

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS  
INDUSTRIALES Y DE TELECOMUNICACIÓN**

**UNIVERSIDAD DE CANTABRIA**



***Proyecto Fin de Carrera***

**Sistema de monitorización de espacios de  
uso público mediante redes de sensores  
(Public spaces monitoring system using  
sensor networks)**

Para acceder al Título de

**INGENIERO DE TELECOMUNICACIÓN**

**Autor: Rafael Castillo Canto  
Marzo - 2013**



E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

## INGENIERÍA DE TELECOMUNICACIÓN

### CALIFICACIÓN DEL PROYECTO FIN DE CARRERA

**Realizado por:** Rafael Castillo Canto

**Director del PFC:** Jorge Lanza Calderón

**Título:** “Sistema de monitorización de espacios de uso público mediante redes de sensores”

**Title:** “Public spaces monitoring system using sensor networks”

**Presentado a examen el día: 26 de Marzo de 2013**

para acceder al Título de

## INGENIERO DE TELECOMUNICACIÓN

### Composición del Tribunal:

Presidente (Apellidos, Nombre): Irastorza Teja, José Ángel

Secretario (Apellidos, Nombre): Lanza Calderón, Jorge

Vocal (Apellidos, Nombre): Mirapeix Serrano, Jesús

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del PFC  
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Proyecto Fin de Carrera Nº  
(a asignar por Secretaría)

# AGRADECIMIENTOS

En primer lugar, quería agradecer el apoyo, la paciencia, la comprensión y la disposición ofrecidas por parte de mi director de proyecto, Don Jorge Lanza Calderón durante estos meses de trabajo. Igualmente a aquellas personas del departamento que cuando he necesitado de ayuda siempre la han prestado.

A todos esos compañeros de carrera, algunos convertidos en amigos, que han dado soporte y ayuda. Finalmente a mi familia que ha sido la auspiciadora incondicional de mis días en la universidad.

# ÍNDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	2
1.2	Objetivos .....	3
1.3	Contenidos de la Memoria.....	3
2	Estado del arte.....	5
2.1	Conceptos generales .....	6
2.1.1	Nodos sensores y actuadores.....	6
2.1.2	Arquitecturas.....	8
2.1.3	Topologías de red .....	9
2.2	Tecnologías de comunicación inalámbrica.....	10
2.2.1	Bluetooth.....	10
2.2.2	Wi-Fi .....	10
2.2.3	Ultra-Wide Band.....	11
2.2.4	Z-Wave .....	12
2.2.5	Wibree - Bluetooth Ultra Low Energy.....	12
2.2.6	IEEE 802.15.4.....	12
2.2.7	Zigbee .....	13
2.2.8	Otras especificaciones.....	13
2.2.9	Comparativa .....	13
2.3	IEEE 802.15.4.....	15
2.3.1	Estándar.....	15
2.3.2	Arquitectura .....	16
2.3.3	Capa física.....	17
2.3.4	Sub-Capa MAC.....	17
2.3.5	Mecanismos de robustez .....	18
2.3.6	Elementos de una red 802.15.4 .....	18
2.3.7	Hardware IEEE 802.15.4.....	19

2.4	Plataformas de nodos sensores .....	20
2.4.1	Crossbow .....	20
2.4.2	Waspmote .....	22
2.4.3	iSense .....	22
2.4.4	Arduino .....	23
2.4.5	Pingüino.....	24
2.4.6	Raspberry .....	24
2.4.7	Picaxe.....	25
3	Protocolo de Gestión de Nodos Sensores .....	26
3.1	Diseño del protocolo .....	26
3.1.1	arquitectura.....	27
3.1.2	operativa .....	28
3.1.3	Primitivas.....	31
3.2	Plataforma de desarrollo.....	34
3.3	Implementación y Test.....	38
3.3.1	Aproximación Módulos XBee .....	38
3.3.2	Acercamiento a la Plataforma Arduino .....	42
3.3.3	Primeros pasos .....	44
3.3.4	Evaluación del protocolo.....	47
4	Control de Presencia.....	50
4.1	Escenario .....	50
4.2	Tecnologías de desarrollo .....	52
4.2.1	Tarjeta Mifare, Tecnología NFC.....	52
4.2.2	REST .....	54
4.2.3	AJAX.....	55
4.2.4	JSON .....	55
4.3	Nodo Sensor Remoto .....	56
4.3.1	Configuración del Lector de Tarjetas .....	56
4.4	Nodo coordinador .....	58
4.4.1	Configuración Hardware .....	58
4.4.2	Funcionamiento en el Coordinador .....	60
4.4.3	Funcionamiento del nodo remoto .....	64

5	Conclusiones y Líneas Futuras .....	65
	Referencias.....	67
	Acrónimos .....	71

# ÍNDICE DE FIGURAS

Figura 2.1 Esquema genérico de un nodo sensor o actuador.....	7
Figura 2.2 Esquema de arquitectura centralizada .....	8
Figura 2.3 Esquema de arquitectura distribuida.....	8
Figura 2.4 Topología en estrella y red peer-to-peer .....	9
Figura 2.5 Modelo OSI.....	16
Figura 2.6 TelosB.....	21
Figura 2.7 MicaZ.....	21
Figura 2.8 Waspote .....	22
Figura 2.9 Módulos iSense .....	23
Figura 2.10 Arduino UNO .....	23
Figura 3.1 Red en estrella.....	27
Figura 3.2 Modos de operación .....	30
Figura 3.3 Arduino UNO .....	35
Figura 3.4 Arduino FIO .....	35
Figura 3.5 Escudo Ethernet .....	36
Figura 3.6 Escudo XBee .....	37
Figura 3.7 Módulo XBee.....	37
Figura 3.8 Tramas IEEE 802.15.4 .....	38
Figura 3.9 Trama TX16Request .....	39
Figura 3.10 Trama RX16Response.....	40
Figura 3.11 Trama Status .....	40
Figura 3.12 Software X-CTU .....	41
Figura 3.13 Placa XBee Explorer USB .....	42
Figura 3.14 IDE de Arduino .....	43
Figura 3.15 Sensor SHT15 .....	46
Figura 3.16 Diagrama funcionamiento Nodo Remoto.....	47
Figura 4.1 Sistema Completo .....	51
Figura 4.2 Sistema lector y tarjeta .....	52
Figura 4.3 Mapa de memoria.....	53

Figura 4.4 Lector de tarjetas .....	56
Figura 4.5 Trama PN532.....	57
Figura 4.6 Acoplo de escudos XBee y Ethernet.....	59
Figura 4.7 Diagrama de flujo .....	60
Figura 4.8 Página web .....	62

# ÍNDICE DE TABLAS

Tabla 2.1 Comparativa de tecnologías radio de transmisión.....	14
Tabla 3.1 Modos de funcionamiento .....	29
Tabla 3.2 Códigos .....	32
Tabla 3.3 Trama Configuration .....	32
Tabla 3.4 Trama Init .....	33
Tabla 3.5 Trama Request .....	33
Tabla 3.6 Trama Response .....	33
Tabla 3.7 Trama Notification.....	34
Tabla 3.8 Direccionamiento .....	39
Tabla 3.9 Parámetros módulo XBee.....	45
Tabla 4.1 Interfaces.....	57
Tabla 4.2 Pines Arduino vs Lector .....	57

# RESUMEN

A día de hoy, para el acceso a ciertos recintos como pueden ser bibliotecas, salas de estudio, o salas de informática es necesaria la identificación e inclusive la reserva previa de un sitio de manera que se haga un uso eficiente y justo de los medios disponibles que ofrece una institución.

Hacer un uso correcto y solidario de los espacios comunes debería ser práctica general entre de los usuarios de los mismos. Pero, a veces, se dan casos en que la utilización de esos recursos es excesiva y/o abusiva y está infrautilizada, sin tener en cuenta que hay otras personas que realmente harían el uso para el que está destinado. En el caso de entornos desatendidos estas malas prácticas son más habituales.

Basándose en estas premisas se pretende desarrollar un sistema que, además de intentar solventar estas situaciones, busque acometer un factor de comodidad al usuario y de ahorro energético para la institución. ¿Cómo? Empleando una red de sensores que sea capaz de detectar e identificar al usuario al tiempo que proporciona información adicional del entorno. De esta forma se logra el objetivo de la mejora de la gestión y uso de la infraestructura disponible.

La comunidad de la Universidad de Cantabria dispone de una tarjeta inteligente que, entre otros muchos usos, se emplea como identificador de la persona dentro de las instalaciones del campus universitario. Haciendo que la red de sensores, mediante este dispositivo, sea capaz de detectar la presencia de un usuario, esta información se podrá enviar a un centro de procesado, quien analizando la información temporal de uso permitirá al supervisor determinar el uso adecuado o indebido del puesto. Además, cada nodo sensor enviará adicionalmente información como la temperatura, humedad, etc. que de manera conjunta puede regular la sensación térmica de la habitación o aula a través del aire acondicionado o la calefacción.

# ABSTRACT

Nowadays, to access to places as libraries, study or computer rooms it is necessary credentials or even a previous reserve in order to achieve an efficient and fair use of the available resources that offers an institution

Making a correct and supportive use of common areas should be a general practise among the students. But, in some cases the utilization of these resources is excessive, abusive and put them under-use, without keep in mind there are other people who really need it. Sadly, these abnormal uses are more common in disregarding environments.

Under these premises the gold of this work is to develop a system which solves these situations, takes up a commodity factor to the user and saves energy to an institution. How? The answer is through sensor networks capable to detect and identify an user at the same time that shows additional information of the environment. In this way, the objective of improvement in the management and the use of the available infrastructure is achieved.

Cantabria's University community counts on an intelligent card; it is used as identifier between their owners in the campus facilities. So the idea is detect the student's presence through the sensor network using the intelligent card. The information receive will be sent to a processing data centre which will allow a supervisor to determine a correct or wrong use of a resource. Additionally, each sensor node, which become part of the sensor network, will send information about temperature or humidity which will help to regulate terminal sensation of a room through the air conditioner or the central heating.

# INTRODUCCIÓN

# 1

Ya antes de iniciar nuestra jornada acudimos a informarnos sobre el tiempo, el estado de las carreteras o la hora a la que pasará aproximadamente el siguiente autobús, entre otras. Todos estos datos se obtienen de sistemas que observan y monitorizan el entorno. Sistemas automatizados cableados o inalámbricos que cada vez están más presentes en nuestra vida cotidiana y nos prestan ayuda.

En muchos de esos sistemas son los sensores quienes juegan un papel importante (sensores de temperatura de estaciones meteorológicas, acelerómetros en móviles y tabletas, sensores infrarrojos en sistemas de alarma o detección de movimiento, etc.). En la mayoría de los casos, la función de estos sensores se limita a realizar una medida de una serie de parámetros del entorno, que se envía a un procesador central dentro del mismo dispositivo electrónico. No obstante, una nueva generación de sensores ha hecho aparición e incorporan, en un mismo dispositivo la alimentación, un módulo de comunicación y una inteligencia propia que permite en su conjunto comunicarse inalámbricamente con sus semejantes u organizarse a sí mismo dentro de su red. Las agrupaciones de este tipo de nodos inteligentes conforman lo que se ha denominado red de sensores inalámbricos (Wireless Sensor Networks, WSN). Estas redes consisten en una gran cantidad de sensores individuales que intercambian información entre sí y/o con un nodo central sin necesidad de cables y mediante un protocolo de comunicación pre-establecido.

Como se ha introducido, este tipo de nodo no sólo acomete la función del sensor sino que tiene la capacidad de procesar esos datos y así, además adquirir la funcionalidad de actuador. Ésto significa que el nodo a través de su capacidad de procesamiento e inteligencia puede desencadenar un proceso sobre un sistema externo en base a la información que ha sentido. Un ejemplo de ello podría ser el control de alumbrado en una ciudad o en un recinto, donde en función de la intensidad de luz medida se procedería a emplear iluminación artificial. De esta forma se puede extender el término WSN y hablar de redes inalámbricas de sensores y actuadores (Wireless Sensor & Actuator Networks, WSAN).

Las redes de sensores tienen su origen en iniciativas militares. Así la red de boyas SOSUS (Sound Surveillance System) [1] instaladas en los Estados Unidos durante la Guerra Fría para

detectar submarinos usando sensores de sonido se puede considerar como el predecesor de este tipo de redes. En la actualidad, su uso se ha extendió más allá de las aplicaciones para el ámbito militar y se utilizan en sistemas de eficiencia energética, entornos de alta seguridad (aeropuertos, centrales nucleares,...), sensorizado de parámetros ambiental e industriales, en medicina (tele asistencia, sensorizado de constantes vitales,...) y domótica.

Su diseño en cuestión de eficiencia e implementación las ha convertido en un área de investigación emergente en los últimos años, debido al gran potencial de estas redes para cubrir grandes áreas con un coste relativamente bajo.

Finalmente, lo que hace más atractivo a este tipo de redes es que aprovechando las capacidades que ofrece internet otorga una interfaz donde los datos, adquiridos por la red de nodos, pueden ser consultados en cualquier punto y momento. Adicionalmente, las capacidades funcionales de los nodos incorporan metodologías que permiten la configuración y gestión de la red de forma remota.

## 1.1 MOTIVACIÓN

A día de hoy, para el acceso a ciertos recintos como pueden ser bibliotecas, salas de estudio, o salas de informática es necesaria la identificación e inclusive la reserva previa de un sitio de manera que se haga un uso eficiente y justo de los medios disponibles que ofrece una institución.

Hacer un uso correcto y solidario de los espacios comunes debería ser práctica general entre de los usuarios de los mismos. Pero a veces se dan casos en que la utilización de esos recursos es excesivo y/o abusivo y está infrutilizado, sin tener en cuenta que hay otras personas que realmente harían el uso para el que está destinado. En el caso de entornos desatendidos estas malas prácticas son más habituales.

Basándose en estas premisas se pretende desarrollar un sistema que, además de intentar solventar estas situaciones, busque acometer un factor de comodidad al usuario y de ahorro energético para la institución. ¿Cómo? Empleando una red de sensores que sea capaz de detectar e identificar al usuario al tiempo que proporciona información adicional del entorno. De esta forma se logra el objetivo de la mejora de la gestión y uso de la infraestructura disponible.

La comunidad de la Universidad de Cantabria dispone de una tarjeta inteligente que, entre otros muchos usos, se emplea como identificador de la persona dentro de las instalaciones del campus universitario. Haciendo que la red de sensores, mediante este dispositivo, sea capaz de detectar la presencia de un usuario, esta información se podrá enviar a un centro de procesado, quien analizando la información temporal de uso permitirá al supervisor determinar el uso adecuado o indebido del puesto. Además, cada nodo sensor enviará adicionalmente información como la temperatura, humedad, etc. que de manera conjunta puede regular la sensación térmica de la habitación o aula a través del aire acondicionado o la calefacción.

Empleando las capacidades como actuador de los nodos, existe otro potencial conjunto de acciones posibles a realizar mediante el nodo sensor el control del puesto de trabajo dotaría de una gestión más eficiente de los recursos y con ello una reducción en los costes. Posibles ejemplos serían habilitar la iluminación, tomas de red, etc. únicamente cuando el usuario esté en el puesto, apagándolas al abandonar el puesto.

El resultado final de todas estas actuaciones es la automatización de la gestión de los recursos del espacio (aula, biblioteca, etc.) convirtiéndolo en un espacio inteligente e independiente.

## 1.2 OBJETIVOS

El objetivo de este proyecto es diseñar e implementar un sistema de monitorización de espacios en interiores basado en una red de sensores inalámbricos con comportamiento autónomo. Además el sistema tendrá la propiedad adicional de proporcionar un punto de compartición de los datos obtenidos y de interacción para la gestión de la red (estado, configuración, etc.) desde el exterior mediante conectividad a internet.

Para la realización del sistema se emplean entornos de código abierto, tanto para la parte software como la parte hardware.

A continuación se detallan las tareas para realizar este proyecto:

- Realizar un estudio de las tecnologías existentes en redes de sensores inalámbricas fijándose en parámetros como consumo o ancho de banda, contrastando los requerimientos y características de la red inalámbrica.
- Estudio del hardware disponible en el mercado y decisión por uno flexible y escalable.
- Familiarización con el hardware, su metodología de programación y funcionamiento. Realización de pequeños programas para testear los dispositivos.
- Adquisición de conocimiento sobre las opciones que ofrece internet en cuanto a comunicación, representación web y su entendimiento con la plataforma hardware elegida.
- Integración global de la red inalámbrica con su interfaz a internet.

## 1.3 CONTENIDOS DE LA MEMORIA

La redacción del proyecto se ha dividido en cinco capítulos los cuales se muestran a continuación:

- Capítulo 1: Introducción.

En el presente capítulo se realiza una introducción que describe tanto las motivaciones que han propiciado la realización del proyecto, como los objetivos que persigue.

- Capítulo 2: Estado del Arte.

Se presentan las bases teóricas sobre las que se sustenta el escrito. Se hace un pequeño repaso a las redes de sensores y sus distintas topologías, para luego ahondar en las diferentes técnicas de comunicación inalámbrica e introducir el protocolo utilizado en la red, IEEE 802.15.4. Finalmente se incluirán algunas de las plataformas más conocidas en este tipo de redes.

- Capítulo 3: Protocolo de Gestión de los Nodos Sensores

El tercer capítulo recoge los aspectos referidos al diseño y posterior implementación y validación de un protocolo de gestión de red de sensores e intercambio de datos entre los mismos. Así como la selección de hardware y de tecnología de comunicaciones.

- Capítulo 4: Control de presencia

En esta parte se describe los pasos dados para hacer operativa una interfaz de gestión remota accesible a través de internet. Además se añade a los nodos sensores la capacidad de detectar presencia a través de lectores de tarjetas, que es un ejemplo de gestión en entornos de interiores con un número limitado de plazas.

- Capítulo 5: Conclusiones y Líneas Futuras

Finalmente se hace balance del proyecto y se reflexiona acerca de las posibles mejoras a realizar a corto y medio plazo.

# ESTADO DEL ARTE

# 2

Hoy en día existen numerosos tipos de sensores formando parte de un gran número de sistemas y dispositivos electrónicos (sensores de temperatura en estaciones meteorológicas, acelerómetros en móviles y tabletas, sensores infrarrojos en sistemas de alarma o detección de movimiento, etc.). La mayor parte de estos sensores se limitan a funcionar como un transductor que realiza un tipo concreto de medición (una o más variables del entorno) y envía dicha información a un procesador central, dentro del mismo sistema electrónico.

Sin embargo, durante los últimos años, ha aparecido una nueva generación de sensores, independientes de un sistema electrónico concreto, que incorporan en un mismo dispositivo el transductor (de la o las variables que interesan medir), la alimentación del propio dispositivo y un módulo de comunicación dotado de cierta inteligencia propia, capaz de organizarse a sí mismo y de interconectarse de forma inalámbrica con otros nodos semejantes. Como resultado de su despliegue surgen las llamadas redes de sensores inalámbricos (Wireless Sensor Networks, WSN), consistentes en redes formadas por multitud de sensores individuales que intercambian información entre sí y/o con un nodo central sin necesidad de cables y mediante un protocolo de comunicación pre-establecido.

Los diferentes nodos pueden operar no solo como sensores, sino que sus capacidades de procesamiento y comunicación permiten que funcionen como actuadores, es decir, mecanismos capaces de activar procesos automatizados sobre sistemas externos bien siguiendo instrucciones remotas o a partir de decisiones tomadas de manera autónoma a partir de la información sensada (i.e. encender luces ante una disminución de la intensidad lumínica). Junto a los sensores, constituyen las redes inalámbricas de sensores y actuadores (Wireless Sensor & Actuator Networks, WSAN).

Las WSN comenzaron siendo utilizadas en aplicaciones militares, más en la actualidad proliferan multitud de aplicativos donde un sistema de control inalámbrico facilita la instalación. Algunos ejemplos son la domótica, medicina o en fábricas industriales o las ambiciosas ciudades inteligentes. El diseño eficiente e implementación de las redes inalámbricas de sensores se ha convertido en un área de investigación emergente en los últimos años, debido al gran potencial de estas redes para cubrir grandes áreas con un coste relativamente bajo.

A continuación se introducirán las características y posibilidades de las redes de sensores inalámbricas, ahondando en las arquitecturas y topologías más usuales, así como en las soluciones tecnológicas empleadas en su despliegue.

## 2.1 CONCEPTOS GENERALES

Una red de sensores inalámbrica o su extensión, una red de sensores y actuadores, es una red inalámbrica ad-hoc formada por un conjunto de dispositivos autónomos e inteligentes y de carácter multifuncional, con capacidades de sensado y recogida de datos, que se comunican entre sí o con una base central con el objetivo de intercambiar la información capturada por ellos mismos y tomar las acciones pertinentes sobre los sistemas bajo su inspección y control. Estas redes deben adaptarse a los cambios (posición, alimentación, etc.) puntuales en los diferentes nodos que las conforman.

En general, en una WSN se pueden encontrar los siguientes elementos:

- Sensores: capaces de medir las condiciones y la información deseada del lugar o del medio en el que se encuentran.
- Nodos: dispositivos que recogen la información recibida por los sensores y la transmiten a otros nodos o a la estación base.
- Pasarelas o Gateways: elementos que conectan los nodos de la red de sensores a una red externa o red troncal.
- Estación base: consistente en un elemento que recoge todos los datos provenientes de la red de sensores. Suele tratarse de un ordenador común o un elemento que lo sustituya.

Usualmente se emplea el término *nodo* para referirse a los nodos en una red de sensores. Este dispositivo integra, al menos, sensores, un micro controlador, un convertidor A/D y un chip de comunicación radio.

### 2.1.1 NODOS SENSORES Y ACTUADORES

La arquitectura de un elemento sensor o actuador se puede dividir en tres bloques funcionales, tal como se refleja en *Figura 2.1*:

- Bloque sensor/Bloque actuador: define la funcionalidad del dispositivo, como sensor (medición de un parámetro) o como actuador (mecanismo activador/interventor).
- Bloque de procesamiento: conjunto de elementos que dotan al dispositivo de inteligencia de computación y toma de decisiones.
- Bloque de comunicación: envía y recibe mensajes hacia y desde los nodos y repetidores vecinos.

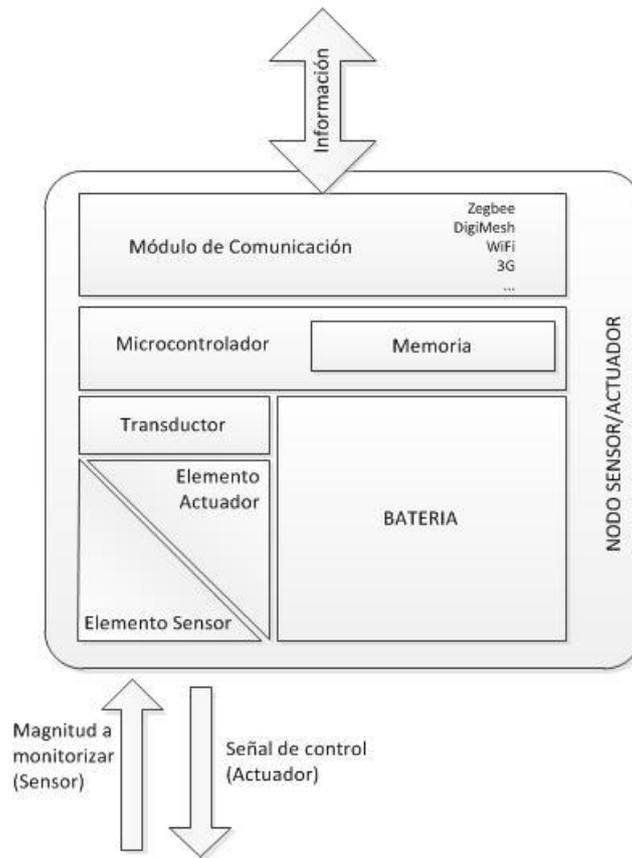


Figura 2.1 Esquema genérico de un nodo sensor o actuador

Se pueden establecer una serie de características que se dan en los nodos:

- Suelen estar limitados por el consumo energético, al estar usualmente alimentados por baterías. Y limitados en memoria y capacidad de proceso para ahorrar costes.
- Son autónomos y operan de forma independiente.
- Tienen una alta probabilidad de fallo, debido a las condiciones a las que se exponen y a su bajo coste.
- Deben incorporar medidas que permitan su adaptación al entorno.

El tamaño de un nodo puede variar, al igual que el coste, desde cientos de euros a unos pocos céntimos, dependiendo de la complejidad requerida. El tamaño y la limitación del consumo en los nodos sensores son los causantes de las limitaciones de recursos como la energía (<50mw), la memoria (<1Mbyte), la velocidad de cálculo (<500 Mhz) y el ancho de banda (250kbps).

Al tener un micro controlador interno, dentro de estos nodos se pueden realizar pequeños procesamientos de los datos con el objetivo de reducir el tráfico circulante por la red; puesto que sólo se enviarán los datos o las informaciones estrictamente necesarias, requiriéndose así una programación eficiente.

## 2.1.2 ARQUITECTURAS

Tomando como elementos de una red los nombrados anteriormente, podemos distinguir:

### 2.1.2.1 ARQUITECTURA CENTRALIZADA

En este tipo de red (Figura 2.2) los nodos se encargan de recoger información, tras lo que enviarán sus datos a la pasarela o gateway más cercano (si es que la hubiera, pues podría hacerlo directamente a la estación base), que dirigirá el tráfico de la red. Esto provocará dos problemas. Uno, el cuello de botella, provocando la lentitud de los envíos de los datos, y dos, como consecuencia un mayor consumo de energía, disminuyendo el tiempo de vida de los nodos.

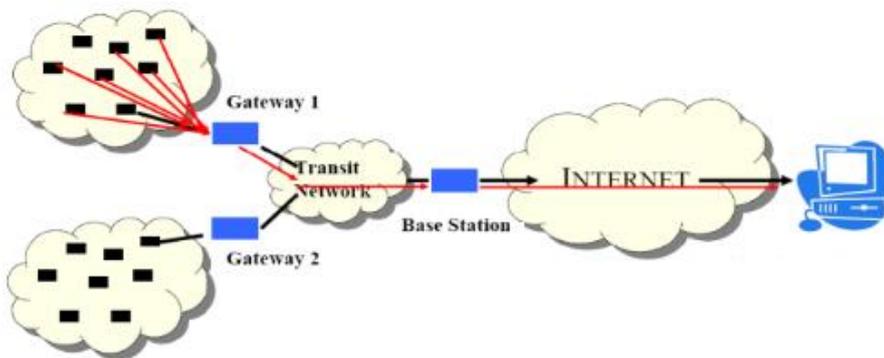


Figura 2.2 Esquema de arquitectura centralizada

### 2.1.2.2 ARQUITECTURA DISTRIBUIDA

Los nodos se comunicarán con sus nodos vecinos, cooperando y obteniendo una respuesta única que será enviada al coordinador del *clúster* que se encargará de comunicar a la estación base (Figura 2.3).

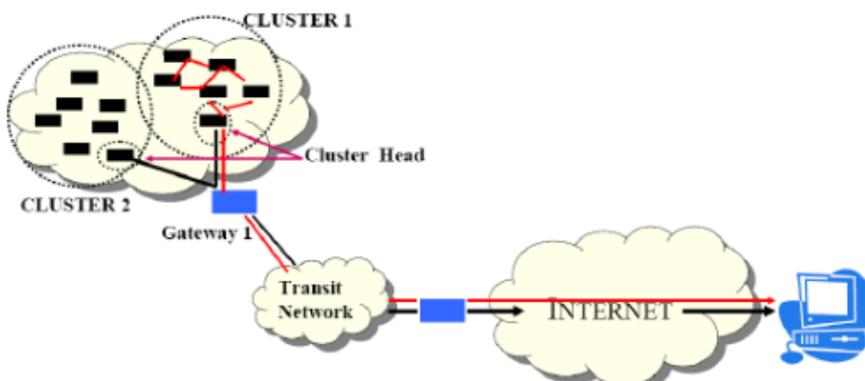


Figura 2.3 Esquema de arquitectura distribuida

### 2.1.3 TOPOLOGÍAS DE RED

Estas redes deben de auto-organizarse y auto-mantenerse para que de esta forma se reduzcan los costes totales y se facilite su uso. Dependiendo de los requerimientos de la aplicación, existen múltiples formas desde el punto de vista lógico en las que disponer los enlaces entre los nodos. A continuación se muestran algunas de las más empleadas (ver Figura 2.4).

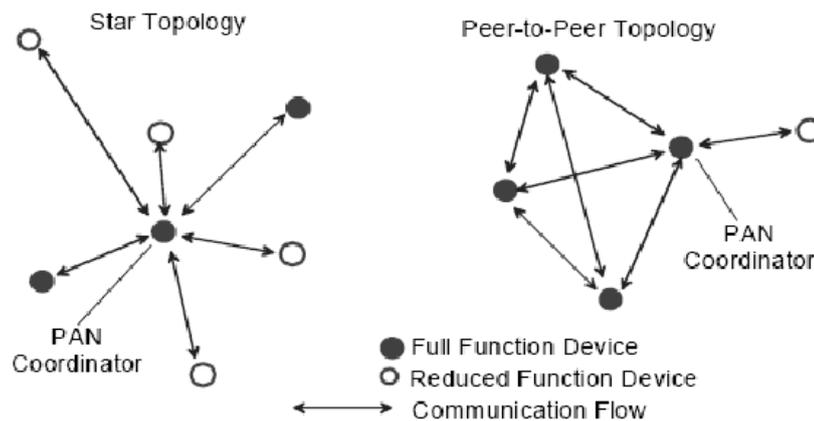


Figura 2.4 Topología en estrella y red peer-to-peer

#### 2.1.3.1 EN ESTRELLA

En este tipo de configuración es necesario un coordinador por cada grupo de nodos o PAN (Personal Area Network), hacia el cual el resto de nodos dirigen su comunicación. Cada dispositivo establece una comunicación bidireccional con el nodo maestro. Dicho coordinador es el encargado de iniciar, terminar o redirigir la comunicación a través de la red. Aparte de esto, realiza la propia función que tenga asignada, por ejemplo, actuar como nodo central de recogida de datos.

#### 2.1.3.2 PEER-TO-PEER

Según esta topología cada dispositivo puede comunicarse con otros de la misma red que estén a su alcance. Conforman redes ad-hoc, es decir, redes sin infraestructura física preestablecida, ni necesidad de un control central que coordine su actividad.

Además pueden ser auto-organizativas, con restauración de caminos, con nodos que puedan trabajar como emisores o receptores y tener la capacidad de establecer caminos de comunicación entre nodos sin visibilidad directa y, modificar estos caminos si alguno de los nodos del encaminamiento falla, etc. Pero estas funciones han de ser establecidas por el protocolo de comunicaciones que se emplee entre los nodos.

Con esta topología de red es posible crear estructuras más complejas, como redes malladas, en las que no existe una jerarquía entre los nodos.

## 2.2 TECNOLOGÍAS DE COMUNICACIÓN INALÁMBRICA

Seguidamente se estudian las tecnologías comunicación inalámbrica de nivel físico y de enlace que potencialmente pueden ser empleadas en redes de sensores.

### 2.2.1 BLUETOOTH

El estándar IEEE 802.15.1[2] o Bluetooth trabaja en la banda ISM (Industrial Scientific & Medical) a una frecuencia de 2,4 GHz con 79 canales de 1 MHz entre 2402 y 2480 MHz. Esto hace muy alta la probabilidad de interferencia debido la proximidad de redes 802.11, hornos microondas, mandos de garaje, etc. Para evitar en lo posible las interferencias, Bluetooth usa un sistema llamado Frequency Hop Spread Spectrum (FHSS). Dicho sistema realiza 1600 saltos de frecuencia por segundo entre los 79 canales radio. Es decir, divide la información en pequeños paquetes, los cuales se transmiten siguiendo una secuencia determinada a través de los 79 canales disponibles.

El rango de alcance de Bluetooth es de 10 metros con un consumo de 1 mW, aunque dicho rango puede alcanzar los 100 metros si aumentamos la potencia de transmisión hasta los 100 mW. La tasa de transmisión es de 1 Mbps cuando se utiliza modulación GFSK (Gaussian Frequency Shift Key). Aunque para versiones más avanzadas del estándar (Bluetooth 2.0) se pueden alcanzar velocidades de hasta 2 Mbps gracias a la combinación de GFSK con PSK (Phase-Shift Keying).

Bluetooth está basado en una arquitectura maestro-esclavo. En la que los dispositivos se organizan en celdas denominadas "piconets", que pueden estar formadas de hasta un máximo de 8 nodos, de los cuales uno es el maestro, encargado de organizar la comunicación entre los demás integrantes de la red o esclavos. Además, cuenta con cinco estados distintos de funcionamiento para reducir el consumo.

### 2.2.2 WI-FI

Esta tecnología se basa en el estándar IEEE 802.11[3] y pretende ser la alternativa inalámbrica a las redes de área local (LAN).

Trabaja en la banda ISM y existen varios estándares Wi-Fi, los principales son:

- 802.11b: trabaja a 2,4 GHz, tiene un alcance de aproximadamente 100 metros con una tasa de transferencia de 11 Mbps. Su principal ventaja es el bajo coste de los dispositivos que lo implementan.
- 802.11a: trabaja a una frecuencia de 5 GHz y su tasa de transferencia puede alcanzar los 54 Mbps. Al ser una frecuencia menos usada, no se dan tantas interferencias. No obstante tiene un rango de unos 30 metros y el coste de los dispositivos es mayor que los de 802.11.

- 802.11g: trabaja a 2,4 GHz, compatible con 802.11b, puede alcanzar velocidades de hasta 54 Mbps en un rango de alcance de unos 100 metros. Por contra, su coste y consumo de potencia son mayores.
- 802.11n: incrementa significativamente la velocidad de transmisión hasta un máximo de 600Mbps.

Existen dos tipos de elementos de red: AP (Access Point) y las estaciones. El primero está conectado a una red Ethernet y, el segundo se trata de estaciones inalámbricas, en general ordenadores personales, teléfonos móviles, entre otros, equipados con un interfaz de red inalámbrico (NIC: Network Interface Card).

Dependiendo de la forma de conectar los elementos de red anteriores se pueden definir diferentes configuraciones para las redes WLAN:

- Infraestructura: cada celda, o BSS (Basic Service Set), es el área de cobertura de un AP que forma parte de una red cableada. Los dispositivos inalámbricos que se encuentran dentro de la celda se conectan a la red o entre ellos mismos a través del AP.
- Ad hoc: en este caso dos o más dispositivos 802.11 se conectan directamente entre ellos sin tener que canalizar previamente la información a través de un AP. A esta configuración también se la conoce como “peer to peer”.

### 2.2.3 ULTRA-WIDE BAND

Basado en los trabajos del IEEE Task Group 802.15.3a[4], Ultra-Wide Band (UWB)[5] es una tecnología en el rango de las PAN (Personal Area Network) que pretende saciar la necesidad del intercambio de datos con una alta tasa de transferencia en distancias cortas, como puede el ejemplo de videocámaras digitales, sistemas home cinema, etc.

UWB hace uso de un espectro muy ancho de frecuencias: desde los 3,1 hasta los 10,6 GHz con un ancho de canal de 500MHz. Pero, al contrario que Bluetooth y Wi-Fi, no realiza el envío de información de forma continua sino en determinados instantes temporales, utilizando gran parte de ese ancho de banda y emitiendo con menos potencia. Así, el consumo de los dispositivos basados en este estándar es relativamente bajo. Además esta tecnología es capaz de hacer uso de un mismo canal sin interferencias.

Con UWB se pueden alcanzar tasas de transmisión de 110 Mbps con un rango de alcance de 10 metros aproximadamente. Sin embargo, en su contra para que esta tecnología se consolide queda por resolver la poca distancia de transmisión, la contaminación electromagnética, la interoperabilidad, entre otros.

## 2.2.4 Z-WAVE

Z-Wave[6] es un protocolo para comunicación inalámbrica diseñado para aplicaciones remotas en el hogar como iluminación, accesos, detectores de humo,... Opera en el rango de los 900MHz perteneciente a la banda ISM, y tiene una cobertura de unos 30 metros.

Permite una topología mallada o mesh, que admite en torno a los 232 dispositivos, en la que puede haber uno o varios coordinadores y donde la tasa de transmisión alcanza los 50Kbps. Cada nodo permite enrutar información hacia otro lo que hace que el radio de cobertura crezca en relación con el de una sola unidad, lo que significa la existencia de ciertos nodos, llamados repetidores, no puedan dormir.

## 2.2.5 WIBREE - BLUETOOTH ULTRA LOW ENERGY

Wibree[7] es una nueva tecnología digital de radio interoperable para pequeños dispositivos. Permite la comunicación entre dispositivos móviles, computadores y otros dispositivos más pequeños (de pila de botón). Está diseñada para que funcione con poca energía y a poca distancia, opera a una frecuencia de 2.4 GHz (ISM) y cuenta con una tasa de transferencia de 1 Mbps en la capa física. Además da soporte de seguridad utilizando el sistema de cifrado AES (Advanced Encryption Standard).

Las principales características de WiBree son que tiene un consumo de energía durante la conexión ordinaria 10 veces menor que Bluetooth 2.1, y es 50 veces más rápido en las transferencias de datos. Wibree tal y como está definido no es compatible con los actuales estándares Bluetooth. Desde junio de 2007, se le conoce como "Bluetooth Low Energy Technology" o "Bluetooth ULP" (Ultra Low Power).

## 2.2.6 IEEE 802.15.4

La norma IEEE 802.15.4[8], implementa las capas física (PHY) y de control de acceso al medio (MAC: Medium Access Control). Esta tecnología está enmarcada dentro de un subgrupo de las redes de área personal, las LR-WPAN (Low Rate WPAN, baja velocidad de transmisión), y está orientada a la implementación de redes de sensores inalámbricos y actuadores.

Dentro de las características más importantes se puede destacar que alcanza tasas de transmisión de hasta 250 Kbps, en un rango de cobertura de entre 10 y 100 metros y que opera en las bandas ISM a las frecuencias de 2.4 GHz, 868 MHz (Europa) y, 915 MHz (Estados Unidos).

Otra característica esencial es la optimización del consumo de los elementos que componen la red. Son los nodos finales (típicamente sensores o actuadores) los elementos con menor consumo, ya que permiten ser configurados de forma que permanezcan inactivos hasta que ocurra un determinado evento o tengan que enviar información. Este protocolo permite dos tipos de topologías en estrella y peer to peer en visión directa.

### 2.2.7 ZIGBEE

ZigBee[9] es un estándar hardware y software que parte del estándar IEEE 802.15.4. Incluye especificaciones en múltiples niveles del modelo de referencia OSI, desde el modo de uso de la capa física y MAC, hasta perfiles de usuario, descubrimiento de topologías, seguridad, y mensajería a niveles de red y aplicación. Al igual que IEEE 802.15.4 trabaja en la banda ISM a 2,4 Ghz, pero tiene una velocidad de transmisión menor.

### 2.2.8 OTRAS ESPECIFICACIONES

Se incluyen a continuación un conjunto de protocolos o soluciones tecnológicas apoyados en IEEE 802.15.4 para dar soporte a la gestión de redes de sensores:

- Wireless HART [10][11]: versión para comunicación inalámbrica del protocolo HART (Highway Addressable Remote Transducer Protocol) usado en aplicaciones industriales y automoción que requieren tiempo real. Puede operar según una topología mallada o estrella y, opera en la banda ISM de 2,4 Ghz.
- ISA - SP100 [12]: también es usado en el ámbito de la automoción. Se caracteriza por su bajo consumo, con una cobertura de hasta 100 metros, estando pensado para un monitoreo periódico y de control de procesos. Desarrollado por la ISA (Industrial Society of Automation), está en proceso de convertirse en estándar
- IETF IPv6 – LoWPAN [13]: (IPv6 over Low power Wireless Personal Area Networks) implementa una adaptación de IPv6 sobre IEEE 802.15.4 para permitir comunicaciones eficientes a un dispositivo acceder desde y hacia internet. (IETF, Internet Engineering Task Force).

### 2.2.9 COMPARATIVA

Antes de proceder con la comparativa de las diferentes tecnologías enumeradas anteriormente, se muestran los requerimientos del sistema objetivo. Una red de sensores se caracteriza por:

- Tráfico no elevado, y no continuo
- Nodo operativo cuando requiera realizar alguna tarea, bien transmisión, recepción o cómputo.
- Baja tasa binaria.
- Complejidad de programación baja.
- Bajo consumo y larga vida de las baterías.

Partiendo de estas premisas se procede el análisis sobre qué tecnología es la más conveniente. Las alternativas inalámbricas anteriormente mencionadas trabajan en la misma banda ISM a excepción de UWB que trabaja desde los 3,1 a los 10,6 Ghz, banda recientemente legalizada.

Wi-Fi requiere una actividad casi ininterrumpida de los dispositivos en la red. Este estándar presenta una elevada tasa binaria, capaz de transmitir gran cantidad de datos entre

dispositivos, lo que supone un gran consumo de energía. Por lo que Wi-Fi queda descartada por su alto consumo.

UWB, por su parte, aunque energéticamente podría ajustarse y proporciona una elevada tasa de transferencia, el radio de cobertura de en torno a 10 metros es pequeño. Esto unido a la escasa disponibilidad de dispositivos en el mercado hace que se descarte.

Otra tecnología que se podría considerar, aunque no haya sido analizada en profundidad, es RFID pasiva, pero su reducida cobertura, de unos pocos centímetros se descarta. RFID activa aunque con mayor cobertura no permite el empleo de dispositivos inteligentes ni la implementación de las características de comunicaciones previstas para el sistema. Igualmente, aunque no se haya mencionado en apartados anteriores, la comunicación infrarroja a pesar de tener una cobertura de hasta 200 metros, presenta como mayor problema la necesidad de tener que evitar la presencia de obstáculos entre los nodos, lo que implicaría que toda la red debería tener en línea de vista directa.

Bluetooth está diseñado para la transferencia de datos a distancias cortas. IEEE 802.15.4 tiene un consumo similar, pero menor consumo en el estado de espera. Esto es debido a que los dispositivos en redes Bluetooth deben dar información a la red frecuentemente para mantener la sincronización, así que no pueden ir fácilmente a modo de sueño. Además el tamaño de las piconets tiene un valor máximo de 8 nodos. No obstante su necesidades de establecer el pareado entre los nodos, dificulta su despliegue en redes de sensores.

IEEE 802.15.4 en cambio, se trata de un protocolo cuya tasa de transmisión no supera los 250Kps, y con una cantidad de datos baja es más que suficiente. No precisa de gran capacidad de cómputo, ni de grandes potencias de transmisión. Precisamente el objetivo buscado. Zigbee por su parte, añade overhead por lo que la velocidad de transmisión bajará. Al contrario que 802.15.4 que permite la comunicación por visión directa, Zigbee enruta paquetes a través de nodos intermedios, es decir permite una topología mucho más compleja, que en el presente proyecto no es necesaria.

En cuanto a consumo, 802.15.4 ofrece mayores prestaciones como se puede observar en la Tabla 2.1:

Tabla 2.1 Comparativa de tecnologías radio de transmisión

Protocolo	802.15.4	Bluetooth	Wi-fi	UWB	Z-Wave	Wibree
Alcance	100m	10m	10-100m	10m	1-30m	1-10m
Banda de frecuencias	2,4Ghz	2,4Ghz	2,4-5 Ghz	3,1-10,6Ghz	900Mhz	2,4Ghz
Régimen binario	250Kbps	1Mbps	11,54,300Mbps	110Mbps	40Kbps	1Mbps
Consumo	1,8 mA en Tx 5.1 µA en reposo	40 mA en Tx 0.2 mA en reposo	400 mA en Tx 20 mA en reposo	--	23 mA en Tx 3 uA en reposo	--

Existen también soluciones propietarias como Z-Wave, Wibree y, otras no descritas como EnOcean[14], ANT[15], MiWi[16], etc., que a pesar de ser robustas y sencillas tienen el inconveniente de ser difícilmente escalables, no siendo interoperables y forzando al cliente a permanecer con un único fabricante, con las limitaciones que eso supone.

Es por ello que si se desea disponer de una solución interoperable y escalable, es imprescindible elegir una tecnología abierta. De entre todas mencionadas anteriormente, resulta especialmente interesante IEEE 802.15.4, que se ajusta más al despliegue de redes o aplicaciones en las que no es necesaria una gran velocidad de transmisión, pero sí es importante minimizar la complejidad, el consumo y el coste, al tiempo que permite la interconexión de una gran cantidad de nodos. Adicionalmente, para los requerimientos establecidos IEEE 802.15.4 es mucho más eficiente energéticamente que todas las tecnologías anteriormente mencionadas y, por lo tanto, mucho más adaptado al uso en redes de sensores.

## 2.3 IEEE 802.15.4

El estándar IEEE 802.15.4, tal como se ha anticipado anteriormente, va a ser la elección como norma de comunicación entre los nodos de la red implementada en este proyecto. Por esta razón a continuación se realiza una descripción más detallada.

### 2.3.1 ESTÁNDAR

IEEE 802.15.4 es el estándar que define la capa de nivel físico y parte de enlace (MAC) para redes inalámbricas de área personal, PAN, y de baja tasa de transmisión de datos (conocidas por su acrónimo en inglés LR-WPAN {Low Rate}).

Nace en 2003 de la necesidad de tener una norma que defina la comunicación en una red de múltiples nodos, los cuales deben tener una baja tasa de datos, baja complejidad, bajo consumo de energía, corto alcance y bajo coste, además de ser sencillos de instalar y hacer uso de bandas no licenciadas.

Entre sus principales características están:

- Tasas de transferencia de datos de 20 Kbps, 40 Kbps, 110 Kbps, 250 Kbps o 851 Kbps dependiendo del rango de frecuencias que se emplee.
- Configuración de red en estrella o mallado total (peer-to-peer).
- Direccionamiento corto de 16 bits o extendido de 64 bits.
- Asignación de intervalos de tiempo garantizados (GTS, Guaranteed Time Slot ).
- Uso de CSMA-CA como técnica de acceso al medio.
- Mecanismo de asentimientos para garantizar fiabilidad en las transmisiones.
- Consumo de energía muy pequeño.
- Control de potencia, e indicación de calidad del enlace (LQI12, Link Quality Indicator).

## 2.3.2 ARQUITECTURA

En la Figura 2.5, se puede observar el nivel de ejecución de 802.15.4 comparándolo con el modelo de capas OSI.

Aplicación	
Presentación	DigiMesh
Sesión	ZigBee
Transporte	ISA SP100
Red	IPv6-LoWAN
Enlace	802.11 LLC
	802.15.4 MAC
Físico	802.15.4 PHY

Figura 2.5 Modelo OSI

Similar a todos los estándares inalámbricos IEEE 802, como se muestra en la figura, el estándar IEEE 802.15.4 normaliza sólo las capas física o PHY y de control de acceso al medio o MAC. Por encima pueden ir otros protocolos como Zigbee.

La capa física se encarga de interactuar con el medio físico de transmisión así como con la capa MAC. Esta capa provee dos servicios a la capa MAC: Servicio de Datos PHY, que habilita la transmisión y recepción de Unidades de Datos de Protocolo PHY (PPDU) a través del canal de radio; y el Servicio de Administración, el cual brinda mecanismos para el control y configuración de la interfaz de radio desde la capa MAC.

La capa MAC, una sub-capa de la capa de enlace, se encarga de brindar acceso a los canales físicos para todo tipo de comunicación. En la se ve que la capa de enlace está formada por la subcapa 802.2 LLC (Logical Link Control) y, la sub-capa MAC. 802.2 LLC es común en todos los estándares IEEE 802 y accede a la sub-capa MAC, sin embargo su explicación está fuera del alcance del estándar IEEE 802.15.4.

La sub-capa MAC ofrece los siguientes servicios a las capas superiores: Servicio de Datos MAC (MCPS, MAC Common Part Layer), el cual permite enviar y recibir datos a la siguiente capa superior; y el Servicio de Administración MAC (MLME, MAC Layer Management Entity), que brinda mecanismos para el control y configuración de comunicaciones, interfaz de radio y creación de redes desde la siguiente capa superior.

## 2.3.3 CAPA FÍSICA

### 2.3.3.1 CARACTERÍSTICAS

Utiliza la técnica DSSS, Direct Sequence Spread Spectrum, para modular la información antes de ser enviada a la capa física. Básicamente cada bit de información se modula en 4 señales distintas, lo que hace que ocupe un gran ancho de banda. En cambio usa una densidad espectral de potencia baja para cada señal. Esto hace que sea inmune a interferencias en la banda de frecuencias de uso, y mejora la relación a ruido, SNR, en el receptor. De manera que la detección y decodificación son más fáciles.

Una característica de las redes sobre IEEE 802.15.4 es que trabajan en bandas no licenciadas. Según la región geográfica las bandas frecuenciales de operación son:

- 868 – 868.6 Mhz (Europa)
- 902 – 928 Mhz (EEUU)
- 2400 – 2483.5 Mhz (Mundial)
- 3100 – 10600 Mhz (variable dependiendo del país)

Se tienen 27 canales, 16 en la banda 2400 MHz, 10 en la banda 915 MHz y 1 en la banda 868 MHz. El coordinador de una red se encarga de escoger el canal adecuado, para esto realiza un escaneo de todos los canales y elige el mejor, que por lo general es el canal con menos interferencias.

### 2.3.3.2 FUNCIONES

Las tareas que realiza la capa física IEEE 802.15.4 son:

- Activación y Desactivación del transceptor de radio.
- Detección del nivel de energía en el canal de trabajo.
- Brindar un Indicador de Calidad de Enlace (LQI, Link Quality Indicator) de los paquetes recibos, basados en su nivel de potencia.
- Realizar la valoración de canal libre (CCA, Clear Channel Assessment), utilizado por la Sub-Capa MAC para el algoritmo CSMA/CA (Acceso Múltiple por Detección de Portadora con Evasión de Colisiones). Elegir el canal de frecuencia de trabajo.
- Transmisión a través del canal físico de los mensajes.

## 2.3.4 SUB-CAPA MAC

Las tareas que realiza la sub-capa MAC IEEE 802.15.4 son:

- En el caso de nodo coordinador, debe generar señales de sincronización, asociación/desasociación.
- Brindar diferentes servicios de seguridad.
- Utilizar el mecanismo CSMA/CA para el acceso al canal.
- Enviar tramas de acuse de recibo.
- Administrar el mecanismo de las ranuras de tiempo dedicadas.
- Administrar la asociación y desasociación de nodos en la red PAN.

Una de las ventajas de la capa MAC en el IEEE 802.15.4 es que sólo dispone de 21 primitivas de servicio o comandos, frente a las 131 primitivas de Bluetooth. Esto hace que el hardware que se utiliza pueda ser más sencillo y económico. El estándar de la capa MAC implementa dos modos de funcionamiento: el modo Beacon y el modo Beaconless.

### 2.3.5 MECANISMOS DE ROBUSTEZ

#### 2.3.5.1 CSMA/CA

Es necesario resaltar la implementación del mecanismo de CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) en la transmisión de mensajes para evitar colisiones con otros mensajes que están siendo transmitidos en el mismo canal por otro nodo.

Cuando se transmite un mensaje, la sub-capa MAC solicita a la Capa Física que realice una Valoración de Canal Libre (CCA, Clear Channel Assessment), si se detecta que el medio está libre entonces se procede a transmitir, de lo contrario, si el medio está siendo utilizado, la sub-capa MAC espera un periodo de tiempo aleatorio (Random Back-Off Period) y después reinicia el proceso.

#### 2.3.5.2 ACUSE DE RECIBO

Al enviar un mensaje, si se ha recibido satisfactoriamente, el nodo receptor se encarga de notificar al origen que se recibió con éxito. Esto se hace mediante un mensaje de acuse de recibo (acknowledgment frame o simplemente ACK). Estos mensajes son enviados sin utilizar CSMA/CA.

#### 2.3.5.3 OTROS

IEEE 802.15.14 incorpora un control de secuencia (FCS) para validar la integridad de los datos.

### 2.3.6 ELEMENTOS DE UNA RED 802.15.4

Existen dos tipos de dispositivos según su hardware y software: los Dispositivos de Función Completa (FFD), que son capaces de hacer uso de todos los servicios de la MAC IEEE 802.15.4; y los Dispositivos de Función Reducida (RFD), que no pueden hacer uso de todos los servicios de la MAC porque tienen hardware limitado.

Además, los nodos de la red pueden ser clasificados según el rol que cumplen:

- **Coordinador PAN:** Se encarga de asignar un ID (identificador) a la red, encontrar el canal de trabajo adecuado, asignarse una dirección corta, administrar la asociación de nuevos nodos y según la topología retransmitir mensajes entre nodos. Por lo que tiene que ser necesariamente un FFD.
- **Router:** En una red con topología de árbol, aparte del coordinador PAN, también puede existir un Coordinador o Router, que se encarga de administrar las solicitudes de asociación a la red y retransmitir mensajes entre nodos. También debe ser un FFD.

- **Dispositivo:** Es un nodo que no cumple funciones de coordinador, puede ser FFD o RFD.

---

#### 2.3.6.1 TOPOLOGÍA DE LA RED

Cómo se comentó anteriormente en el apartado 2.1.3, los nodos se pueden agrupar lógicamente en estrella o formar una red peer-to-peer.

### 2.3.7 HARDWARE IEEE 802.15.4

---

#### 2.3.7.1 MODULOS XBEE DIGI

XBee[16] describe una familia de módulos radio, compuestos básicamente por un transmisor/receptor RF y un microprocesador. Los módulos están diseñados para dar interoperabilidad entre múltiples plataformas, incluyendo topologías punto-multipunto, Zigbee/Mesh, basadas en el estándar 802.15.4. Trabajan en 2.4 GHz y 900 MHz con una tasa binaria de 250 Kbps.

Actualmente, y en función del módulo RF implementado, XBee puede trabajar con:

- **ZigBee:** Módulos XBee, XBee-PRO ZigBee y XBee-PRO ZB ofrecen solución a redes ZigBee mesh. La familia ZigBee PRO son compatibles con otros dispositivos ZigBee.
- **DigiMesh:** es un protocolo propietario de Digi, basado en el estándar 802.15.4, para redes malladas. Este protocolo incluye funciones de descubrimiento de nodos, detección de nodos caídos, y soporte para nodos routers que pueden entrar en estados de bajo consumo, entre otras características. Por todo ello es una solución adecuada en donde el uso de baterías es crítico.
- **IEEE 802.15.4:** Módulos XBee y XBee-PRO 802.15.4 OEM RF usan el protocolo 802.15.4 para redes “peer to peer” y punto multipunto. Están diseñados para aplicaciones de alto throughput que requieren baja latencia y tiempos predecibles de comunicación.

Los módulos XBee permiten comunicaciones robustas en configuraciones punto a punto, peer to peer y multipunto/estrella y una fácil configuración maximizando el rendimiento en la comunicación. Además, pueden ser fácilmente configurados desde un ordenador utilizando el programa X-CTU[17], desarrollado por Digi, o bien desde un microcontrolador. Estos módulos incluyen dos modos de funcionamiento, API (Application Programming Interface) y AT (o modo comandos). Por todo ello, hoy en día tienen una alta aceptación entre los desarrolladores y proveedores de servicios.

---

#### 2.3.7.2 OTROS FABRICANTES

Además del mencionado anteriormente, existen una serie de fabricantes que desarrollan módulos con características similares por ejemplo:

- Módulo RP-M100 de FirmTech[19].
- ProBee-ZE10 y ProBee-ZE20S de Sena Technologies [20].
- ProFLEX01 y SiFLEX02 de LS Research[21].

## 2.4 PLATAFORMAS DE NODOS SENSORES

Una vez estudiado su hardware de comunicación, se plantean diferentes plataformas que permitan implementar la inteligencia de las motas. Tanto el hardware como el software de las motas debe ser configurable con cierta flexibilidad, lo que deriva en que existan en el mercado multitud de sistemas.

### 2.4.1 CROSSBOW

Crossbow[22][23] es un fabricante de motas especializado que desarrolla plataformas hardware y software que da soluciones para las redes de sensores inalámbricas. Entre sus productos se encuentran las plataformas Mica, Mica2, MicaZ, Mica2dot, Telos y TelosB.

Las motas formas redes tipo Adhoc, alguna de sus características son que permite el descubrimiento de rutas, y que son auto-organizativas. Además, utiliza el sistema operativo TinyOS encargado de las comunicaciones radio y, de minimizar el consumo manteniendo la mayor parte del tiempo al sistema durmiendo, y sólo cuando ocurre un evento se despierta para atenderlo. El lenguaje utilizado para programar las motas es el “NesC”[24], parecido a C, y es compilado por Cygwin[25] (un emulador de Linux).

Además de Crossbow existen otras soluciones como MOTEIV, que desarrolla las plataformas Tmote Sky y Tmote Invent. Y Shockfish que desarrolla la plataforma TinyNode[26].

A continuación se ahondará en algunas de las plataformas mencionadas.

#### 2.4.1.1 TELOSB

TelosB[27] es una plataforma para aplicaciones en redes de sensores de muy bajo consumo y alta recolección de datos. Lleva integrada sensores de temperatura, humedad y radiación, una antena, un microcontrolador y, pueden ser fácilmente programados.

El microcontrolador es el MSP430 F1611 (procesador RISC de 16 bits) que permanece en estado de sueño la mayor parte del tiempo, se activa tan rápido como puede para procesar, enviar y, una vez finalizado vuelve al estado de sueño. De forma que su consumo en modo activo es de 1,8 mA y, de 5,1  $\mu$ A en estado de sueño.

Utiliza un controlador USB del fabricante FTDI para comunicarse con un ordenador y lleva el módulo de radio CC2420 (de Chipcon), el cual envía y recibe bajo el estándar IEEE 802.15.4, a una frecuencia de 2,4Ghz y con una velocidad de transmisión de 250Kbps. La antena puede ser apagada para ahorrar energía y, tiene una cobertura de 50 metros en interiores y, 125 metros en exteriores. En la Figura 2.6 se puede observar una mota TelosB:



Figura 2.6 TelosB

Utiliza el sistema operativo TinyOS<sup>1</sup> en el enrutamiento e implementación de las comunicaciones. Y se alimenta con 2 baterías de tipo AA, aunque igualmente lo puede hacer a través de su puerto USB (que además sirve para programación). También proporciona la capacidad de añadir dispositivos adicionales mediante los dos conectores de expansión de los que dispone, estos pueden ser configurados para controlar sensores analógicos, periféricos digitales y displays LCD. Entre las distintas casas que lo fabrican se encuentran MoteIV y, Crossbow.

#### 2.4.1.2 MICAZ

MicaZ [28] es un tipo de mote fabricado por la empresa Crossbow Technology, que puede ser utilizado para formar redes inalámbricas de sensores de bajo consumo. Entre algunas de sus características está que trabaja a la frecuencia de 2,4Ghz, cumple la especificación del IEEE 802.15.4, alcanza una tasa binaria de 250Kbps, cuenta con entradas analógicas, digitales, soporte de I2C, SPI (Serial Peripheral Interface) y UART (Universal Asynchronous Receiver-Transmitter), y utiliza el sistema operativo TinyOS.

En la Figura 2.7 podemos ver el aspecto exterior de este mote, donde al igual que en modelos de otros fabricantes el mayor espacio es el ocupado por las 2 baterías del tipo AA.



Figura 2.7 MicaZ

<sup>1</sup> TinyOS es un sistema operativo de código libre, pensado para crear códigos que optimicen los escasos recursos de un microcontrolador que controla una red de sensores y un módem inalámbrico.

## 2.4.2 WASPMOTE

En el año 2009 la plataforma Wasmote[29] fue lanzada por la empresa Libelium (Spin-off de la Universidad de Zaragoza). El proyecto se centró en conseguir una plataforma de muy bajo consumo, alcanzando así los 0,7uA en modo hibernación. Hace uso de los módulos radio Xbee, lo que le permite trabajar a las frecuencias de 2,4GHz, 900Mhz y 868MHz (bandas ISM), y con los protocolos IEEE 802.15.4 y Zigbee. En la Figura 2.8 se muestra el mote con un módulo de radio, entrada USB, tarjeta SIM, lector de tarjetas MMC/SD y el conector de expansión.

Estos módulos están basados en una arquitectura modular por lo que es sencillo ampliar sus capacidades. Para ello se han desarrollado diferentes escudos o shields siendo algunos de los más utilizados los de GPS, GPRS, lectores de tarjetas SD, sensores para monitorización ambiental CO<sub>2</sub>, O<sub>2</sub>, presión atmosférica, de luminosidad, vibración, nivel de líquidos y módulos de entrada salida genéricos como ADC, entradas digitales, etc.



Figura 2.8 Wasmote

## 2.4.3 ISENSE

iSENSE[30] es una plataforma hardware y software para redes inalámbricas. Están constituidos principalmente con microcontroladores RISC de 32 bits y trabajan con la norma IEEE 802.15.4. Pero, además, está disponible un protocolo propietario mesh o para redes en malla, que también incluye en capas superiores soporte para conexión a internet tanto en IPv4 como IPv6. Son programables a través del lenguaje C y, están fabricados por la empresa Coalsenses. Existe una variedad de módulos según su función como “Core Module 3”, o “Gateway Module” y distintos escudos o shields que los complementan. En la Figura 2.9 se pueden observar algunos de ellos. Para más información se puede acudir a la página web del fabricante[31].



Figura 2.9 Módulos iSense

#### 2.4.4 ARDUINO

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. No es más que una placa con un circuito que comunica un procesador con diferentes puertos de entrada o salida, y con una memoria E<sup>2</sup>PROM. Esta memoria actúa a modo de pequeño disco duro, donde se almacenan los programas diseñados a ejecutar. Su unidad de procesamiento está basada en los microcontroladores Atmega168 ó Atmega8.



Figura 2.10 Arduino UNO

Arduino puede tomar información del entorno a través de transductores conectados a los múltiples pines de entrada y, operar como actuador mediante los interfaces de salida a los que se pueden conectar luces, motores, etc. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino[33], basado en Wiring[34], empleando el entorno de desarrollo Arduino, basado en Processing[35]. Tanto el diseño de la placa como el software de programación están disponibles bajo licencia abierta y cualquiera es libre de adaptarlo a sus necesidades. Un ejemplo de ello son los sistemas de Libelium, como el WaspMote que se pueden considerar un fork del sistema Arduino.

Lo que hace tan atractivo Arduino es su facilidad para aprender a utilizarlo y, su sencillez al programarlo, además del gran número de librerías disponibles, fruto de la gran comunidad de usuarios de que dispone.

Existen una gran variedad de diseños de placas Arduino, creadas según requerimientos específicos como la Arduino FIO, la EthernetDUINO, entre otras [32] y de proyectos basados en Arduino, como Mosquino, ZArduino, Imaguino... creadas para una función específica y, totalmente compatibles con el IDE de Arduino[36].

#### 2.4.5 PINGÜINO

Pingüino[37][39] dio sus primeros pasos en el año 2008, se trata de una plataforma de hardware y software libre para la experimentación con microcontroladores. Es similar a Arduino pero basada en un microcontrolador PIC18F2550, de estructura RISC (Reduced Instruction Set Computer), que incluye un módulo USB nativo y una UART para la comunicación serie. Cuenta con su propio Entorno de Desarrollo Integrado (IDE) de uso y apariencia similar al de Arduino, válido para Windows, Linux y MAC OS X.

El módulo USB integrado permite comunicarse directamente con el PC y reduce el costo del hardware, dejando además libre el puerto UART del microcontrolador para otras aplicaciones.

El circuito básico Pingüino es muy simple y sólo precisa de unos pocos componentes; todo el software necesario está disponible para bajarse de internet gratuitamente. Su IDE está basado en Python[39]. Cuenta con un preprocesador que traduce instrucciones Arduino directamente a C, lo que reduce la longitud del código y velocidad ejecución.

PowerJaguar es otro proyecto parecido que utiliza el mismo controlador PIC. Pero aún está en desarrollo. No cuenta aún con IDE propio.

#### 2.4.6 RASPBERRY

Raspberry Pi[40] es una placa computadora (SBC) de bajo coste desarrollada en Reino Unido por la Fundación Raspberry Pi.

El diseño incluye un procesador ARM1176JZF-S a 700 MHz y 256 MB de memoria RAM con el objetivo de ejecutar un sistema operativo Linux o RISC OS.2 3. Utiliza una tarjeta SD para el almacenamiento permanente. (ARM: Advanced RISC Machine).

Se trata de un sistema embebido que se puede conectar a un monitor o a un televisor, cuenta con puertos USB, slot SD (donde se carga el sistema operativo), puerto HDMI, micro USB para alimentación y un puerto Ethernet. Dada su juventud aún no se encuentra debidamente probada para su uso como nodos sensores.

### 2.4.7 PICAXE

Este tipo de soluciones consiste en adquirir un controlador y montar el resto de la placa según tus necesidades. Picaxe[41] consiste en un microcontrolador basado en un chip PIC, que tiene cargado un bootloader que permite ser programado fácilmente con un simple programa en BASIC o un diagrama de flujo.

Existen múltiples versiones de este chip con diferente número de pins (8, 14, 18, 20, 28 y 40), dependiendo del requerimiento, configurables como salida/entrada digital y, entrada analógica... Además cuenta con interfaces PWM, I<sup>2</sup>C, SPI y RS232.

# PROTOCOLO DE GESTIÓN DE NODOS SENSORES

# 3

Una vez analizadas las características básicas de las redes de sensores, tanto a nivel de arquitectura de comunicaciones como de plataformas hardware y software, se procede a describir el trabajo desarrollado en este proyecto.

El objetivo del presente trabajo es implementar una red de nodos sensores que permita la recolección de datos y la accesibilidad a dicha información en cualquier lugar. A lo anterior se debe incorporar la posibilidad de que los nodos sensores deben en algunos casos operar como nodos actuadores.

Seguidamente y, basándose en el estudio realizado en el capítulo anterior, se plantea la selección del hardware y de la tecnología de comunicaciones a emplear en el desarrollo de la red de sensores objetivo. Adicionalmente, se lleva a cabo el diseño y posterior implementación y validación de un protocolo de gestión de la red de sensores y, la transferencia de información entre ellos.

## 3.1 DISEÑO DEL PROTOCOLO

La red que se quiere desarrollar intercambia información, siendo más concreto, una serie de nodos finales se comunican con un nodo central o coordinador y viceversa. Para ello es necesario conocer cómo se envía esa información, cómo se representa. Así pues surge la necesidad de la implementación de un protocolo.

En el capítulo anterior se mencionó el protocolo IEEE 802.14.5 como vehículo de las comunicaciones inalámbricas, el que se va definir se encapsula dentro de él.

En las siguientes líneas se verá que el protocolo llevado a cabo es capaz de gestionar y descubrir los nodos existentes en la red, de almacenar los datos que recolectan del medio en el que miden sus sensores y, finalmente, de la configuración de diferentes modos de operación de los nodos finales.

### 3.1.1 ARQUITECTURA

La funcionalidad que se quiere dar a la red es la de recolectar datos y almacenarlos para su posterior disponibilidad, siendo una arquitectura en estrella la que más se ajusta a la funcionalidad prevista por sus requerimientos de centralización de las comunicaciones. Esta arquitectura en caso necesaria se podrá extender a varios clústers que se comunicarán entre sí, pero dentro del ámbito de este proyecto se considera únicamente la comunicación en un único clúster.

Como se observó en el capítulo anterior los módulos XBee de la serie 1 con el protocolo IEEE 802.14.5 pueden alcanzar una cobertura de hasta 100 metros en exteriores y 30 metros en interiores. El potencial uso del sistema planteado se circunscribe a un entorno de interiores con unas dimensiones inferiores, como puede ser una habitación o un aula.

Si se quisiera aumentar la cobertura de la red, en ese caso, se podrían añadir nodos intermedios, pero siempre teniendo en cuenta que el destino final es el nodo central que computa toda la información de la red. Para ese caso convendría utilizar otro protocolo como, Zigbee o Digimesh dado que con los módulos XBee utilizando el protocolo IEEE 802.15.4 solo permite dos tipos de funcionamiento de los nodos: coordinador y remoto.

Tal como se ha introducido, el sistema resultante quedaría de la siguiente manera: un único clúster formado por un nodo central o maestro y varios nodos esclavos formando una topología en estrella. En la Figura 3.1 se muestra la forma de la red.

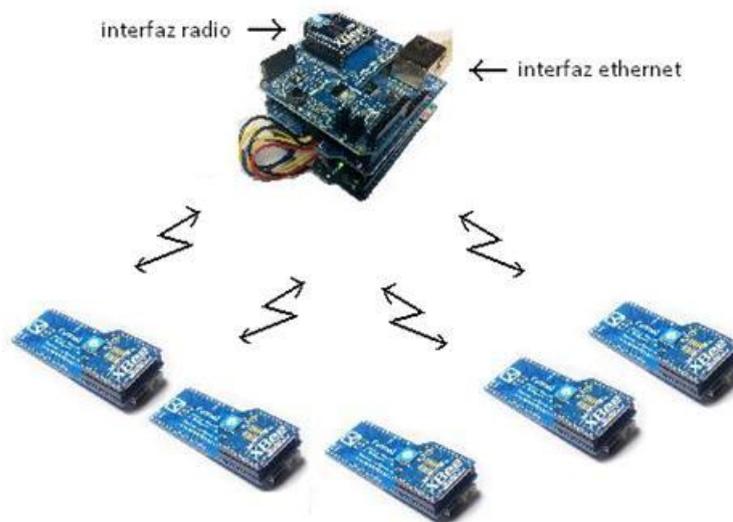


Figura 3.1 Red en estrella

El nodo central o coordinador cuenta con una interfaz Ethernet y una interfaz radio:

- La interfaz Ethernet permite al sistema conectarse a internet pudiendo así estar disponibles datos de la red si así se deseara.
- La interfaz radio permite la comunicación con el resto de nodos.

Por su parte, los nodos esclavos también cuentan con una interfaz inalámbrica a través de la cual mandan información al nodo central. Al tratarse de una red en estrella los nodos esclavos no se comunican entre ellos, sólo con su nodo central. En cambio este último puede

comunicarse con cualquiera de ellos. Sobre esta estructura de red, es el nodo esclavo quién define la funcionalidad o propósito del sistema, pues en él es donde se pueden incluir sensores de todo tipo, luminosidad, presión, etc. y cualquier dispositivo. En cualquier caso tan solo basta con conocer cómo se comunica el dispositivo en cuestión, ya sea un sensor u otro, como manejar los datos extraídos del mismo para luego simplemente enviarlos al nodo central como se haría con cualquier tipo de dato.

### 3.1.2 OPERATIVA

La estructura de la red plantea dos tipos de nodos bien diferenciados. Por un lado un nodo coordinador o también llamado central o maestro y, por otro el nodo esclavo, denominado igualmente remoto o final.

Los nodos maestros son los encargados de gestionar y recolectar la información proveniente de los esclavos. Seguidamente se introducirá el modo de operación de los mismos y la relación estrecha que tiene con el propio protocolo.

#### 3.1.2.1 NODO REMOTO

Cuando un nodo remoto inicia su funcionamiento lo primero que hace es tomar los valores de los sensores o dispositivos conectados a él, y a continuación lo notificará a su nodo central. De esta manera el nodo coordinador tiene conocimiento de la existencia del nodo remoto al recibir datos provenientes de él.

El nodo remoto nunca hace peticiones al nodo central, sólo o bien le responde por haber recibido de él una petición o bien notifica, en tramas generadas automáticamente con los parámetros de configuración del nodo y aquellos que pueda medir de sus sensores.

Antes de continuar con la explicación de la forma de actuar del nodo, al estar fuertemente relacionado, se introduce la existencia de dos variables, a las que se ha dado el nombre de *Asynchronous* y *Sleep\_Cycle*, que definen los diferentes modos de funcionamiento del nodo. Según los valores que tomen, se pueden diferenciar hasta cuatro estados o modos de funcionamiento distintos. *Asynchronous* puede valer 0 ó 1 y, *Sleep\_Cycle*, como su nombre indica se trata de un ciclo de sueño, que tomará un valor mayor o igual que cero (expresando segundos). Estas variables hacen referencia a la posibilidad de mantener en bajo consumo (o dormir) el módulo radio, siendo extensible al nodo por completo. De forma general, el modo de funcionamiento de un nodo remoto sigue la siguiente metodología:

- Escucha a través de su interfaz radio por si tiene alguna petición del coordinador y, en caso de que la hubiera la respondería.
- Mide los valores de los sensores o dispositivos conectados a él.
- A partir de este instante es donde las variables anteriormente introducidas juegan su papel, pues evaluándolas se define un comportamiento distinto en el nodo.

En la *Tabla 3.1* se muestran los diferentes modos de funcionamiento previstos y que se explicarán en detalle a continuación.

Tabla 3.1 Modos de funcionamiento

Modo	SLEEP_CYCLE	ASYNCHRONOUS
1	0	0
2	0	1
3	t	1
4	t	0

Analizando con detenimiento la tabla expuesta y, concretamente fijándose en la variable Sleep\_Cycle = 0 se diferencian dos modos:

- En el modo 1, la variable Asynchronous tiene como valor 0. Por lo que el nodo remoto permanecerá encendido<sup>2</sup> escuchando si es que hubiere alguna petición de su coordinador y, si ésta existiera la respondería como se puede observar en la Figura 3.2a.
- Con el modo 2, la variable Asynchronous toma el valor 1. El nodo permanece despierto, solo envía información si hubo algún cambio en los valores medidos de sus sensores y, además responde a peticiones si es que las hubiere. Este caso se puede comprobar en la Figura 3.2b.

Una vez visto los dos primeros modos, igualmente, prestando atención al parámetro Sleep\_Cycle si es distinto de cero, se pueden distinguir los dos restantes, en los que la dinámica de comportamiento es: escuchar al medio por si tiene alguna petición que contestar, medir los parámetros de sus sensores y, finalmente entra en modo sueño durante el periodo que indique la variable Sleep\_Cycle. Al despertar evalúa el valor del parámetro Asynchronous y, dependiendo de lo que éste marque el nodo enviará o no los datos leídos. Durante el tiempo que permanezca despierto volverá a medir esos parámetros y evaluará si los envía o no hasta que de nuevo vuelva a dormirse.

- En el modo 3 la variable Asynchronous es igual a 1. Por lo que luego de que el nodo entra en modo ahorro de energía durante t segundos, al salir de ese estado, lee sus sensores, compara los valores actuales con medidos antes de dormirse y, si fueran distintos enviaría los nuevos datos al nodo central. En la Figura 3.2c se pueden observar dos casos, cuando los datos recogidos comparados con los anteriores son iguales y al contrario.
- Finalmente, el modo 4. Aquí la variable Asynchronous vale 0 y el comportamiento es el siguiente: durante t segundos el nodo entra en modo de bajo consumo, al salir de él, lee los valores de sus sensores y los envía al coordinador, sean o no iguales que los anteriormente medidos como se indica en la Figura 3.2d.

<sup>2</sup> Cuando se hace referencia a permanecer encendido quiere decir que no se implementa ahorro de consumo de energía, como es el caso de los modos 1 y 2

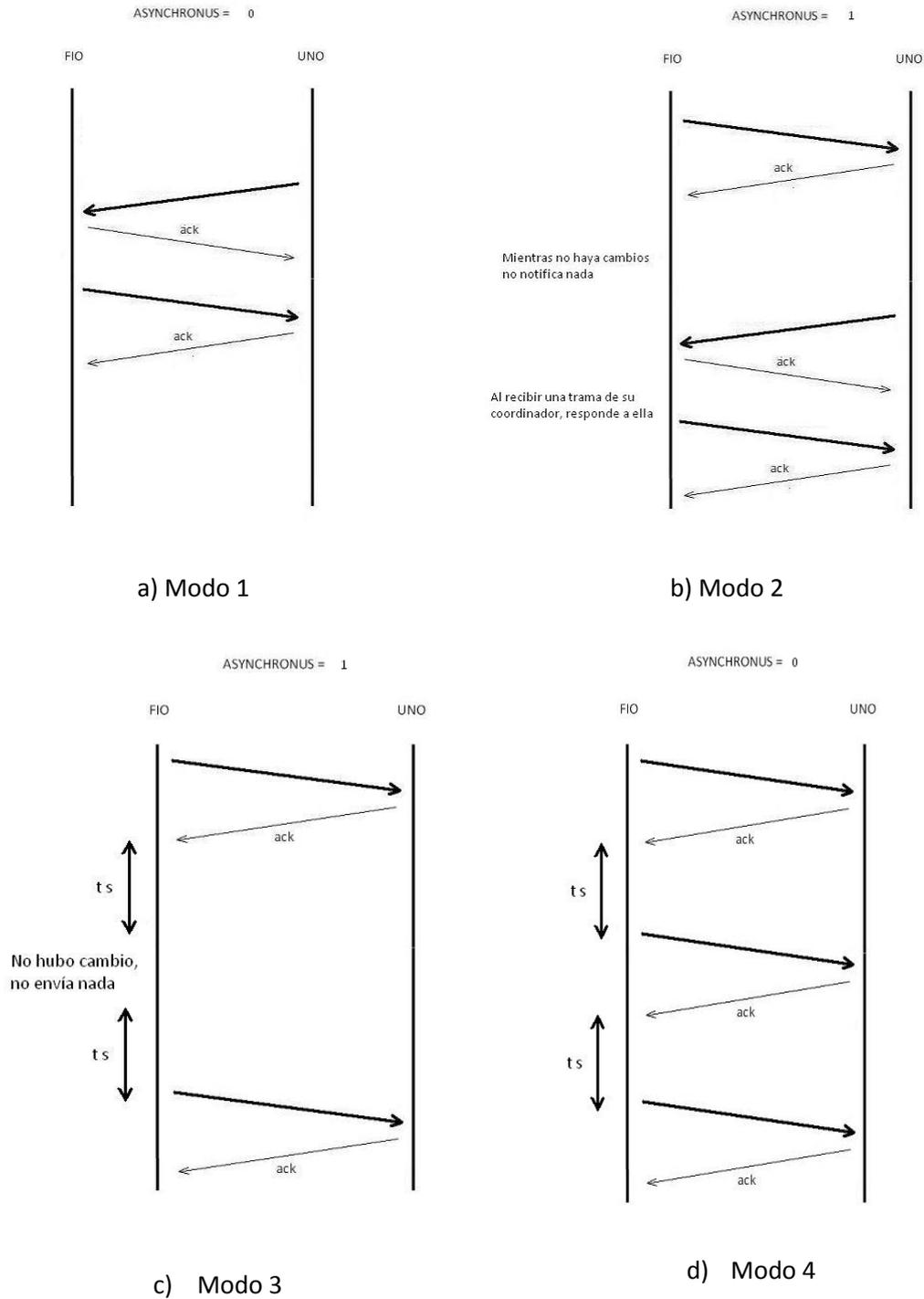


Figura 3.2 Modos de operación

### 3.1.2.2 NODO COORDINADOR

El nodo central o coordinador nada más iniciar su funcionamiento envía una trama de tipo broadcast, en la que pide información a los nodos de su red (en su radio de cobertura) para conocer los datos que puedan ofrecer en ese instante (parámetros de configuración, parámetros de los sensores, entre otros). Solo responderán aquellos nodos que compartan el mismo identificador de red y además estén escuchando en el mismo canal de trabajo. A medida que los nodos finales responden a esa trama, el coordinador recolecta la información y genera un histórico con ella.

El coordinador remite, cuando sea necesario, peticiones a los nodos sensores bien para proceder a su reconfiguración o para requerir el envío de datos. Estas peticiones, en el caso de que los nodos sensores puedan estar en modo de ahorro de energía, es posible que no obtengan respuesta. En ese caso serán encoladas y eventualmente se reenviarán. Por último cuando recibe información de un remoto, el coordinador la actualiza extrayendo los datos de la trama que reciba. Además aprovechará para comparar la dirección del remitente con una lista que confecciona poco a poco a medida que recibe datos de los nodos que pertenecen a su red.

### 3.1.3 PRIMITIVAS

Una vez explicado el funcionamiento de los nodos queda por definir cómo se entienden entre ellos, es decir, las primitivas del protocolo de comunicación que utilizan. El nodo coordinador necesita disponer de la capacidad de poder enviar tramas de configuración y de petición hacia los nodos finales y, estos últimos, para poder responder lo hagan a través de las correspondientes tramas de respuesta. Por último, un tipo de trama especial que permite a los nodos remotos dar información de su estado de manera autónoma y, que su coordinador es capaz de entender. Todas las tramas del protocolo cuentan un campo llamado tipo de trama o Frame Type y según el valor que tome distingue en cinco tipos de tramas:

- Trama CONFIGURATION o de configuración.
- Trama INIT
- Trama REQUEST o de petición.
- Trama RESPONSE o de respuesta.
- Trama NOTIFICATION o de información/notificación.

Cabe resaltar, que los campos de dirección (tanto destino como origen), identificador de trama, checksum, etc forman parte de las tramas IEEE 802.15.4 y, que lo aquí expuesto está encapsulado dentro del campo de datos, por lo que no hace falta indicarlo.

A continuación se explicará brevemente el formato y el valor de los campos de las primitivas del protocolo. Se han representado en tablas de tres filas, en las que la segunda fila se indica el nombre del parámetro y, en la tercera su valor. Para el encapsulado de datos se ha empleado el formato TLV (Tipo - Longitud - Valor), dado que permite extender de forma sencilla el conjunto de parámetros en caso de que se necesiten implementar nuevas funcionalidades o recolectar diferente información. No obstante, al trabajar con datos de longitud fija y para comprimir más la información se ha suprimido el campo longitud reduciendo el tamaño de datos enviados. Por último, en los siguientes apartados se observarán que cada campo en las tramas tiene un número asignado, en la [Tabla 3.2](#) a continuación reúne la mayoría de los utilizados en este trabajo. Como se observa hay una serie de códigos que son fijos y otros que se dejan a voluntad del usuario para futuros usos (RFU – Reserved for Future Use).

Tabla 3.2 Códigos

Código	Valor
CONFIGURATION	0x00
INIT	0x01
REQUEST	0x02
RESPONSE	0x03
NOTIFICATION	0x04
Asynchronous	0x01
Sleep_Cycle	0x02
...	
RFU	>0x0A

3.1.3.1 TRAMA CONFIGURATION

Este tipo de trama es enviada por el coordinador cuando quiere cambiar el funcionamiento de los nodos remotos, según los valores de las variables Sleep\_Cycle y Asynchronous configura al nodo remoto en uno de los 4 modos explicados anteriormente. La Tabla 3.3 muestra el formato de la trama.

Tabla 3.3 Trama Configuration

Frame Type	Length	Data Type	Value	Data Type	Value	...	Data Type	Value
CONFIGURATION		SLEEPCYCLE_FIELD		ASYNCHRONOUS_FIELD				
0x00	4	0x02	≥ 0	0x01	0 ó 1			

Como se observa podría haber más parámetros de configuración, según el usuario necesite. Como ejemplo, solo se tendría en cuenta los dos citados anteriormente.

Cabe recordar que la variable Sleep\_Cycle es un entero mayor que cero que mide segundos y, Asynchronous puede tomar los valores 0 ó 1.

Queriendo disminuir la carga de la trama, se podría compactar Sleep\_Cycle y Asynchronous en un único código, pues Asynchronous toma únicamente el valor 0 ó 1, (con un bit, bastaría) y Sleep\_Cycle valores no mayores de tres cifras por lo que con 7 bits sería de sobra. Por ejemplo, en un byte el bit  $b_7$  equivaldría a Asynchronous y  $b_6 b_5 b_4 b_3 b_2 b_1 b_0$  a Sleep\_Cycle. La trama es confirmada con un ack de nivel IEEE 802.15.4. Pero además, como se explicó en el apartado 3.1.2.1 al haber un cambio, en este caso de parámetros Sleep\_Cycle y/o Asynchronous, el nodo remoto ha de notificarlo con una trama (Notification, que se explica a continuación) de manera confirma que el cambio.

### 3.1.3.2 TRAMA INIT

Se trata de un mensaje broadcast enviado por el coordinador cuando entra en el sistema o cuando se inicia, a modo de conocer el número de nodos que forman parte de su red. La trama es interpretada por los nodos remotos de manera que responderán con una trama (Notification) que contiene la configuración actual del nodo y los parámetros de los sensores o dispositivos conectados a él. A continuación en la Tabla 3.4 se muestra la trama que como la trama CONFIGURATION, tiene un tamaño fijo.

Tabla 3.4 Trama Init

Frame Type	Length
INIT	
0x01	0x00

### 3.1.3.3 TRAMA REQUEST

Las tramas *REQUEST* son enviadas por el nodo maestro hacia uno o varios nodos esclavos para conocer un parámetro, o varios. Tienen la estructura que se observa en la Tabla 3.5:

Tabla 3.5 Trama Request

Frame Type	Length	Data Type	Value	...	Data Type	Value
REQUEST				...		
0x02	L					

### 3.1.3.4 TRAMA RESPONSE

Este tipo de mensaje es generado por los nodos remotos en respuesta a una trama tipo *REQUEST*, su campo de datos contiene los valores solicitados por el coordinador. Sigue el formato mostrado en la Tabla 3.6:

Tabla 3.6 Trama Response

Frame Type	Length	Data Type	Value	...	Data Type	Value
RESPONSE				...		
0x03	L					

### 3.1.3.5 TRAMA NOTIFICATION

Una trama de notificación es generada por los nodos remotos bien porque han sido sondeados por su coordinador (en respuesta a la trama INIT) o de acuerdo con alguno de los modos de funcionamiento explicados anteriormente (modo autónomo). Esta trama tiene un tamaño fijo y contiene los valores de todos los sensores o dispositivos que hayan conectado a él.

Tabla 3.7 Trama Notification

Frame Type	LENGTH	Data Type	Value		Data Type	Value
NOTIFICATION		Sleep_Cycle		... ..	Temperature	
0x04	L	0x			0x	t°C

En la Tabla 3.7 se ha indicado a modo de ejemplo la temperatura, pero dependería de la aplicación que se quiera implementar y, por lo tanto, los respectivos parámetros que mida. Como se puede observar el formato es idéntico a la trama RESPONSE, pero se optó por dar un identificador diferente para facilitar su gestión.

### 3.1.3.6 TRAMA ACK

Cuando se envía una trama IEEE 802.14.5, desde el destinatario es respondida con una trama de confirmación o acknowledgement al emisor. De esa forma se sabe que la trama se recibió correctamente. Si no llegara a su destino el protocolo implementado incluye un método de retransmisión. Y si aún así no fuera asentida, entonces, se almacenaría para su posterior envío. Lo aquí expuesto se profundizará más en el siguiente apartado.

## 3.2 PLATAFORMA DE DESARROLLO

A raíz del estudio realizado se ha identificado la solución basada en IEEE 802.15.4 como la más adecuada para su utilización en redes de sensores. Esto es así dado que su modo de operación supone un consumo de energía muy bajo manteniendo un buen balance con el rendimiento de las comunicaciones, tanto en alcance como en ancho de banda, para los requerimientos de este tipo de redes.

Si bien existen capas de control y gestión específicas que operan sobre IEEE 802.15.4, en este proyecto se ha preferido trabajar de forma nativa para así tener un mayor control sobre la operativa del protocolo y de esa manera contar con la oportunidad de tener mayores posibilidades de experimentación.

El presente trabajo busca realizar una prueba de concepto de las posibilidades de una red de sensores desatendida y aplicarla a un caso real como puede ser la gestión de aulas. En este sentido y, por el número de nodos que potencialmente se pueden desplegar, se busca un bajo coste. Dado que los requerimientos computacionales que se van a exigir a cada uno de los nodos no van a ser excesivos, se considera que dispositivos con un microprocesador de 8 bits ofrecen la potencia suficiente de cálculo necesaria para ejecutar las operaciones de recogida de datos y gestión de las comunicaciones. Estos microprocesadores proporcionan un reducido consumo de energía, lo que hace que el sistema se pueda considerar como adecuado para la implementación de nodos sensores.

De entre las diferentes plataformas hardware con estas características, por su reducido coste y capacidades de conexión y expansión, se ha decido realizar los desarrollos del proyecto empleando la plataforma Arduino. Se trata además de una plataforma cuyo hardware y software sigue filosofía de código abierto, lo que hace que en la actualidad exista una amplia comunidad de usuarios. Señalar que empresas con diversos sistemas desplegados en diversas

ciudades, como Libelium [42], parten de un fork de la plataforma Arduino, lo que demuestra la versatilidad de la elección.

De entre las diferentes versiones oficiales de Arduino existentes [43] se han seleccionado dos, que se ajustan a los tipos de nodos identificados para la red que se va a desarrollar:

- Arduino Uno.

Se trata del dispositivo Arduino estándar, que cuenta con un conector USB para su alimentación y programación como se puede observar en la Figura 3.3. Utiliza un procesador ATmega328 y, además, permite la interacción con otros dispositivos gracias a sus 14 pines digitales de entrada/salida, a sus 6 entradas analógicas y buses I<sup>2</sup>C y SPI. Es fácilmente ampliable en funcionalidades gracias a los denominados escudos o shields, conjunto de placas básicas que permiten incorporar capacidades adicionales (conexión Ethernet, GPS, comunicaciones inalámbricas, tarjetas de memoria, etc.).



Figura 3.3 Arduino UNO

- Arduino Fio

Similar en términos de funcionalidades a Arduino Uno, aunque con un microcontrolador menos potente. Se trata de una versión diseñada para trabajar en sistemas inalámbricos IEEE 802.15.4, incluyendo en su diseño un zócalo para conectar el módulo radio XBee correspondiente, como se puede observar en la Figura 3.4. Sus menores capacidades (opera a 3.3V y 8 Mhz frente a los 16 Mhz de Arduino Uno) y mayor nivel de integración, unido a la posibilidad de operar con batería le hacen un dispositivo idóneo para implementar sensores.

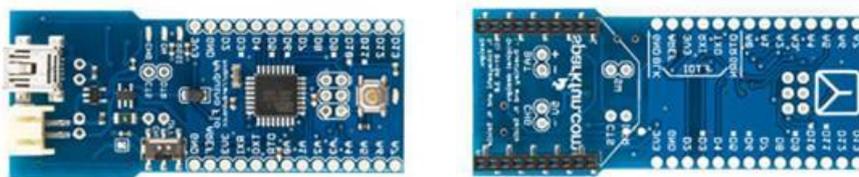


Figura 3.4 Arduino FIO

Según esto parece claro que Arduino Uno será el hardware empleado para desplegar los nodos coordinadores o agregadores de información, mientras que las Arduino Fio se encargarán de medir las métricas que se seleccionen. Aunque el nodo coordinador podría operar en términos de cómputo sobre una Arduino Fio, las capacidades de expansión casi nulas de esta hace que sea necesario emplear una Arduino Uno.

El nodo coordinador se encargará de recolectar la información de los diferentes nodos, realizar un pequeño procesado y almacenamiento de esta y, hacerla accesible a través de internet. Es por esto que debe tener conexión Ethernet con la pila de protocolos IP, además de memoria de almacenamiento. Por supuesto, debe también disponer de un modo de conexión con los nodos bajo su control, por lo que requiere conectividad IEEE 802.15.4.

Todas estas capacidades de expansión y otras más se encuentran disponibles a través de los escudos. Dichos escudos pueden interconectarse entre ellos para su uso simultáneo. Para el caso de este proyecto se han empleado los siguientes escudos:

- Escudo Ethernet

Permite a una placa Arduino conectarse a internet a través de un cable RJ45. Está basada en el chip Ethernet Wiznet W5100 [44] el cual está provisto de la pila de red IP que soporta TCP/UDP con hasta cuatro conexiones simultáneas. Incluye un slot para una tarjeta micro SD, que puede servir para almacenar archivos. Las versiones más recientes permiten la alimentación a través del clave UTP 5, simplemente añadiendo un módulo PoE (Power over Ethernet). La Figura 3.5 muestra un escudo Ethernet.

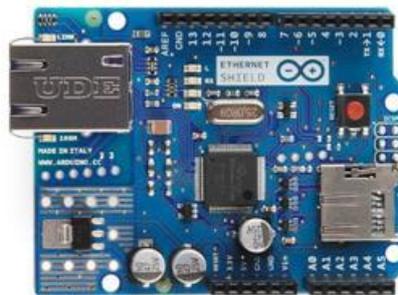


Figura 3.5 Escudo Ethernet

Existen librerías para desarrollar programas que se conecten a Internet usando el escudo [45] y, para gestionar el almacenaje en la memoria micro SD [46].

- Escudo XBee

El escudo XBee permite comunicar una placa Arduino de forma inalámbrica con tan solo apilarla sobre ella. Se trata básicamente de una placa que contiene un zócalo que se puede usar con módulos XBee 802.15.4 (Series 1) y XBee ZNet 2.5 (Series2). Fue creada con la colaboración de Libelium quienes la desarrollaron para utilizarlo con sus motas SquidBee [47]. El escudo se puede observar en la Figura 3.6.

Como se ha podido adivinar, el hecho de seleccionar la plataforma Arduino como hardware para implementar los nodos implica que la elección del fabricante de comunicaciones IEEE 802.15.4 sea Digi International, dada su integración en los diferentes escudos y las librerías de programación disponibles.



Figura 3.6 Escudo XBee

Los módulos XBee utilizados pertenecen a la Serie 1 [48] de la empresa Digi International, ya que la Serie 2 [49] opera empleando ZigBee y no nativamente sobre IEEE 802.15.4. Son módulos con antena integrada de chip, preparados para introducirse en placas de desarrollo. Trabajan en la banda de los 2.4 GHz e implementan la pila de protocolos IEEE 802.14.5. Su radio de cobertura alcanza los 100 metros en exteriores y 30 metros en interiores. Además la velocidad de transmisión es de unos 250Kbps. Disponen de dos modos de operación, bien a través de comandos AT o mediante acceso directo en modo API (Application Programming Interface). Para su configuración se puede utilizar el software que ofrece Digi, X-CTU, conjuntamente con una placa llamada XBee Explorer de Sparkfun. La Figura 3.7 muestra el tipo de módulo utilizado en el presente proyecto.



Figura 3.7 Módulo XBee

Para profundizar más sobre el software, uso, configuraciones, esquemático de los módulos XBee se recomienda acudir a la documentación online que se encuentra disponible en la página del fabricante. En el caso de Arduino, la principal fuente de información está en su página oficial, aunque también se puede acudir a una amplia literatura [50][51].

## 3.3 IMPLEMENTACIÓN Y TEST

Después de introducir el protocolo y los dispositivos que se han elegido para la implementación del sistema, en este apartado se explicará en primer lugar la configuración de algunos de ellos para más tarde mostrar los primeros pasos que se dieron con las placas antes de montar la red ejemplo final. En el presente capítulo, además, se hablará de la parte radio del sistema y, todo lo referido a las comunicaciones inalámbricas.

### 3.3.1 APROXIMACIÓN MÓDULOS XBEE

En esta sección se quiere dar una visión global de los módulos XBee elegidos para el desarrollo de este proyecto. Para ello se va a pasar por los siguientes puntos: sus modos de funcionamiento y cuál es el que más se ajusta a nuestras necesidades, el software y hardware disponible para su configuración y, finalmente, el montaje necesario para alcanzar un bajo consumo, los cuales se desarrollan a continuación, todos ellos necesarios para conseguir el objetivo al incluirlos en una red de sensores.

#### 3.3.1.1 MODO API DE XBEE

El protocolo explicado en el apartado anterior se puede implementar atendiendo a la configuración del módulo XBee. Por lo que antes de hablar del montaje y de la configuración de los propios módulos convendría comentar los dos modos en los que éstos pueden trabajar:

- Transparente o modo AT: los datos recibidos por el interfaz serie son encolados para su transmisión vía RF. Se trata del modo de funcionamiento por defecto.
- Modo API (Application Programming Interface): los datos tanto entrantes como salientes están contenidos en tramas.

El modo API resulta idóneo pues permite construir un protocolo en la sección de datos de la trama. Pero además con este modo se puede:

- Enviar datos a múltiples destinatarios sin entrar en modo AT.
- Recibir notificación debido al éxito o al fallo al enviar una trama.
- Identificar el origen del paquete (en modo AT la configuración es punto a punto)

Para que el módulo trabaje en modo API es necesario cambiar un parámetro del mismo, el llamado AP (API Parameter). Como se comentó anteriormente el hecho de trabajar con tramas facilita la construcción de nuestro protocolo. A continuación se muestra su formato en la Figura 3.8.

Start Delimiter	Length		Frame Data	Checksum
0x7E	MSB	LSB		1 byte

Figura 3.8 Tramas IEEE 802.15.4

La información enviada es empaquetada de la siguiente forma: se delimita, se especifica la longitud de los datos y, se incluye un control de errores o Checksum. Todo lo mencionado anteriormente se realiza de manera automática por lo que es transparente al programador.

Es en el campo de datos o Frame Data donde se encapsula el protocolo diseñado que es lo que interesa. Pero antes de continuar convendría introducir el direccionamiento de los módulos, que es, además, una manera de identificar y diferenciar un nodo de otro dentro de la red. Existen dos tipos: de 16 bits y de 64 bits. Con la primera opción el programador puede asignar la dirección a su voluntad en el rango 0 a 65533 y, la segunda es la dirección que viene de fábrica que es única y que coincide con el número de serie.

En este proyecto se optó por la primera, pues es una forma de identificar y llevar cuenta de los nodos más fácil de manejar que con un número tan largo. Además, debido a que la red se puede instalar en un aula, se consideró que con un solo byte es más que suficiente teniendo un total de 256 direcciones, desde la 0 a la 255. De esta manera se considera un único clúster formado un nodo maestro y de hasta 256 nodos remotos (siendo un número más que suficiente para una clase de tamaño normal). En la Tabla 3.8 se muestra la relación nodo-dirección.

Tabla 3.8 Direccionamiento

Nodo	Coordinador	Nodo1	Nodo2	...	Nodo 255	Broadcast
Dirección	0x1234	0x01	0x02	...	0xFF	0xFFFF

Volviendo a la estructura de las tramas, la librería XBee proporciona una serie de formatos que permite diferenciar si se trata de datos de entrada, de salida o reconocimientos (ack). A continuación se mostrará brevemente esos formatos para el caso de direccionamiento de dos bytes. Como se puede deducir existen otros para el caso de 64bits pero en el presente trabajo no es relevante, aun así, para más información se puede acudir al manual de usuario de los módulos XBee de la serie 1 [52]. Las siguientes tramas están incluidas en el campo Frame data que se explicó antes.

#### 3.3.1.1.1 TX16REQUEST

La estructura de trama empleada para transmitir un paquete con dirección destino de tamaño de 16 bits se observa a continuación en la Figura 3.9:

API Id	Frameld	@destination	Option	Data
0x01	1 byte	2 bytes	1 byte	<100 bytes

Figura 3.9 Trama TX16Request

Como se observa se definen los siguientes campos:

- API Id que identifica el tipo de trama, en este caso TX16REQUEST.
- Frameld es el identificador del paquete enviado (y de sus consecuentes *status response* o ack's tramas explicadas en el apartado 3.3.1.1.3).
- Option: si toma el valor 1 desactiva los status response, si toma el valor 0 el Frameld es 1 y activa los ack o asentimientos. Si es 4 se trata de envío broadcast a la PAN (Personal Area Network – Red de Área Personal).

### 3.3.1.1.2 RX16RESPONSE

La estructura de trama para la recepción de datos entrantes al módulo con una dirección de 16bits se comprueba en la Figura 3.10 siguiente:

API Id	@source	Rssi	Option	data
0x81	2 bytes		1byte	<100 bytes

Figura 3.10 Trama RX16Response

Igual que antes, se diferencian los siguientes campos:

- API Id identifica el tipo de trama, RX16RESPONSE.
- Rssi (Receive Signal Strength Indication): calidad de la señal recibida.
- Option: Se trata de un byte en el que cada bit indica una opción  $b_7b_6b_5b_4b_3b_2b_1b_0$ ,  $b_1$  dirección broadcast,  $b_2$  PAN broadcast, etc.

### 3.3.1.1.3 TXSTATUS

Cuando un mensaje TxRequest es enviado, el módulo genera este mensaje que indica si fue recibido correctamente o hubo algún fallo. Su formato es el mostrado en la Figura 3.11:

API Id	Frame ID	Status
0x89	1 byte	1 byte

Figura 3.11 Trama Status

Como en las otras tramas, cuenta con un campo API Id que indica que se trata de una trama TXSTATUS, un Frame Id que la identifica y, finalmente el campo Status que según su valor indicará el estado del envío, por ejemplo 0 si hubo éxito.

### 3.3.1.2 ESTADO DE BAJO CONSUMO XBEE

Los módulos XBee cuentan con varios modos de funcionamiento: transmisión, recepción, comandos, hibernación y espera. De todos ellos el modo que permite al módulo entrar en estado de bajo consumo es el de hibernación. Anteriormente se habló de unos parámetros llamados AT. Con ellos se puede cambiar la configuración del dispositivo. Hay dos maneras de hacerlo: a través del software de Digi, X-CTU o, por ejemplo, a través de un terminal en Linux. Volviendo al parámetro a cambiar, se trata del SM (Sleep Mode) que debe de tomar el valor 1 (Pin hibernate). Este modo usa el pin SLEEP\_RQ, pin 9 del módulo que se activa o desactiva por voltaje, por lo que:

- Si SLEEP\_RQ “asserted” → el módulo terminará cualquier transmisión/recepción, entrará en modo espera y finalmente en el estado de sueño. Durante ese periodo de sueño no responderá a ningún tipo de señal serie o RF.
- Si SLEEP\_RQ “de-asserted” → el módulo despertará y estará preparado para transmitir o recibir luego de un tiempo de recuperación.

Según las especificaciones en el estado de hibernación o sueño se consigue un consumo de entorno de  $10\mu\text{A}$ , mientras que en estado de espera o recepción es de  $45\text{mA}$  y en transmisión  $50\text{mA}$ . Lo que supone un ahorro considerable de energía.

### 3.3.1.3 X-CTU

X-CTU es un software propiedad de Digi International, disponible para sistema operativo Windows. Este programa permite la configuración de los módulos XBee cambiando sus parámetros AT a través del puerto serie de un ordenador. Como se puede observar en la Figura 3.12 de las cuatro pestañas con las que cuenta, la que más interesa es *Modem configuration*, a continuación se explicará porqué.

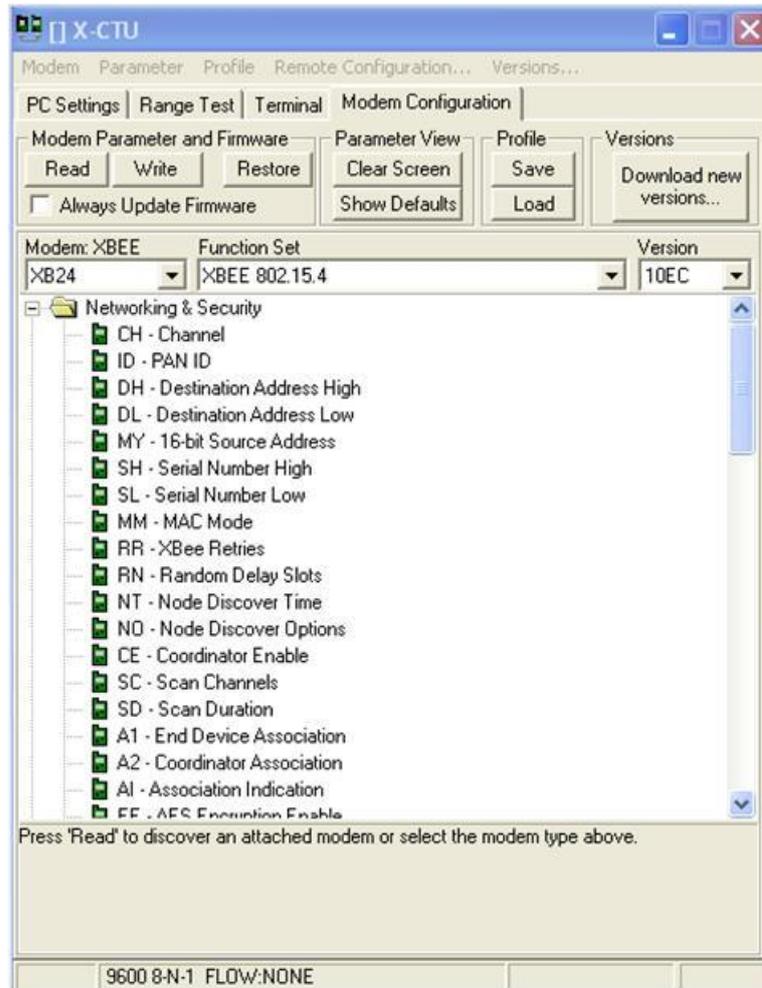


Figura 3.12 Software X-CTU

En esta pestaña se puede encontrar entre otras funcionalidades (imprescindibles en la configuración de un módulo radio): actualizar el firmware, guardar o cargar un perfil, restaurar los valores iniciales de fábrica y, leer y escribir parámetros. Ésta última opción es la que define el funcionamiento y caracteriza al módulo, por ejemplo permite conocer la dirección de fábrica, establecer el canal de transmisión, activar o desactivar el modo de sueño, trabajar en modo API, entre otros.

Para más información sobre los parámetros del módulo radio se debe acudir a la hoja de características del mismo [52] y, para el manejo del programa X-CTU en la página web de Digi se encuentra documentación sobre su uso [53]. Aunque en ambos casos existen foros y tutoriales en internet que ofrecen la misma información.

#### 3.3.1.4 XBEE EXPLORER

Se trata de una placa que permite conectar y utilizar cualquier módulo XBee directamente mediante un puerto USB. Basta con acoplar un módulo XBee a la placa y conectarlo a través de un cable mini USB al PC, de manera que a través del programa X-CTU se puede cambiar la configuración de los módulos, como se comentó en el apartado anterior. La Figura 3.13 muestra el módulo XBee Explorer fabricado por la compañía Sparkfun [54].

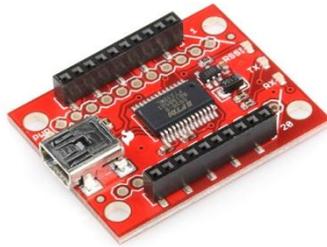


Figura 3.13 Placa XBee Explorer USB

### 3.3.2 ACERCAMIENTO A LA PLATAFORMA ARDUINO

A continuación al igual que se ha hecho con los módulos RF, se presentará el entorno software necesario para comenzar a utilizar Arduino, además de las librerías que hay disponibles y que ayudan a la creación de programas. Más tarde se incluirán los primeros pasos y ejemplos que se dieron durante el aprendizaje del entorno. Para finalmente incluir el envío de datos a través de la interfaz RF que está embebida en las placas Arduino.

#### 3.3.2.1 IDE ARDUINO

El software de Arduino permite generar programas o sketches, compilarlos e introducirlos en la placa. Es de distribución libre y se encuentra disponible para Windows, Linux y Mac OS X. En el caso particular que ocupa, durante el proceso se utilizó la versión 022 para Windows, actualmente está disponible la versión 1.0.3.

El lenguaje de programación que utiliza es parecido a C pero con algunas diferencias muy pequeñas. Además está en disposición del usuario múltiples librerías que complementan toda la plataforma, no solo para las placas oficiales, que al ser libre permite que cualquier usuario incluya mejoras en las mismas y que surjan otras. La Figura 3.14 muestra la ventana principal del programa.

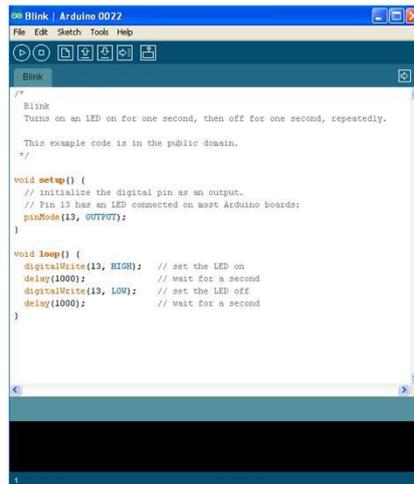


Figura 3.14 IDE de Arduino

### 3.3.2.2 BAJO CONSUMO DE LA PLACA

Para reducir el consumo de una placa, se puede utilizar diversas librerías creadas por usuarios y disponibles en la página web de Arduino. Algunas de ellas son: Enerlib [55], Narcoleptic [56], Power, Sleep, (las dos últimas incluidas en el repertorio de Arduino) todas ellas hacen uso del nivel de registro para reducir el consumo.

La técnica consiste en despertar la placa a base de interrupciones, en el caso que ocupa esta interrupción se produce cuando un contador alcanza una cifra. La placa puede realizar la cuenta con cuatro contadores counter 0, counter 1, counter 2 y, el contador Watchdog.

Así que simplemente utilizando instrucciones propias de la librería la placa entra en modo de bajo consumo. Lo malo de estos contadores es que no alcanzan valores de tiempo muy elevados. Para ello podría utilizarse un contador externo de mayor cuenta, pero haría necesario un controlador para él. Debido a esto la opción elegida fue el contador Watchdog que permite contar 1, 2, 4, y 8 segundos. Para valores distintos o mayores es necesaria una combinación de los básicos, por ejemplo 10 segundos equivaldría a dormir la placa primero durante 8 y luego 2 segundos o al revés. Para evitar este tipo de combinaciones, se ha fijado que los ciclos de sueño sean 8, 40 y, 64 segundos. Resultando así ciclos que consistirían en 1, 5, y, 8 intervalos de 8 segundos, respectivamente. Una vez alcanzado el valor deseado permanecerá despierto un tiempo que se ha fijado en 3 segundos, en el cual realizará las tareas que tenga programadas para una vez expirado el tiempo vuelva a dormirse.

### 3.3.2.3 LIBRERÍAS

Para el hardware mencionado en este capítulo ha hecho falta el uso de librerías que contienen instrucciones específicas que facilitan su manejo. Muchas de ellas contienen ejemplos de uso y, simplemente basta con agregarlas al IDE de Arduino para poder comenzar a utilizarlas. En el caso particular de desarrollo de las comunicaciones inalámbricas se ha usado la librería Xbee.h[57], que como su propio nombre indica controla el módulo radio. Para la gestión de la placa Arduino las librerías Timer2.h, Power.h y AVR.h (incluidas en el software de Arduino) que facilitan que entre en modo de bajo consumo.

### 3.3.3 PRIMEROS PASOS

Se empezaron los primeros contactos con el hardware y software Arduino a través de la guía rápida de instalación y uso disponible en la página oficial. Allí se explica cómo instalar el entorno de desarrollo (IDE, Integrated Development Environment) y, los drivers FTDI<sup>3</sup>. Referente a los drivers, en el caso particular de la placa Arduino UNO si se utiliza en un equipo con sistema operativo Windows hay que añadir un driver especial, cosa que no ocurre si se trabaja en el entorno Linux.

Una vez completados estos primeros pasos se inició la fase de aprendizaje del lenguaje y la metodología de programación. En el propio IDE se encuentran numerosos ejemplos sencillos de funcionamiento y control de dispositivos como, por ejemplo, un led.

En general un programa en Arduino consta de una estructura formada por tres bloques:

```
// Librerías

// Definición/inicialización de variables

void setup() {
    ...
}

void loop() {
    ...
}
```

En el primer bloque incluye librerías y definiciones e inicializaciones de variables. El segundo bloque corresponde con una función llamada setup () en la que se inicializan pines como entradas o salidas, o las comunicaciones serie, radio, Ethernet, entre otras. Finalmente, la función loop () es donde se define el comportamiento del nodo que se repite continuamente, a diferencia de setup () que sólo lo hace una vez. Es importante remarcar que el funcionamiento una vez compilado e introducido el programa en la placa es el siguiente: ejecuta una vez la función setup y, después, repite la función loop continuamente.

El entorno Arduino, como se mencionó incluye una serie de ejemplos de funcionamiento. Parte de la familiarización con la plataforma se probaron esos ejemplos, entre ellos:

- Manejo de la memoria E<sup>2</sup>PROM de la placa, escritura, lectura y borrado de datos.
- Control de funcionamiento de un diodo led - encendido, apagado, intensidad de luz desde un ordenador, introduciendo un carácter o un número a través de la interfaz serie de Arduino y visible a través de su monitor.
- Lectura en el monitor del IDE de Arduino de los valores de tensión en un pin que a su vez estaba conectado a un potenciómetro.
- Familiarización y configuración de los módulos XBee. Junto con ello, se probó la librería NewSoftSerial[58] que adiciona puertos para la comunicación serie. Por defecto la placa cuenta con dos: Rx y TX puertos 1 y 2, con ellos no se puede leer y escribir a la vez. Lo que ésta librería permite es utilizar otros pines como puertos serie de manera

<sup>3</sup> FTDI Future Technology Devices International, fabricante del micro controlador que lleva Arduino

que se pueden conectar otros dispositivos. Un ejemplo de uso fue introducir órdenes desde un ordenador de manera que desde un monitor se permitía enviar esas peticiones, los datos eran leídos y entendidos por la placa, para luego generar las tramas de petición. Y desde el otro monitor que interpretaba los datos de los nuevos puertos serie, se leían los datos que llegaban vía RF como respuesta.

A continuación se ahonda en la comunicación inalámbrica, así como en los parámetros programables que permiten la programación de los módulos y el formato de un programa en Arduino.

### 3.3.3.1 PRUEBAS CON MÓDULO XBEE

En el apartado 3.3.1.1 se hizo una introducción a las características de los módulos XBee, en él se comentó que el modo de funcionamiento a utilizar era el modo API y, que para ello era necesario cambiar los parámetros del módulo que definen ese comportamiento. Una forma de hacerlo es usando el software X-CTU junto con la placa USB-Xplorer, introducidos en apartados anteriores.

Continuando con la configuración de los parámetros, luego de estudiar el funcionamiento a través del manual que ofrece Digi como también tutoriales en internet, se pasó a programar los módulos XBee. En el caso de este ejemplo la configuración es la mostrada en la [Tabla 3.9](#).

Tabla 3.9 Parámetros módulo XBee

Coordinador	Nodos Remotos
AP=2	AP=2
CE=1	CE=0
MY=1234	MY=00ii
ID=3332	ID=3332
	SM = 1
CH=C	CH=C
RR=3	RR=3

Antes de configurar los módulos conviene restaurar los valores por defecto, por si se hubiera tocado algún parámetro que pudiera ocasionar un comportamiento no deseado.

En cuanto a los parámetros que se observan en la tabla, el parámetro AP hace referencia a que se va a utilizar la configuración API. El parámetro CE habilita funcionamiento en modo coordinador si toma el valor uno, sino será cero para el resto de nodos. Los parámetros MY, ID y CH son la dirección del módulo, la dirección de la PAN (Personal Area Network) y, el canal a utilizar, respectivamente.

El parámetro SM (Sleep Mode) con valor igual a uno, selecciona el tipo de modo de sueño hibernación, el módulo se despierta/duerme a través de voltaje en el pin 9 del módulo XBee

(explicado en el apartado 3.3.1.2). Este parámetro es configurado en los nodos remotos pues son éstos lo que van a llevar a cabo el modo de ahorro de energía. Y finalmente, el parámetro RR que indica el número de retransmisiones en caso de fallo, en este caso se ha fijado a 3 en ambos tipo de nodo.

Una vez se ha configurado el módulo, se pasa a implementar la comunicación. Para ello se hizo uso de la librería XBee [57] que contiene funciones que permiten el entendimiento entre nodos en modo API.

Así pues conocida la configuración de los módulos radio se va a ver un ejemplo que en realidad es parte del código utilizado en este proyecto. Simplemente se trata de enviar información de un lado a otro, la complejidad se puede encontrar en cómo se mide o se calcula ese dato. En este caso se utilizó un sensor de temperatura y humedad. Se trata del sensor SHT15 de la empresa SENSIRION cuyo rango de medidas está entre los -0.3 y 23°C, y en el caso de la humedad entre 0 y 100%, con una precisión de 14 bits y 12 bits respectivamente. A continuación se muestra en la Figura 3.15 el módulo utilizado.



Figura 3.15 Sensor SHT15

Este sensor junto con la placa Arduino Fio y, el módulo de comunicaciones radio forman el nodo remoto. Para poder hacer las lecturas del sensor es necesaria una serie de funciones que interactúan con él. Una vez medidos esos parámetros simplemente hay que empaquetarlos para enviarlos como se haría con cualquier otro dato. A continuación en la Figura 3.16 se muestra el funcionamiento resumido de lo que el nodo realiza.

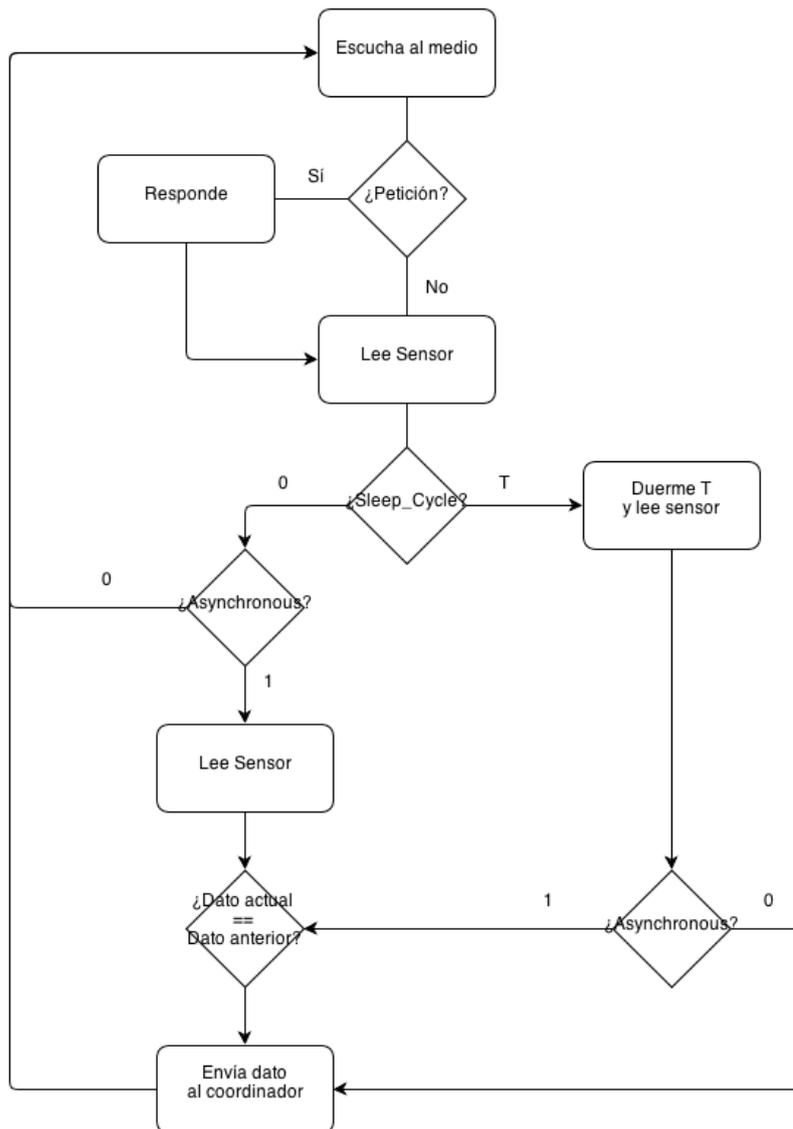


Figura 3.16 Diagrama funcionamiento Nodo Remoto

### 3.3.4 EVALUACIÓN DEL PROTOCOLO

El ejemplo utilizado ha sido el de una red que mide temperatura y humedad por lo que es lógico pensar que su funcionamiento normal sea aquel en el que se transmitan datos siempre y cuando haya habido un cambio en las lecturas.

También se comentó que un nodo se podría configurar de manera que envíe información de forma continua luego de despertar, haya o no cambio en la medida de los parámetros de temperatura y/o humedad. Con esto se pretende ver que en el peor de los casos si muchos nodos envían información de golpe al nodo coordinador, evidentemente, es muy probable que no sea capaz de leer todas las tramas en el caso de que la red tenga una cantidad de nodos grande y, por lo tanto se pierdan algunas de ellas, pero aún así la capacidad de lectura es bastante rápida. Durante las pruebas se experimentó a enviar muchas tramas desde los nodos que se disponía para ver la respuesta del coordinador. Se observó que la mayoría de ellas llegaban. Pero retomando la idea anterior, los cambios de temperatura y humedad serán paulatinos y no abruptos en una situación normal.

Dado que el sensor tiene una sensibilidad de decimales, se optó a quedarse con el valor entero de la medida, de esta forma se evita un número excesivo de cambios de manera que la sensibilidad sea de un grado en el caso de la temperatura y de un 1% para la humedad.

En el caso de que tramas no lleguen a su destinatario, el nodo remoto y también el coordinador cuentan con un sistema de reenvío y, si no recibe un asentimiento entonces lo intentará hasta tres veces (este parámetro podría ser configurable y añadirse a Sleep\_Cycle y Asynchronous).

En el caso del nodo remoto, la pérdida de una trama no es importante. Desde el punto de vista del coordinador, éste puede hacer una petición y encontrarse que no recibe respuesta del nodo preguntado. Eso significaría que está dormido o bien no operativo. Por lo que esa petición se almacenaría en la memoria y cuando el nodo se despertara si es que envía información de cambio de temperatura o humedad, el nodo maestro sabría que está despierto por lo que entregaría la petición que se guardó. Inicialmente se utilizó la memoria EPROM como sistema de almacenaje, pero su capacidad reducida no la hace apropiada para ello, más si la red está formada por una cantidad de nodos considerable, por lo que se empleará un almacenaje en tarjetas SD de memoria externa.

En las pruebas iniciales se probó a enviar tramas pidiendo la temperatura o la humedad, para comprobar el correcto funcionamiento de la implementación. No obstante, dicha operativa se recomienda para casos en los que se requiera adquirir un conjunto más grande de parámetros, cuyos valores no se obtengan por procedimientos automáticos. Para el caso broadcast, como no hay asentimiento, si algún nodo se encuentra en modo de bajo consumo no le llegaría la información por lo que esa orden se perdería.

En las siguientes versiones, igualmente si un nodo por lo que sea no estaba disponible cuando se envió la petición se almacena y cuando llega información de él se chequea si hay alguna petición pendiente que entregarle. Pero esto es algo que se explicará con mayor detenimiento en el siguiente capítulo.

Anteriormente se ha hablado de reenvíos y/o almacenamiento en el caso de que no sea confirmada. Como se introdujo en el apartado 3.1.3.6 se utiliza las tramas de confirmación del protocolo 802.15.4 en lugar de crear una al nivel del protocolo utilizado, disminuyendo así la cantidad de datos en tránsito.

El tiempo de sueño de un nodo no sobrepasará el minuto por lo que en algún momento enviará un cambio de temperatura y, el coordinador podrá entregar la petición. En cualquier caso si no es posible transmitir dichas tramas existe un mecanismo de control periódico que descarta las tramas definitivamente.

Tal y como está configurado el sistema, enviar una trama broadcast y que ésta sea escuchada por todos los nodos de la red se daría solo si todos están despiertos, sino, si alguno está dormido no la escucharía. Por lo que es conveniente que se duerman todos de golpe con el mismo ciclo de sueño. La idea implementada consiste en que cuando se duerme al grupo por completo una variable contiene el tiempo de sueño, cuando éste se alcance se envía la información pertinente. Hay un tiempo en el que los remotos permanecen despiertos, así que durante esa ventana se podría cambiar la configuración. Este es un modo operativo para redes pequeñas sencillo que emula sincronización y hacer que la red duerma durante un tiempo

determinado. Ciertamente existe ya un protocolo que hace uso del protocolo IEEE 802.15.4 que sincroniza y duerme la red entre otras opciones y, se trata del propietario DigiMesh.

Un aspecto importante es conocer por parte del coordinador los nodos que tiene bajo su control. Para tener una idea de los nodos existentes en la red, el coordinador envía una trama broadcast que se llamó INIT que junto con las tramas periódicas de notificación es capaz de conocer el número de nodos presentes, así como su dirección y parámetros de los sensores.

Finalmente, tanto los nodos remotos como el coordinador están programados, al añadir un nuevo nodo se debe de saber que direcciones libres hay.

# CONTROL DE PRESENCIA 4

En el capítulo anterior se introdujo el protocolo de gestión de la red de sensores y su implementación haciendo uso de la interfaz inalámbrica. De esta forma se da soporte al intercambio de información entre los nodos remotos y el nodo coordinador, tanto para su configuración como para la adquisición de datos.

Los desarrollos realizados hasta el momento requerían de un entorno de control sustentado en la comunicación serie entre un computador y las placas Arduino. No obstante, en los objetivos planteados para este proyecto se pretende que el sistema sea autónomo y disponga de un interfaz de gestión remoto accesible desde internet. Seguidamente se describen los desarrollos realizados para hacer operativo dicho interfaz.

Adicionalmente se plantea e implementa un aplicativo de uso del sistema de gestión mediante sensores. La solución escogida en este caso pretende ayudar en la gestión de aulas, bibliotecas o cualquier otro entorno de interiores que tengan un número limitado de plazas.

## 4.1 ESCENARIO

Hoy en día para acceder a espacios públicos compartidos como bibliotecas, salas de informática, etc. es necesario identificarse, e incluso en algunos casos reservar un sitio, de forma que se haga un uso eficiente y balanceado de los recursos disponibles.

En los casos que este tipo de entornos están desatendidos, normalmente los usuarios realizar un uso excesivo de los recursos (mesas de estudio, ordenadores, etc.) sin tener en cuenta que hay otros usuarios esperando o con voluntad de usarlos. Además existen situaciones donde los usuarios dejan el puesto durante un periodo de tiempo muy largo y que nuevamente perjudica el correcto uso de las instalaciones.

Basándose en estas premisas, el sistema desarrollado pretende solventar estas situaciones mediante el uso de redes de sensores. Para ello, supone que el elemento identificador del usuario es una tarjeta inteligente sin contactos, tal como ocurre con el personal y alumnado de la Universidad de Cantabria.

Los nodos remotos serán capaces de mediante un lector de tarjeta inteligente detectar e identificar al usuario y transmitir esa información a un nodo que se encargará de visualizar la identidad y tiempo de uso del puesto. En caso de que el usuario abandone su puesto, el nodo sensor remitirá una notificación. De esta forma se tiene conocimiento de cuánto tiempo queda el puesto sin uso para que el personal encargado pueda proceder a liberar el puesto si el tiempo es excesivo y permitir su uso a otro usuario.

Los nodos sensores proporcionan además información sobre la temperatura y humedad en los puestos. Cruzando esta información con el número de usuarios existentes en la sala se puede realizar una gestión del aire acondicionado o calefacción.

Otro conjunto de acciones posibles a realizar mediante el nodo sensor el control del puesto de trabajo, es decir, habilitar la iluminación, tomas de red, etc. únicamente cuando el usuario esté en el puesto. De esta forma, al abandonar el puesto se apagaran las luces y se liberarán los recursos de red permitiendo una disminución en el gasto.

En la Figura 4.1 se observa un ejemplo de la disposición de un clúster de nodos en una sala.

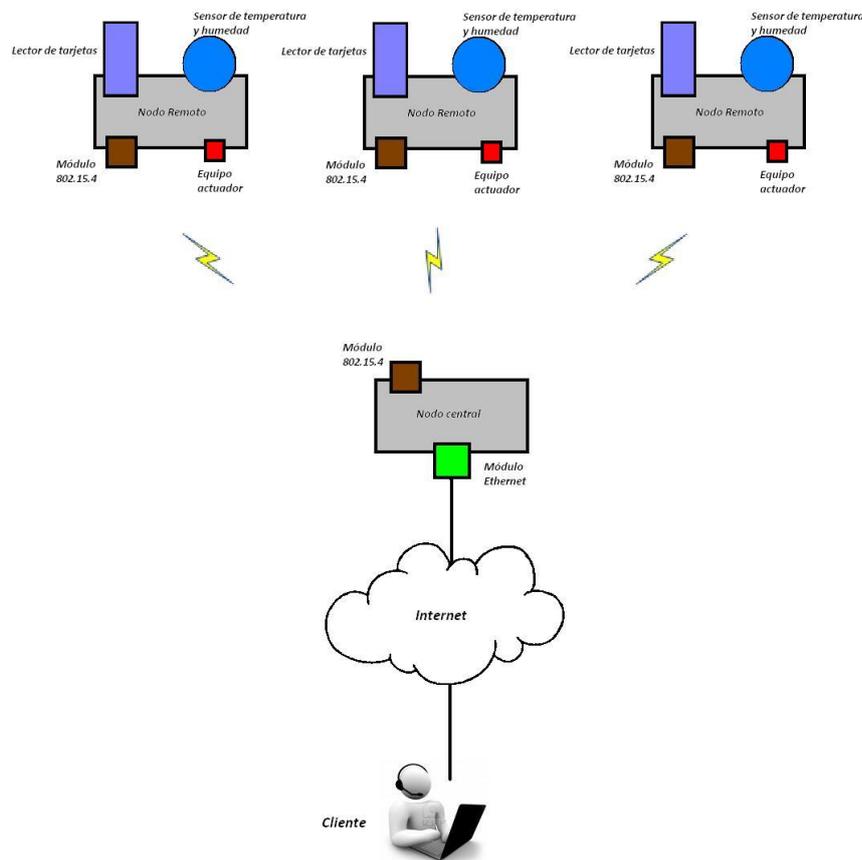


Figura 4.1 Sistema Completo

La conexión con los nodos de control se realiza mediante una conexión Ethernet, lo que facilita su gestión desde cualquier emplazamiento mediante un interfaz web. Estos nodos recolectan y almacenan la información proveniente de los nodos remotos para poder ser analizada y reportada a posteriori por los administradores. Dado que proporcionan un interfaz de gestión remoto, los nodos de control implementan una funcionalidad de servidor web, el cual atenderá a las peticiones que se realicen por los administradores.

Dar a la placa coordinadora, Arduino UNO, la capacidad de salir a internet supone incluir un escudo más, el escudo o shield Ethernet. Pero el hecho de poder implementar el servicio de administración y visualización de datos implica otra serie de conocimientos y manejo de la web, como Ajax y JSON. Es por eso que en los siguientes apartados se ahondará en ello.

## 4.2 TECNOLOGÍAS DE DESARROLLO

A continuación se introducen algunos aspectos de interés de las tecnologías empleadas para la implementación del sistema anteriormente comentado. Primeramente se hace una aproximación de las tarjetas sin contactos empleadas, analizando sus características y disposición de memoria.

Seguidamente, se describen las tecnologías de desarrollo empleadas para proporcionar el acceso remoto a la información a través de redes IP. Se ha optado por emplear soluciones web basadas en un servicio web REST y encapsulando los datos en el formato JSON. Todo ello en combinación con AJAX, permite optimizar la carga de datos transmitidos por la red y la descarga de funcionalidades y necesidades de cómputo a los servidores implementados en los nodos remotos.

### 4.2.1 TARJETA MIFARE, TECNOLOGÍA NFC

Como se ha introducido, se decidió adjuntar un lector de tarjetas a los nodos remotos. Es por ello que a continuación se hará una breve introducción a la arquitectura de las tarjetas Mifare que son las utilizadas en el presente trabajo y que funcionan bajo la tecnología NFC (Near Field Communication).

#### 4.2.1.1 TARJETAS MIFARE

Las tarjetas Mifare [60] son un tipo de tarjetas sin contacto que utilizan tecnología de radiofrecuencia para la transmisión de información, tanto para leer como escribir, con lectores de proximidad. Mifare es una tecnología descrita en el estándar ISO/IEC 14443 Tipo A [61], tratándose más que de una tarjeta de memoria sin procesador. En la Figura 4.2 se muestra un sistema típico formado por un lector y una tarjeta en el que el rango típico de funcionamiento es de en torno a 10 cm. Y donde ambos dispositivos cuentan con una antena y la comunicación se realiza mediante radiofrecuencia.

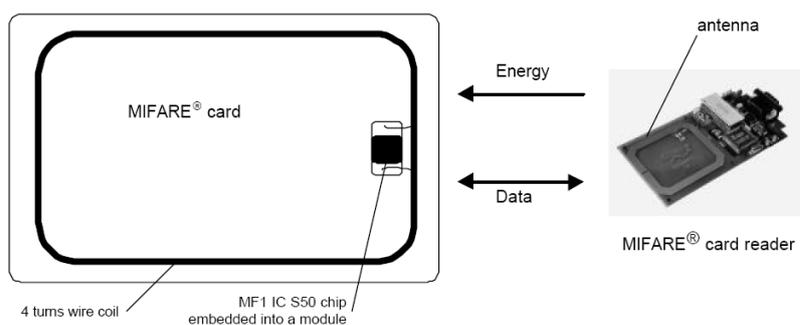


Figura 4.2 Sistema lector y tarjeta

Las tarjetas Mifare estándar cuenta con una memoria de 1Kbyte, la cual está distribuida en 16 sectores que constan de 4 bloques de 16 bytes cada uno como se representa en la Figura 4.3. Además, cada sector está compuesto por 3 bloques dedicados a almacén de información.

Sector	Block	Byte Number within a Block														Description	
		0	1	2	3	4	5	6	7	8	9	10	11	12	13		14
15	3	Key A					Access Bits				Key B					Sector Trailer 15	
	2	Data															
	1	Data															
	0	Data															
14	3	Key A					Access Bits				Key B					Sector Trailer 14	
	2	Data															
	1	Data															
	0	Data															
:	:																
:	:																
:	:																
1	3	Key A					Access Bits				Key B					Sector Trailer 1	
	2	Data															
	1	Data															
	0	Data															
0	3	Key A					Access Bits				Key B					Sector Trailer 0	
	2	Data															
	1	Data															
	0	Manufacturer Block															

Figura 4.3 Mapa de memoria

Antes de realizar una operación sobre la memoria es obligatoria la ejecución de la autenticación mutua y securización del canal de comunicaciones entre el lector y la tarjeta. Este procedimiento se lleva a cabo mediante un triple intercambio definido en el ISO 9798-2[62], que consiste en la generación de un número aleatorio que combinado con claves alojadas en la tarjeta es enviado al lector. Éste último responde con un mensaje de confirmación. La operación descrita se realiza tres veces para verificar que la tarjeta leída es válida, si lo es entonces se puede tener acceso a la información de los bloques de memoria.

Según los requerimientos de los usuarios empleando el mismo interfaz de comunicación, se puede encontrar los siguientes modelos de tarjetas:

- Mifare Mini: empleadas en aplicaciones que requieren poca capacidad de memoria, 320 bytes (de memoria EEPROM), además cuenta con un número de serie único de 4 bytes y, 5 sectores para distintas aplicaciones. Un ejemplo de uso sería el control de acceso o tarjetas de identificación.
- Mifare Standard 1Kb/4Kb: utilizadas en servicios de transporte público multi-modales, dispone de una memoria EEPROM de 1Kbytes/4Kbytes, organizados en 16/40 sectores y con un número de serie único de 4 o 7 bytes.
- Mifare DESFire: ofrece las mismas funcionalidades que Mifare Estándar pero con mayor flexibilidad, seguridad y rapidez, pero emplean el algoritmo DES para el cifrado de la comunicación.

#### 4.2.1.2 NEAR FIELD COMMUNICATION

NFC[63], es una tecnología de comunicación e identificación de corto alcance o campo cercano. Está basada en la Identificación por Radio Frecuencia o RFID (Radio Frequency Identification), que utiliza la inducción de campo magnético para hacer posible la comunicación entre dispositivos electrónicos. Entre sus características se encuentran que opera a una frecuencia de 13.56 MHz (banda ISM), con un rango de funcionamiento que varía entre los 0 y 20 cm, alcanzando una tasa de transferencia de datos de 106, 212 y 424 Kbps y permitiendo comunicaciones half-dúplex, es decir, en ambos sentidos pero no de manera simultánea. NFC se podría calificar como una tecnología segura, debido a su corto alcance ya que prácticamente los dispositivos tienen que tocarse para llevar a cabo la comunicación. Adicionalmente, podría decirse que proporciona un modo de acceso intuitivo: “Si quieres un servicio, tócalo”.

Por último, NFC se encuentra regulada en los estándares ISO 18092[64], EC-MA 340[65] y ETSI TS 102 190[66], que especifican capacidades básicas como velocidad de transferencia, codificación de bits, modulación y protocolo de transporte. Además de los modos de comunicación pasivo y activo que se explicarán a continuación.

Existen varios modos de operación de los dispositivos NFC, pudiendo actuar como lector, emular el comportamiento de una tarjeta o bien establecer comunicaciones entre pares (P2P).

Se dice que un dispositivo opera en modo activo cuando genera su propio campo de radiofrecuencia, por ejemplo un lector de tarjetas. Mientras que un dispositivo es pasivo cuando utiliza el acoplamiento inductivo para transmitir datos, por ejemplo una tarjeta Mifare. Es decir, la tarjeta al aproximarse al lector utiliza el campo RF que éste genera y realiza la transferencia de información.

#### 4.2.2 REST

REST [67] (Representation State Transfer) es un tipo de servicio Web. Un servicio Web se puede definir como un conjunto de protocolos y estándares que intercambian datos entre aplicaciones software desarrolladas en lenguajes de programación diferentes y ejecutados sobre cualquier plataforma.

Es bueno aclarar que REST no es una tecnología, ni siquiera una arquitectura, sino una familia de arquitecturas. Y aunque REST no sea un estándar, está basado en otros estándares:

- HTTP (Hypertext Transfer Protocol)
- URL (uniform resource locator)
- Representación de los recursos: XML/HTML/GIF/JPEG/JSON...
- Tipos MIME: text/xml, text/html, text/json,...

En REST uno de los elementos clave es el concepto de recurso. Se define entonces un recurso como una entidad que puede ser accedida públicamente. Por ejemplo un documento, una imagen, el tiempo hoy en Nueva York... Cada recurso tiene un identificador único, por ejemplo un URI (Uniform Resource Identifier). A través de él, es posible acceder a una copia desde el exterior, en el caso de cliente/servidor, un cliente quiere leer un recurso, lo que lee

en realidad es la copia del original (que está almacenado en el servidor). La forma en la que se puede leer es a través de su representación asociada y, puede estar implementado a través de un documento XML, HTML, JSON, entre otros. Incluyendo la lectura se definen cuatro operaciones que actúan sobre un recurso: create, delete, read y update, o lo que es lo mismo crear, borrar, leer y actualizar un recurso, respectivamente.

Un ejemplo típico y donde se reconoce REST es en la web. En este caso la aplicación cliente típica es el navegador, que operarían sobre las URI (identificador único) a través del protocolo HTTP operando con sus primitivas. En el caso habitual cada página sólo admite una representación, típicamente HTML, PDF, texto plano o imágenes. Pero eso no significa que una misma URI no pueda soportar distintas representaciones. Por lo que la web no es más que un ejemplo perfecto de servicios distribuidos a nivel global totalmente interoperable.

Los sistemas que siguen los principios REST se llaman con frecuencia Restful. Además de REST, aunque no compete con este proyecto, existen otros estilos de servicios web como RPC (Remote Procedure Calls) y SOA (Service-Oriented Architecture).

### 4.2.3 AJAX

Ajax, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o también llamadas RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene una comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas de forma local sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones y por tanto mejorar la calidad de experiencia (Quality of Experience – QoE) del usuario.

Ajax es una tecnología asíncrona, ya que los datos adicionales que se soliciten al servidor se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y, utilizable en muchos sistemas operativos y navegadores, dado que está basada en estándares abiertos como JavaScript y Document Object Model (DOM) [68].

### 4.2.4 JSON

JSON[69] [68] es el acrónimo de JavaScript Object Notation y, se trata de un formato ligero de intercambio de datos. Es fácil de escribir y leer para humanos, de parsear y generar para máquinas.

A diferencia de XML, donde la información independiente de su tipo es tratada como un string, el objeto JSON permite que sus datos tengan tipo: string, número, array, booleanos..., etc. De manera que el objeto JSON no es más que una estructura, de parejas de nombres y

valores. Este formato está soportado por una gran cantidad de lenguajes de programación, por ejemplo en JavaScript es un tipo de objeto y su uso está bastante generalizado como alternativa a XML en AJAX.

Está delimitado por corchetes, cada par de nombre/valor está separado por comas, y el valor está antecedido por dos puntos. El siguiente ejemplo muestra un objeto JSON sencillo:

```
{
  "id": "0001",
  "type": "donut",
  "name": "Cake"
}
```

## 4.3 NODO SENSOR REMOTO

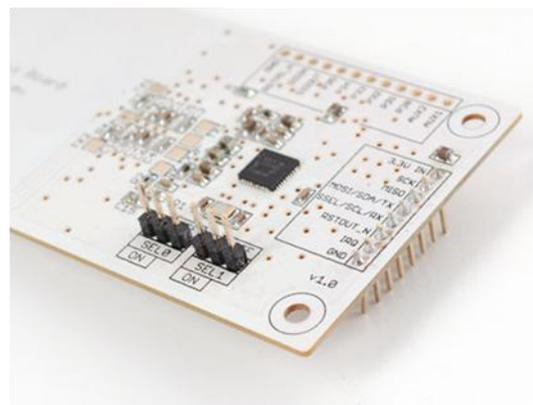
El funcionamiento del nodo remoto es el mismo que se explicó en el capítulo anterior, la única diferencia es que se incluye un lector de tarjetas. Lo que significa que además de hacer una lectura de los parámetros de temperatura y humedad se incluyen también los de este dispositivo. Es por eso que a continuación se hace una breve introducción a la configuración del lector para que pueda interactuar con la placa Arduino.

### 4.3.1 CONFIGURACIÓN DEL LECTOR DE TARJETAS

El lector de tarjetas empleado es el PN532 Breakout Board perteneciente a la empresa Adafruit [70], está basado en la tecnología NFC (Near Field Communications), permitiendo no solo la lectura de tarjetas sino también la operación en los otros modos NFC. Utiliza el chip PN532, bastante conocido y extendido, que permite la comunicación serie UART, o a través de los buses I<sup>2</sup>C y SPI. Necesita una tensión de alimentación de 3,3 V y es capaz de comunicarse con tarjetas inteligentes sin contactos y dispositivos NFC. La Figura 4.4a muestra el lector de tarjetas basado en el chip PN532.



a) Lector



b) Ampliación de interfaz selección

Figura 4.4 Lector de tarjetas

En el proyecto actual se eligió la comunicación SPI por ser la más rápida de las tres. En su contra tiene que usa una mayor cantidad de conexiones. Para seleccionar cualquiera de ellas, la placa incluye unos jumpers que según su posición se elige la interfaz deseada como se observa en la Figura 4.4b.

En la Tabla 4.1 se puede comprobar según la combinación de SEL0 y SEL1 como se seleccionaría una interfaz u otra. El modo UART solo se requiere los pines GND, VIN (3,3v), MOSI y SSEL. Este modo está más orientado a la comunicación con un host que la soporte (por ejemplo, un ordenador), en cambio los otros dos pueden ser también usados desde Arduino.

Para la comunicación SPI además de los pines referidos anteriormente hay que añadir los SCK y MISO. Por lo que la conexión con la placa Arduino FIO quedaría según se muestra en la Tabla 4.2.

Tabla 4.1 Interfaces

	SEL0	SEL1
UART	OFF	OFF
SPI	OFF	ON
I <sup>2</sup> C	ON	OFF

Tabla 4.2 Pines Arduino vs Lector

Lector	Arduino FIO
SS	10
MOSI	11
MISO	12
SCK	13

La configuración del lector sigue un Baudrate de 115200, modo asíncrono con 8 bits de datos, ningún bit de paridad y, 1 bit de parada.

Los comandos intercambiados<sup>4</sup> entre el lector y la tarjeta tienen el formato que a continuación se muestra en la Figura 4.5:

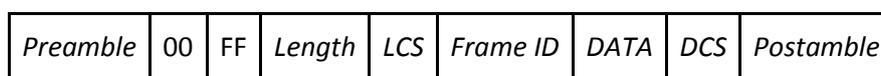


Figura 4.5 Trama PN532

Donde cada campo se indica a continuación brevemente:

- Preamble o preámbulo: 0x00
- Starcode o código de inicio de trama: 0x00 0xFF
- Length o longitud.
- Control de errores: LCS toma un valor de manera que la suma entre LCS y la longitud sea cero.
- Frame ID o identificador de trama:

<sup>4</sup> Para obtener más información sobre las tramas se aconseja acudir al manual de usuario del chip PN532 [71].

- 0xD4 Si se trata de una trama que va desde el host a la placa PN532.
- 0xD5 Si se trata de una trama que va desde la placa PN532 al host.
- Data: Data[0] ... Data[length-1]
- DCS es el byte más a la derecha de la suma de Frameld + DATA + DCS
- Postamble: 0x00

Volviendo a la funcionalidad del lector en el presente trabajo, ésta simplemente consiste en identificar al usuario. Para ello se utilizaron tres comandos. Dos son de configuración: *SAM Configuration* (carga la configuración de la tarjeta) y *RFCONFIGURATION* (evita que atasque el programa que contiene el uso del lector), por último, el tercero que permite la funcionalidad deseada: leer el identificador de la tarjeta.

Inicialmente se planteó el uso directo del lector de tarjetas implementando los comandos necesarios para la lectura de la tarjeta. Sin embargo, finalmente y con la intención de facilitar la implementación y tener una mayor generalidad para futuros desarrollos, se han empleado las librerías de Adafruit. Adafruit pone a servicio de los usuarios las librerías necesarias para poder manejar el lector según la interfaz deseada I<sup>2</sup>C [72], SPI [73] y, UART [74] así como un tutorial para su instalación [75].

## 4.4 NODO COORDINADOR

El nodo coordinador, como se explicó en el capítulo anterior, cuenta con dos interfaces. La radio para comunicarse con los nodos remotos de su red inalámbrica, y la interfaz Ethernet que no sólo permite el acceso de la información que la red inalámbrica ofrece sino que concede la opción de cambiar la configuración de la misma desde un equipo conectado a internet. Por ello en los siguientes apartados se hará una introducción a la configuración necesaria y una explicación detallada del funcionamiento del nodo maestro.

### 4.4.1 CONFIGURACIÓN HARDWARE

Interesa que el nodo central o coordinador sea capaz de recolectar datos, procesarlos, almacenarlos y ponerlos en disposición en internet. Para implementar estas características es necesario un escudo XBee y otro Ethernet. Por lo que el nodo central consistirá en una placa Arduino UNO, un escudo Ethernet y un escudo XBee.

Durante el montaje se ha de prestar cuidado a la manera de hacerlo, por lo que primero hay que montar el escudo Ethernet sobre la Arduino UNO debido a que permite la conexión de otros escudos u otros dispositivos al dejar libres los pines de la placa. Aunque esto sea trivial, lo que interesa y dónde se presenta el problema es cuando se conecta el escudo XBee sobre el escudo Ethernet, pues al igual que este último se alimenta de los 6 pines ICSP (In-Circuit Serial Programming) de la placa Arduino. Por lo que hay que hacer un puente que conecte los pines ICSP con el escudo XBee como muestra la *Figura 4.6*.

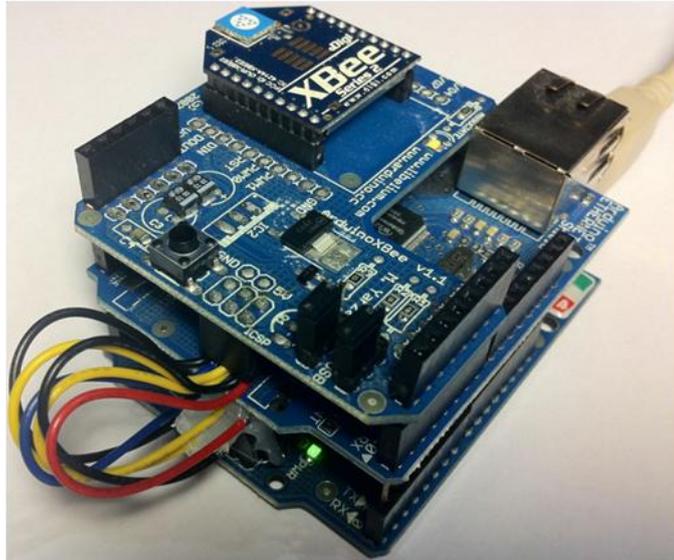


Figura 4.6 Acoplo de escudos XBee y Ethernet

Para poder implementar las funcionalidades de comunicación del nodo de control se han utilizado las librerías Ethernet.h [76] y SD.h [77]. El escudo Ethernet posee un slot para albergar una tarjeta de memoria micro SD, la cual permite el almacenamiento de datos. En concreto, los datos recogidos por la red de sensores y, el fichero HTML de la página web.

Antes de hablar de la configuración del sistema se ha de tener en cuenta una serie de consideraciones con respecto a los pines de conexión a la placa Arduino UNO:

- La placa utiliza los pines digitales 10, 11, 12, y 13 (pines SPI) para comunicarse con el chip W5100 del escudo Ethernet. Por lo que estos pines no pueden ser usados para E/S genéricas.
- Cuando se trabaja con la librería SD.h, el pin SS corresponde con el pin 4 de la placa Arduino, por lo que este pin también queda reservado.

Dado que el chip W5100 y la tarjeta micro SD comparten el bus SPI, no pueden funcionar a la vez. Por lo que si ambos son usados, si no se utiliza uno de los dos, es aconsejable deseleccionarlo. Es decir que para la tarjeta SD, el pin 4 debe ser definido como salida a nivel alto y, en el caso del chip W5100 el pin 10 también como salida a nivel alto.

Una vez se tiene en cuenta que ciertos pines han de respetarse para el correcto funcionamiento del escudo se verá cómo se consigue que el sistema esté conectado a internet.

#### 4.4.2 FUNCIONAMIENTO EN EL COORDINADOR

El escudo puede funcionar tanto como servidor capaz de aceptar conexiones entrantes, o como cliente permitiendo realizar conexiones de salida. Para el caso que nos ocupa se implementará la funcionalidad en modo servidor. Así pues el coordinador es iniciado como servidor, asignándole una dirección MAC y, una dirección IP (teniendo en cuenta no utilizar alguna ya disponible en la red). De esta forma el nodo escucha por el puerto 80 peticiones de sus clientes, pudiendo haber hasta seis conexiones diferentes.

El funcionamiento del nodo coordinador puede resumirse en el diagrama de la [Figura 4.7](#). Señalar que los datos de cada nodo sensor son almacenados en local haciendo uso de una tarjeta micro SD. En dicha tarjeta se alojan ficheros con la información adquirida de cada sensor, que será leída y remitida a través de Internet según las peticiones realizadas.

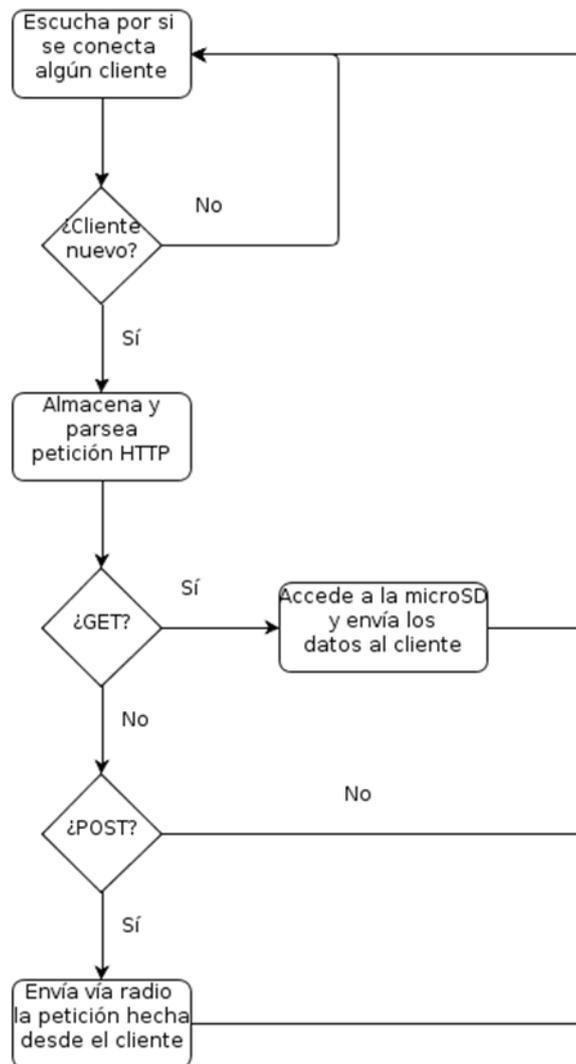


Figura 4.7 Diagrama de flujo

El coordinador en cada ciclo escucha la interfaz radio, la interfaz Ethernet y chequea por si hay alguna petición pendiente de atender. Pero en particular con respecto a la interfaz Ethernet, cuando un cliente inicia un navegador e introduce la dirección IP del servidor se establece una comunicación entre ambos. Como se puede observar en el diagrama de flujo, si

un cliente se conecta al servidor éste último almacena la trama HTTP que recibe del usuario en una cadena de caracteres. De todas las tramas que puedan llegar solo analiza las siguientes, pues son las direcciones o URLs que se han habilitado en el servidor:

- GET / HTTP/1.1
- GET /fioList HTTP/1.1
- POST /changes HTTP/1.1

Para el resto de los casos se devuelve un mensaje de error HTTP 404 Not Found error señalando que el recurso buscado no está disponible en el servidor. Las facilidades que ofrece la librería Ethernet no incluyen la implementación del servidor, sino que este debe implementarse a bajo nivel analizando los diferentes mensajes HTTP que se remiten.

Siguiendo la filosofía REST, el método GET se emplea para las situaciones en las que el cliente requiere información del servidor, mientras que con POST el cliente envía información al servidor. Los métodos PUT y DELETE no se han implementado pues los recursos existentes no se pueden modificar ni borrar. Al introducir la dirección del servidor en el navegador se está enviando una petición GET que el nodo coordinador identifica como la solicitud de la página web en la raíz del servidor, por lo que accede a la tarjeta de memoria micro SD y envía el código de la página requerida. Dicho código incluye tanto elementos HTML como código Javascript. A medida que el navegador interpreta el código, una de las funciones implementadas pide al servidor la lista de los nodos presentes en la red con sus respectivos valores, para completar la representación con esos datos. De nuevo el servidor accede a la tarjeta de memoria y lanza los datos que tiene disponibles en el formato ligero JSON como se muestra a continuación:

```
{ "fioList": [  
  { "name": "fio1", "address": "0x01", "temperature": "24", "humidity": "85", "Asynchronous": "0",  
    "sleepCycle": "20", "uid": "0000", "state": "0"  
  },  
  .  
  .  
  .  
  { "name": "fio6", "address": "0x06", "temperature": "24", "humidity": "85", "Asynchronous": "0",  
    "sleepCycle": "20", "uid": "0000", "state": "2"  
  }  
]  
}
```

Como se puede observar se trata de una lista que está formada por varias estructuras, cada estructura corresponde a un nodo con los parámetros, nombre, dirección, temperatura, humedad, Asynchronous, sleepCycle y uid. Una vez se reciben estos datos en el navegador los representa como se muestra en la Figura 4.8:

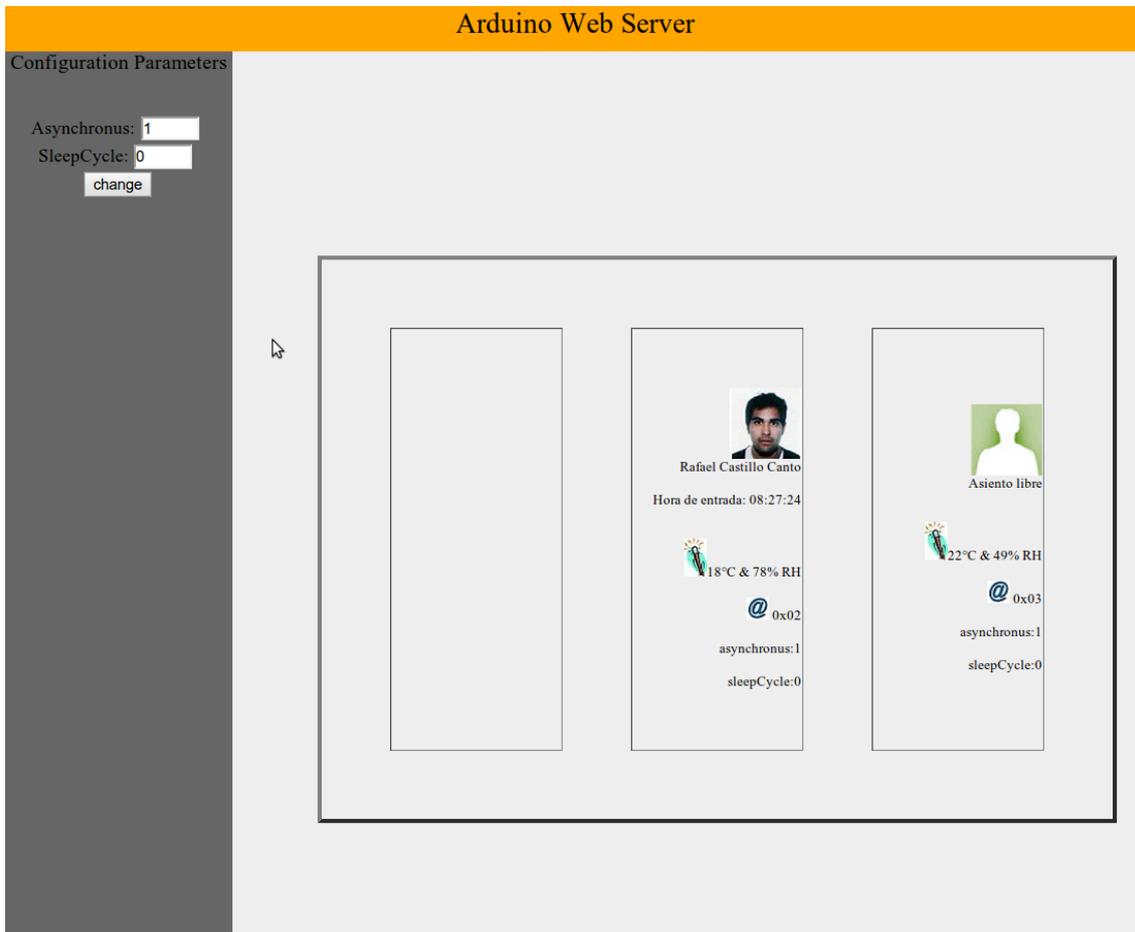


Figura 4.8 Página web

Se han incluido dos campos que permiten cambiar el valor de las variables ASINCRONUS y SLEEP\_CYCLE. Al hacerlo se enviará una petición al nodo coordinador que recibe una trama HTTP con la función POST que incluye los parámetros y los valores que deben de tomar. Se ha implementado que cada minuto se haga una petición de la lista de nodos con sus valores por si hubiera habido algún cambio. Esta filosofía sigue la técnica AJAX introducida anteriormente, es decir, desde el navegador se pide actualizar los valores de los sensores pero sólo los valores, nada más, reduciendo el volumen de datos transmitidos.

Con respecto a la página web, y como ya se ha comentado antes, está programada en HTML. Pero además se utilizan funciones incluidas en la librería JQuery y, en la representación visual de la página CSS. Dado el tamaño de dicha librería, se ha referenciado a un servidor externo de forma que no se sobrecargue la placa Arduino coordinadora con la descarga del fichero.

Ahora, teniendo una visión global del sistema se puede extraer dos casos de uso que clarifican su funcionamiento:

- Si un nodo sensor detecta un cambio en temperatura o, en la humedad o, bien el lector de tarjetas reconoce a un usuario, entonces envía una trama al coordinador con los valores que acaba de medir. El nodo maestro lee esa trama, extrae su información y, crea un fichero JSON (como el mostrado anteriormente) que es almacenado en la tarjeta de memoria.
- Si desde la página web, se cambia los valores de los parámetros *Asynchronous* y/o *sleep\_cycle*, una trama http que contiene un POST con esos valores es enviada al servidor. Éste almacena la petición en una cadena de caracteres y analiza su contenido, al tratarse de un POST crea una trama (en el capítulo anterior se llamó trama de configuración) con los valores que se desean cambiar y, es enviada. En el caso de que no sea respondida, se almacenará en memoria para luego ser reenviada.

En la tarjeta de memoria se crean tantos archivos como nodos haya, pero en lugar de enviarlos uno a uno, se leen todos y se envían en un único mensaje como cómo un sólo archivo. Desde el punto de vista de la red inalámbrica, si se envían datos desde un determinado nodo, en el coordinador se destruye el archivo anterior (si es que existía) y se crea uno nuevo con su mismo nombre. El nombre del archivo es exactamente el mismo que la dirección del nodo con una F delante, es decir por ejemplo, al nodo con dirección 0x04 le correspondería el nombre archivo F4.JS. Como dato informativo, el formato de nombre de archivos que soporta la tarjeta micro SD es el 8.3, es esa otra razón adicional por la que se eligió el direccionamiento de 16bits, pues el de 6 excedería tal formato. Para evitar que desde la página web colapse al coordinador, por ejemplo presionando repetidas veces el botón de enviar cambios en los parámetros de *Asynchronous* y *Sleep\_Cycle*, se ha programado de manera que solo se puede presionar una vez antes del siguiente refresco de esta forma se evita que se envíen tramas de manera abrupta lo que en consecuencia haría que el nodo coordinador no pueda continuar escuchando a través de su interfaz radio. Este último comentario da pie a introducir la idea de que no existe simultaneidad en las acciones, no se hace uso de threads o hilos, sino que las operaciones se ejecutan una tras otra. Lo que ocurre y como ya se explicó, es que la placa Arduino se encuentra en un ciclo infinito de la llamada función *loop*.

Destacar que se ha hecho uso del servidor Apache2 como servidor externo para acceder a las imágenes y las fotos de los usuarios, pues la placa Arduino tendría que tratar con demasiada información lo que podría ralentizar su funcionamiento. Además, de adicionarle la posibilidad de interpretar código php, con la idea de acceder a una base de datos sencilla MySQL. De manera que, el parámetro UID que se puede observar en la estructura JSON sirve para identificar el nombre y la foto a quién corresponde ese UID, y con el parámetro *state* se puede llevar un control de las horas de llegada y salida de un usuario.

### 4.4.3 FUNCIONAMIENTO DEL NODO REMOTO

El modo de actuar de los nodos remotos cambia ligeramente con respecto a lo explicado en el capítulo anterior. A lo largo del presente capítulo se ha hablado del lector de tarjetas, pero no del papel que juega en el sistema y como varía el comportamiento de los nodos.

Al incluir el lector, hay que cambiar el modo de funcionamiento por uno en el que el nodo no duerma. La idea de incluir este sistema en un aula permite evitar la preocupación de las baterías por lo que se trabajaría en el modo 2, en el que permanece despierto escuchando por peticiones desde el coordinador y enviando datos si es que hubiera algún suceso que hacer notificar. Antes, sólo se contaba con el sensor de temperatura/humedad y sus cambios permitían mantener al nodo en bajo consumo si así se deseaba. En cambio la presencia de una persona no permite a primera vista dormir al nodo, sino que debe permanecer despierto por si se diera el caso de que alguien se identificara en la red.

Adicionar el nuevo dispositivo lo único que implica es la lectura del mismo en cada ciclo de programa. Pero ¿qué se lee? La respuesta es el UID o identificador de tarjeta. Cada vez que es posada sobre el lector se compara con una variable que almacena el valor anterior. Tanto la primera vez como con una tarjeta distinta a la predecesora, significaría que hay una nueva presencia y por ello una trama de información será enviada con los nuevos datos al coordinador, incluyendo los de temperatura y humedad. En la siguiente repetición del programa, se volverá a leer de la tarjeta, si los valores actual y almacenado coincidieran significaría que es la misma persona y, si no hubo cambios de temperatura ni de humedad no se tendría porque avisar al coordinador. En cambio si se retirara la tarjeta, el lector no leería nada por lo que se notificaría al nodo central que la persona ha abandonado el puesto en el que estaba. Por ello se diferencian una serie de estados: llegada, ausencia temporal, regreso, expiración del tiempo permitido fuera del puesto,... por ello se ha añadido una variable que indica estos estados cuyo nombre es state. De esta forma conociendo el estado se puede iniciar la lectura del tiempo y, saber cuánto tiempo lleva siendo utilizado el recurso.

El funcionamiento es exactamente el mismo que en el apartado 3.1.2 del capítulo anterior, con la excepción de que además de leer la temperatura y la humedad, ahora se incluye la lectura de la tarjeta cuyo parámetro leído es el UID que se incluye en la trama de notificación.

# CONCLUSIONES Y LÍNEAS FUTURAS

# 5

Para finalizar con esta memoria, se resumirán las principales conclusiones extraídas tras la realización del trabajo, planteando además unas posibles líneas de actuación futuras, que surgen a partir de la información que este Proyecto de Fin de Carrera ha aportado.

Por una parte, se ha hecho un análisis de las diversas soluciones de comunicación en redes de sensores eligiendo al protocolo IEEE 802.15.4 como una de las que se ajusta más a las necesidades inicialmente planteadas: bajo tráfico de datos, funcionamiento en rango de frecuencias ISM, relación del tráfico con el baudrate.

Además se ha estudiado el hardware que lo puede soportar, decantándose por la plataforma Arduino, que aunque no sea la mejor opción, se trata de una solución hardware y software libre, caracterizada por su escalabilidad e interoperabilidad, además de estar respaldada tanto por una gran cantidad de bibliografía como una comunidad amplia en internet que está en continuo desarrollo. A todo esto hay que unir que la sencillez de uso permite establecer una curva de aprendizaje rápida.

Una vez conocida la plataforma y la tecnología de comunicaciones se ha experimentado con ella, para finalmente construir un protocolo propio adecuado a la aplicación que se buscaba: comunicar una serie de nodos, caracterizados por un comportamiento independiente e inteligente, con otro nodo central llamado coordinador que gestionará la información vertida por ellos.

A continuación llegó la parte de dar a la red inalámbrica una interfaz de comunicación con Internet. Esto se ha conseguido a través de un complemento o escudo apilable en la placa Arduino UNO del nodo central. En dicho nodo se ha implementado un servidor web de baja capacidad que permite por un bajo coste disponer de la información obtenida dentro de la red de sensores. Se han observado las limitaciones de dicho servidor, donde la imposibilidad de Arduino para ejecutar hilos, hace que no se puedan atender múltiples peticiones en paralelo. Adicionalmente la reducida capacidad de proceso de la placa hace que los datos que se procesen deban ser lo más sencillo posible. Para conseguir ese objetivo, se ha hecho uso de conceptos como la técnica AJAX, el formato ligero de intercambio de datos JSON, delegando parte del procesado en el nodo final, que tendrá una mayor capacidad de cómputo. En este sentido, tiene interés analizar nuevas alternativas que doten de mayor capacidad de proceso

concurrente al hardware (i.e. *protothreads*) o analizar nuevas arquitecturas para reorganizar la manera de acceder a la información desde el exterior. En una primera aproximación durante el proyecto, toda la información estática y de gran volumen se ha obtenido de servidores más potentes mediante servidor Apache y programación dinámica y una base de datos sencilla como MySQL. De esta manera se completa el funcionamiento y las capacidades del sistema.

La utilización del servidor Apache da al sistema más dinamismo permitiendo almacenar ya no sólo la página web sino funciones, fotos, e información que agiliza el funcionamiento de la placa Arduino (al quitarle un peso de ejecución a un sistema de velocidad de procesamiento y almacenaje mucho menor). Junto con ello y la base de datos MySQL, podría aprovecharse para contabilizar un registro de actividad del aula, durante diferentes periodos, es decir, un historial que permita obtener una estadística de comportamiento de los usuarios. Conocer a usuarios de malos hábitos y poder sancionarlos, afluencia del aula, entre otros.

La forma de acceder a la información de identificación del usuario se ha habilitado empleando la tecnología de tarjetas inteligentes sin contactos Mifare. En este sentido se ha trabajado con la propia tarjeta de la Universidad de Cantabria, validando la operativa de la solución planteada en este entorno.

En cuanto a la gestión de la red de sensores se ha planteado una solución para el descubrimiento, configuración y control de la red. Se ha diseñado un protocolo sobre el que se ha sustentado toda la plataforma. Adicionalmente, destacar que se ha definido un procedimiento para dormir los nodos de la red y establecer un modo de bajo consumo de estos. No obstante se han observado problemas a la hora de mantener la sincronización entre todos los nodos de la red. Aunque la solución planteada opera sobre nodos IEEE 802.15.4 nativos, hay que reseñar que existen soluciones propietarias en el mercado que solventan en cierta medida este problema. Es por esto que aunque la implantación de estos nuevos protocolos pudiera suponer una pérdida de versatilidad, pueda ser recomendable su uso en determinadas circunstancias.

En la aplicación real del sistema en un aula, no todos los nodos tienen porqué tener sensores, podría elegirse una serie de ellos colocados en puntos físicos estratégicos para obtener valores representativos de algunos de los parámetros que pueden medirse.

Durante el trabajo se ha considerado la funcionalidad de los nodos como actuadores para controlar dispositivos externos (luces, etc.). Se ha validado esta solución conectando leds gestionados en función de la presencia de usuario. Restaría implementar adaptaciones para aplicar el resultado del procesado de la información capturada (temperatura, humedad, presencia, etc.) al control ambiental de las salas (aire acondicionado, calefacción, etc.).

## REFERENCIAS

- [1] Sistema SOSUS, <http://es.wikipedia.org/wiki/SOSUS>
- [2] Estándar 802.15.1, Bluetooth, <http://www.ieee802.org/15/pub/TG1.html>
- [3] Estándar 802.11, Wi-Fi, <http://ieee802.org/11/>
- [4] Estándar 802.15.3, <http://www.ieee802.org/15/pub/TG3a.html>
- [5] Ultra-Band Wide: ECMA Std. 368: "High Rate Ultra Wideband PHY and Standard", 3<sup>rd</sup> Edition, December 2008
- [6] Página oficial de Z-Wave Alliance, <http://www.z-wavealliance.org>
- [7] Ultra Low Power Bluetooth o Wibree, <http://www.eetimes.com/design/microwave-rf-design/4012680/Introduction-to-Bluetooth-ULP>, <http://vic-en-6to.blogspot.com.es/2012/05/dispositivos-moviles-wibree-ulp.html>
- [8] Estándar 802.15.4, <http://www.ieee802.org/15/pub/TG4.html>
- [9] Página oficial de Zigbee Alliance, <http://www.zigbee.org/>
- [10] Página oficial protocolo HART, <http://www.hartcomm2.org/index.html>
- [11] Artículo interés sobre redes wireless HART, <http://rii.galeon.com/aficiones2095541.html>
- [12] Página oficial protocolo ISA, <http://www.isa.org>
- [13] Página oficial protocolo IETF IPv6, <http://ietfreport.isoc.org/ids-wg-6lowpan.html>
- [14] Página oficial de EnOcean, <http://www.enocean.com/home/>
- [15] Página oficial ANT, <http://www.thisisant.com/>
- [16] MiWi Wireless Networking Protocol Stack, David Flowers y Yifeng Yang Octubre 2011: [http://www.microchip.com/stellent/idcplgIdcService=SS\\_GET\\_PAGE&nodeId=1824&appName=en520606](http://www.microchip.com/stellent/idcplgIdcService=SS_GET_PAGE&nodeId=1824&appName=en520606)
- [17] Digi International, <http://www.digi.com/>
- [18] X-CTU, <http://www.digi.com/support/kbase/kbaseresultdet?id=2125>
- [19] Firmtech, [http://www.firmtech.co.kr/01pro/main\\_eng.php?index=100&proinfo=103](http://www.firmtech.co.kr/01pro/main_eng.php?index=100&proinfo=103)
- [20] Sena Technologies, [http://www.sena.com/products/industrial\\_zigbee/index.php?tab\\_menu=OEM](http://www.sena.com/products/industrial_zigbee/index.php?tab_menu=OEM)

- [21]LS Research: <http://es.mouser.com/lresearch-prosiFLEX/>
- [22]Crossbow, <http://www.xbow.com/>
- [23]Tutorial uso de motas crossbow,  
<http://www.pages.drexel.edu/~kws23/tutorials/motes/motes.html>
- [24]Lenguaje de programación NESC, <http://standards.ieee.org/about/nesc/index.html>
- [25]Compilador CYGWIN, <http://www.cygwin.com/>
- [26]Shockfish, plataforma Tinynode, <http://www.tinynode.com/>
- [27]TelosB, <http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=252>
- [28]MicaZ, <http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=164>
- [29]Waspmote, <http://www.libelium.com/products/waspmote>
- [30]Módulos iSense, <http://www.coalesenses.com/index.php?page=isense-hardware>
- [31]Coalesenses, <http://www.coalesenses.com/>
- [32]Placas Arduino, <http://arduino.cc/es/Main/Hardware>
- [33]Lenguaje Arduino, <http://arduino.cc/es/Reference/HomePage>
- [34]Lenguaje Wiring, <http://wiring.org.co/>
- [35]Processing, <http://www.processing.org/>
- [36]Placas similares a Arduino, <http://arduino.cc/playground/Main/SimilarBoards>
- [37]Página oficial proyecto pinguino,  
[http://www.hackinglab.org/pinguino/index\\_pinguino.html](http://www.hackinglab.org/pinguino/index_pinguino.html)
- [38]Tutorial de uso, <https://sites.google.com/site/pinguinotutorial/home>
- [39]Lenguaje Python, <http://www.python.org/>
- [40]Página oficial proyecto Raspberry, <http://www.raspberrypi.org/>
- [41]Picaxe, <http://www.picaxe.com>
- [42]Página web de la empresa Libelium, <http://www.libelium.com/>
- [43]Listado de placas oficiales de Arduino, <http://arduino.cc/en/Main/Hardware>
- [44]Chip Wiznet w5100,  
[http://www.wiznet.co.kr/Sub\\_Modules/en/product/Product\\_Detail.asp?cate1=&cate2=&cate3=&pid=1011](http://www.wiznet.co.kr/Sub_Modules/en/product/Product_Detail.asp?cate1=&cate2=&cate3=&pid=1011)

- [45]Libería Ethernet, <http://arduino.cc/es/Reference/Ethernet>
- [46]Librería SD, <http://arduino.cc/en/Reference/SD>
- [47]SquidBee, [http://www.libelium.com/squidbee/index.php?title=Main\\_Page](http://www.libelium.com/squidbee/index.php?title=Main_Page)
- [48]Módulos XBee Serie 1, <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/XBee-series1-module#overview>
- [49]Módulos XBee Serie 2, <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/point-multipoint-rfmodules/XBee-series1-module#docs>
- [50]O'Reilly, Making Things Talk 2nd Edition Sep. 2011 by Tom Igoe
- [51]O'Reilly, Getting Started with the Internet of Things 1st Edition May 2011 by Cuno Pfister
- [52]Product Manual v1.xEx - 802.15.4 Protocol, API Operation, [http://ftp1.digi.com/support/documentation/90000982\\_G.pdf](http://ftp1.digi.com/support/documentation/90000982_G.pdf)
- [53]Software X-CTU, <http://www.digi.com/support/productdetail?pid=3352>
- [54]Página fabricante XBee Xplorer USB, <http://www.sparkfun.com/products/8687>
- [55]Librería Enerlib, <http://arduino.cc/playground/Code/Enerlib>
- [56]Librería Narcoleptic, <http://code.google.com/p/narcoleptic/>
- [57]Librería XBee, <http://code.google.com/p/xbee-arduino/>
- [58]Librería NewSoftSerial, <http://arduiniana.org/libraries/newsoftserial/>
- [59]Sensor de humedad SHT15 de Sparkfun, <http://www.sparkfun.com/products/8227>
- [60]Tarjetas Mifare, [http://www.nxp.com/documents/data\\_sheet/MF1S503x.pdf](http://www.nxp.com/documents/data_sheet/MF1S503x.pdf)
- [61]Estándar ISO/IEC 14443 Tipo A, [http://www.iso.org/iso/home/search.htm?qt=ISO%2FIEC+14443&sort\\_by=rel&type=simple&published=on](http://www.iso.org/iso/home/search.htm?qt=ISO%2FIEC+14443&sort_by=rel&type=simple&published=on)
- [62]Estándar ISO 9798-2, [http://webstore.iec.ch/preview/info\\_isoiec9798-2%7Bed3.0%7Den.pdf](http://webstore.iec.ch/preview/info_isoiec9798-2%7Bed3.0%7Den.pdf)
- [63]Tecnología NFC,
- [64]Estándar ISO 18092, [http://www.iso.org/iso/catalogue\\_detail.htm?csnumber=38578](http://www.iso.org/iso/catalogue_detail.htm?csnumber=38578)
- [65]Estándar EC-MA 340, <http://www.ecma-international.org/publications/standards/Ecma-340.htm>

- [66]Estándar ETSI TS 102 190,  
[http://www.etsi.org/deliver/etsi\\_ts/102100\\_102199/102190/01.01.01\\_60/ts\\_102190v010101p.pdf](http://www.etsi.org/deliver/etsi_ts/102100_102199/102190/01.01.01_60/ts_102190v010101p.pdf)
- [67]REST, [http://es.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://es.wikipedia.org/wiki/Representational_State_Transfer)
- [68]DOM: Document Object Model, <http://www.w3.org/DOM/>
- [69]Json, <http://www.json.org/>
- [70]Página oficial de Adafruit Industries, <http://adafruit.com/>
- [71]Datasheet lector de tarjetas, <http://www.microbuilder.eu/Projects/PN532.aspx>
- [72]Librería I2C, [https://github.com/adafruit/Adafruit\\_NFCShield\\_I2C](https://github.com/adafruit/Adafruit_NFCShield_I2C)
- [73]Librería SPI, <https://github.com/adafruit/Adafruit-PN532>
- [74]Librería UART, <http://code.google.com/p/libnfc/downloads/list>
- [75]Tutorial lector de tarjetas, <http://www.ladyada.net/products/rfidnfc/mifare.html>
- [76]Librería Ethernet, <http://arduino.cc/en/Reference/Ethernet>
- [77]Librería SD, <http://arduino.cc/en/Reference/SD>
- [78]Librería Xbee, <http://code.google.com/p/xbee-arduino/>
- [79]Manual de usuario lector de tarjetas, <http://www.adafruit.com/products/789>
- [80]Tutorial HTML, CSS, <http://www.w3schools.com/html/default.asp>
- [81]Tutorial JQuery, <http://www.genbetadev.com/frameworks/30-dias-para-aprender-jquery>
- [82]Tutorial JQuery, AJAX, <http://jquery.com/>

## ACRÓNIMOS

ACK	Acknowledgement
AES	Advanced Encryption Standard
AJAX	Asynchronous JavaScript And XML
AP	Access Point
API	Application Program Interface
ARM	Advanced RISC Machine
BSS	Basic Service Set
CCA	Clear Channel Assessment
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CSS	Cascading Style Sheets
DOM	Document Object Model
DSSS	Direct Sequence Spread Spectrum
E <sup>2</sup> PROM	Electrically Erasable Programmable Read-Only Memory
FCS	Frame Check Sequence
FFD	Full Function Device
FHSS	Frequency Hop Spread Spectrum
FIO	Funnel I/O
FTDI	Future Technology Devices International
GFSK	Gaussian Frequency Shift Key
GPRS	General Packet Radio Service
GPS	Global Positioning System
HART	Highway Addressable Remote Transducer Protocol

HTTP	Hypertext Transfer Protocol
I <sup>2</sup> C	Inter-Integrated Circuit
IPv6-LoWPAN	IPv6 over Low power Wireless Personal Area Networks
IDE	Integrated Development Environment
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISA	Industrial Society of Automation
ISM	Industrial Scientific & Medical
JSON	JavaScript Object Notation
LAN	Local Area Network
LLC	Logical Link Control
LQI	Link Quality Indicator
LR-WPAN	Low Rate Wireless Personal Area Network
MAC	Media Access Control
MCPS	MAC Common Part Layer
MIME	Multipurpose Internet Mail Extensions
MLME	MAC Layer Management Entity
NFC	Near Field Communication
NIC	Network Interface Card
OSI	Open System Interconnection
PAN	Personal Area Network
PIC	Peripheral Interface Controller
PPDU	PHY Protocol Data Unit
PSK	Phase-Shift Keying
PWM	Pulse-Width Modulation
REST	Representation State Transfer

RFD	Reduced Function Device
RFU	Reserved for Future Use
RIA	Rich Internet Applications
RISC	Reduced Instruction Set Computing
RPC	Remote Procedure Calls
SBC	Single-Board Computer
SOA	Service-Oriented Architecture
SPI	Serial Peripheral Interface
TCP	Transport Control Protocol
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram Protocol
ULP	Ultra Low Power
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UWB	Ultra-Wide Band
WLAN	Wireless Local Area Network
WSAN	Wireless Sensor & Actuator Networks
WSN	Wireless Sensor Network
XML	eXtensible Markup Language