanto el intercambio de paquetes de contenido informativo cifrado entre nodo GSN y móvil, mientras que en los servicios GSM la encriptación sólo se utiliza en el interfaz radio. De todas formas, el control y la solicitud del procedimiento de encriptación son siempre funciones de la red.

Una ulterior función del nodo GSN es el de asignar y comunicar a las estaciones móviles un identificador lógico que lo distinga dentro del área de localización (location area) gestionada por el nodo; este identificador se llama **TLLI** (Temporary Logical Link Identifier) y se utiliza para diferenciar la conexión de nivel dos entre el nodo y la estación móvil (el TLLI, de hecho, es uno de los parámetros del header de los tramos de nivel dos y en base a esto, el móvil es capaz de recibir paquetes dirigidos a aquel).

Sobre la base de estas informaciones, tanto el nodo visitado como el móvil memorizan el contexto de la conexión, de la que ofrecemos un cuadro resumen:

- Contexto memorizado por la **VSN**
- Identidad del usuario móvil (**IMSI**)
- Estado del móvil (**Espera/Activo**)
- Dirección de la backbone network del **HSN**
- **TLLI**
- Área de Encaminamiento (Routing Area, **RA**) en modalidad **Espera**
- Identificador de la casilla en que se encuentra el móvil y el canal GPRS utilizado en modalidad **Activa**

Contexto memorizado por el móvil GPRS:

- Estado actual del móvil (**Espera/Activo**)
- Tipo de **PDP** (Packet Data Protocol), dirección PDP (una pareja para cada protocolo gestionado)
- Indicador (Flag) de habilitación del algoritmo de compresión de datos
- **TLLI**
- Routing Area (**RA**) actual (en modalidad Espera)
- Identificador de la casilla en que se encuentra el móvil y el canal GPRS utilizado en modalidad **Activa**

Entre las informaciones destacables de la modalidad Espera destaca la de Routing Area (RA). El concepto de routing area es un concepto nuevo y está estrechamente relacionado con el GPRS y en particular a la necesidad de efectuar el paging del móvil GPRS en un tiempo más breve que el característico del GSM: para reducir este tiempo es necesario enviar los mensajes de paging por los canales físicos que gestionan el tráfico de paquetes y, en esta perspectiva, con el fin de reducir la extensión del área en que se transmiten los mensajes de paging. Con este fin, se vuelve útil la subdivisión en subáreas de la clásica location area, cada una de las cuales adopta el nombre de routing area.

Por tanto, la red conoce la posición de la estación móvil en Espera con una resolución limitada a una routing area; tal conocimiento presupone que entre las funciones desempeñadas por un móvil en Espera se encuentra la de actualización de la posición que se solicita cada vez que se produce un cambio de una modalidad a otra.

### La modalidad Espera

La primera función utilizada en cuanto una estación móvil GPRS entra en Espera es la de actualización del contexto memorizado bajo el **HSN** y en particular por la dirección de red vertebral del nodo VSN en cuya área se encuentra la estación móvil.

De hecho, esta información es básica para poder impulsar de nuevo los paquetes dirigidos al móvil, los cuales, en base a una valoración de su dirección PDP, son dirigidos en una primera fase hacia el **HSN**. El VSN notifica al **HSN**, mediante un oportuno mensaje MAP, su dirección de red vertebral, pero antes de realizar esta asignación determina el **HSN** al que está asociado el móvil que ha solicitado el procedimiento de conexión al servicio GPRS. Para realizar esto se pueden seguir distintos procedimientos. Una primera alternativa es la de determinar el **HSN** en modo algorítmico teniendo como acceso a la identidad del usuario móvil (**IMSI**); otro método podría ser el de memorizar en la tarjeta SIM del usuario móvil la identidad del **HSN**.

Una tercera solución podría ser la de memorizar en el GPRS register la identidad del **HSN** del que depende el móvil, poniéndola en relación con el IMSI. Determinada la identidad de la **HSN**, el VSN le envía un mensaje MAP con los parámetros de su dirección de red vertebral y el IMSI del usuario móvil GPRS que ha realizado la conexión al servicio.

Recibido ese mensaje, el **HSN** actualiza el contexto relativo a la estación móvil sobre la base de las informaciones en él contenidas; las informaciones memorizadas son en síntesis las siguientes:

- Contexto memorizado por el **HSN**.
- Identidad del usuario móvil (**IMSI**).
- Información necesaria para la tarificación.
- Tipo de PDP, dirección **PDP** (una pareja por cada PDP gestionado por el móvil).
- Clave de encriptación.
- Indicador de habilitación del algoritmo de compresión.
- Dirección de la red vertebral del **VSN**.

La función de gestión de la conexión lógica entre la estación móvil y el nodo VSN permite establecer, mantener y suprimir la conexión lógica, identificada por el TLLI, que se establece durante la fase de conexión al servicio y que permite la transmisión a nivel dos de los tramos entre **MS** y **VSN**.

La función de gestión de la movilidad de la estación que se encuentra en **Espera** permite la actualización de las informaciones necesarias para su localización.

La estación móvil utiliza este procedimiento una vez que es reconocida, entre la informaciones transmitidas en producción en la casilla, un identificador de RA diferente del memorizado por el móvil.

Se pueden verificar dos situaciones:

- Pasaje a una RA del mismo **GSN**
- Pasaje a una RA de otro **GSN**

En el primer caso es necesario actualizar el identificador de RA memorizado en el contexto del único VSN, operación que se efectúa mediante un mensaje de RA update (con el **TLLI** y los identificadores de la vieja y la nueva RA) predefinido en el protocolo GPRS. El VSN responde con un mensaje en el que se notifica el resultado de la operación solicitada.

El segundo caso es un poco más complicado en cuanto debe ser actualizado también el contexto del **HSN**, visto que existe el cambio a un área de competencia de un nuevo GSN. La estación móvil envía al nuevo GSN el mismo mensaje de RA update, descrito en el anterior caso, y este último, descubriendo que tal usuario proviene de un área bajo control de otro GSN, solicita al viejo GSN (mediante un mensaje MAP) el IMSI del usuario al que está asociado el TLLI contenido en la solicitud de RA update. Conocido el IMSI, el nuevo GSN es capaz de determinar el **HSN** del que depende el móvil y de enviarle un mensaje MAP (con el IMSI y su dirección de red vertebral), como señal de actualización del contexto relativo al usuario.

El **HSN** actualiza el contexto modificando la dirección de red del nodo visitado por el usuario y envía al viejo nodo VSN el comando de cancelación del contexto del usuario memorizado por él.

Otra función desempeñada por el nodo GSN es la de gestionar los relojes que indican la finalización del intervalo de tiempo en que el usuario se puede encontrar en Espera.

### **Transición Espera/Activo**

Esta transición puede producirse por uno de estos dos motivos:

- El nodo VSN recibe paquetes dirigidos al usuario móvil
- El usuario móvil debe enviar paquetes al destinatario

En el primer caso se transmite un mensaje de paging (con la TLLI) en la RA cuyo identificador esta memorizado en el contexto del usuario dentro del VSN;

La estación móvil responde al mensaje notificando al VSN el canal GPRS en que está sintonizada y este último utiliza esta información para actualizar el contexto del usuario yendo, además, a variar de Espera a Activo el identificador de estado operativo de la estación móvil.

En el segundo caso es la estación móvil la que notifica al VSN el cambio de estado operativo mediante un mensaje predefinido por el protocolo GPRS (en el que se suministra, además del TLLI, también el canal GPRS al que está sintonizada la estación).

### ***La modalidad Activa***

En esta modalidad la estación móvil intercambia paquetes con el VSN, por lo que viene activada la función de tarificación, que permite, en función del recuento de los bytes transferidos, a los parámetros de QOS requeridos y al tipo de servicio solicitado para tarificar la llamada GPRS (en un servicio **CONS**, al contrario de un servicio **CLNS**, la tarificación se realiza también por la base horaria desde el momento en que se utilizan por toda la duración de la comunicación de los recursos lógicos, como los circuitos virtuales).

Los **PDU** son transferidos entre el VSN y el TE después de ser encapsulados en tramos GPRS, mediante la función G-Relay.

Los **PDU** son llevados por el SAP (Service Access Point) relativo al protocolo al que se refieren, y son elaborados mediante la función G-Relay que, en base al header del **PDU** recibido, genera un header GPRS y por tanto un paquete GPRS (en donde el SDU es el **PDU** recibido por el SAP).

Los **PDU**, antes de ser incluidos como **SDU** en los paquetes GPRS, son eventualmente comprimidos y cifrados; los paquetes GPRS obtenidos así son gestionados por la función **PRT** (Packet Routing and Transfer), que es esencialmente una función de nivel dos.

La función **G-Relay** prevé también la eventual segmentación de los **PDU** que tienen dimensiones superiores al espacio previsto por el campo de datos dentro de una trama GPRS.

Por la otra vertiente de la conexión lógica **VSN/MS** se toman los paquetes GPRS del **PRT** SAP y se realiza una eventual descompresión y descriptación de los **SDU** y, sucesivamente, son transferidos al **SAP** oportuno.

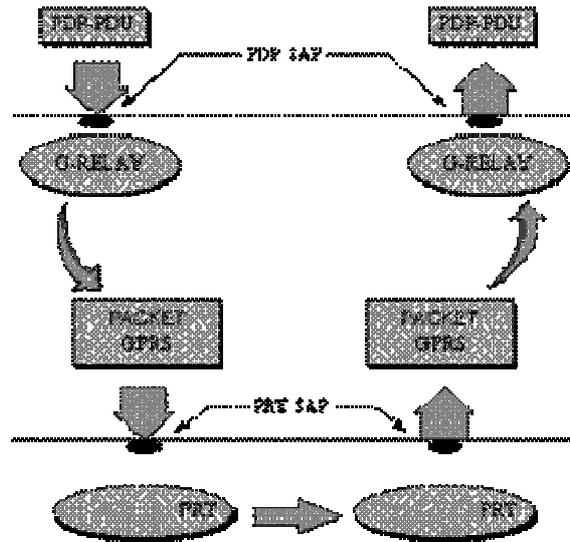


Figura 3.4. Modalidad Activa

Cuando la estación móvil descubre que la calidad de señal es mejor en otro canal GPRS, envía al nodo VSN un mensaje predefinido del protocolo GPRS con el solicita la actualización del contexto memorizado en él, relativo a la indicación de canal GPRS en el que está sintonizado el móvil. Tal mensaje contiene el TLLI y la indicación del nuevo canal GPRS en que está sintonizada la estación móvil.

La actualización puede estar determinada por el traslado del móvil entre las casillas de áreas controladas por nodos GSN distintos, o bien de la misma área.

En el segundo caso basta con actualizar el contexto del usuario bajo el VSN, además de lo memorizado en el móvil.

En el primer caso, por el contrario, el nuevo VSN, descubierta que la solicitud de update proviene de un usuario que ha efectuado la conexión al servicio en otro nodo GSN, inicia una serie de indicaciones totalmente idéntica a la descrita en el caso de Inter. GSN RA UPDATE, con la única diferencia que en este caso lo que debe ser actualizado es el identificador de la casilla en la que se encuentra el móvil y la indicación del canal GPRS al que está sincronizado.

Una función ulterior que se desempeña en la modalidad Activa es la función de **PTA** (Packet Terminal Adaption), mediante la cual es posible adaptar los paquetes generados por el TE, con el fin de poderlos transmitir por la **GPRS PLMN** (Public Land Mobile Network), y los paquetes dirigidos al TE. Tal función es una función de nivel uno y se realiza por el MT. Con este fin, para permitir, mediante el interfaz radio, el acceso a la GPRS PLMN a diferentes terminales, es verosímil que se desarrollen diversos MT, cada uno de los cuales dote de un acceso al TE mediante un interfaz estándar.

Por tanto, estarán disponibles MT que garanticen a terminales asincrónicas el acceso a la GPRS PLMN, mediante un interfaz serial asincrónico estándar y un PAD (por vez primera el PAD no estará lejano del terminal de datos asincrónico), y serán disponibles MT que garanticen el acceso a terminales sincrónicos a través de interfaz

serial sincrónico. Este hecho constituye una gran innovación relativa al acceso al mundo del paquete de informaciones, que actualmente prevé el acceso a un PAD remoto (respecto al usuario final) mediante una conexión a conmutación de circuito, con todas las desventajas que de esto se derivan, sobre todo para las aplicaciones de usuario de tipo bursty.

Resumiendo, por lo que respecta a las conexiones de datos a conmutación de paquete, el servicio GPRS del GSM será el primero que garantice una conexión a paquete end-to-end entre las dos (o más de dos en el caso de servicios PTM) entidades implicadas en la comunicación.

No se excluye la posibilidad de que las funcionalidades de la MT sean integradas en el terminal TE, utilizando interfaces API (Application Program Interface) para permitir al usuario desarrollar sus aplicaciones.

### **3.5.2. Tipología Punto Multipunto (PTM):**

Esta tipología de servicio permite distribuir las informaciones basándose en la ubicación de los usuarios a donde están direccionados y no sobre la base de la que es la identidad del usuario individual destinatario, como sucede en los servicios punto-punto.

Entre los servicios punto-multipunto se distinguen los servicios **broadcast** y los servicios **multicast**. Los primeros prevén la ramificación de los mensajes en toda el área especificada por su emisor sin que estos manifiesten ninguna otra forma de direccionamiento, mientras que los segundos prevén la ramificación de los mensajes en todo el área especificada por su emisor con la diferencia, respecto a los servicios de producción, que estos están direccionados únicamente a un conjunto de usuarios que se encuentren en ese momento en el área.

- El direccionamiento del área geográfica a la que tienen que ser transmitidos los mensajes
- El direccionamiento del grupo de usuarios a los que están destinados los mensajes.

Para los servicios **broadcast** se aprovecha únicamente el direccionamiento geográfico, en el momento en que los mensajes se direccionan, en esta tipología de servicios, a todos los usuarios pertenecientes al área especificada.

Para los usuarios **multicast**, por el contrario, se aprovechan ambas formas de direccionamiento, siendo los usuarios destinatarios de los mensajes un conjunto de todos los usuarios localizados en ese momento en el área especificada.

Una ulterior característica distintiva de las dos tipologías de servicios es la de la comprobación (acknowledgement) a la recepción del mensaje; de hecho, en general,

los servicios de broadcast son sin comprobación, mientras que los multicast prevén que los destinatarios de los mensajes respondan al emisor con mensajes de verificación.

Los **GPRS MS** pueden ser subdivididas en tres clases en base a su capacidad de recibir los mensajes **PTM** (Punto Multipunto) a él dirigidos mientras están activos:

- **Clase A:** los **GPRS MS** pertenecientes a esta clase poseen la capacidad de recibir mensajes **PTM** a él dirigidos incluso si ya están ocupados en una llamada GSM, sin que tal llamada experimente merma alguna.
- **Clase B:** los **GPRS MS** pertenecientes a esta clase no pueden recibir mensajes **PTM** sin que los servicios GSM que están usando sufran merma (por ejemplo el throughput)
- **Clase C:** los **GPRS MS** de esta clase no pueden recibir de ninguna manera **PTM** cuando están utilizando servicios GSM

La información relativa a la clase de pertenencia de la GPRS MS constituye parte integrante del perfil de inscripción del abonado (se trata por tanto de un fragmento informativo necesario para la caracterización del perfil de usuario y no para la del aparato utilizado).

Normalmente el usuario que solicita un servicio punto-multipunto es un usuario de red fija, incluso si no se excluye la posibilidad de que al realizar una solicitud de servicio sea usuario de la **GPRS PLMN**. El acceso a la red **GPRS** por parte de usuarios de red externa se produce a través de un interfaz estandarizado Gi de nivel uno al siete mediante el cual comunican las redes GPRS y los centros de servicios gestionados por el proveedor de servicios (o por el operador de red).

La definición de un interfaz estándar permite a redes distintas disponer de un único interfaz común para acceder (a través de un gateway GSN) a la red GPRS. Por tanto, la misión de los centros de servicios es poner a disposición de los usuarios externos a la **GPRS PLMN** los interfaces oportunos.

Según las solicitudes de servicio **PTP** los centros de servicios efectúan operaciones de bajo nivel, en particular prosiguen un simple encapsulamiento de los paquetes de nivel tres que le vienen de las redes interconectadas.

En oposición, los paquetes originados por solicitudes de servicio **PTM** son elaborados mediante los fundamentos de servicio definidos en los estratos aplicativos de la pila de protocolos que caracteriza al interfaz estándar.

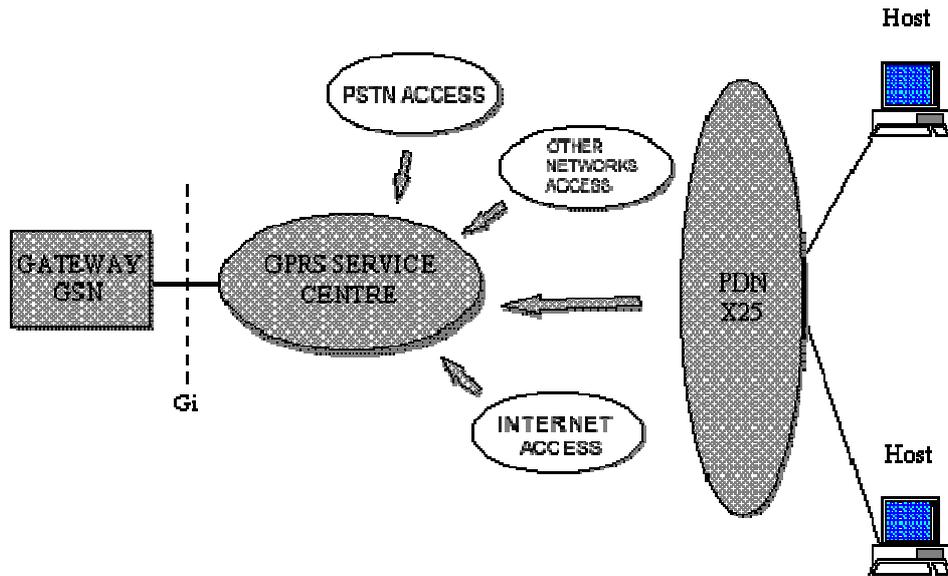


Figura 3.5. Centro de servicio GPRS

El interfaz estándar entre **GPRS PLMN** y centros de servicio estipula los siguientes tres fundamentos de servicio:

- **PTM Data Transfer**
- **PTM Kill** (Data Transfer)
- **PTM Status** (Data Transfer)

El fundamento **PTM Data Transfer** es utilizada por el usuario que necesita el servicio **PTM** con el fin de instaurar una nueva transacción **PTM** en la GPRS PLMN. Este fundamento utiliza los siguientes parámetros:

- Identificador del solicitante del servicio (service requester)
- Identificador del grupo de usuarios destinatarios de los mensajes PTM
- Descriptor del área geográfica interesada en la transacción PTM
- Información de programación temporal (scheduling)
- Parámetros de **QOS**
- Mensajes por distribuir
- Identificador de la transacción bajo el emisor de la misma

La red responde a una solicitud de por parte de un service requester comunicando a este último si el perfil del servicio solicitado (caracterizado por las informaciones de scheduling y los parámetros de **QOS** especificados en la **PTM Data Transfer**) es compatible con la disponibilidad de los recursos necesarios y con el perfil de suscripción memorizado en el GPRS register y acordado entre el usuario y el proveedor de servicio, en el momento de la suscripción del abono del servicio PTM.

Además, la red asocia a cada transacción **PTM** un identificador. La respuesta de la red a un **PTM Data Request** contiene los siguientes parámetros:

- Identificador de la transacción asignado por la red
- Identificador de la transacción bajo el emisor de la misma
- Resultado de la operación solicitada por el service requester
- Información de scheduling
- Parámetros de **QOS**

La red retorna al identificador de la transacción sólo cuando esta ha sido aceptada, mientras que en caso contrario retorna la respuesta negativa de la operación. En el caso de que el perfil de servicio requerido por el service requester no pueda ser garantizado, la red comunica al service requester aquello que puede garantizar (por tanto, la respuesta de la red a una **PTM Data Request** contendrá sólo en este caso las informaciones de scheduling y los parámetros de **QOS**).

El fundamento **PTM Kill** (Data Transfer) permite abortar la transacción antes del término previsto por la información de scheduling. Cuando el service requester solicita este fundamento los parámetros que se facilitan son los siguientes:

- Identificador del service requester
- Identificador del sistema de la transacción

La red responde a la solicitud del service requester con un mensaje que indica si la solicitud ha sido aceptada o no.

El fundamento **PTM Status** (Data Transfer) permite al service requester obtener informaciones relativas al estado en que se encuentra la transacción. Tales informaciones pueden ser dadas a conocer al service requester bajo su explícita solicitud o bien por indicación de la red. En cualquier caso, lo que el service requester recibe de la red es una pareja de parámetros:

- El identificador del sistema de la transacción
- El estado de la transacción

Estas tres operaciones definen, como hemos visto, una serie de parámetros de los que se facilita a continuación una breve descripción. El identificador del service requester es necesario para poderles adeudar el costo de la transacción requerida. El identificador del grupo de usuarios destinatarios de los datos incluidos en la transacción **PTM** es un parámetro que se utiliza para direccionar una transacción de tipo multicast y se compone de dos partes: una la asigna el operador de red y una parte significativa sólo en el ámbito de un área de cobertura del servicio.

La descripción del área geográfica es una descripción a alto nivel que se abstrae totalmente de lo que es la distribución de las casillas de cobertura del servicio; este tipo de descripción del área en que se debe transmitir el mensaje **PTM** permite evitar que los usuarios conozcan las informaciones relativas a la infraestructura de red. Por otra parte este método de direccionamiento comporta que los nodos de la **GPRS**

**PLN (GSN)** tramiten la función de conversión de tal descripción a alto nivel en términos de direccionamiento de las casillas de cobertura.

El sistema de referencia usado en la codificación de la descripción geográfica es el World Geodetic System 1984 (**WGS 84**), el mismo que se utiliza en el sistema **GPS** (Global Positioning System). Tal sistema de referencia tiene origen en el centro del elipsoide WGS (obtenido de la rotación de una elipse en torno a su eje menor orientado en dirección norte-sur; la dimensión del eje menor es de 6.356.752.314 metros, mientras que las dimensiones de su eje mayor son de 6.378.137 metros).

Las coordenadas de un punto sobre la superficie de este elipsoide son indicadas en términos de latitud y longitud respecto al ecuador y el meridiano principal.

El área geográfica en la que el mensaje **PTM** debe ser transmitido puede ser descrita por el service requester mediante una serie de figuras geométricas predefinidas y la precisión con que se describe la posición de un punto genérico es de un metro al ecuador (lo que conlleva que los ángulo de latitud y longitud estén codificados con 26 bit).

Las informaciones de scheduling están compuestas por la frecuencia de repetición de los mensajes y por los instantes en que se debe iniciar y parar la transmisión del mensaje **PTM**. Si los valores de estos parámetros son contemporáneamente nulos, la transacción **PTM** a la que se refieren es una transacción que prevé la transmisión sin repetición del mensaje.

Los datos del usuario constituyen el mensaje **PTM** que debe ser reenviado a los usuarios indicados por el identificador de grupo válido en el área geográfica descrita. Al contrario que en el servicio SMS CB (Short Message Service Cell Broadcast) no se establece un límite a las dimensiones del mensaje.

El identificador de transición bajo el service requester es necesario cuando este solicita a la red distintas transacciones **PTM**; en base a este identificador, de hecho, el service requester es capaz de distinguir los mensajes relativos a las transacciones requeridas que la red le devuelve.

El identificador de transacción asignado por la red, por el contrario, permite individuar unívocamente la transacción con todos los parámetros a ella asociados. En definitiva, se prevén dos campos informativos en donde se codifican respectivamente: el resultado de una operación **PTM** y el estado de la transacción **PTM**.

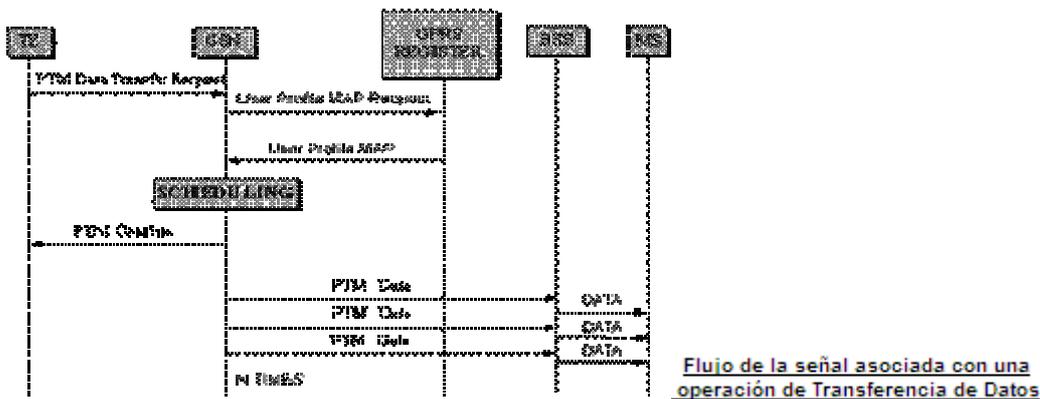
En general una transacción **PTM** implica un área más grande que la gestionada por el GSN al que está dirigida la transacción, por lo que se habla de split **PTM** transaction, esto es, de una transacción que involucra a diferentes nodos GSN. El nodo al que llega la solicitud de transacción **PTM** se denomina anchor GSN y el que gestiona la solicitud de servicio del service requester.

De hecho, cuando un nodo GSN recibe una solicitud de transacción **PTM** se activa una secuencia de operaciones que pueden ser resumidas en cinco puntos:

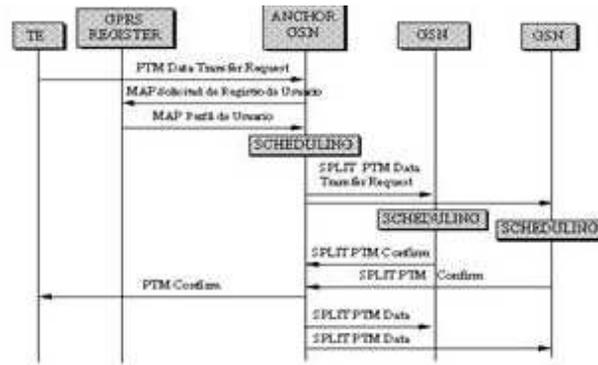
- Paso 1: validación del perfil de servicio requerido por el service requester
- Paso 2: Encaminamiento (routing) geográfico
- Paso 3: activación del algoritmo de scheduling
- Paso 4: señalización de transacción **PTM** a los GSN involucrados en la solicitud
- Paso 5: señalización al service requester de confirmación de la transacción PTM

Las Slip **PTM** Data Transfer Request presentan, además de los parámetros un a **PTM** Data Transfer Request normal, también el identificador del anchor GSN y el identificador de transacción asignado por el anchor GSN en el paso 3. Los nodos GSN que reciben tal solicitud del anchor GSN, después de haber realizado la lista de las casillas en la que se debe transmitir el mensaje y después de haber lanzado el algoritmo de scheduling, responden al anchor GSN comunicando el resultado de la solicitud y, al mismo tiempo, memorizando los parámetros característicos de la transacción (el identificador de transacción asignado por el anchor GSN, la dirección de red del anchor GSN, la dirección del service requester, la lista de casillas en las que se transmite el mensaje, una copia del mensaje a transmitir).

Si el anchor GSN recibe de todos los nodos GSN contactados por las confirmaciones que son satisfactorias desde el punto de vista de los parámetros de **QOS** ofrecidos, se memoriza el set de parámetros que caracterizan la transacción y se envía al service requester la **PTM** Data Transfer Confirm que lo advierte de la aceptación de su solicitud. Si, por el contrario, algún GSN responde negativamente a la solicitud avanzada por el anchor GSN, este último realiza una **PTM** Kill Request anulando la transacción bajo los GSN que la han aceptado; sucesivamente, el anchor GSN responde negativamente al service request



Flujo de la señal asociada con una SPLIT PTM DATA TRANSFER REQUEST\*



### 3.6. Protocolo GPRS:

El protocolo **GPRS** es un protocolo de nivel tres, transparente para todas las entidades de red comprendidas entre el terminal móvil **MT** y el nodo **GSN** al que el móvil está, lógicamente, conectado; las entidades entre las que se establece una conexión a este nivel están, de hecho, localizadas en el terminal móvil **MT** y en el nodo **GSN**. Este protocolo soporta tanto el intercambio de informaciones de control como de paquetes **PDP-PDU** (Packet Data Protocol - Protocol Data Unit) entre el móvil y el nodo al que éste está conectado (los PDP-PDU son, de hecho, encapsulados en las tramas GPRS).

El formato de una trama GPRS prevé los siguientes campos:

- Identificador del protocolo GPRS
- Identificador del protocolo de los **PDU** (identificador de **PDP**)
- Mensaje GPRS

El identificador del protocolo **GPRS** es una información numérica cuyo objetivo es el de distinguir los burst que contienen paquetes GPRS, de los burst que contienen informaciones GSM.

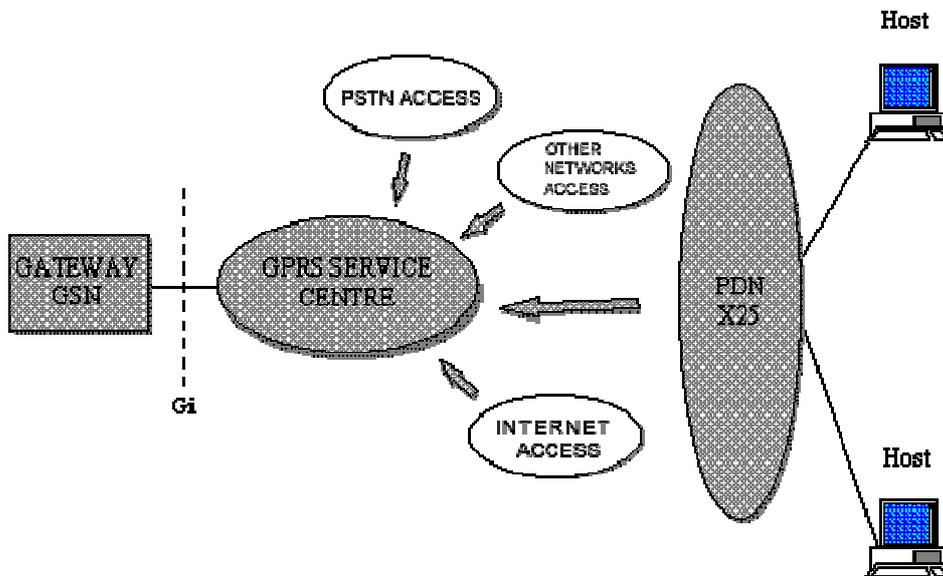
El identificador del protocolo de los **PDU** encapsulados en las tramas GPRS es necesario para direccionar éstos, en cuanto son desencapsulados, hacia el correcto **SAP** (Service Access Point); también esta información es de tipo numérico. Se tendrá, por tanto, un valor que define los paquetes **X25**, uno que define los paquetes **IP** (Internet Protocol), uno que define los paquetes **CLNP** (Connectionless Network Protocol) y así sucesivamente. Además, dicha información permite la interpretación del GPRS contenido en la trama GPRS; de hecho, como ya se anticipaba, las tramas GPRS son utilizadas tanto para el transporte de mensaje de control como para el transporte de paquetes de datos, por lo tanto, se hace necesario el uso de un indicador que permita distinguir a cuál de las dos categorías posibles pertenece el mensaje GPRS. Los mensajes GPRS de control son definidos por un valor preestablecido del identificador de **PDP**.

Algunos de los posibles mensajes de control se enumeran a continuación:

- Petición de log-on (**LOG-ON REQUEST**)
- Respuesta a una petición de log-on (**LOG-ON RESPONSE**)
- Activación del modo de transmisión cifrado (**SET GPRS CIPHERING MODE**)
- Petición de actualización de las informaciones de routing (**ROUTING UPDATE REQUEST**)
- Respuesta a una petición de actualización de las informaciones de routing (**ROUTING UPDATE RESPONSE**)
- Petición de actualización del indicador de routing area (área de encaminamiento) (**GPRS RA UPDATE REQUEST**)
- Respuesta a una petición de actualización del indicador de routing area (GPRS RA UPDATE RESPONSE)

El nodo **GSN**, antes de encaminar en la backbone network los **PDU** de nivel tres desencapsulador de las tramas GPRS recibidos a través del interfaz Gb, los encapsula (como SDU - Service Data Unit) en PDU del protocolo de red utilizado en la backbone network para el transporte de paquetes del usuario.

Obviamente, realiza la operación inversa para los paquetes dirigidos al usuario móvil.



### 3.7. Calidad del servicio GPRS:

Cuando se habla de calidad de servicio, QOS, sólo se hace referencia a un conjunto de parámetros de prestaciones que pueden ser observados directamente y medidos desde el punto de acceso al servicio utilizado por el usuario.

Los criterios que se usan para valorar la calidad de una prestación de servicio son principalmente:

- Velocidad
- Cuidado
- Fiabilidad

La velocidad con que es servida una petición de servicio puede ser valorado en términos de bit-rate con que las informaciones son transportadas o bien en términos de intervalo de tiempo para terminar la petición de servicio.

El cuidado se refiere, sin embargo, al grado de corrección con el que se atiende una petición de servicio.

La fiabilidad del servicio sintetiza la disponibilidad del servicio sin tener en cuenta la velocidad ni el cuidado con que se atienden las peticiones de servicio.

En relación a cada uno de estos tres criterios de valoración es posible distinguir diferentes clases de servicio:

- Servicios de altas prestaciones (en relación al criterio de valoración utilizado) en los que la variación del parámetro prestacional medido es irrelevante (guaranteed service)
- Servicios caracterizados por buenas prestaciones (en relación al criterio de valoración utilizado) y por un valor preciso de la variación del parámetro prestacional (predictive service)
- Servicios caracterizados por discretas prestaciones (en relación al criterio de valoración utilizado) y por un valor no precisado de la variación del parámetro prestacional (best effort service)

Los parámetros significativos de la valoración prestacional, en términos de velocidad, son los siguientes:

- Velocidad neta del flujo binario (throughput)
- Tiempo de transferencia de las informaciones

La caracterización del throughput en un canal puede hacerse en base al ritmo binario medio y al ritmo binario más alto ofrecido a todos los usuarios que acceden a él. El tiempo necesario para la transferencia de las informaciones del usuario es la suma del tiempo necesario para acceder al canal radio, del tiempo necesario para la propagación en el canal radio (irrelevante) y del tiempo necesario para la transferencia a través de la red.

Los parámetros característicos en la valoración prestacional en términos de cuidado son los siguientes:

- Probabilidad de pérdida de un paquete.
- Probabilidad de recepción de paquetes equivocados.
- Probabilidad de duplicación de un paquete.
- Probabilidad de secuencia equivocada en la recepción de los paquetes

Los parámetros característicos en la valoración prestacional en términos de fiabilidad son los siguientes:

- Probabilidad de fallo en la negociación de la QOS entre el usuario y la red.
- Probabilidad de que la QOS establecida en fase de negociación no sea garantizada durante la terminación de la petición de servicio.
- Disponibilidad del servicio.
- Tiempo medio entre dos back out sucesivos del servicio.
- Duración media de un back out del servicio

Por lo que respecta a los servicios orientados a la conexión de parámetros prestaciones son constituidos por el tiempo medio necesario para establecer una conexión y por el tiempo medio necesario para la finalización de la misma.

### **3.8. Módem utilizado (Nokia N12)**

El módulo GSM Nokia N12 ha sido diseñado para aplicaciones M2M y otras aplicaciones inalámbricas. El concepto M2M consiste en utilizar datos inalámbricos para proporcionar enlaces eficaces y económicos entre sistemas, dispositivos remotos e individuos.

Es un dispositivo de banda dual por lo que puede trabajar a 900 y 1800 MHz, y soporta GSM, GPRS, EGPRS, HSCSD, CSD y SMS.



Figura 3.8. Módulo GSM Nokia N12

El módulo Nokia N12 puede ser usado en muchas aplicaciones, dado que posee tres modos de funcionamiento. Se pueden implementar fácilmente aplicaciones de Entrada/Salida usando el modo de control de usuario, que ofrece la posibilidad de personalizar los mensajes, mensajería segura, y calendario para aplicaciones controladas por SMS. En el modo AT, el nokia N12 funciona como un módem GSM controlado mediante comandos AT. En el modo módem, es compatible con todas las portadoras disponibles. Además, es compatible con la plataforma M2M nokia, y en modo M2M, el N12 se comunica con un servidor de aplicaciones a través del Gateway Nokia M2M, y se pueden implementar múltiples aplicaciones.

Además de estos modos de funcionamiento, en el nokia N12 se han implementado la pila de protocolos TCP/IP que permiten conexiones de datos GPRS entre una aplicación final y un servidor. Además de los protocolos TCP/IP, existen disponibles APIs para funciones HTTP y Sockets.

El módulo nokia N12 soporta numerosas APIs de Java, mediante las cuales se puede proveer de funcionalidad para servicios de localización mediante un módulo externo GPS, aportar fiabilidad y seguridad con características como AutoPIN, encriptación GSM y códigos seguros, reset del módulo, y autenticación con la plataforma M2M de Nokia. Además, la tecnología Java permite la actualización remota del software de aplicación (IMlet).

Para configurar el Nokia N12 y explotar su funcionalidad, es necesario conectarlo a través de la placa de conexión, la cual provee de alimentación, adaptador de antena, conector para la tarjeta SIM, entrada y salida para señales de audio, adaptadores de puertos serie, además de interfaz de conexión con un PC para configuración mediante el software Nokia Configurator.

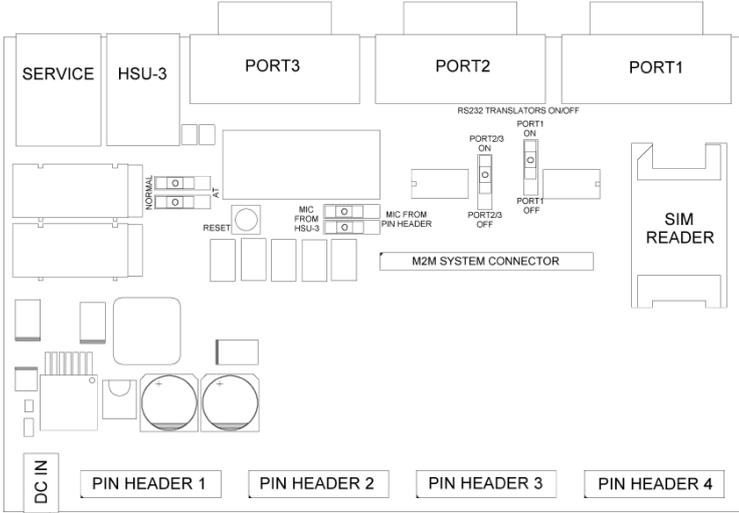


Figura 3.9. Placa de conexión del Nokia N12.

## 4. Breve descripción del sistema GPS

El **Global Positioning System (GPS)** o **Sistema de Posicionamiento Global** (más conocido con las siglas *GPS*, aunque su nombre correcto es **NAVSTAR-GPS<sup>1</sup>**) es un Sistema Global de Navegación por Satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave, con una precisión hasta de centímetros, usando GPS diferencial, aunque lo habitual son unos pocos metros. Aunque su invención se atribuye a los gobiernos francés y belga, el sistema fue desarrollado e instalado, y actualmente es operado, por el Departamento de Defensa de los Estados Unidos.

El GPS funciona mediante una red de 27 satélites (24 operativos y 3 de respaldo) en órbita sobre el globo, a 20.200 km, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la posición y el reloj de cada uno de ellos. En base a estas señales, el aparato sincroniza el reloj del GPS y calcula el retraso de las señales; es decir, la distancia al satélite. Por "triangulación" calcula la posición en que éste se encuentra. La triangulación en el caso del GPS, a diferencia del caso 2-D que consiste en averiguar el ángulo respecto de puntos conocidos, se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que llevan a bordo cada uno de los satélites.

La antigua Unión Soviética tenía un sistema similar llamado GLONASS, ahora gestionado por la Federación Rusa.

Actualmente la Unión Europea está desarrollando su propio sistema de posicionamiento por satélite, denominado Galileo.

### 4.1 Historia

En 1957 la Unión Soviética lanzó al espacio el satélite Sputnik I, que era monitorizado mediante la observación del Efecto Doppler de la señal que transmitía. Debido a este hecho, se comenzó a pensar que, de igual modo, la posición de un observador podría ser establecida mediante el estudio de la frecuencia Doppler de una señal transmitida por un satélite cuya órbita estuviera determinada con precisión.

La marina estadounidense rápidamente aplicó esta tecnología, para proveer a los sistemas de navegación de sus flotas de observaciones de posiciones actualizadas y precisas. Así surgió el sistema TRANSIT, que quedó operativo en 1964, y hacia 1967 estuvo disponible, además, para uso comercial.

Las actualizaciones de posición, en ese entonces, se encontraban disponibles cada 40 minutos y el observador debía permanecer casi estático para poder obtener información adecuada.

Posteriormente, en esa misma década y gracias al desarrollo de los relojes atómicos, se diseñó una constelación de satélites, portando cada uno de ellos uno de estos relojes y estando todos sincronizados con base a una referencia de tiempo determinada.

En 1973 se combinaron los programas de la Marina de EE.UU. y el de la USAF (este último consistente en una técnica de transmisión codificada que proveía datos precisos usando una señal modulada con un código de sonidos pseudo-aleatorios (PRN = Pseudo-Random Noise), en lo que se conoció como Navigation Technology Program, posteriormente renombrado como NAVSTAR GPS.

Entre 1978 y 1985 se desarrollaron y lanzaron once satélites prototipo experimentales NAVSTAR, a los que siguieron otras generaciones de satélites, hasta completar la constelación actual, a la que se declaró con «capacidad operacional inicial» en diciembre de 1993 y con «capacidad operacional total» en abril de 1995.

En 1994, este país ofreció el servicio normalizado de determinación de la posición para apoyar las necesidades de la OACI, y ésta aceptó el ofrecimiento.

#### 4.2. Características técnicas y prestaciones

Este Sistema Global de Navegación por Satélite lo componen:

1. *Sistema de satélites.* Está formado por 24 unidades con trayectorias sincronizadas para cubrir toda la superficie del globo terráqueo. Más concretamente, repartidos en 6 planos orbitales de 4 satélites cada uno. La energía eléctrica que requieren para su funcionamiento la adquieren a partir de dos paneles compuestos de celdas solares adosados a sus costados.
2. *Estaciones terrestres.* Envían información de control a los satélites para controlar las órbitas y realizar el mantenimiento de toda la constelación.
3. *Terminales receptores:* Indican la posición en la que están; conocidas también como Unidades GPS, son las que podemos adquirir en las tiendas especializadas.



Figura 4.1. Operadora de satélites controlando la constelación NAVSTAR-GPS, en la Base Aérea de Schriever.



Figura 4.2. Lanzamiento de satélites para la constelación NAVSTAR-GPS mediante un cohete Delta

#### 4.2.1. Segmento espacial

- Satélites en la constelación: 24 (4 X 6 órbitas)
  - Altitud: 20.200 km
  - Período: 11 h 56 min
  - Inclinación: 55 grados (respecto al ecuador terrestre).
  - Vida útil: 7,5 años
- Segmento de Control (estaciones terrestres)
  - Estación principal: 1
  - Antena de tierra: 4
  - Estación monitora (de seguimiento): 5

- Señal RF
  - Frecuencia portadora:

Civil - 1 575.42 MHz (L1). Utiliza el Código de Adquisición Aproximativa (C/A)

Militar – 1227.60 MHz (L2). Utiliza el Código de Precisión (P), cifrado.

- Nivel de potencia de la señal: -160 dBW (en superficie tierra)

- Polarización: circular dextrógira

- Exactitud
  - Posición: aproximadamente 15 m (el 95%)
  - Hora: 1 ns

- Cobertura: mundial
- Capacidad de usuarios: ilimitada
- Sistema de coordenadas:
  - Sistema Geodésico Mundial 1984 (WGS84)
  - Centrado en la Tierra, fijo.
- Integridad: tiempo de notificación 15 minutos o mayor. **NO ES SUFICIENTE PARA LA AVIACIÓN CIVIL**
- Disponibilidad: 24 satélites - 70 % y 21 satélites - 98 % **NO ES SUFICIENTE COMO MEDIO PRIMARIO DE NAVEGACIÓN**

### 4.3. Evolución del sistema GPS

El GPS está evolucionando hacia un sistema más sólido (GPS III), con una mayor disponibilidad y que reduzca la complejidad de las aumentaciones GPS. Algunas de las mejoras previstas comprenden:



Figura 4.3. Estación y receptor GPS profesionales para precisiones centimétricas.

El GPS está evolucionando hacia un sistema más sólido (GPS III), con una mayor disponibilidad y que reduzca la complejidad de las aumentaciones GPS. Algunas de las mejoras previstas comprenden:

- Incorporación de una nueva señal en L2 para uso civil.
- Adición de una tercera señal civil (L5): 1176.45 MHz
- Protección y disponibilidad de una de las dos nuevas señales para servicios de Seguridad Para la Vida (SOL).
- Mejora en la estructura de señales.

- Incremento en la potencia de señal (L5 tendrá un nivel de potencia de -154 dB).
- Mejora en la precisión (1 – 5 m).
- Aumento en el número de estaciones monitorizadas: 12 (el doble)
- Permitir mejor interoperabilidad con la frecuencia L1 de Galileo

El programa GPS III persigue el objetivo de garantizar que el GPS satisfará requisitos militares y civiles previstos para los próximos 30 años. Este programa se está desarrollando para utilizar un enfoque en 3 etapas (una de las etapas de transición es el GPS II); muy flexible, permite cambios futuros y reduce riesgos. El desarrollo de satélites GPS II comenzó en 2005, y el primero de ellos estará disponible para su lanzamiento en 2012, con el objetivo de lograr la transición completa de GPS III en 2017. Los desafíos son los siguientes:

1. Representar los requisitos de usuarios, tanto civiles como militares, en cuanto a GPS.
2. Limitar los requisitos GPS III dentro de los objetivos operacionales.
3. Proporcionar flexibilidad que permita cambios futuros para satisfacer requisitos de los usuarios hasta 2030.
4. Proporcionar solidez para la creciente dependencia en la determinación de posición y de hora precisa como servicio internacional.

#### 4.4. Funcionamiento

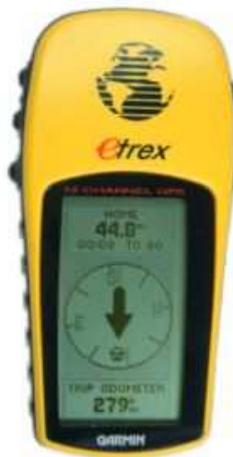


Figura 4.4. Receptor GPS común

1. La situación de los satélites es conocida por el receptor con base en las efemérides (5 parámetros orbitales Keplerianos), parámetros que son transmitidos por los propios satélites. La colección de efemérides de toda la constelación se completa cada 12 min y se guarda en el receptor GPS.
2. El receptor GPS funciona midiendo su distancia a los satélites, y usa esa información para calcular su posición. Esta distancia se mide calculando el tiempo que la señal tarda en llegar al receptor. Conocido ese tiempo y basándose en el hecho de que la señal viaja a la velocidad de la luz (salvo algunas

- correcciones que se aplican), se puede calcular la distancia entre el receptor y el satélite.
3. Cada satélite indica que el receptor se encuentra en un punto en la superficie de la esfera, con centro en el propio satélite y de radio la distancia total hasta el receptor.
  4. Obteniendo información de dos satélites se nos indica que el receptor se encuentra sobre la circunferencia que resulta cuando se intersectan las dos esferas.
  5. Si adquirimos la misma información de un tercer satélite notamos que la nueva esfera solo corta la circunferencia anterior en dos puntos. Uno de ellos se puede descartar porque ofrece una posición absurda. De esta manera ya tendríamos la posición en 3-D. Sin embargo, dado que el reloj que incorporan los receptores GPS no está sincronizado con los relojes atómicos de los satélites GPS, los dos puntos determinados no son precisos.
  6. Teniendo información de un cuarto satélite, eliminamos el inconveniente de la falta de sincronización entre los relojes de los receptores GPS y los relojes de los satélites. Y es en este momento cuando el receptor GPS puede determinar una posición 3-D exacta (latitud, longitud y altitud). Al no estar sincronizados los relojes entre el receptor y los satélites, la intersección de las cuatro esferas con centro en estos satélites es un pequeño volumen en vez de ser un punto. La corrección consiste en ajustar la hora del receptor de tal forma que este volumen se transforme en un punto.

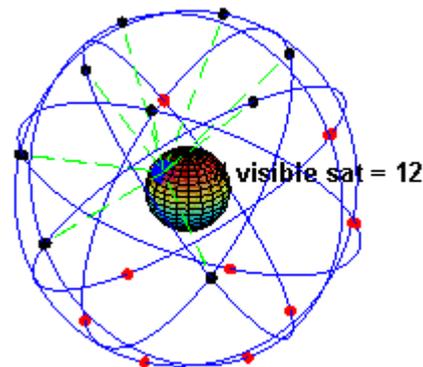


Figura 4.5. Constelación GPS alrededor de la Tierra.

#### 4.5. Fiabilidad de los datos

Debido al carácter militar del sistema GPS, el Departamento de Defensa de los EE.UU. se reservaba la posibilidad de incluir un cierto grado de error aleatorio, que podía variar de los 15 a los 100 m. La llamada disponibilidad selectiva (S/A) fue eliminada el 2 de mayo de 2000. Aunque actualmente no aplique tal error inducido, la precisión intrínseca del sistema GPS depende del número de satélites visibles en un momento y posición determinados.

Con un elevado número de satélites siendo captados (7, 8 o 9 satélites), y si éstos tienen una geometría adecuada (están dispersos), pueden obtenerse precisiones inferiores a 2,5 metros en el 95% del tiempo. Si se activa el sistema DGPS llamado SBS (WAAS-

EGNOS-MSAS), la precisión mejora siendo inferior a un metro en el 97% de los casos. (estos sistemas SBS no aplican en Sudamérica, ya que esta parte del mundo no cuenta con este tipo de satélites geoestacionarios)

**4.5.1. Fuentes de error**

La posición calculada por un receptor GPS requiere el instante actual, la posición del satélite y el atraso medido de la señal recibido. La precisión es dependiente en la posición y atraso de la señal.

Al introducir el atraso, el receptor compara una serie de bits (unidad binaria) recibida del satélite con una versión interna. Cuando se comparan los límites de la serie, las electrónicas pueden meter la diferencia a 1% de un tiempo BIT, o aproximadamente 10 nanosegundos por el código C/A. Desde entonces las señales GPS se propagan a la velocidad de luz, que representa un error de 3 metros. Este es el error mínimo posible usando solamente la señal GPS C/A.

La precisión de la posición se mejora con una señal P(Y). Al presumir la misma precisión de 1% de tiempo BIT, la señal P(Y) (alta frecuencia) resulta en una precisión de más o menos 30 centímetros. Los errores en las electrónicas son una de las varias razones que perjudican la precisión (ver la tabla).

Fuentes de error	
Fuente	Efecto
Ionosfera	± 5 m
Efemérides	± 2,5 m
Reloj satelital	± 2 m
Distorsión multibandas	± 1 m
Troposfera	± 0,5 m
Errores numéricos	± 1 m o menos

1. Retraso de la señal en la ionosfera y la troposfera.

2. Señal multirruta, producida por el rebote de la señal en edificios y montañas cercanos.
3. Errores de orbitales, donde los datos de la órbita del satélite no son completamente precisos.
4. Número de satélites visibles.
5. Geometría de los satélites visibles.
6. Errores locales en el reloj del GPS.

#### 4.6. Aplicaciones

1. Navegación terrestre (y peatonal), marítima y aérea. Bastantes coches lo incorporan en la actualidad, siendo de especial utilidad para encontrar direcciones o indicar la situación a la grúa.
2. Topografía y geodesia.
3. Localización agrícola (agricultura de precisión), ganadera y de fauna.
4. Salvamento y rescate.
5. Deporte, acampada y ocio.
6. Para localización de enfermos, discapacitados y menores.
7. Aplicaciones científicas en trabajos de campo (geomática).
8. Geocaching, actividad deportiva consistente en buscar "tesoros" escondidos por otros usuarios.
9. Se utiliza para rastreo y recuperación de vehículos.
10. Navegación Deportiva.
11. Deportes Aéreos: Parapente, Ala delta, Planeadores, etc.
12. Existe quien dibuja usando tracks o juega utilizando el movimiento como cursor (común en los gps garmin).
13. Sistemas de gestión y seguridad de flotas.



Figura 4.6. Navegador GPS de pantalla táctil de un vehículo con información sobre la ruta, así como las distancias y tiempos de llegada al punto de destino.

Actualmente la Unión Europea está desarrollando el sistema Galileo, iniciativa surgida para desarrollar un Sistema Global de Navegación por Satélite, de titularidad civil, que proporcione a Europa independencia respecto a los sistemas actuales: GPS y GLONASS. La componente espacial de Galileo está constituida por 30 satélites repartidos en tres planos orbitales de 23.600 Km. de altura y 55° de inclinación, diseño que mejora su cobertura en latitudes extremas (cerca de los polos) con respecto a los sistemas actuales.

El sistema se complementa con una serie de estaciones terrestres encargadas de controlar y gestionar la constelación, así como de proporcionar servicios de valor añadido como datos de integridad, servicios de búsqueda y rescate e incluso información comercial.

Inicialmente Galileo iba a estar disponible en el 2008 aunque el proyecto acumula ya tres años de retraso y se encuentra aún en fase de despliegue, y no podrá comercializar sus primeros servicios hasta 2011.

## 5. La tecnología Java J2ME

La plataforma J2ME es una familia de especificaciones que definen varias versiones minimizadas de la plataforma Java 2; estas versiones minimizadas pueden ser usadas para programar en dispositivos electrónicos; desde teléfonos celulares, en PDAs, hasta en tarjetas inteligentes, etc. Estos dispositivos presentan en común que no disponen de abundante memoria ni mucha potencia en el procesamiento, ni tampoco necesitan de todo el soporte que brinda el J2SE, (la plataforma estándar de Java usada en sistemas de escritorio y servidor)

J2ME es la versión de Java orientada a los dispositivos móviles. Debido a que los dispositivos móviles tienen una potencia de cálculo baja e interfaces de usuario pobres, es necesaria una versión específica de Java destinada a estos dispositivos, ya que el resto de versiones de Java, J2SE (para sistemas servidor) o J2EE (para sistemas de escritorio), no encajan dentro de este esquema. J2ME es por tanto, una versión “reducida” de J2SE.

La plataforma Java ME representa la única solución verdaderamente abierta para construir aplicaciones móviles. Esta tecnología permite la portabilidad de aplicaciones entre plataformas, y las inversiones se reducen debido a la capacidad de reutilización. Además, la plataforma J2ME sigue en constante crecimiento debido a las demandas por parte de la industria de nuevos componentes y APIs de Java. A su vez, los desarrolladores crean nuevas capacidades que hacen que puedan surgir nuevas aplicaciones basadas en el lenguaje. Esta evolución constante de demandas y de capacidades forma el ecosistema Java ME, y es la razón más importante del éxito de la plataforma de Java y hace que se asegure que continuará desarrollándose para satisfacer las necesidades, tanto de los desarrolladores como de los usuarios.

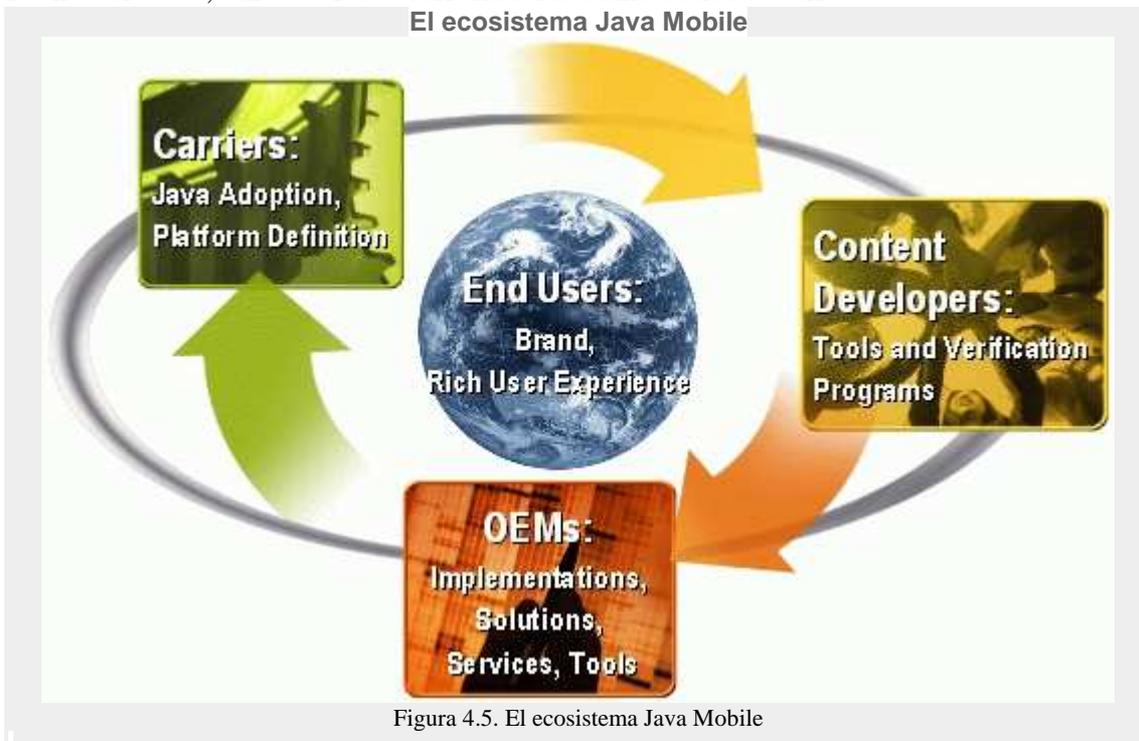


Figura 4.5. El ecosistema Java Mobile

### 5.1 La arquitectura J2ME

La arquitectura Java™ 2 Micro Edition está orientada a pequeños dispositivos y sistemas embebidos como son teléfonos móviles, PDAs, Set-Top Boxes, máquinas expendedoras y un largo etcétera de productos existentes o futuros. Al igual que sucede con J2EETM, que está orientado a entornos corporativos o J2SETM, orientado a sistemas de sobremesa, la arquitectura J2ME está formada por un conjunto de APIs estándares que permiten que las aplicaciones desarrolladas se beneficien de las características multiplataforma de Java y que abren la puerta a la distribución de aplicaciones a millones de dispositivos.

La arquitectura J2ME propuesta por SUN sigue consideraciones de modularización, personalización y estrictas restricciones en las características de los dispositivos (memoria y ancho de banda limitado, alimentación mediante baterías, tamaño físico de la pantalla y conectividad intermitente).

Las especificaciones J2ME de SUN son lo suficientemente flexibles para soportar:

- Gran cantidad de tipos de dispositivos y configuraciones hardware
- Diferentes modelos de uso: dispositivos operados con voz, teclado numérico, con pantalla táctil...
- Tecnología que está mejorando y cambiando constante y muy rápidamente
- Gran cantidad de aplicaciones y funcionalidades
- La necesidad de que las aplicaciones y sus capacidades evolucionen y crezcan con rapidez

Para soportar este tipo de personalización y extensibilidad, tres conceptos esenciales son definidos en la arquitectura J2ME:

- **Configuration.** Define una plataforma Java para un grupo de dispositivos con los mismos (mínimos) requerimientos de memoria y hardware. Especifica:
  - Características del lenguaje Java soportadas
  - Características de la máquina virtual Java soportadas
  - Librerías y APIs Java soportadas

SUN propone dos especificaciones de Configuration: la CLDC (Connected Limited Device Configuration) y la CDC (Connected Device Configuration).

- **Profile.** Proporciona un toolkit completo para implementar aplicaciones para un dispositivo especificado. Se encuentra por encima de una Configuration (de hecho la extiende). Incluye librerías que son más específicas del dominio. Garantiza interoperabilidad entre un conjunto de dispositivos heterogéneo. Un dispositivo puede soportar varios Profiles.
- **Optional Packages.** Además de las funcionalidades proporcionadas por Configuration y Profile, son necesarias otras librerías de propósito general que solucionen funcionalidades específicas. La definición de estas librerías se realiza en el grupo de trabajo de Optional Packages.

Al tratarse de especificaciones, tanto las Configuration, como los Profiles, se convierten en un contrato entre la aplicación, el programador y el fabricante del dispositivo. Los fabricantes deviden qué Profile(s) soportan en cada uno de sus dispositivos y se encargan de implementar todas las características de los Profile(s) seleccionados.

En el siguiente diagrama, podemos ver los dos grandes bloques de arquitecturas en los que se divide la plataforma J2ME. En función de la familia de dispositivos tomaremos una u otra opción.

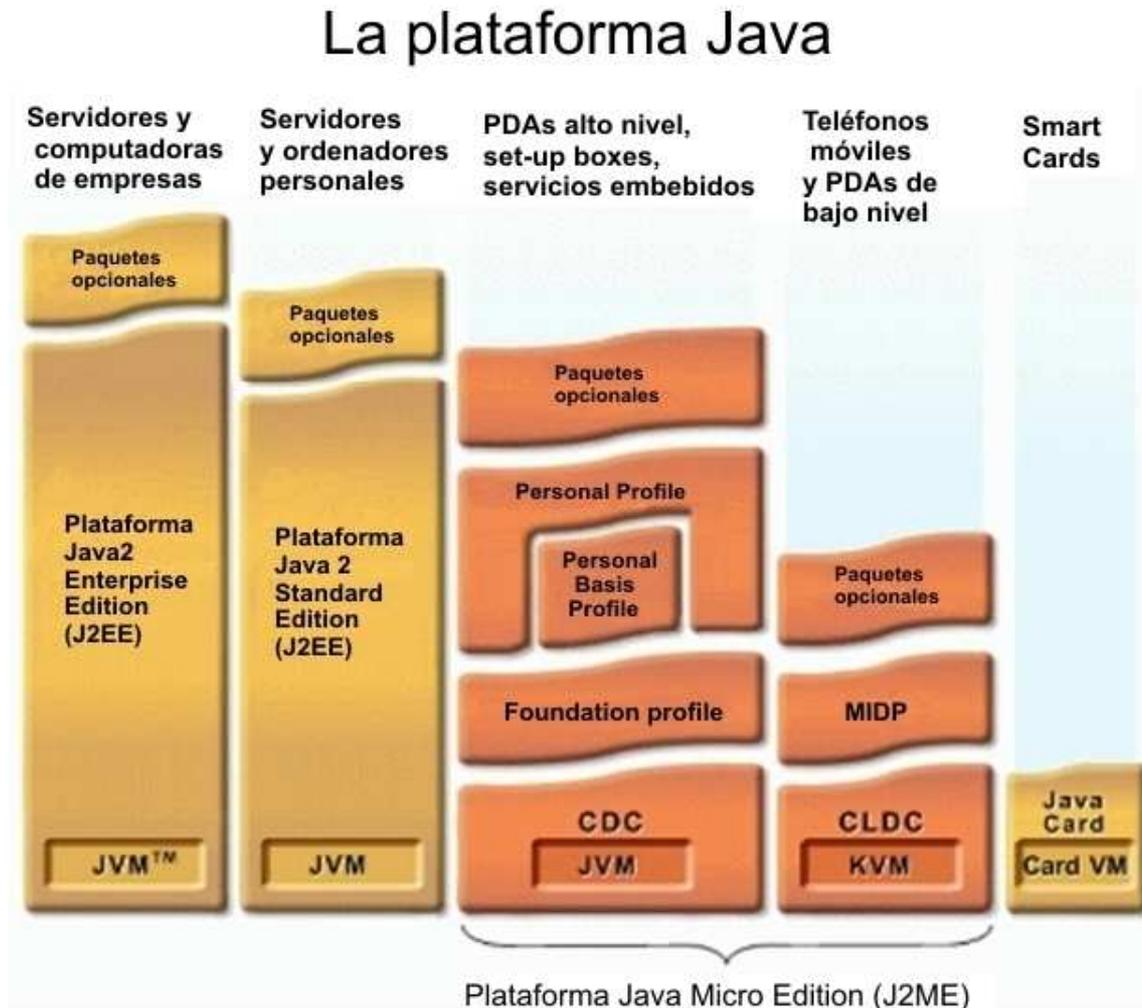


Figura 4.2. La plataforma Java J2ME

## 5.2 Especificaciones

### 5.2.1 Configuraciones

Las configuraciones se componen de una máquina virtual y un conjunto mínimo de bibliotecas de función. Proporcionan la funcionalidad básica para un conjunto de dispositivos que comparten características similares, tales como gestión de memoria o

conectividad a la red.

En la actualidad existen dos configuraciones J2ME:

Connected Limited Device Configuration (CLDC)

Connected Device Configuration (CDC)

### ***CLDC***

Esta configuración está diseñada para dispositivos con conexiones de red intermitentes, procesadores lentos y memoria limitada: teléfonos móviles, asistentes personales (PDAs), etc. Es habitual que estos dispositivos tenga CPUs de 16 o 32 bits y un mínimo de entre 128 y 256 KB de memoria disponible para la implementación de la plataforma Java y sus aplicaciones asociadas. Está basada en la máquina virtual K (K Virtual Machine, KVM).

### ***CDC***

Esta configuración está diseñada para dispositivos que tienen más memoria, procesadores más rápidos y un ancho de banda mayor, como Set-top boxes, pasarelas residenciales, asistentes personales de gran capacidad, etc. Incluye una máquina virtual Java completa (Java Virtual Machine, JVM) y un subconjunto de APIs de la arquitectura J2SE mucho mayor. Se orienta a dispositivos con CPU de 32 bits y un mínimo de 2 MB de memoria disponible para la plataforma Java y aplicaciones asociadas

## **5.2.2 Profiles**

Para conformar un entorno de ejecución completo orientado a una categoría de dispositivos, las configuraciones se han de combinar con un conjunto de APIs de un nivel más alto, llamadas perfiles, que van un paso más allá en la definición del modelo de ciclo de vida de las aplicaciones, la interfaz de usuario y acceso a las propiedades específicas de los dispositivos.

En la actualidad existen los siguientes perfiles asociados a J2ME:

***Mobile Information Device Profile (MIDP)*** Está diseñado para teléfonos móviles y PDAs con capacidades básicas. Ofrece la funcionalidad básica para las aplicaciones móviles, incluyendo la interfaz de usuario, conectividad a redes, almacenamiento local de datos y gestión del ciclo de vida de las aplicaciones.

Al combinarlo con la configuración CLDC, MIDP proporciona un entorno de ejecución Java completo que incrementa la capacidad de los dispositivos móviles y que reduce el consumo de memoria y energía.

### ***Foundation Profile (FP)***

Los perfiles CDC están organizados en capas de forma que permitan la agregación según se precise para proporcionar funcionalidad a las aplicaciones para distintos tipos de dispositivos. El FP es el perfil de más bajo nivel para el CDC. Proporciona una implementación lista para el trabajo en red que se puede emplear en implementaciones embebidas que carecen de interfaz de usuario. También se puede combinar con los perfiles Personal Basis y Personal para los dispositivos que precisan de una interfaz gráfica de usuario (IGU).

***Personal Profile (PP)*** El perfil Personal, es el perfil para CDC orientado a dispositivos que requieren una IGU (Interfaz gráfica de usuario) completa o capacidad de ejecutar

applets de Internet, como por ejemplo PDAs de gama alta, consolas de juegos, etc. Incluye todas las bibliotecas de funciones de la Java Abstract Window Toolkit (AWT) y ofrece fidelidad Web, permitiendo la ejecución de applets diseñados para utilización en entornos de sobremesa. PP reemplaza la tecnología PersonalJavaTM.

### ***Personal Basis Profile (PBP)***

El perfil Personal Basis es un subconjunto del perfil Personal y proporciona un entorno de aplicación para dispositivos con conexión que tolerar un nivel de presentación gráfica básico o que precisan de conjuntos de herramientas (toolkits) gráficas especializadas para aplicaciones específicas. Al igual que el perfil Personal, está pensado para ejecutarse sobre la configuración CDC.

### ***Paquetes opcionales***

La plataforma J2ME se puede ampliar combinando varios paquetes opcionales con CLDC y CDC junto con sus perfiles. Estos paquetes se han creado para responder a requisitos concretos de mercado y ofrecen un conjunto de APIs estándares para utilizar tanto tecnologías existentes como emergentes; entre estas se incluyen localización GPS, Bluetooth, servicios Web, mensajería wireless, capacidades multimedia o conectividad a bases de datos. Dado que son modulares, los fabricantes de dispositivos pueden incorporarlos según los vayan necesitando para mejorar las características soportadas.

El diseño de la plataforma Java ME hace posible la portabilidad de aplicaciones de dispositivos limitados, con conexión intermitente a las redes de comunicación, a dispositivos móviles más avanzados orientados a conexión, de una manera eficiente y flexible, de modo que pueda coexistir el mismo tipo de aplicaciones en todos los dispositivos móviles.

## **5.3 El MIDlet.**

Las aplicaciones desarrolladas con MIDP son denominadas MIDlets. Estas clases MIDlets son almacenadas en clases java en ficheros "class".

Los Midlet necesitan ser empaquetados antes de ser instalado en el dispositivo de destino. Se utilizará un fichero .JAR(Java ARchive) donde se tendrá el Midlet principal y todas aquellas clases, imágenes u otros ficheros que nos puedan ser necesarios en tiempo de ejecución. También se incluirá en el .JAR información (en el fichero *manifest*) que le explique al dispositivo el contenido del fichero .JAR. Esta misma información también se incluye en el fichero .JAD (Java Application Descriptor).

Las clases del Midlet que son empaquetadas en el .JAR, deben estar compiladas en .class (obviamente) y verificadas antes de su utilización en un dispositivo, para verificar que no realizan ninguna operación no permitida. De hecho, las únicas operaciones 'no permitidas' que permite el API de MIDP es el método *exit()* de las clases *System* o *Runtime*, y como vimos requieren usar la excepción *SecurityException* para poder ser utilizado. Esta verificación se debe a hacer, debido a las limitaciones de los dispositivos, donde añadir esta parte de seguridad en la máquina virtual sería muy costoso en memoria. Es por ello, que los usuarios de estos dispositivos tiene que tener mucho

cuidado con los .JAR que instalan, ya que si es de una fuente no fiable, podría contener código malicioso no verificado y que escapase al control de la máquina virtual.

Un midlet siempre heredará de la clase *javax.microedition.midlet.MIDlet*. Los métodos de esta clase permiten a nuestra aplicación crear, empezar, parar y destruir un Midlet, estos métodos son la interfaz del Midlet, que va a permitir a nuestro dispositivo poder manejar múltiples Midlets, sin tener que estar todos ejecutándose en el mismo entorno. El sistema de nuestro dispositivo puede seleccionar que Midlet está activo usando los métodos correspondientes para empezar o pausar.

El ciclo de vida de un midlet se compone de lo siguientes estados: **Pausado**, **Activo** o **Destruído**. Sólo puede estar en un estado a la vez. La figura 1 muestra como se pasa de uno a otro:



Figura 4.3. Ciclo de vida de un Midlet.

#### 5.4 Consideraciones a tener en cuenta en el desarrollo de una aplicación J2ME.

- **Gestión de la comunicación entre la parte servidora (aplicación web) y cliente (dispositivo móvil).** Se hace necesario identificar la manera de intercambiar información entre ambos sistemas, por un lado para descargar información al dispositivo móvil y por otra para enviar los cambios operados en el dispositivo móvil al servidor.
- **Persistencia de información en la parte cliente (dispositivo móvil).** Debido a que la conectividad de dispositivos móviles a veces no está garantizada, la

información con la que trabajará la parte cliente debe persistir en el dispositivo, de tal manera que el usuario final pueda seguir trabajando, aún cuando no pueda conectarse al repositorio final, en el servidor.

- **Detección en la parte cliente de la existencia de conectividad.** Como la conectividad no está garantizada, es necesario poder preguntar en todo momento (antes mismo de realizar las operaciones de comunicación con la parte servidora) por su estado y no depender de timeouts de conexión que ralentizarían la aplicación.
- **Detección en la parte servidora de los dispositivos móviles conectados.** Necesidad de conocer y controlar los N dispositivos móviles conectados al sistema.
- **Cifrado de información en el dispositivo móvil.** El dispositivo móvil está sujeto a posibles pérdidas, extravíos o robos, por ello al almacenar internamente información confidencial, se hace necesario el cifrado de cierta información.
- **Gestión de la validación de usuario.** En el caso de una aplicación web tradicional, una vez se ha realizado la primera petición al servidor, el servidor mantiene información del estado de los clientes a través de la sesión, encargándose el navegador de mantenernos asociados a nuestra sesión. En un cliente J2ME, por defecto, esta gestión de estado no la tenemos; si es necesaria esta funcionalidad deberemos programarla. A su vez los accesos al servidor por parte de cualquier dispositivo móvil deben estar asegurados, por lo que se hace necesario un mecanismo de validación que contemple este escenario.
- **Capa de presentación en la parte cliente.** Se hace necesario la elección de la tecnología para gestionar la capa de presentación en la parte cliente, teniendo en cuenta las limitaciones de un dispositivo móvil.
- **Selección del IDE.** Necesidad de un entorno de desarrollo que te permita crear fácilmente aplicaciones de movilidad, depurarlas, probarlas y desplegarlas.

### 5.5 Elección de las tecnologías/productos

- **Selección del Perfil J2ME.** Aunque no aparece en el apartado anterior como una necesidad identificada a la hora de desarrollar aplicaciones de movilidad, es un requisito intrínseco al desarrollo de aplicaciones en dispositivos móviles con Java. La elección del Profile nos condicionará la elección a alguna de las siguientes componentes. En este enlace podréis encontrar información de la edición micro de la plataforma Java (J2ME), así como de sus configuraciones y perfiles, las diferencias entre ellos, los pros y los contras, que os ayudarán a decidirlos. De momento nos tenemos que decantar por uno de los siguiente perfiles: **MIDP 2.0**

con **CLDC 1.1** o **Personal Profile 1.0** (compatible jdk 1.3)

- **Máquina virtual.** Si se ha optado por MIDP, el dispositivo móvil elegido deberá ser compatible con este Profile y la aplicación cliente se ejecutará en el contenedor, Middlet Manager, que incorpore; no hay que instalar nada más por defecto en el dispositivo. Si se ha optado por Personal Profile 1.0, habrá que instalar en nuestro dispositivo una máquina virtual compatible con él, AME recomienda la jvm Creme.
- **Gestión de la comunicación entre la parte servidora (aplicación web) y cliente (dispositivo móvil).** AME ofrece una serie de componentes para cubrir esta necesidad. Dependiendo del tipo de información, del tamaño de los mensajes intercambiados y el condicionante de pérdida de conexión, AME ofrece dos alternativas: servicios web (axis-burlap-hessian), para intercambio de información puntual, y Funambol para servicios de sincronización de información.
- **Persistencia de información en la parte cliente (dispositivo móvil).** Si se ha optado por MIDP, la persistencia se realizará con el Record Management System propio de la especificación; sin embargo si se opta por Personal Profile 1.0, AME ofrece hsq como base de datos embebida, la cual habrá que incorporar a nuestro aplicativo cliente.
- **Detección en la parte servidora de los dispositivos móviles conectados.** A veces se hace necesario, conocer qué dispositivos móviles están encendidos y conectados al sistema, así como enviarles mensajes en un momento preciso.
- **Cifrado de información en el dispositivo móvil.** Independientemente del profile elegido (MIDP o Personal Profile) AME recomienda la librería bouncycastle, para el tratamiento de cifrado. Esta librería trae varias distribuciones en función de las necesidades de nuestra aplicación, j2se, j2me,...
- **Gestión de la validación de usuario.** Independientemente del perfil elegido, la aplicación cliente tiene que gestionar el envío de las credenciales a la parte servidora, ya que en las aplicaciones móviles con cliente rico el concepto de "sesión" no existe. La mejor manera de realizarlo es a través de las cabeceras estándar http. Del lado del servidor como siempre utilizaremos Acegi, como framework de seguridad.
- **Capa de presentación en la parte cliente.** Si elegimos MIDP, desarrollaremos Midlets en el lado del cliente, y si elegimos Personal Profile, utilizaremos Thinlets.
- **Entorno de desarrollo.** En función del perfil elegido, utilizaremos
  - MIDP. Netbeans con el paquete de movilidad, para desarrollar la parte del dispositivo móvil, incluida la capa de presentación con Midlets. Eclipse

para el desarrollo de la parte servidora. Para realizar las pruebas el paquete de movilidad de Netbeans ofrece emuladores.

Personal Profile. ThinG para el desarrollo de la capa de presentación con Thinlet, Eclipse (con el jre 1.3.1) para el desarrollo de la parte móvil y la parte servidora, y el emulador de la Creme, para realizar las pruebas.

## 6. Prototipo desarrollado

El objetivo de este proyecto era el desarrollo de un prototipo que actuara como estación recolectora de los datos generados por una red de sensores inalámbricos, y que a su vez fuera capaz de transmitir los datos a través de la red de telefonía móvil GSM, a un destino conectado a la red de Internet. Además, se quería dotar de capacidad de actuación al sistema, y poder controlar de manera remota los parámetros sensorizados por los sensores, mediante la activación de líneas digitales integradas en nuestro prototipo.

Se partía de la base de un prototipo implementado con anterioridad, que constaba de las siguientes partes:

- La red de sensores Mote Kit 410CB, con dos motes y la placa receptora MIB 510.
- Una FPGA SPARTAN-3 BOARD de interconexión.
- Un módem Sony-Ericsson GT47 transmisor de datos.

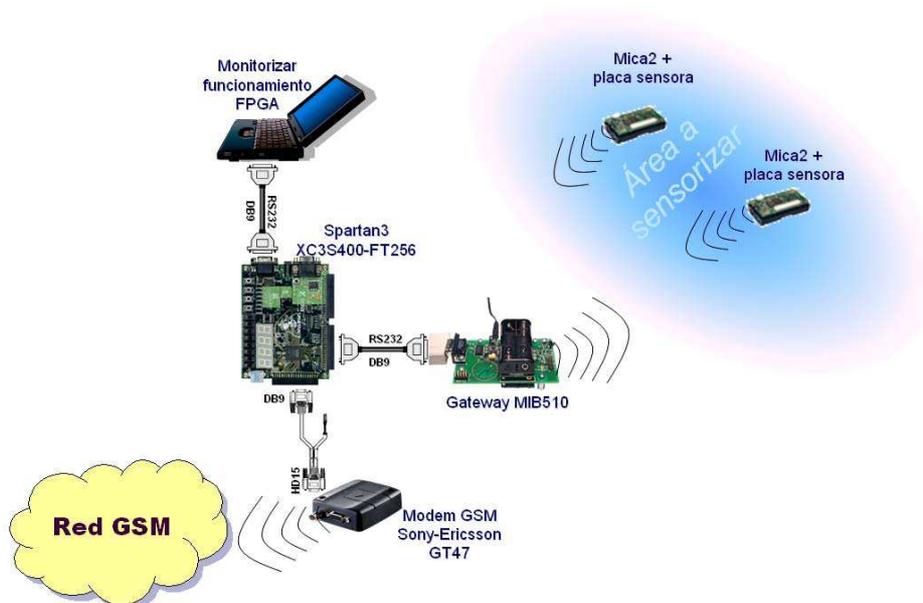


Figura 6.1. Prototipo previo

El funcionamiento de este prototipo previo era el siguiente:

Los sensores mica2 captaban valores relativos a los siguientes parámetros: luz, temperatura, batería del mote, aceleración en los ejes x e y, magnetismo en los ejes x e y. Dichos valores eran encapsulados en una trama de 66bytes de longitud, y enviados al gateway MIB510.

Estos datos se pretendían transmitir a través del envío de mensajes cortos SMS mediante el módem GT47 a un destino. Este módem disponía de una entrada de datos



En el prototipo previo existente, se ha sustituido el módem GT47 por un módem Nokia N12 de mayores capacidades, programable en tecnología Java J2ME, y con comunicación GPRS. Este módem sí es capaz de recibir datos en formato byte, pero se ha seguido usando la FPGA existente para dotar al sistema de cierta capacidad de cómputo, si fuera necesaria. Se ha usado un dispositivo comercial de la marca Teltónica, el FM3101, el cual integra éste módem N12, así como un módulo receptor GPS, y conectividad GSM/GPRS, y a la vez provee de salidas digitales configurables para la funcionalidad de actuación. De este modo, el esquema del prototipo será el siguiente:

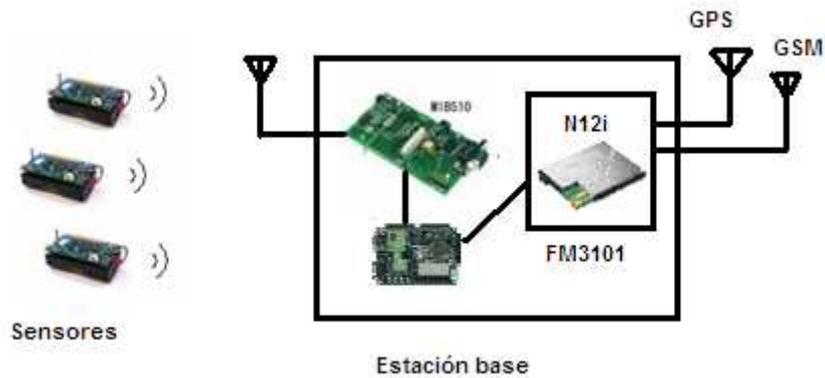


Figura 6.3. Esquema del prototipo

A continuación se muestra una imagen real del prototipo y de los sensores:

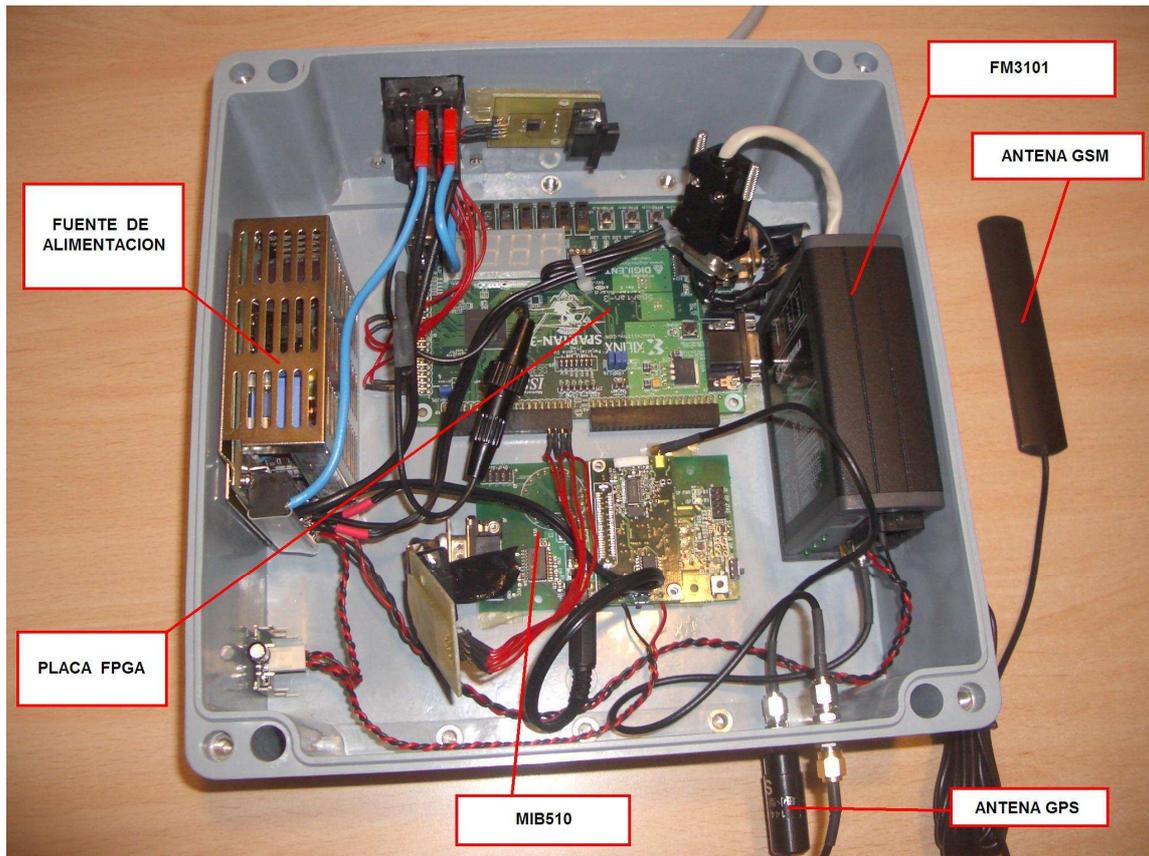


Figura 6.4. Fotografía del prototipo



Figura 6.5. Fotografía de los motes.

## 6.1 Configuración de la red de sensores y el Gateway

Mediante el uso del software de configuración MoteConfig suministrado por Crossbow, se programan los nodos con el programa *Surge Reliable*, el cual hace que los motes envíen una trama cada diez segundos con la información relativa a luz, temperatura, batería, aceleración y campo magnético.

## 6.2 Placa FPGA. Funcionalidad. Reuso del anterior prototipo.

Una FPGA (Field Programable Array) es un dispositivo semiconductor que contiene elementos lógicos programables e interconexiones programables entre ellos. Los componentes lógicos programables pueden ser programados para duplicar la funcionalidad de las puertas básicas o funciones combinatoriales más complejas.

La FPGA utilizada corresponde al modelo SPARTAN-3 BOARD [29]. Este dispositivo ha sido diseñado para permitir la rápida construcción de circuitos y el testeado de éstos últimos. Este modelo se caracteriza por poseer 400.000 puertas lógicas, tres reguladores de voltaje de 3.3, 2.5 y 1.2 VDC, dispositivos de entrada-salida (puerto serie, puerto VGA y un puerto PS/2 ratón/teclado), una memoria asíncrona de 1MB (SRAM), tres conectores de expansión de 40 pines y diversos LEDS, pulsadores y switches. La programación de este dispositivo se realiza a través de un cable de programación JTAG y cables P4 & MULTIPRO de Xilinx. Estas características hacen de esta FPGA una plataforma perfecta de experimentar con nuevos diseños.



Figura 6.6. FPGA Spartan 3 integrada en placa de pruebas.

A través de un programa denominado EDK de Xilinx, se configura la FPGA con tres puertos serie, uno de entrada y dos de salida. Al tener únicamente un puerto serie, la FPGA se emplea para examinar si está funcionando correctamente. Los dos puertos serie restantes se configuran en dos conectores de expansión de 40 pines.

Una vez realizada esta configuración, se carga en la FPGA el programa conversor para la traducción de caracteres hexadecimales a caracteres ASCII.

El programa traductor es muy sencillo y está escrito en lenguaje de programación C. Este programa recoge los caracteres hexadecimales que llegan por el puerto serie de entrada, los transforma y los envía por un puerto serie de salida byte a byte.

Para evitar programar la FPGA cada vez que se quiera poner en funcionamiento, se ha cargado el programa en una memoria flash que posee, de manera que cada vez que se alimente la FPGA comenzara a ejecutarse el programa conversor.

### 6.3 Dispositivo Teltónika.

El FM3101 es un terminal con conectividad GSM y GPS, capaz de adquirir las coordenadas del dispositivo y transmitir las vía la red GSM. Este equipo es especialmente adecuado para aplicaciones donde se requiere la localización de sistemas remotos, pudiendo realizar seguimiento de objetos remotos (vehículos, máquinas...) fácilmente.

Dispone de una batería recargable y un controlador especial para gestión de la alimentación, de modo que puede funcionar durante un período de 3-6 horas sin una fuente de alimentación externa. Además este dispositivo soporta multitud de conexiones, de modo que se puede elegir el modo de transmisión según la velocidad, fiabilidad, simplicidad o precio. Este dispositivo GSM/GPS tiene una arquitectura abierta, totalmente programable, y puede realizar tareas en sistemas remotos, ya que tiene capacidad de medición, control y actuación, con sus interfaces analógico/digital.

En caso de perder conectividad, el FM3101 puede almacenar datos durante una semana, y una vez la conexión se reestablezca, el dispositivo enviará todos los datos almacenados vía GPRS, de manera que la información almacenada, tal como parámetros medidos del entorno, o coordenadas geográficas, no se pierde.

Su carcasa de aluminio le hace ser muy robusto y adecuado para su instalación en entornos hostiles.

Es una solución avanzada y versátil a un precio muy razonable.

#### Características principales:

- Soporta las siguientes portadoras GSM:
  - EDGE clase 6 (hasta 177,4 kbps);
  - GPRS clase 10 (hasta 85,6 kbps);
  - HSCSD (hasta 43,2 kbps);
  - CSD (hasta 14,4 kbps);
  - SMS (texto/datos);
  - USSD (datos).
  
- Soporta multitud de conexiones, de modo que se puede elegir el modo de transmisión según la velocidad, fiabilidad, simplicidad o precio.
- Banda dual:
  - Versión europea (y asiática) - 900 MHz / 1800 MHz.
  - Versión americana - 850 MHz / 1900 MHz.

- La carcasa de aluminio hace que el dispositivo sea muy resistente y adecuado para instalar en entornos no delicados, como pueden ser entornos industriales, en vehículos o barcos, o expuesto a las condiciones meteorológicas.
- Batería interna recargable con controlador de carga.
- El FM3101 tiene tres entradas digitales, tres salidas digitales y otras tres entradas analógicas, que pueden ser usadas para llevar a cabo tareas de medición, control y actuación en sistemas remotos, como medición de parámetros ambientales y activación de sistemas de mantenimiento, control de parámetros en vehículos, activación de puertas, o envío de alarmas cuando un evento se produzca.
- El FM3101 tiene integrado el protocolo 1-Wire® I/O protocol para medición de la temperatura o identificación mediante código.
- Dispone de interfaz serie RS232 que puede ser usada para adquisición de datos de periféricos externos. Precisamente en este proyecto usaremos este puerto para conectar el FM3101 con el gateway que recoge la información de los sensores.
- El FM3101 es un dispositivo de arquitectura abierta totalmente programable, de modo que si se quiere realizar una tarea especial, se pueden programar las funciones necesarias adaptando el dispositivo a nuestras necesidades, escribiendo un programa MIDLET en java J2ME. Además los protocolos TCP/IP UDP están implementados en el dispositivo.
- La función AutoPIN introduce el código PIN de la tarjeta SIM cada vez que el dispositivo se enciende.
- Funciones de voz (opcional).
- 3 indicadores LED indicators: “Power”, “Estado ” and “Navigate”.
- Se incluye antena GSM y cables de conexión.



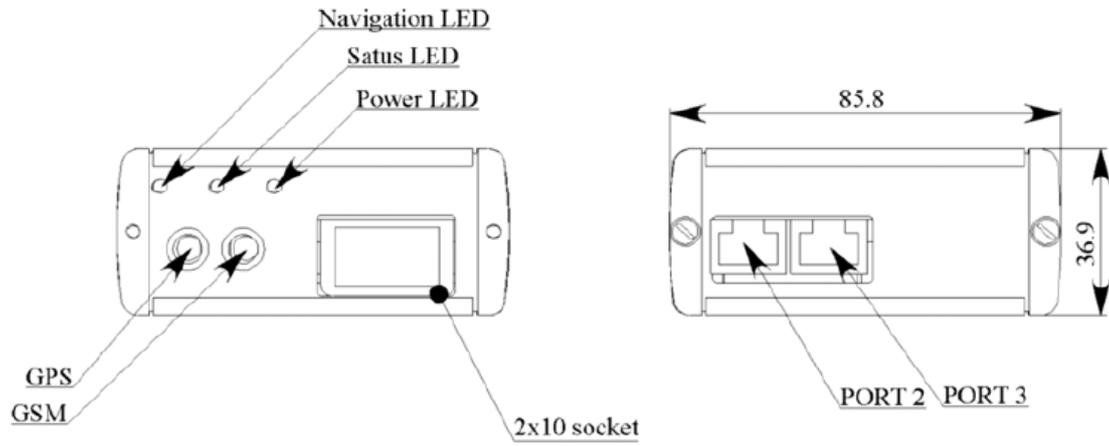


Figura 6.7. Dispositivo FM3101 y conexiones externas.

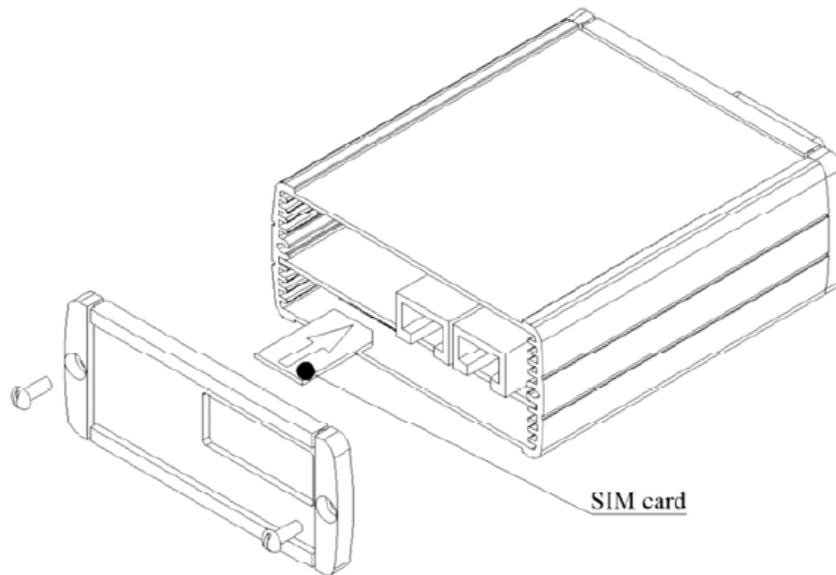


Figura 6.8. Introducción de la tarjeta SIM

*Arquitectura interna del FM3101*

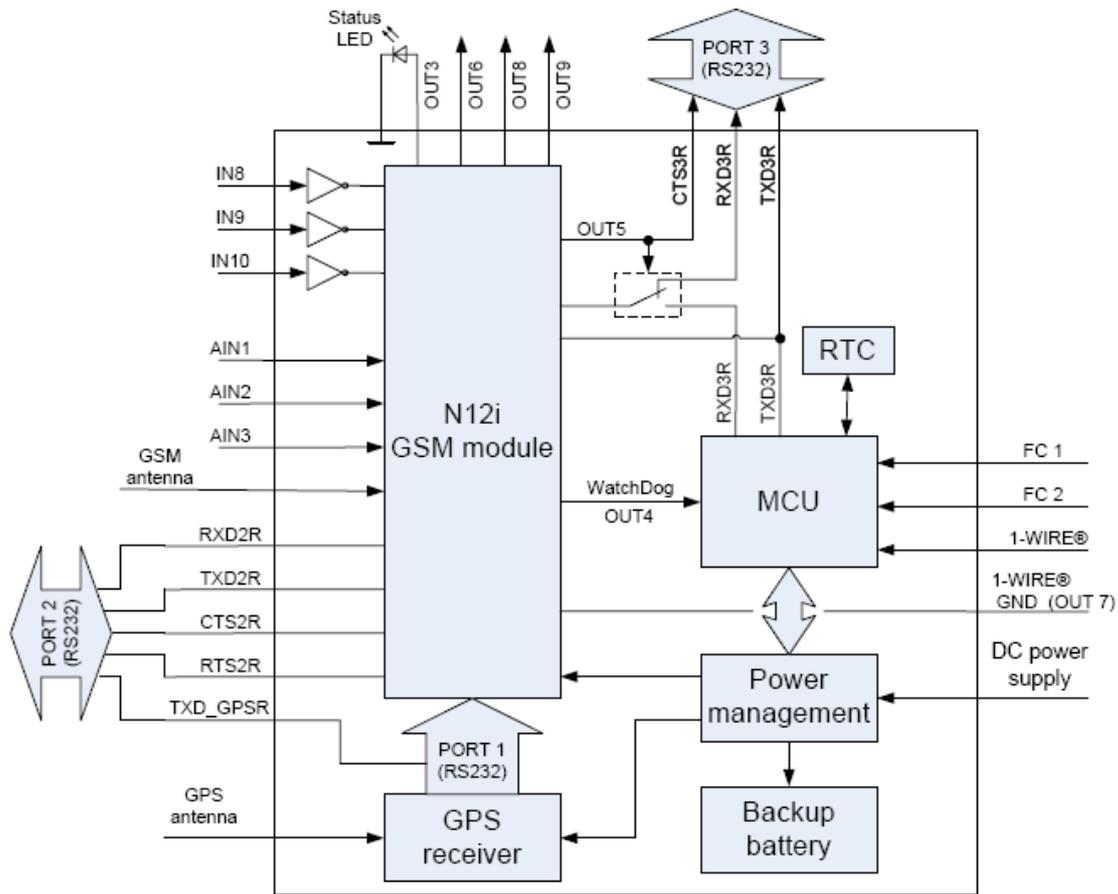
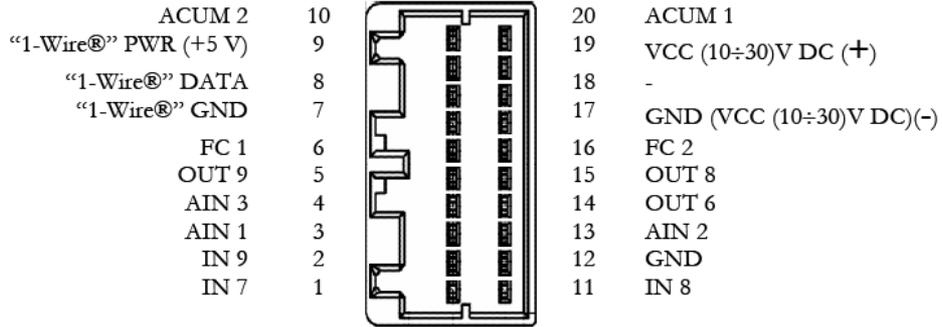


Figura 6.9. Esquema interno del FM3101

MCU: Micro controller unit. Es el corazón del dispositivo. Aquí se encuentra instalada la Máquina Virtual de Java, y es donde se carga el programa que diseñamos para controlar todos los componentes del equipo, el MIDlet. El módem nokia N12i recibe las 6 entradas (3 analógicas y 3 digitales) de que dispone el equipo, y controla a su vez las tres salidas digitales. El puerto 2 se usa para configuración del equipo y para cargar el MIDlet implementado, y el puerto 3 es de entrada/salida y es controlado por el microcontrolador y será el que usemos para comunicar el dispositivo con la red de sensores.

A continuación vemos la disposición de pines en el conector “2x10 Socket”.



Pin Nr.	Pin Name	Description
1	IN 7	Digital input, channel 7. *
2	IN 9	Digital input, channel 9. *
3	AIN 1	Analog input, channel 2. Input range: 0-10V DC **
4	AIN 3	Analog input, channel 3. Input range: 0-10V DC **
5	OUT 9	Digital output. Channel 9. Open collector output. Max. --- 500mA.
6	FC 1	Fuel Counter (Quick counter digital input, channel 1)
7	"1-Wire®" GND	Digital output Channel 7, used for Dallas 1-Wire® devices GND (purpose: output could be OFF-ON-OFF to reset device)
9	"1-Wire®" PWR (+5 V)	+ 5 V output (not only) for Dallas 1-Wire® devices. (max 100mA)
10	ACUM 2	This pin is used connected with pin ACUM 1. Function of those pins is to disconnect the internal accumulator during shipment or storage. When ACUM 1 and ACUM 2 are connected, the internal accumulator is on, while disconnected - the internal accumulator is off.
11	IN 8	Digital input, channel 8. *
12	GND	Ground pin.

13	AIN 2	Analog input, channel 2. Input range: 0-10V DC **
14	OUT 6	Digital output. Channel 6. Open collector output. Max --- 500mA.
15	OUT 8	Digital output. Channel 8. Open collector output. Max --- 500mA.
17	GND	Ground pin. (10÷30)V DC ( - )
18	-	-
19	+ (10÷30) V	Power supply for module. Power supply range (10...30) V DC Energy consumption: GPRS: --- 400 mA r.m.s Max., Nominal: --- 100 mA r.m.s..
20	ACUM 1	This pin is used connected with pin ACUM 2. Function of those pins is to disconnect the internal accumulator during shipment or storage. When ACUM 1 and ACUM 2 are connected, the internal accumulator is on; while disconnected, the internal accumulator is off.

\* Input is inverted.

\*\* N12i module have input divider of 4.08, that meant if input equals to 10V, then N12i will receive 2,45V

Figura 6.10. Conector "2x10" y leyenda.

Los pines que nos interesarán son 19 y 17 (alimentación y masa), 5, 14 y 15 (las tres salidas digitales),

Además, monitorizaremos también el valor de las entradas analógicas( pines 3, 4 y 13 ), enviando el valor en la trama junto con el resto de la información.

Disposición de pines en los puertos 2 y 3:

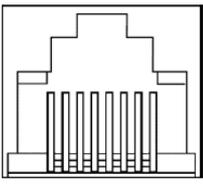
RJ-45 socket		PORT 2		PORT 3	
		Pin Nr.	Description	Pin Nr.	Description
		1	-	1	-
		2	TXD_GPS	2	-
		3	-	3	-
		4	GND	4	GND
		5	RXD2	5	RXD3
		6	TXD2	6	TXD3
		7	CTS2	7	RTS3
		8	RTS2	8	-

Figura 6.11. Disposición de pines en los puertos 2 y 3:

- Port 2. Conectado al puerto 3 del modem N12i. Este puerto se usa para configuración del equipo mediante protocolo M2M.
- Port 3. Conectado al puerto 3 del modem N12i. Es controlado por la aplicación Java.

### ***Configuración del equipo***

Para configurar el dispositivo, lo conectamos a un PC con interfaz RS232 mediante el puerto 2 y usamos el software Nokia Configurator. Tiene este aspecto:

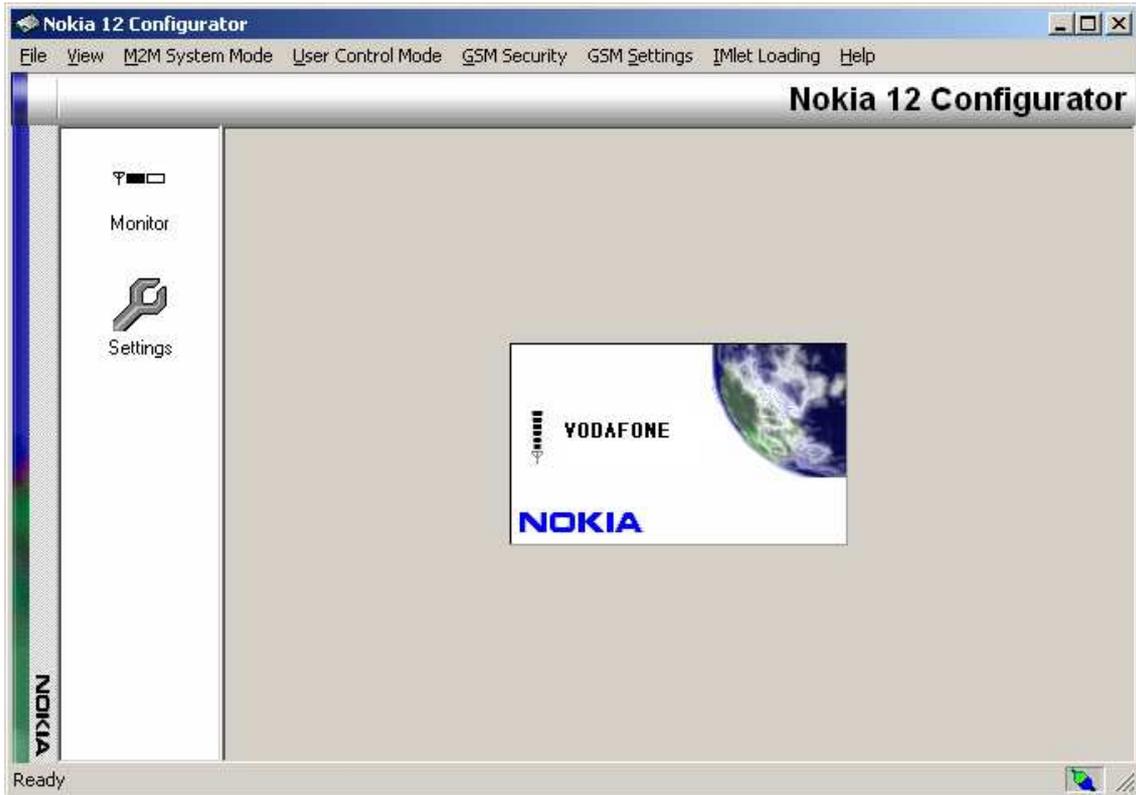
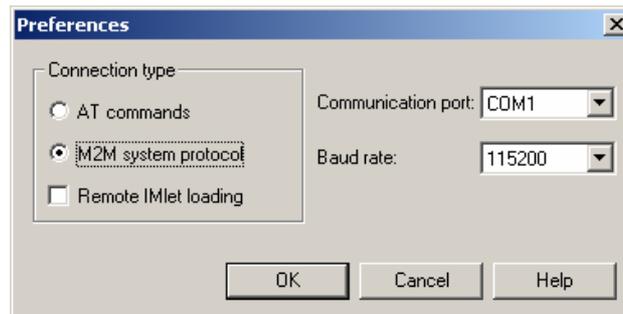


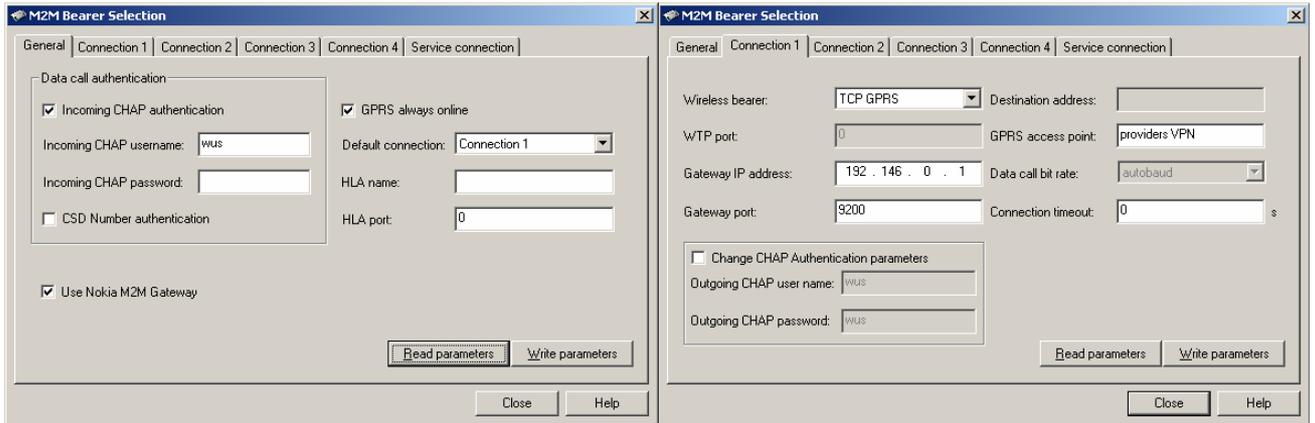
Figura 6.12. Nokia 12 Configurator

En la pantalla se muestra la operadora que da servicio al equipo (según la tarjeta SIM que se haya insertado), y la cobertura de que dispone el aparato.

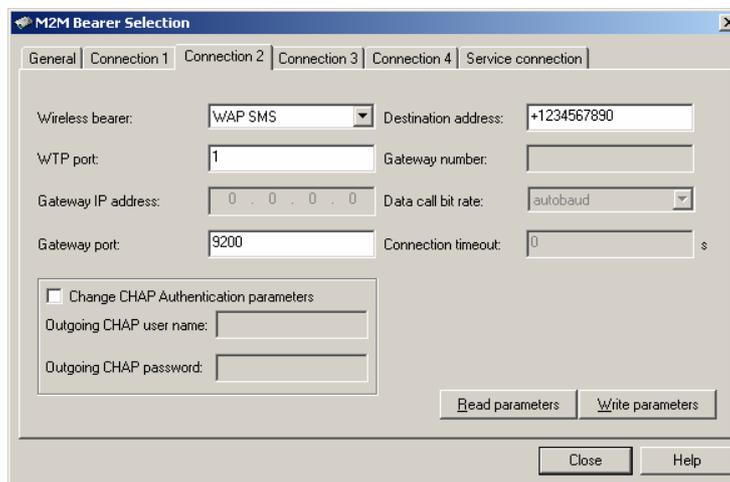
Se puede seleccionar el tipo de conexión que se va a usar (en este caso protocolo M2M), así como el puerto a usar y la velocidad de interconexión con el PC.



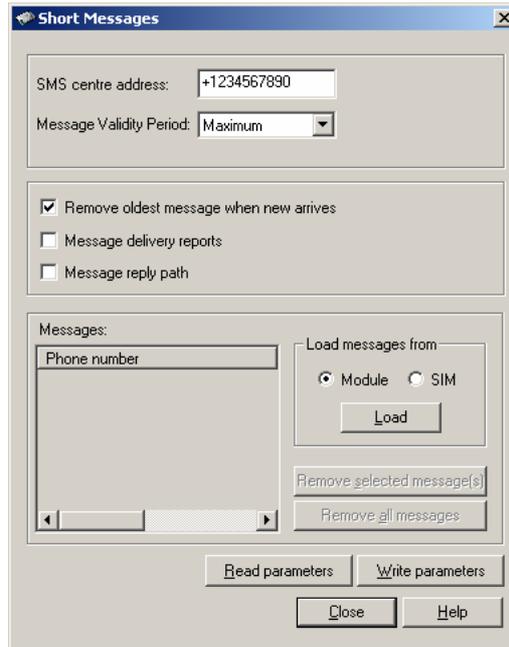
Dispone de múltiples ajustes de la conexión GPRS que se va a usar:



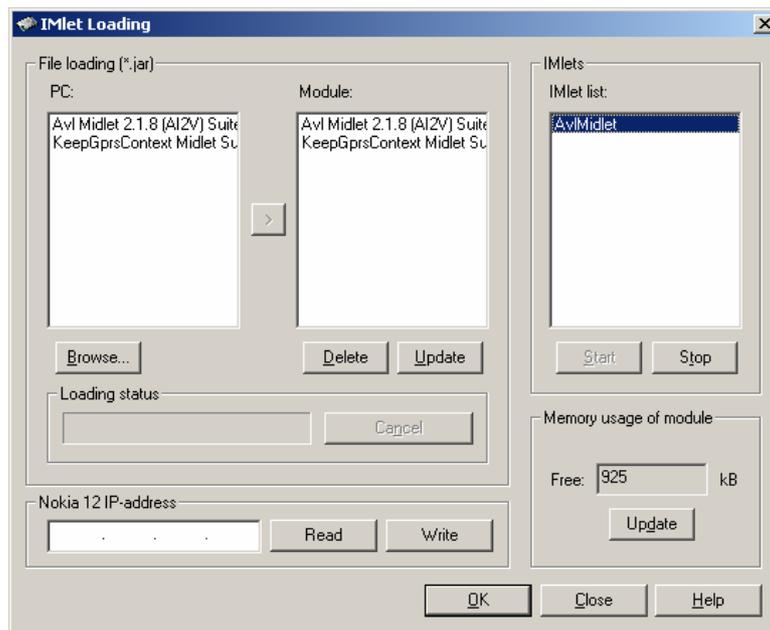
Además de la conexión GPRS que vamos a usar, se pueden definir múltiples conexiones, como por ejemplo una conexión WAP:



Para poder usar el servicio de mensajería SMS, hemos de configurar los parámetros como lo haríamos en un teléfono móvil, siendo el parámetro imprescindible el número del centro de mensajes de la operadora:



A continuación se muestra la ventana de carga de IMlets en el dispositivo. Se pueden cargar tantos IMlets como se quieran, siempre y cuando haya memoria disponible en el equipo, aunque sólo uno de ellos puede estar funcionando.



#### 6.4. Programa desarrollado en tecnología J2ME.

Como se explicó en el apartado de la tecnología J2ME, un MIDlet es una aplicación para dispositivos móviles, que consta de una estructura definida. En nuestro programa, habrá una clase principal que declarará varios threads o hilos de ejecución que correrán en paralelo, y que serán cíclicas, esto es, se ejecutarán repetidamente indefinidas veces. Se creará un hilo por cada tarea que debe realizar nuestro programa, de manera que se separarán las tareas. Para implementar toda la funcionalidad de nuestro MIDlet, se usarán las numerosas librerías disponibles en Java J2ME.

Las clases principales de nuestro MIDlet serán por tanto las siguientes:

<b>Programa</b>	<ul style="list-style-type: none"> <li>Inicializar watchdog de seguridad</li> <li>Abrir conexiones</li> <li>Leer datos de memoria permanente</li> <li>Crear temporizadores y tareas</li> <li>Arrancar temporizadores</li> </ul>
<b>RS232</b>	<ul style="list-style-type: none"> <li>Comprobar la recepción de una trama de los sensores</li> <li>Añadir información de GPS en ese momento</li> <li>Añadir valor de entradas y salidas digitales</li> <li>Insertar en almacén de tramas para envío</li> </ul>
<b>Read GPS</b>	<ul style="list-style-type: none"> <li>Inicialización de módulo GPS</li> <li>Adquisición de datos</li> <li>Construcción de la trama con los datos</li> </ul>
<b>IO</b>	<ul style="list-style-type: none"> <li>Adquisición de estado de las entradas y salidas digitales</li> </ul>
<b>Send_GPRS</b>	<ul style="list-style-type: none"> <li>Envío de las tramas que se encuentren almacenadas en el equipo</li> <li>Comprobación de datos de entrada GPRS</li> <li>Actualización de parámetros y reset del equipo</li> </ul>

A continuación se explicará el funcionamiento de cada clase más detalladamente, y se mostrarán algunos fragmentos de código interesantes.

La clase principal que implementa el MIDlet es Programa.java.

En ella se definen los distintos thread que se ejecutaran en el MIDlet. Estas tareas están temporizadas, y se van ejecutando organizadamente cuando el temporizador asociado a cada una de las tareas así lo indica.

```
public class Programa extends MIDlet
```

En esta clase se define el temporizador Watchdog de seguridad. Este temporizador se ejecuta ajeno al transcurso del programa, y si alcanza el fin de su cuenta, resetea el dispositivo. Nos será útil por si el dispositivo tiene algún fallo de ejecución y queda en estado irrecuperable. Cada vez que una tarea se ejecute correctamente, se reiniciará la cuenta del watchdog.

Lo primero que aparece en esta clase es, por tanto:

```
public static WatchdogTimer wd = new WatchdogTimer();
```

A continuación se definen los parámetros de funcionamiento por defecto de nuestro MIDlet. Son los referentes a tiempos de actualización de las coordenadas GPS del equipo, intervalo de envío por GPRS, tiempo de lectura por RS232, tiempo de actualización del estado de las entradas/salidas digitales del equipo, así como dirección IP destino y puerto al que se enviarán los datos mediante un socket HTTP al que se accederá por GPRS.

```
public String t_gps = "1500";  
public String t_gprs = "8000";  
public String t_rs232 = "1500";  
public String t_io = "1500";  
public static String ip = "62.43.188.94";  
public static String puerto = "1114";
```

Los tiempos se especifican en milisegundos.

El usuario también tiene la opción de introducir los parámetros en el archivo de manifiesto, que se incluye en un programa java empaquetado en .jar. De esta forma, el aspecto del archivo Manifest.mf sería el siguiente:

Manifest-Version: 1.0 MicroEdition-Configuration: CLDC-1.0 MIDlet-Name: Programa Created-By: 1.4.2_14-b05 (Sun Microsystems Inc.) Ant-Version: Apache Ant 1.7.0 MIDlet-Vendor: Javier Belando MIDlet-Version: 1.0.0 MicroEdition-Profile: MIDP-1.0	} Generado por el compilador
timer_gps: 20000 timer_gprs: 60000 timer_rs232: 20000 timer_io: 20000 ip: 62.43.188.94 puerto: 1114	} Introducido por el usuario

Y se obtendrían dichos parámetros del archivo de manifiesto mediante las siguientes líneas de código:

```
public String t_gps = this.getAppProperty("timer_gps");
public String t_gprs = this.getAppProperty("timer_gprs");
public String t_rs232 = this.getAppProperty("timer_rs232");
public String ip = this.getAppProperty("ip");
public String puerto = this.getAppProperty("puerto");
```

A continuación se implementa el método **StartApp( )**, que, como se comentó anteriormente, es el corazón de la aplicación.

```
protected void startApp() throws MIDletStateChangeException {
    wd.setTimeout(5 * 60); // Watchdog: 5 minutos

    try{
        //Espera para que el dispositivo se conecte a la red GSM
        Thread.sleep(30000);

        // Conexiones de lectura y escritura por RS232
        conn =
(StreamConnection)Connector.open("comm:3;baudrate=115200",Connector.RE
AD_WRITE);

        is = conn.openInputStream();
        os = conn.openOutputStream();

        // Se construye la dirección destino
        String ip = Programa.ip;
        String puerto = Programa.puerto;
        String url = "socket://" + ip + ":" + puerto;
```

Nuestro dispositivo dispone de una memoria permanente de almacenamiento, a la cual se puede acceder para almacenar datos de forma invulnerables a fallo de alimentación

del equipo. La usaremos para almacenar los parámetros de funcionamiento anteriormente descritos, de manera que el programa comprobará en su inicio si dispone de parámetros escritos en la memoria, y los usará para arrancar las tareas. En caso de que la memoria se encuentre vacía, usará los parámetros por defecto introducidos en las líneas anteriores. El usuario podrá enviar nuevos parámetros al equipo mediante una instrucción GPS, los cuales se almacenarán en dicha memoria, y serán los que se usen a partir de ese momento. A dicha memoria se accede mediante un objeto *RecordStore* de la siguiente manera:

```

if( RecordStore.listRecordStores()!=null){
    RecordStore rs = RecordStore.openRecordStore("aaa", false);

    Thread.sleep(1000);
    int i = rs.getNumRecords();

if(i>0){
    String encoding = "US-ASCII" ;
    byte[] a=rs.getRecord(1);
    byte[] b=rs.getRecord(2);
    byte[] c=rs.getRecord(3);
    byte[] d=rs.getRecord(4);
    byte[] f=rs.getRecord(5);

    if (a!=null) t_gps = new String(a, encoding);
    if (b!=null) t_gprs = new String(b, encoding);
    if (c!=null) t_rs232 = new String(c, encoding);
    if (d!=null) ip = new String(d, encoding);
    if (f!=null) puerto = new String(f, encoding);

    rs.closeRecordStore();
    }
}

```

A continuación se crean las tareas que se ejecutan en el MIDlet:

```

// Declaración de tareas y temporizadores asociados

Timer timer_Read_GPS = new Timer();
TimerTask Read_GPS = new Read_GPS();

Timer timer_Send_GPRS = new Timer();
TimerTask Send_GPRS = new Send_GPRS();

Timer timer_RS232 = new Timer();
TimerTask RS232 = new RS232();

Timer timer_SMS = new Timer();
TimerTask SMS = new SMS();

Timer timer_io = new Timer();
TimerTask io = new io();

Integer t1 = Integer.valueOf(t_gps);
Integer t2 = Integer.valueOf(t_gprs);
Integer t3 = Integer.valueOf(t_rs232);
Integer t4 = Integer.valueOf(t_io);

long f1 = t1.longValue();

```

```

long f2 = t2.longValue();
long f3 = t3.longValue();
long f4 = t4.longValue();

timer_Read_GPS.schedule( Read_GPS, 30000, f1 );
timer_Send_GPRS.schedule( Send_GPRS, 32000, f2 );
timer_RS232.schedule( RS232, 28000 );
timer_io.schedule( io, 28000, f4 );
timer_SMS.schedule(SMS, 15000);

}
catch (Exception e) {
}

```

Finalmente, declaramos los métodos obligatorios que debe tener todo MIDlet, aunque no los usemos en este caso:

```

protected void pauseApp() {
    // Not needed.
}

protected void destroyApp(boolean unconditional)throws
MIDletStateChangeException {
    // Not needed.
}
}

```

A continuación veremos la clase:

```

public class RS232 extends TimerTask {

```

Esta clase gestiona el puerto 3 del FM3101. Se ocupa de captar a través de éste puerto las tramas recogidas por la estación base MIB510 receptora de los mensajes de los nodos.

```

public void run() {

    while(true){

        try{

```

Se comprueba que el mensaje contenido en el buffer de entrada sea de la longitud adecuada y comience y acabe con 7E, como marca el formato estándar de los sensores:

```

        if((received = Programa.is.available() )>=66){

            data=new byte[received];
            /**
             * Read received bytes from the serial port
             * to byteArray data from offset 0 and as much as
             * data received.
             */

```

```

Programa.is.read(data,0,received);

rs232_received = new String(data, encoding);

if(rs232_received.startsWith("7E") &
rs232_received.endsWith("7E")){

```

Se concatena el mensaje recibido con la trama disponible en la clase Read\_GPS, la cual contendrá las coordenadas geográficas actuales y el estado de las salidas/entradas digitales, y se inserta en un “almacén” de tramas, que no es más que una cola donde se van depositando las tramas a enviar. La tarea Send\_GPRS se encargará de obtener las tramas de éste almacén y enviarlas por GPRS.

```

if(rs232_received.length()==66){
Read_GPS.trama = Read_GPS.trama + rs232_received;

Read_GPS.almacen.insertar(Read_GPS.trama);

    }
}

} catch (Exception e) {
}

```

Y por último, si se alcanza el final de este proceso, se reinicia el watchdog de seguridad si todo ha ido bien.

```

Programa.wd.resetTimer();
}
}

```

```

public class Read_GPS extends TimerTask {

```

Esta clase gestiona la recepción de la información del GPS. Cada ejecución del thread equivale a la adquisición de nuevas coordenadas geográficas.

Creamos el almacén de tramas:

```

public static Almacen_Tramas almacen = new Almacen_Tramas();

public void run() {

    try {
        // Get refrence to the GPS module.
        String url =
"corbaloc::127.0.0.1:19740/ORB/OA/IDL:gps/GpsModule:1.0";
        org.omg.CORBA.Object ref = SMS.orb.string_to_object(url);
        GpsModule gpsModule = GpsModuleHelper.narrow(ref);

        String url2 =
"corbaloc::127.0.0.1:19740/ORB/OA/IDL:ET:1.0";

```

```
org.omg.CORBA.Object ref2 = SMS.orb.string_to_object(url2);
ET et = ETHelper.narrow(ref2);
String imei = et.IMEI();
```

Hemos obtenido también el IMEI del dispositivo, para integrarlo en la trama e identificar los mensajes del equipo.

Hacemos una llamada al módulo GPS para que nos devuelva la información:

```
try{
    data = gpsModule.getGpsData();
    }
    catch(NoGpsDataAvailableException ngps){
        nodata = true;
    }
    //Velocidad
    velocidad = String.valueOf(data.speed.speed);

    //satélites visibles
    num_sat = String.valueOf(data.visibleSatellites);

    //Altura
    altitud = String.valueOf(data.alt);

    //Rumbo
    rumbo = String.valueOf(data.speed.angle);

    //Fecha y hora
    long segundos = data.acquisitionTime;
    String timest = String.valueOf(segundos*1000);
```

Y constituimos la trama añadiendo a éstos parámetros las variables que representan el estado de las salidas/entradas analógicas y digitales, que se encuentran en la clase *IO*

```
trama =
"$"+imei+", "+timest+", "+lati+", "+longi+", "+velocidad+", "+num_sat+", "+i
o.A1+", "+io.A2+", "+io.estado+", ";
}
```

Si la información del gps no fuera válida (el sistema puede estar sin cobertura GPS momentáneamente), constituimos una trama con coordenadas vacías, en la cual, en lugar de indicar la fecha y hora adquirida del satélite, colocamos la fecha y hora actual del dispositivo (se sincroniza con la red GSM y actualiza su reloj periódicamente).

```
else{

    long segundos = System.currentTimeMillis();
    String timest = String.valueOf(segundos);

    String lati = "0";
    String longi = "0";
    trama =
"$"+imei+", "+timest+", "+lati+", "+longi+", "+0+", "+0+", "+io.A1+", "+io.A2
+", "+io.estado+", ";

}
```

Finalmente, reseteamos el watchdog:

```
Programa.wd.resetTimer();
}
```

```
public class io extends TimerTask {
```

Esta clase monitoriza el valor actual de las entradas y salidas (I/O) digitales/analógicas que posee el equipo:

Codificaremos el valor de las tres entradas y las tres salidas digitales en un único byte para acortar la trama y optimizar la transmisión, en una variable llamada *estado*.

```
public static int estado=0;
public static int A1 =0;
public static int A2 =0;

public void run() {
    try{

        // Entradas digitales
        boolean Din7 = IOControl.getInstance().getDigitalInputPin(7);
        boolean Din8 = IOControl.getInstance().getDigitalInputPin(8);
        boolean Din9 = IOControl.getInstance().getDigitalInputPin(9);

        // Salidas digitales
        boolean Dout6 = IOControl.getInstance().getDigitalOutputPin(6);
        boolean Dout8 = IOControl.getInstance().getDigitalOutputPin(8);
        boolean Dout9 = IOControl.getInstance().getDigitalOutputPin(9);

        estado = 0;

        // LECTURA
        if (Dout9 == true){ estado =estado+32;}// si V=0, false
        if (Dout8 == true){ estado =estado+16;}// V=+5, true
        if (Dout6 == true){ estado =estado+8;}
        // Las entradas se encuentran en lógica negativa
        if (Din9 == false){ estado =estado+4;}// V = +5, false
        if (Din8 == false){ estado =estado+2;}// V = 0, true
        if (Din7 == false){ estado =estado+1;}

        //      MSB                                LSB
        //      [Dout9 Dout8 Dout6 Din9 Din8 Din7 ];

        //Entradas analógicas
        A1 = IOControl.getInstance().getAnalogInputPin(1);
        A2 = IOControl.getInstance().getAnalogInputPin(2);
```

Las entradas analógicas nos servirán como multímetro, para medir tensiones con una precisión de milivoltios.

```
catch (Exception e) {
    }
}
```

```
public class Send_GPRS extends TimerTask {
```

Mediante esta clase se realiza el envío de las tramas almacenadas. En cada ejecución de este thread, se vacía el almacén, enviando secuencialmente todas las tramas disponibles. Si por alguna razón no fuera posible el envío por fallos de cobertura GSM u otras causas, las tramas se almacenarían en un almacén de “no enviadas”, reenviándose cuando el servicio esté nuevamente disponible.

```
Almacen_Tramas NoEnviadas = new Almacen_Tramas();
```

```
public void run() {
    try {
        while(!NoEnviadas.isEmpty()){
            Read_GPS.almacen.insertar(NoEnviadas.retirar());
        }
        while (!Read_GPS.almacen.isEmpty()){
```

Abrimos las conexiones si se encuentran cerradas:

```
        if (Programa.conn2==null){Programa.conn2 =
(StreamConnection) Connector.open(url);}
        if (Programa.out2==null){Programa.out2 =
Programa.conn2.openOutputStream();}
        if (Programa.in2==null){Programa.in2 =
Programa.conn2.openInputStream();}

        String msg = String.valueOf(Read_GPS.almacen.retirar());

        try{
            Programa.out2.write(msg.getBytes());
            Thread.sleep(1000); //Dejamos un segundo de
separación entre el envío de dos tramas para prevenir errores.
```

Después del envío, gestionamos los datos de entrada que ha podido recibir el equipo por GPRS, que estarán almacenados en un buffer.

```
int received = Programa.in2.available();
if(received!=0){
    byte[] data=new byte[received];
    Programa.in2.read(data,0,received);
```

```
String GPRS_received = new String(data, encoding);
```

Llamamos al método que analiza dichos datos de entrada:

```
MessageObserverImpl.procesa_mensaje(GPRS_received); }
}
```

Se ha pensado la programación del dispositivo para que se puedan enviar comandos de configuración de los parámetros, comandos para ordenar un reset del dispositivo, o para fijar el valor de una de las salidas digitales y actuar sobre un posible autómeta.

A continuación se detalla el formato de dichos comandos y las funciones que realizan:

Comando	Formato	Ejemplo
Reset del dispositivo	<b>reset</b>	reset
Configuración de parámetros de ejecución	<b>program</b> fgps UUUUU fgprs VVVVV frs232 WWWW fio XXXXX ip YYY.YYY.YYY.YYY puerto ZZZZ	program fgps 20000 fgprs 60000 frs232 20000 fio 20000 ip 62.43.188.94 puerto 1114
Set de Outputs Digitales	<b>Dout</b> pin ON/OFF	Dout 6 ON

El código del método que analiza estos datos es el siguiente:

```
public static void procesa_mensaje(String msg){
try{
```

Si queremos enviar una instrucción al dispositivo para que se resetee, deberá contener reset como mensaje:

```
String clave = "reset";
if (msg.equals(clave)) {
SMS.device.reset();
}
```

Si el mensaje recibido es de programación de nuevos parámetros en el equipo, se llevarán las siguientes acciones a cabo:

```
String clave2 = "program";
if (msg.substring(0,7).equals(clave2)) {
String p1 =getValor(msg, "fgps");
String p2 =getValor(msg, "fgprs");
String p3 =getValor(msg, "frs232");
String p4 =getValor(msg, "ip");
String p5 =getValor(msg, "puerto");
```

Actualizamos los parámetros recibidos en la memoria permanente del equipo:

```
if( RecordStore.listRecordStores()!=null){
RecordStore.deleteRecordStore("aaa");}
RecordStore rs2 = RecordStore.openRecordStore("aaa", true);
```

```

        if(p1!=null){rs2.addRecord(p1.getBytes(), 0,
(p1.getBytes()).length);Thread.sleep(1000);}
        if(p2!=null){rs2.addRecord(p2.getBytes(), 0,
(p2.getBytes()).length);Thread.sleep(1000);}
        if(p3!=null){rs2.addRecord(p3.getBytes(), 0,
(p3.getBytes()).length);Thread.sleep(1000);}
        if(p4!=null){rs2.addRecord(p4.getBytes(), 0,
(p4.getBytes()).length);Thread.sleep(1000);}
        if(p5!=null){rs2.addRecord(p5.getBytes(), 0,
(p5.getBytes()).length);Thread.sleep(1000);}

        byte[] a=rs2.getRecord(1);
        byte[] b=rs2.getRecord(2);
        byte[] c=rs2.getRecord(3);
        byte[] d=rs2.getRecord(4);
        byte[] f=rs2.getRecord(5);

        rs2.closeRecordStore();
        Thread.sleep(1000);

```

Y reseteamos el dispositivo para que funcione con los nuevos parámetros:

```

        SMS.device.reset();

    }

```

Por último, podemos enviar un mensaje para conmutar una de las tres salidas digitales de que dispone nuestro equipo. Es aquí precisamente donde se ejerce la capacidad de actuación del sistema.

```

String clave3 = "Dout";
    if (msg.substring(0,4).equals(clave3)){
        int pin = Integer.parseInt(msg.substring(5,6));

        if (msg.substring(7,9).equals("on")){

            IOControl.getInstance().setDigitalOutputPin(pin,true)
            ;}
        if (msg.substring(7,10).equals("off")){
            IOControl.getInstance().setDigitalOutputPin(pin,false);
        }
    }
}
    }
    catch(Exception e){

    }

}
}

```

Para aumentar las posibilidades de nuestro equipo, se ha implementado un manejador de mensajes SMS, de modo que nuestro sistema sensor será capaz de enviar SMS a un destino alertando de posibles valores anómalos de las magnitudes sensorizadas, así como recibir órdenes también por SMS, al igual que se ha visto para instrucciones recibidas por GPRS.

El código implementado para dotar al sistema de dicha funcionalidad es el siguiente:

```

public class SMS extends TimerTask {
    static ORB orb;
    static Device device;
    public void run() {

        try {

            Hashtable props = new Hashtable();
            props.put("com.nokia.m2m.orb.UseM2MGateway", "no");
            orb = ORB.init(null, props);

            // Get the RootPOA.

            org.omg.PortableServer.POA rootPoa;
            rootPoa =
POAHelper.narrow(orb.resolve_initial_references("RootPOA"));

            //
            // Create a new POA. Use persistent policies
            // so that servants activated under
            // this POA will always get the same IOR. This
ensures
            // that when the IMlet is restarted
            // after Nokia 12 module reboot, the new registration
will
            // overwrite the existing registration.
            //
            Policy[] policies =
                new Policy[] {

                    rootPoa.create_id_assignment_policy(IdAssignmentPolicyValue.USER
_ID),

                    rootPoa.create_id_uniqueness_policy(IdUniquenessPolicyValue.UNIQ
UE_ID),

                    rootPoa.create_lifespan_policy(LifespanPolicyValue.PERSISTENT)};

            //
            // Name the new POA as 'SmsObserverPOA' this name
will be part of
            // the listener objects object key. Object key can be
freely chosen.
            //
            POA observerPoa =
rootPoa.create_POA("SmsObserverPOA", rootPoa.the_POAManager(),
policies);

            // Resolve the Embedded Terminal object. This is
needed to send response shortmessages.
            String etUrl =
"corbaloc::127.0.0.1:19740/ORB/OA/IDL:ET:1.0";
            org.omg.CORBA.Object etRef =
orb.string_to_object(etUrl);
            ET et = EThelper.narrow(etRef);

```

```

        String deviceUrl =
"corbaloc::127.0.0.1:19740/Device";
        org.omg.CORBA.Object deviceRef =
orb.string_to_object(deviceUrl);
        device=DeviceHelper.narrow(deviceRef);

        //
        // Initialize the servant object. Give embedded
terminal to the observer so that it can
        // use et methods to send response shortmessage back
to the sender. Object id can be freely
        // chosen.
        //
        MessageObserverImpl observerImpl = new
MessageObserverImpl(et);

        observerPoa.activate_object_with_id("SMObserver".getBytes(),
observerImpl);

        //
        // Activate the POAManager.
        // Activation starts request processing in all POAs
that
        // use this POAManager.
        //
        rootPoa.the_POAManager().activate();

        //
        // The observer is now ready to receive events.
        // Register the listener into the Module ORB.
        //

        // Get CORBA object reference from servant
        Object observerObj =
observerPoa.servant_to_reference(observerImpl);

        // Start to listen to the SignalQuality parameter.
        device.setEventObserver("SMS-RECEIVED-TEXT",
EventObserverHelper.narrow(observerObj));

        // Activate also SMS-RECEIVED-UNRESOLVED if you want
to receive binary short messages.
        // Same observer object can be set to listen multiple
different events
        device.setEventObserver("SMS-RECEIVED-UNRESOLVED",
EventObserverHelper.narrow(observerObj));

        orb.run();

    } catch (Exception e) {
        try{

            Programa.wd.resetTimer();
        }
        catch(Exception e1){
        }
    }
}
}

```

```

    public static ORB getORB() {
        return orb;
    }
}

```

De manera que cuando se reciba un mensaje SMS se ejecutará el siguiente método, y se analizará el mensaje recibido con el método *ProcesaMensaje* explicado anteriormente.

```

public void eventFired(String eventName, Any eventData) {
    try {
        // Use event name to resolve the type of received event
        if (eventName.equals("SMS-RECEIVED-TEXT")) {
            SMS_RECEIVED_TEXT msg =
SMS_RECEIVED_TEXTHelper.extract(eventData);

            phoneNumber = msg.phoneNumber;

            procesa_mensaje(msg.content);
        }
    }
}

```

Para el envío de mensajes SMS, se ha implementado el siguiente método, el cual recibirá como parámetros el mensaje a enviar y el número destino:

```

public void sendSMS(String phoneNumber, String text){
    try{
        IntHolder ih = new IntHolder();

        et.sendShortMessage(phoneNumber, text, ih);
    }
    catch(Exception e) {
    }
}

```

Con el programa desarrollado, el prototipo envía las siguientes tramas:

```

$,imei,timestamp,latitud, longitud, velocidad, numero de satélites
visibles, Analog input 1, Analog input 2, Estado entradas/salidas
digitales,datos sensor;

```

A continuación se muestra un ejemplo de las tramas enviadas por el prototipo:

```

$352540003270716, 1220634075000, +37.5852140, -
1.0759448, 0, 8, 3, 3, 0,

```

7E42FFFF117D5D16020002005401FF000000FFFF20080  
F3C496FF0001C7B51F7E

### 6.5. Estudio de consumo de energía de los motes.

Se calculó el tiempo en sleep en que debe mantenerse el mote para cumplir ciclos de trabajo del 1% y del 0,5% con una herramienta de cálculo suministrada por el fabricante de la red de sensores, Crossbow:

**Computes sleep time required to achieve duty cycle given number of TOS msgs to rcv/xmit while awake**

	value	units
<b>Specifications</b>		
Msg Size		40 bytes
Msg Preamble		16 bytes
Baud Rate	38400	baud
Duty Cycle	0,5	%
# of msgs to rcv/re-xmit during wake time		5 msg
<b>Computed Values</b>		
Time to xmit/rcv 1 msg		11,7 msec
Time to rcv/xmit all msgs		116,7 msec
Required sleep time to maintain duty cycle		23,22 sec

Usando la longitud total de paquete que nos da el fabricante (56 bytes), obtenemos que para un ciclo de trabajo del 0,5%, se debe mantener un tiempo en reposo entre transmisiones de al menos 23,22 segundos. Para el caso de un ciclo de trabajo del 1%, se obtuvo 46,44 segundos. Los sensores fueron programados con el programa *Surge Reliable*, según el cual no transmiten más de una trama por minuto, de modo que en cualquier caso cumplimos los tiempos mínimos de sleep.

**Duración de bacteria estimada frente a capacidad de corriente de la batería para los dos ciclos de trabajo estudiados (0,5% y 1%)**

<b>SYSTEM SPECIFICATIONS</b>					
	<b>Currents</b>		<b>Duty Cycles</b>		
	value	units	Model 1	Model 2	units
<b>Micro Processor (Atmega128L)</b>					
current (full operation)	6	ma	1	0,5	%
current sleep	8	ua	99	99,5	%

Radio			
current in receive	8 ma	0,75	0,4 %
current xmit	12 ma	0,25	0,1 %
current sleep	2 ua	99	99,5 %
Logger			
write	15 ma	0	0 %
read	4 ma	0	0 %
sleep	2 ua	100	100 %
Sensor Board			
current (full operation)	5 ma	1	0,5 %
current sleep	5 ua	99	99,5 %
Battery Specifications			
Capacity Loss/Yr	3 %		

Computed mA-hr used each hour	Model 1	Model 2
uP	0,0679	0,0380
Radio	0,0920	0,0460
Flash Memory	0,0020	0,0020
Sensor Board	0,0550	0,0300
<b>Total current(ma-hr) used</b>	<b>0,2169</b>	<b>0,1159</b>

Computed battery life vs battery size	Model 1	Model 2
---------------------------------------	---------	---------

Battery Capacity (ma-hr)	Battery Life - months	Battery Life - months
250	1,58	2,95
500	3,16	5,91
1000	6,32	11,82
1500	9,48	17,73
2000	12,63	23,63
3000	18,95	35,45

	After Battery Capcacity Loss	After Battery Capcacity Loss
250	1,57	2,93
500	3,13	5,82
1000	6,22	11,47
1500	9,25	16,94
2000	12,24	22,24
3000	18,05	32,31

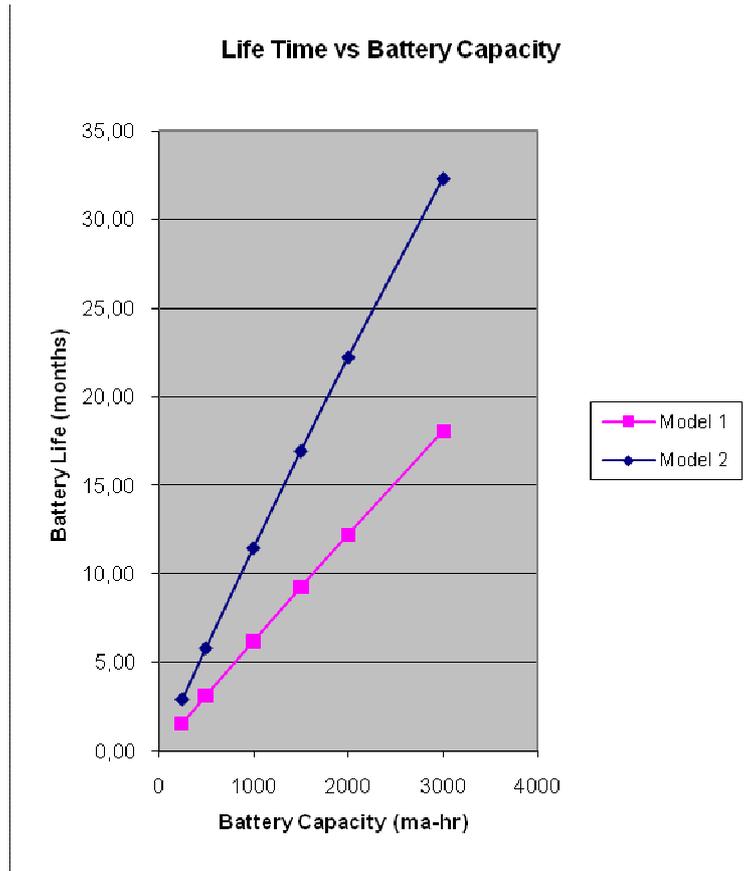


Figura 6.5. Autonomía de la batería.

Las baterías normales LR6 de 1,5V que usan los sensores suelen tener una capacidad de 2000 mA/hora, y vemos que la duración será de más de un año tanto para un ciclo de trabajo del 0,5% como para un 1%, con lo cual no será necesario el uso de baterías especiales, ya que se alcanza una cifra de duración bastante buena.

Además, se fabricó una fuente de alimentación única que suministrara las tensiones necesarias para alimentar la placa receptora MIB510 y el dispositivo Teltónika FM3101.

Los consumos de estos dispositivos son:

- Estación base receptora (MIB510):  
5V @ 50mA usando alimentación externa.
  
- Consumo del módulo FM3101  
Tensión de alimentación requerida: 9-30 V DC

Consumo de energía:

Modo reposo: 30mA

Consumo en transmisión GPRS: max RMS 590 mA, picos de hasta 2A.

Con estos datos se fabricó la fuente de alimentación. A continuación vemos una imagen de la fuente alimentando todos los componentes:

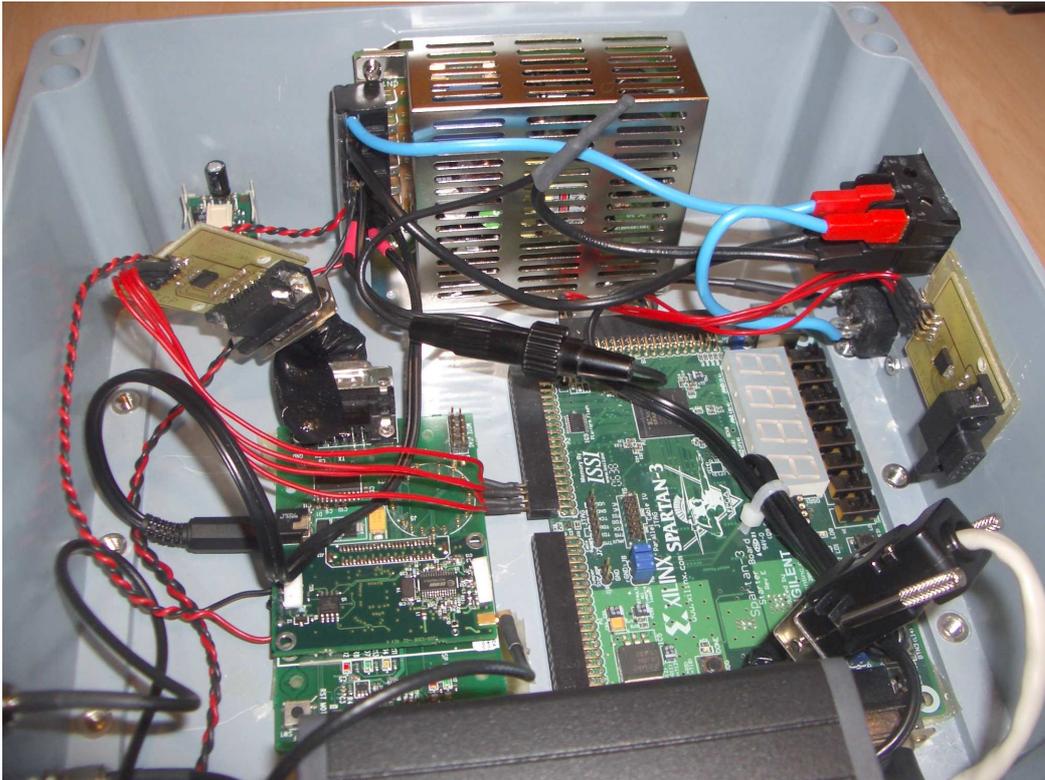


Figura 6.6. Fuente de alimentación

## 7. Conclusiones y futuras líneas de trabajo

En el primer capítulo se describieron los objetivos de este proyecto, éstos eran:

- Interconectar remotamente una red de sensores vía el sistema GSM/GPRS, a través del desarrollo de un prototipo compuesto por distintos elementos.

En el capítulo dos se ha hecho un profundo análisis de las redes de sensores inalámbricas y las posibilidades que éstas ofrecen hoy en día.

En el capítulo tres se ha hecho una completa revisión de los estándares de comunicación GSM y GPRS, conociendo en profundidad las funcionalidades de estos sistemas para optimizar la comunicación de nuestro equipo.

En el capítulo cuarto, se ha hecho un breve resumen del funcionamiento del sistema de posicionamiento que usa nuestro prototipo, el GPS.

En el capítulo quinto se ha analizado la tecnología J2ME, que nos permitió la programación y explotación de todas las posibilidades de que disponía el equipo FM3101.

En el capítulo sexto se ha explicado profundamente como se ha llevado a cabo el desarrollo de este prototipo, detallando la configuración de cada una de las partes (red de sensores, placa FPGA, FM3101).

Se partía de la base de un prototipo previo, el cual tenía ciertas limitaciones, las cuales se han podido mejorar bastante.

Se han aumentado las capacidades de conectividad del dispositivo, siendo ahora capaz de comunicarse por GPRS, además de por SMS.

También se ha dotado al sistema de geolocalización, por lo que es posible conocer su ubicación remotamente.

Se ha capacitado al sistema para que pueda recibir órdenes y ser gestionado y reprogramado en remoto, así como para actuar sobre otros dispositivos mediante la activación de líneas de tensión digitales.

Cabe mencionar, que éste prototipo fue probado en su desarrollo por una empresa del sector de las comunicaciones móviles, quedando ampliamente satisfecha con los resultados, e integrando éste dispositivo a sus procesos productivos.

Como **futuras líneas de trabajo** o aspectos a mejorar del prototipo desarrollado podríamos mencionar:

- Incorporación de nuevos sensores que midan mayor cantidad de parámetros.
- Búsqueda de aplicaciones comerciales para el prototipo desarrollado (Prevención de incendios, agricultura de precisión, control de aguas, control de contaminación...)
- Incorporación de alimentación autónoma del prototipo, mediante la inclusión de un sistema de células solares fotovoltaicas, para su uso en exteriores.
- Reducción del tamaño de los componentes del prototipo, ya que se ha desarrollado a nivel experimental, y cara a una fabricación en serie del producto podría ser optimizado y reducir sus costes.

## Bibliografía

[1] [http://www.euroresidentes.com/Blogs/avances\\_tecnologicos/2004/06/redes-de-sensores-sin-cable.htm](http://www.euroresidentes.com/Blogs/avances_tecnologicos/2004/06/redes-de-sensores-sin-cable.htm)

[2] Redes de sensores inalámbricos, [Jan Erik Frey](#), [Niels D. Aakvaag](#). [Revista ABB](#), ISSN 1013-3135, Nº 2, 2006, pags. 39-42. <http://dialnet.unirioja.es/servlet/articulo?codigo=2027180&orden=96623&info=link>

[3] <http://www.cs.utah.edu/classes/cs6935/papers/sensNet1.pdf>

[4] [http://www.cs.cmu.edu/~sensing-sensors/S2004/L2004-03-data\\_acquisition/L2004-03-motes-Jen\\_Morris.pdf](http://www.cs.cmu.edu/~sensing-sensors/S2004/L2004-03-data_acquisition/L2004-03-motes-Jen_Morris.pdf)

[5] *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, Mohammad Ilyas and Imad Mahgoub. CRC Press LLC. ISBN: 0-8493-1968-4

[6] [http://www.ewh.ieee.org/soc/icss/pdf/haenggi\\_martin\\_lecture1.pdf](http://www.ewh.ieee.org/soc/icss/pdf/haenggi_martin_lecture1.pdf)

[7] [http://www.i3a.uclm.es/documentos/2/congresos/Congreso2\\_147\\_teresaolivaresmontespdf](http://www.i3a.uclm.es/documentos/2/congresos/Congreso2_147_teresaolivaresmontespdf)

[8] <http://www.xbow.com>

[9] <http://www.atmel.com>

[10] <http://usuario.cicese.mx/~cruiz/PWSN.pdf>

[11] Guía de usuario de Crossbow: “*Getting Started Guide*”. Agosto 2004.

[12] Guía de usuario de Crossbow: “*MPR-MIB Series User’s Manual*”. Agosto 2004.

[13] Guía de usuario de Crossbow: “*MTS-MDA Series User’s Manual*”. Junio 2006.

[14] Hoja de características: “*Mica2 Datasheet*”.

[15] Guía de usuario de Crossbow: “*Mote-View User’s Manual*”. Noviembre 2006.

[16] *Deciphering TinyOS Serial Packets*: <http://www.octavetech.com/pubs/TB5-01%20Deciphering%20TinyOS%20Serial%20Packets.pdf> [17][1] GSM, GPRS

& EDGE Performance. Evolution towards 3G/UMTS. Timo Halonen, Javier Romero y Juan Melero. Ed. John Wiley & Sons, Ltd. ISBN: 0-470-86694-2.

[17] *GSM and UMTS: The Creation of Global Mobile Communication*. Friedhelm Hillerbrand. Ed. John Wiley & Sons, Ltd. ISBN: 0-470-84322-5

[18] <http://www.info-ab.uclm.es/labelec/Solar/Comunicacion/>

Telefonia\_movil/index\_archivos/Page828.htm

[19] *GSM and UMTS. Evolution towards 3rd Generation Systems*. Zoran Zvonar, Peter Jung y Karl Kammerlander. Ed. Kluwe Academic Publishers. ISBN: 0-792-8351-6

[20] *GSM and Personal Communications Handbook*. Seigmund M. Redl, Matthias K. Weber y Malcolm W. Oliphant. Ed. Artech House, Inc. ISBN: 0-89006-957-3

[21] *GSM, cdmaOne and 3G Systems*. Raymond Steele, Chin-Chun Lee and Peter Gould. Ed. John Wiley & Sons, Ltd. ISBN: 0-471-49185-3.

[22] *GSM Networks: Protocols, Terminology, and Implementation*. Gunnar Heine. Ed. Artech House, Inc. ISBN: 0-89006-471-7.

[23] <http://fermat.eup.udl.es/~cesar/xc1/Treballs/GPRS-pres.ppt>

[24] *Comunicaciones Móviles*. Jose María Hernando Rábanos. Ed. Centro de Estudios Ramón Areces. ISBN: 84-8004-231-1.

[25] [http://www.it.uc3m.es/~gavilan/apuntes/gsm\\_intro\\_2000.pdf](http://www.it.uc3m.es/~gavilan/apuntes/gsm_intro_2000.pdf)

[26] <http://leo.ugr.es/J2ME/MIDP/aplicaciones.htm>

[27] <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-132/paper18.pdf>

[28] <http://www.digilentinc.com>

[29] *Manual Spartan-3 Starter Kit Board User Guide*. Mayo 2005.

[30] <http://www.xilinx.com>

[31] [http://www.escet.urjc.es/~jmartine/CHS/\\_private/TutorialEDK.pdf](http://www.escet.urjc.es/~jmartine/CHS/_private/TutorialEDK.pdf)

[32] <http://www.ii.uam.es/~igonzale/recursos/>

[33] T-Box\_SMS\_Control\_Guide\_Amplified\_Engineering.pdf

[34] FM3101 user manual v1.3.pdf

[35] T-BoxN12R Quick Start Guide.pdf

[36] M2M\_System\_Protocol2\_Socket\_Interface\_User\_Manual\_v1\_0.pdf

[37] Nokia\_12\_Software\_Developers\_Guide\_v1\_0.pdf

[38] Nokia\_M2M\_Platform\_Configuration\_Manual\_v1\_0.pdf

[39] Nokia\_12\_IMlet\_Programming\_Guide\_v1\_0.pdf

[40] Nokia\_12\_Remote\_I\_O\_Control\_Guide\_v2\_0.pdf

[41] Tutorial\_MicroBlaze\_uCLinux\_jcra2004.pdf

[42] Proyecto Final de Carrera: *Desarrollo e implementación de una aplicación para la gestión de una red de sensores conectada remotamente vía el sistema GSM*. Concepción García Pardo. Universidad Politécnica de Cartagena. Marzo 2007.

