



Universidad Politécnica de Madrid
Escuela Técnica Superior De Ingenieros Industriales
Grado de Ingeniería en Tecnologías Industriales

**Integración de Smart Wearables y Redes de
Sensores Inalambricas para aplicaciones
Colaborativas**

Trabajo Fin de Grado

Jaime Zornoza Sánchez-Lozano

2017



Escuela Técnica Superior De Ingenieros Industriales
Grado de Ingeniería en Tecnologías Industriales
Universidad Politécnica de Madrid

**Integración de Smart Wearables y Redes de
Sensores Inalambricas para aplicaciones
Colaborativas**

Trabajo Fin de Grado

Autor:
Jaime Zornoza Sánchez-Lozano

Tutores:
Gabriel Mujica
Jorge Portilla
2017

Agradecimientos

Este trabajo está decidido principalmente a las tres personas que más me han ayudado durante estos duros años de carrera: En primer lugar a mi abuela, con sus constantes rezos en mis periodos de exámenes. En segundo lugar a mi madre, que posiblemente haya sufrido mas lo que es la Ingeniería Industrial que yo mismo, y por último a mi pareja por aguantarme, animarme y comprenderme.

Me gustaría darle las gracias también a mi padre, cuya opinión sobre los agradecimientos en este tipo de documentos es que no deberían existir. Las expectativas que él tiene sobre mi, y la presión indirecta que estas crean, posiblemente hayan sido una de las claves del éxito de mi paso por la ETSII.

A mis compañeros con los que he compartido días enteros de biblioteca, litros y litros de café, alegrías y algún que otro palo.

Por último a mis tutores Jorge Portilla y Gabriel Mujica por guiarme durante este año e introducirme en el apasionante y prometedor campo de las *Redes de sensores inalámbricas*.

Resumen

Con la realización de este proyecto se busca estudiar las posibilidades de la integración de un dispositivo inteligente y además *wearable* dentro de una red de sensores inalámbrica, así como el desarrollo de una serie de aplicaciones que hagan uso de la arquitectura resultante de dicha integración y resalten su potencial.

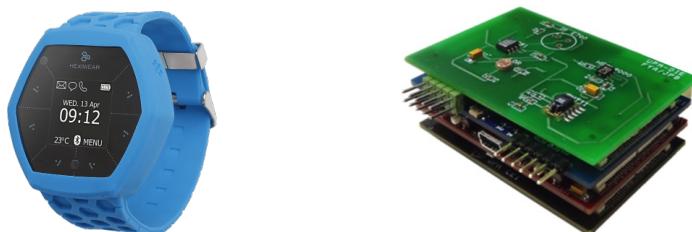
Al igual que hace varias décadas el nacimiento de *internet* supuso una revolución que transformó la infraestructura global de las comunicaciones, la revolución actual del ***Internet de las Cosas*** está transformando la forma de interactuar con el entorno físico que nos rodea y con los distintos agentes que participan en las redes de comunicaciones. Objetos hasta ahora cotidianos, están adquiriendo capacidades de recolección de datos y transmisión de información, dando lugar a una serie de aplicaciones y servicios sin precedentes. Dentro del ecosistema del *Internet de las Cosas*, las **Redes de Sensores Inalámbricas** son una de las tecnologías más prometedoras, debido a su ubicuidad, bajo coste y sencillez. Este tipo de redes se basa en el uso de dispositivos de consumo reducido y de bajo coste, que han de contar con una extensa vida útil, ya que en la mayoría de los casos una vez desplegados no están sujetos a ningún tipo de mantenimiento o sustitución. La misión de estos dispositivos es recopilar información de su entorno, y transmitirla a través de la red hasta una interfaz de salida que pone dicha información a disposición de los usuarios finales. Para que los nodos de este tipo de redes tengan la eficiencia energética mencionada previamente, en los últimos años se han desarrollado una serie de estándares de comunicación inalámbrica, diseñados específicamente para dar soporte a la comunicación entre dispositivos de estas características. Entre estos, el estándar mas utilizado hoy en día en las redes de sensores inalámbricas es el **IEEE 802.15.4**, que define las capas física (PHY) y de acceso al medio (MAC) para redes inalámbricas de área personal con bajas tasas de transmisión de datos, y en las cuales es necesaria una alta eficiencia energética.

En este trabajo, dicho estándar de comunicaciones se implementa en un dispositivo inteligente, de coste reducido, con capacidades propias de sensado, y una alta eficiencia energética, posibilitando su comunicación directa con los nodos de una red

de sensores inalámbrica. Estos nodos, a su vez, no son unos nodos corrientes para redes de sensores inalámbricas, sino una plataforma HW/SW modular y reconfigurable desarrollada por el Centro de Electrónica Industrial (CEI) de la Universidad Politécnica de Madrid, con el fin de contar con una elevada capacidad de control del nodo inalámbrico y una gran adaptabilidad ante cualquier entorno de aplicación. A estos nodos se los conoce como nodos **Cookie**.

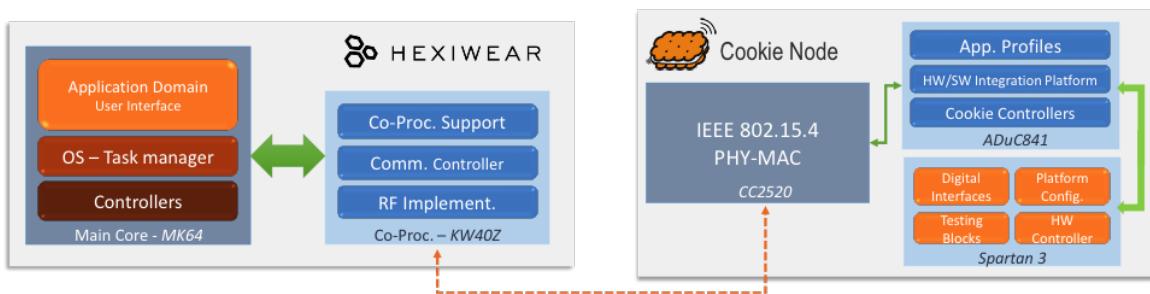
El dispositivo *wearable* inteligente es **Hexiwear**, desarrollado por *MikroElektronika* en asociación con NXP con el objetivo de dar soporte al prototipado de aplicaciones para el *Internet de las cosas*. Hexiwear es completamente *Open Source*, lo que significa que todos los esquemáticos de los circuitos que lleva dentro, así como de los códigos integrados en sus dos microcontroladores, están disponibles para ser descargados y modificados, pudiendo dotar al dispositivo de casi cualquier finalidad. El hardware de Hexiwear cuenta con dos microcontroladores (de los cuales uno se encarga exclusivamente de gestionar las comunicaciones inalámbricas y el otro de la gestión del resto de periféricos) una amplia gama de sensores, una pantalla OLED, un motor haptico de vibración y un LED RGB. En cuanto al *software*, Hexiwear incorpora en su microcontrolador principal el sistema operativo en tiempo real *FreeRTOS*, que permite separar cada una de las tareas del dispositivo en hilos de procesamiento independientes, mejorando la reacción de este ante eventos en tiempo real y aumentando el aprovechamiento de sus recursos hardware.

A su vez, los nodos *Cookie* han sido diseñados con una estructura modular dividida en capas, cada una de las cuales satisface una de las principales necesidades de los nodos para las redes de sensores inalámbricas. Estas necesidades, que dan nombre a las respectivas capas son: procesamiento, alimentación, comunicaciones y sensado. La modularidad resultante de la división del nodo de esta forma permite el uso de las *Cookies* en una gran variedad de entornos de aplicación, ya que para cambiar de una aplicación en la que, por ejemplo, se mida el pH de una sustancia, a otra en la cual interese tomar medidas sobre la humedad de un cultivo determinado, solo es necesario cambiar la capa de sensado del nodo. De la misma forma, para pasar de un despliegue donde se este utilizando IEEE 802.15.4 a otro donde se requiera la utilización de *Wi-Fi* u otro protocolo, solo sería necesario intercambiar la capa de comunicaciones del nodo. En la siguiente figura se muestra una imagen de estos dos dispositivos: A la izquierda Hexiwear y a la derecha un nodo *Cookie*.



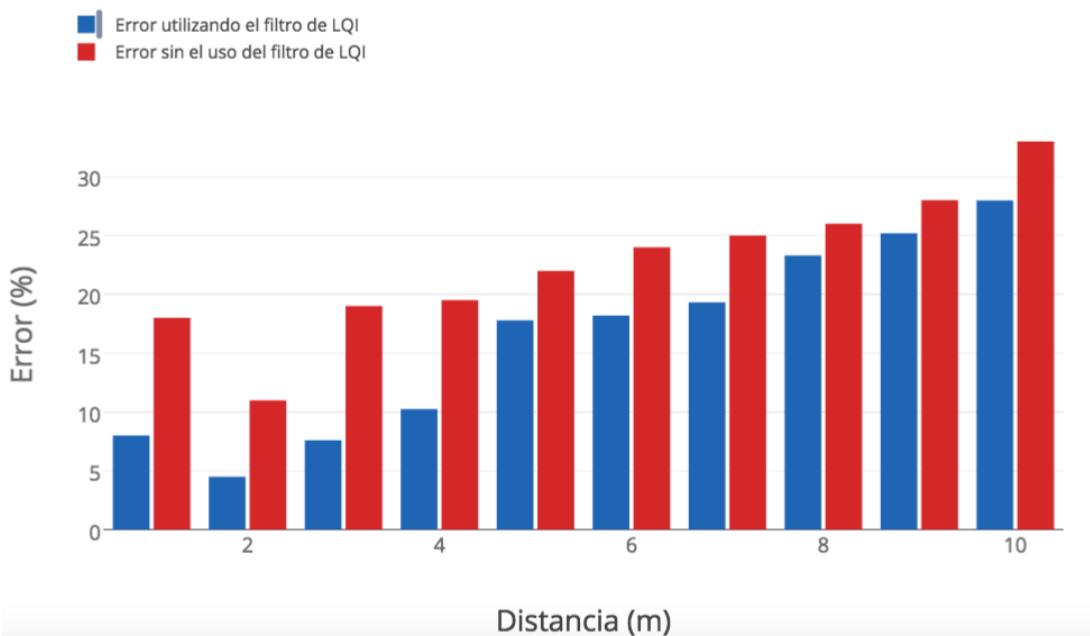
RESUMEN

Hexiwear con sus dos núcleos de procesamiento, su interfaz gráfica y su sistema operativo, y las *Cookies* con sus capacidades de adaptabilidad y su modularidad, se han combinado para formar una plataforma dinámica, configurable, y potente que pueda dar respaldo a una gran variedad de aplicaciones para las redes de sensores inalámbricas, y con la cual se busca la integración de este tipo de redes en el ecosistema del Internet de las Cosas. La arquitectura resultante de esta combinación da lugar a un sistema robusto y flexible, que abre un amplio abanico de nuevas aplicaciones para las redes de sensores inalámbricas, alejándose de las propuestas tradicionales de recolección de parámetros del entorno hacia aplicaciones más dinámicas y con mayor participación por parte de los usuarios. La siguiente figura muestra la arquitectura conseguida, resaltando sus distintos módulos: a la izquierda Hexiwear y sus dos núcleos de procesamiento, y a la derecha el nodo *Cookie* con su módulo de comunicaciones, su procesador y su FPGA.



Esta arquitectura se ha implementado en distintos entornos de aplicación y pruebas experimentales, con la intención de estudiar sus virtudes y conocer sus limitaciones. En primer lugar se buscó realizar la propuesta de un nuevo modelo o mecanismo que mejore la precisión en las estimaciones de distancia para las redes de sensores inalámbricas, donde la forma más común de calcular la separación entre dos nodos consiste en utilizar un parámetro denominado **Indicador de fuerza de la señal recibida** o **RSSI** (Received Signal Strength Indicator) que es un indicador de la potencia con la que se recibe una determinada señal inalámbrica. En la propuesta desarrollada se utiliza RSSI para estimar la distancia pero con la asistencia de otro parámetro: el **Indicador de Calidad de enlace** o **LQI** (Link Quality Indicator) que indica la calidad del enlace en las transmisiones de radiofrecuencia.

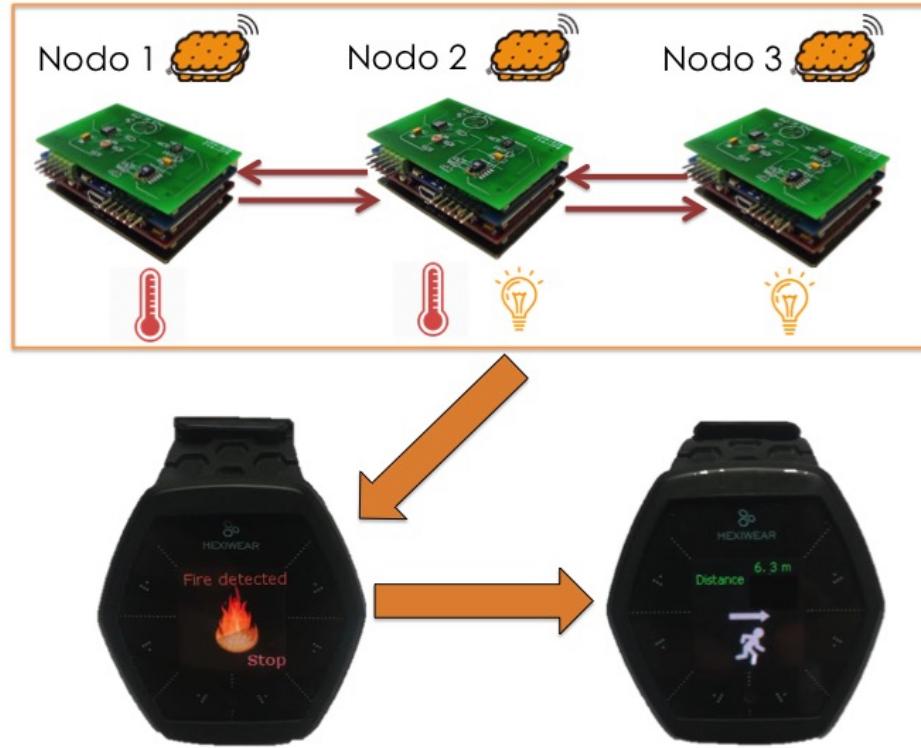
Dicha propuesta se basa en el uso de un filtro de LQI para restringir los paquetes que se utilizan en el proceso de estimación de la distancia. Como se puede apreciar en el gráfico que se muestra a continuación, el uso del filtro propuesto reduce el error en las estimaciones de distancia para este tipo de redes.



Los resultados obtenidos del uso de dicho filtro han sido utilizados para la redacción de un artículo científico de investigación para el congreso Internacional *XXXII IEEE Design of Circuits and Integrated Systems* (DCIS 2017), en el que también se caracteriza la arquitectura resultante de la integración de Hexiwear con los nodos *Cookies*. Dicho artículo se encuentra actualmente bajo revisión.

Posteriormente, la arquitectura propuesta se ha utilizado para el desarrollo de una serie de aplicaciones, con el objetivo de resaltar las ventajas de integrar un dispositivo inteligente como Hexiwear dentro de una red de sensores inalámbrica, y de poner dicha arquitectura a prueba. La primera aplicación, pretende también fomentar la cooperación entre los nodos de las redes de sensores inalámbricas, que en sus aplicaciones más tradicionales se limitan a enviar la información que recopilan a través de la red, sin conocer los datos recopilados por el resto de los nodos. A las redes en las que los nodos colaboran entre sí con un fin global, se las conoce como **Redes Colaborativas**, y suelen llevar asociadas una serie de ventajas, como mejoras en la eficiencia energética de la red, la posibilidad de desarrollo de aplicaciones de mayor complejidad, y mejoras en el tráfico de datos dentro de dicha red. La aplicación desarrollada tiene como fin la detección de eventos de riesgo, como puede ser un incendio, por parte de los nodos *Cookie*, y la notificación de la aparición de dicho evento al usuario del Hexiwear. Colaborando entre sí, nodos que individualmente no tienen la capacidad de detectar un evento así, debido a sus limitadas capacidades de sensado, pueden detectar eventos de distintas naturalezas, y comunicárselos al usuario encargado de monitorizar la red de forma rápida y eficiente, reduciendo el tiempo de respuesta ante dichos eventos. La siguiente imagen muestra un esquema del funcionamiento de la aplicación desarrollada.

RESUMEN



En esta aplicación 3 nodos *Cookies* comparten la información recopilada por sus sensores, y al detectar cualquiera de ellas que todas las medidas de dichos sensores están por encima de un cierto umbral, se envía un mensaje al Hexiwear, que produce una vibración y muestra una pantalla de alerta. Posteriormente, se calcula la distancia a un cuarto nodo *Cookie*, situado en una salida de emergencia, utilizando el filtro de LQI propuesto y un algoritmo de triangulación en caso de ser necesario. Este algoritmo de triangulación se ha implementado para reducir el error en la estimación de distancias lejanas al nodo de emergencia, y es otro ejemplo más de las ventajas que aporta la colaboración entre los nodos de una red de sensores inalámbrica.

Finalmente se ha implementado una aplicación basada en la provisión de distintos servicios por parte de los nodos de una red de sensores inalámbrica a un usuario que utiliza un dispositivo Hexiwear para monitorizar y gestionar dicha red. Según el usuario de Hexiwear se desplaza por el edificio en el que está desplegada la red, al entrar en las áreas de servicio de cada uno de los nodos, va recibiendo distintos tipos de información. Los objetivos de esta última aplicación son, por una parte, mostrar los servicios que pueden ofrecer las redes de sensores inalámbricas, más allá de la recopilación de parámetros físicos del entorno y, por otro lado, resaltar la forma en la que un dispositivo inteligente como Hexiwear puede recibir estos servicios de forma dinámica, descentralizada y sin los costes derivados de la utilización de equipos como ordenadores para la monitorización de la red desplegada.

Índice general

Dedicatoria	III
Resumen	V
Índice de figuras	XIII
Índice de tablas	XVII
1. Introducción	1
1.1. Antecedentes y justificación	1
1.2. Objetivos	3
1.3. Estructura del documento	4
2. Estado del Arte	7
2.1. Redes de sensores inalámbricas	7
2.2. La evolución de los <i>wearables</i>	10
2.3. Elementos inteligentes en redes de sensores inalámbricas	12
2.4. Introducción a la localización en redes de sensores inalámbricas	13
3. Plataformas utilizadas	19
3.1. Hexiwear	19
3.1.1. ¿Qué es Hexiwear?	19
3.1.2. Hardware	20
3.1.3. Software	22
3.1.4. IEEE 802.15.4	24
3.2. Plataforma Cookies	29

3.2.1. ¿Qué son las Cookies?	29
3.2.2. Las distintas capas	30
4. Arquitectura, verificación funcional y interoperabilidad	35
4.1. Arquitectura	35
4.2. Interoperabilidad y verificación funcional de las comunicaciones	42
4.2.1. Objetivos	42
4.2.2. Metodología y configuraciones	42
4.2.3. Resultados	45
5. Localización en redes de sensores inalámbricas	49
5.1. Formulación del problema	49
5.2. ¿Qué son el RSSI y el LQI?	50
5.2.1. Recieved Signal Strenght Indicator – RSSI	50
5.2.2. Link Quality Indicator – LQI	52
5.3. Correspondencia del RSSI con la distancia	53
5.4. Correspondencia del LQI con la distancia	56
5.5. Solución propuesta	58
5.5.1. El filtro de LQI	59
5.5.2. Resultados: Estimación de distancias utilizando el <i>Log Normal Shadowing Model</i>	60
6. Redes Colaborativas y Cooperación en Redes de Sensores Inalámbricas	65
6.1. Redes Colaborativas	65
6.2. Implementación de colaboración para la detección de incendios.	67
6.2.1. Metodología	69
6.3. Implementación de aplicación colaborativa para la detección de incendios con localización	75
7. Provisión de servicios en una red de sensores inalámbrica mediante el uso de un <i>smart wearable</i>	83
7.1. Implementación de la aplicación de provisión de servicios	83
8. Conclusiones, líneas futuras y gestión del proyecto.	95
8.1. Conclusiones generales	95
8.2. Consecución de objetivos	97
8.3. Líneas futuras de investigación	98
8.4. Gestión del proyecto	100
8.4.1. Planificación temporal	100
8.4.2. Presupuesto	104

Índice de figuras

2.1.	Gráfica del consumo asociado a distintas actividades en el funcionamiento de un nodo de una red de sensores inalámbrica	9
2.2.	Arquitectura general del <i>hardware</i> de un nodo para una red de sensores inalámbrica	10
2.3.	Reloj Calculadora de <i>Casio</i> - El predecesor de los <i>wearables</i> de hoy en día	11
2.4.	Esquema de un algoritmo de triangulación usando 3 nodos ancla y un nodo móvil.	15
2.5.	Gráfica cualitativa del consumo energético asociado a distintos algoritmos de localización	18
2.6.	Gráfica cualitativa que muestra el coste de implementación de distintos algoritmos de localización para redes de sensores inalámbricas . .	18
3.1.	Hexiwear	20
3.2.	Diagrama de la interconexión de los componentes del Hexiwear.	21
3.3.	Imagen del Hexiwear conectado a la <i>docking station</i> y de las <i>Click Boards</i>	22
3.4.	Esquema de la arquitectura por capas del IEEE 802.15.4 y el <i>Zigbee</i> .	25
3.5.	Distintas topologías que permiten las redes de IEEE 802.15.4: Estrella (Izq) y P2P (Dcha)	26
3.6.	Imagen de la estructura de la trama definida por el IEEE 802.15.4 . .	27
3.7.	Imagen de un nodo Cookie donde se pueden visualizar las cuatro capas principales acopladas	30
3.8.	Capa de procesamiento de un nodo Cookie	31
3.9.	Capas de comunicaciones que implementan Zigbee (arriba) y Bluetooth (abajo)	32
3.10.	Imagen de una capa de alimentación mediante conexión USB	33

3.11. Dos posibles capas de sensado para los nodos <i>Cookie</i> : Temperatura, humedad, y luz (arriba) y galga extensiometrica (abajo)	34
4.1. Código que implementa la estructura de los paquetes de que transmiten por UART entre los dos microcontroladores del Hexiwear.	36
4.2. Secuencia de pasos de la comunicación via UART de los dos uCs de Hexiwear	38
4.3. Esquema global de la arquitectura propuesta en el presente trabajo, compuesta por Hexiwear y un nodo <i>Cookie</i>	40
4.4. Imagen del Hexiwear y dos nodos <i>Cookies</i> , elementos que componen la arquitectura descrita en este capítulo.	41
4.5. Aplicación de linterna y pantalla principal del Hexiwear	43
4.6. Configuraciones de la PAN escogidas para la prueba de comunicación, mostradas a través de la pantalla del Hexiwear.	44
4.7. Pantallas <i>custom</i> de acceso a las funcionalidades del IEEE 802.15.4 (izq) y de comienzo de la red de área personal (dcha)	45
4.8. Instancia del terminal del PC al realizar la prueba de comunicación entre Hexiwear y las Cookies.	46
5.1. Gráficas que muestran distintas señales inalámbricas (rojo) y el ruido en su entorno (azul)	53
5.2. Esquema de las pruebas realizadas para la caracterización del RSSI. .	54
5.3. Gráficas que muestran el comportamiento del RSSI con la distancia para $0dbm$ (arriba) y $-15dBm$ (centro) para mensajes enviados de un nodo <i>Cookie</i> al Hexiwear. La gráfica inferior muestra la variación del RSSI con la distancia para mensajes enviados de Hexiwear a un nodo <i>Cookie</i>	55
5.4. Gráficas que muestran el comportamiento ideal del RSSI con la distancia (arriba) y la línea de tendencia para la variación de este parámetro con la distancia para los mensajes enviados de la <i>Cookie</i> al Hexiwear (abajo)	56
5.5. Gráficas que muestran la variación del LQI con la distancia de los mensajes enviados del Hexiwear a un nodo <i>Cookie</i> (arriba) y el comportamiento ideal de este parámetro con la distancia (abajo)	57
5.6. Gráfica que muestra el error obtenido en la estimación de distancias sin el uso del filtro de LQI (rojo) y con el (azul)	62
6.1. Esquema que muestra el despliegue de nodos <i>Cookie</i> con su identificador y las medidas que recopilan: T_i para temperaturas y L_i para luz ambiente	68
6.2. Captura de pantalla de los terminales de ordenador asociados a los nodos <i>Cookie</i> utilizados en la implementación anterior	70
6.3. Imagen de la pantalla de alerta del Hexiwear	71

Índice de figuras

6.4.	Captura de pantalla de los terminales de ordenador asociados a los nodos <i>Cookies</i> utilizados en la implementación de la prueba descrita en el Paso 3	72
6.5.	Flujograma del comportamiento de cualquiera de los tres nodos <i>Cookie</i> para la implementación de la aplicación colaborativa	74
6.6.	Pantallas de Hexiwear utilizadas en la aplicación de detección de riesgo con distancia.	75
6.7.	Despliegue realizado en el CEI para la estimación de distancias en la aplicación colaborativa de localización.	77
6.8.	Despliegue de la aplicación de detección de riesgos más localización utilizando un nodo de asistencia.	78
6.9.	Esquema del algoritmo de triangulación utilizado para asistir a Hexiwear en la estimación de la distancia al nodo de emergencia	79
7.1.	Pantalla de estación meteorológica de Hexiwear	84
7.2.	Imagen de los elementos que intervienen en el servicio de detección de presencia	85
7.3.	Pantallas de Hexiwear utilizadas en el servicio de detección de presencia.	86
7.4.	Despliegue realizado para la implementación del servicio de detección de presencia.	87
7.5.	Flujograma del funcionamiento de los dos nodos <i>Cookie</i> utilizados para el servicio de detección de presencia.	88
7.6.	Flujograma del funcionamiento de los dos nodos Hexiwear utilizados para el servicio de detección de presencia.	89
7.7.	Imagen de la distribución de los ejes de la capa de sensado que usa el acelerómetro y de una de las pantallas de Hexiwear implementadas para este servicio	90
7.8.	Gráfica que muestra la evolución de la aceleración medida por el nodo <i>Cookie</i> en el proceso de cierre de la puerta.	92
7.9.	Gráfica que muestra las oscilaciones en la aceleración medida por el nodo <i>Cookie</i> en el momento de cierre de la puerta.	92
8.1.	Diagrama de Gantt del presente proyecto.	103
8.2.	Presupuesto del presente proyecto	104

Índice de tablas

3.1.	Tabla comparativa de distintos aspectos de los protocolos de comunicación más comunes.	27
4.1.	Tabla que muestra la cantidad de paquetes recibidos en la comunicación de Hexiwear y un nodo <i>Cookie</i> al enviar 100 mensajes desde distintas distancias.	46
5.1.	Tabla que muestra los valores de RSSI y LQI de los paquetes anómalos con respecto a la media para distintas distancias.	59
5.2.	Tabla que muestra los valores de los parámetros utilizados en el <i>Log Normal Shadowing Model</i>	60
5.3.	Tabla que muestra las distancias estimadas por el <i>Log Normal Shadowing Model</i> junto con el error en la estimación para distancias de uno a diez metros sin el uso del filtro de LQI	61
5.4.	Tabla que muestra las distancias estimadas por el <i>Log Normal Shadowing Model</i> junto con el error en la estimación para distancias de uno a diez metros utilizando el filtro de LQI	61
5.5.	Tabla que muestra las distancias estimadas por el <i>Log Normal Shadowing Model</i> usando solo paquetes con un LQI de 235	63
6.1.	Tiempos de reacción de la aplicación colaborativa de detección de riesgos ante la detección del evento usando dos nodos <i>Cookie</i>	71
6.2.	Tiempos de reacción de la aplicación colaborativa de detección de riesgos ante la detección del evento usando tres nodos <i>Cookie</i>	73
6.3.	Distancias estimadas al nodo de emergencia usando el <i>Log normal shadowing model</i> con el filtro de LQI para una distancia de referencia de dos metros.	76

6.4. Distancias estimadas al nodo de emergencia usando el <i>Log normal shadowing model</i> con el filtro de LQI para una distancia de referencia de doce metros.	76
6.5. Tabla que muestra los resultados para las distintas medidas utilizadas en el algoritmo de triangulación.	80
8.1. Tabla que muestra el desglose de las distintas fases del proyecto . . .	104

Capítulo 1

Introducción

En este capítulo se pretende dar una visión global del *Internet de las Cosas* y de cómo las redes de sensores inalámbricas están llamadas a ser una tecnología fundamental para dicho ecosistema. A su vez, se plantearán los objetivos de este trabajo, y se desglosará la estructura del documento para facilitar la posterior lectura y comprensión del mismo.

1.1. Antecedentes y justificación

Hace varias décadas, el nacimiento del hoy imprescindible *internet* puso al alcance de todo aquel que quisiese acceder a él, un enorme universo de información digital, y fue el origen de una revolución que transformó el mundo de las comunicaciones. Ahora está empezando una nueva revolución, que tendrá un impacto similar o mayor al que tuvo el internet en su día: La revolución del *Internet de las Cosas*.

El Internet de las cosas o IoT (del inglés *Internet of things*) conecta el mundo físico en el que vivimos, y cualquier objeto dentro de él, con la infraestructura ya establecida del internet, creando así una forma completamente nueva de interactuar con la realidad. El concepto de IoT, nació en el Instituto Tecnológico de Massachusetts (MIT) y consiste en un cambio radical en las relaciones entre los objetos y las personas, o entre los propios objetos, que adquieren la capacidad de comunicarse entre sí y con la red, para ofrecer todo tipo de información. Cualquier objeto cotidiano, dotado del *hardware* necesario, podrá entrar en la red del internet de las cosas, adquiriendo la capacidad de recopilar información (mediante el primer elemento de este hardware: sensores) y de transmitirla de forma inalámbrica (mediante el segundo elemento necesario: un módulo de comunicaciones). De forma progresiva, en los próximos años, objetos hasta ahora cotidianos, se convertirán en dispositivos inteligentes con el fin de proporcionar una serie de servicios y aplicaciones sin prece-

1.1. Antecedentes y justificación

dentes. Entre estas aplicaciones están los *Smart Grids*, los Sistemas de Transporte Inteligentes o ITS, y las *Smart Homes*. Estas aplicaciones están respaldadas por sistemas embebidos inteligentes, denominados *Cyber Physical Systems* (CPS) o como se conocen en castellano *sistemas ciberfísicos*, que permiten monitorizar y gestionar el ambiente que les rodea. Los sistemas *ciberfísicos* integran capacidades de computación, almacenamiento, y comunicación junto con capacidades de seguimiento y/o control de objetos en el mundo físico, y están normalmente conectados entre sí y a su vez conectados con el mundo virtual de las redes digitales globales.

En este marco del Internet de las Cosas, las Redes de Sensores Inalámbricas o WSN (*Wireless Sensor Networks*) están llamadas a ser una de las principales tecnologías que dan soporte a la conexión transparente y sencilla de todo tipo de dispositivos. Estas redes se basan en el uso de dispositivos pequeños, sencillos, y de bajo coste que recopilan información de su entorno mediante el uso de sensores, para posteriormente transmitirla de forma inalámbrica hasta uno o varios nodos especiales, denominados *gateways*, que la ponen a la disposición de los usuarios finales. La integración de redes de este tipo dentro del Internet de las Cosas posibilita que la información que recopilan pueda ser accedida desde cualquier lugar y en cualquier momento, convirtiendo a las redes de sensores inalámbricas en los ojos y oídos del IoT. De esta forma, la información recopilada por las redes de sensores inalámbricas puede ser visualizada y gestionada desde dispositivos inteligentes como *smartphones* y *tablets*, lo que se traduce en una amplia variedad de nuevas aplicaciones para este tipo de redes, ya que los usuarios ahora no solo pueden ser informados de los parámetros que la red mide, sino que pueden también interactuar con ella de una manera sencilla y transparente.

Desgraciadamente, esta integración de distintos dispositivos en el Internet de las cosas no es trivial, ya que según las tecnologías que rodean al IoT se desarrollan y maduran, el espectro de variedad de elementos *hardware* y *software* del que este ecosistema hace uso crece exponencialmente, siendo la compatibilidad entre todos estos distintos elementos uno de los principales desafíos tecnológicos actuales, que ha de ser evaluado desde distintas perspectivas. La primera es desde el punto de vista de la comunicaciones entre los distintos dispositivos, y la segunda desde el punto de vista del procesado y interconexión de datos. En el mundo de las redes de sensores inalámbricas el estándar de comunicaciones más extendido es el IEEE 802.15.4 que define las capas física y MAC para la comunicación de bajo coste, baja tasa de datos, y alta eficiencia energética entre dispositivos. La integración de este estándar de comunicaciones en cualquier dispositivo inteligente permite el uso de éste dentro de una red de sensores inalámbrica, y su comunicación directa con los nodos de esta.

Algunos trabajos previos exploran la integración de dispositivos inteligentes dentro de redes de sensores inalámbricas, como [1], donde un dispositivo *wearable* inteli-

gente se usa dentro de una red de sensores inalámbrica para medir parámetros físicos de un sujeto y fusionarlos con los parámetros ambientales que recopila la red, que trás analizar ambos *datasets* sugiere al usuario distintas rutinas de entrenamiento. Dicho trabajo utiliza *Bluetooth* para comunicar el *wearable* con los nodos, lo que resta eficiencia a la red, al no estar este protocolo optimizado para WSN. [2] realiza una integración similar pero esta vez con el fin de crear un sistema de alerta para industrias en las que se llevan a cabo actividades de riesgo, desarrollando para la aplicación un nodo *wearable* personalizado.

1.2. Objetivos

En este trabajo se propone un sistema similar a [2] desde el punto de vista de la integración de un *wearable* dentro de una red de sensores inalámbrica, pero en el caso de la integración propuesta en el presente trabajo, el *wearable* no es otro nodo más, sinó un dispositivo inteligente con las características fundamentales de un nodo para WSN, además de un sistema operativo a tiempo real, y una interfaz gráfica de usuario desde la cual gestionar y monitorizar la red. De esta forma, el objetivo global de este trabajo es estudiar la integración de un dispositivo inteligente de estas características con una plataforma HW / SW modular para redes de sensores inalámbricas para crear una arquitectura flexible, dinámica y potente que pueda dar respaldo a una gran cantidad de servicios y aplicaciones, orientadas sobre todo a la localización y colaboración mejorada de los nodos de las redes de sensores inalámbricas.

Este fin global del proyecto se puede expresar en función de tres objetivos individuales:

- **Objetivo 1** - Conseguir exitosamente integrar un dispositivo inteligente, de bajo coste, consumo reducido y *wearable* en una red de sensores inalámbrica, de forma que se pueda comunicar utilizando el IEEE 802.15.4, estándar definido con el objetivo de cubrir las necesidades de los nodos para este tipo de redes.
- **Objetivo 2** - Implementar un nuevo modelo o mecanismo que permita mejorar las estimaciones de distancias para algoritmos de localización en redes de sensores inalámbricas basados en el uso de el Indicador de Fuerza de la señal recibida o RSSI (*Received Signal Strength Indicator*). Se propondrá el uso de un procedimiento que involucre también al Indicador de Calidad del Enlace o LQI (*Link Quality Indicator*) en el cálculo de estas estimaciones.
- **Objetivo 3** - Evaluación del sistema propuesto mediante el desarrollo de una serie de aplicaciones que a su vez resalten las posibilidades de la integración *smart wearable*/WSN. Estas aplicaciones contarán con dos enfoques: Un enfoque colaborativo, en el cual se estudiarán las virtudes de la cooperación entre

los nodos de una red de sensores inalámbrica, y las ventajas de incluir un dispositivo inteligente en la red de tal forma que se fomente esta cooperación mediante el desarrollo de una aplicación de detección de eventos de riesgo con un elemento de localización que ponga en uso el modelo que se pretende desarrollar como segundo objetivo de este trabajo. En el segundo enfoque, se implementará una aplicación de provisión de servicios, en la cual el usuario de Hexiwear recibe distintos paquetes de información según se desplaza por un edificio en el cual hay una red de sensores inalámbrica desplegada. Con esta segunda aplicación se muestra la variedad de posibles servicios que puede ofrecer una red de sensores inalámbrica, así como la forma en la cual un dispositivo inteligente como Hexiwear es capaz de recibir la información provista por dichos servicios de forma dinámica y descentralizada.

1.3. Estructura del documento

En este apartado se va a explicar el contenido de los diferentes capítulos del documento teniendo en cuenta que el actual capítulo (**Capítulo 1**) es la introducción.

Capítulo 2: Estado del arte

En este capítulo se hace una explicación detallada de las redes de sensores inalámbricas, de la evolución de los dispositivos conocidos hoy en día como *wearables*, del estado del arte de la integración de dispositivos inteligentes en las WSN, y por último una introducción a los algoritmos de localización para redes de sensores inalámbricas.

Capítulo 3: Plataformas utilizadas

En este capítulo se describen en profundidad las dos plataformas que forman la arquitectura propuesta en este trabajo y se explica en detalle el estándar de comunicaciones que ambas utilizarán para comunicarse: El IEEE 802.15.4.

Capítulo 4: Arquitectura, verificación funcional y interoperabilidad

Este capítulo detalla la arquitectura conseguida con la integración del dispositivo inteligente escogido con la plataforma HW/SW para WSN utilizada. Se detallan las características de comunicación para distintos elementos de cada dispositivo a bajo nivel, así como la perspectiva del funcionamiento global a nivel de aplicación. También se detallan las pruebas realizadas para comunicar los dos dispositivos descritos en el anterior capítulo: El nodo inteligente y *wearable* y la plataforma HW/SW modular para redes de sensores inalámbricas.

Capítulo 5: Localización en redes de sensores inalámbricas

En este capítulo se plantea el problema de localización en las redes de sensores inalámbricas, se describen los principales parámetros utilizados en este tipo de redes para la estimación de distancias, el RSSI y el LQI, con sus ventajas e inconvenientes,

Capítulo 1. Introducción

se exponen distintos resultados experimentales de la medición de dichos parámetros para la arquitectura propuesta, y por último se plantea una solución que mejore las estimaciones de distancias en redes de este tipo.

Capítulo 6: Redes Colaborativas y Cooperación en Redes de Sensores Inalámbricas

En este capítulo se explica qué son las redes colaborativas, se resaltan las ventajas de la cooperación entre los nodos de las redes de sensores inalámbricas, y se exponen dos aplicaciones colaborativas implementadas con el fin de resaltar las ventajas que el uso de un dispositivo inteligente brinda a este tipo de redes. La primera aplicación consiste en una aplicación de detección de eventos de riesgo mediante la cooperación de los nodos de la red, y la segunda es una variante de ésta que incorpora además un elemento de localización que pone en uso la solución propuesta en el anterior capítulo, así como un algoritmo de triangulación.

Capítulo 7: Provisión de servicios en una red de sensores inalámbrica mediante el uso de un *smart wearable*

En este capítulo se exponen las ventajas de utilizar un dispositivo inteligente para recibir los servicios que provienen de una red de sensores inalámbrica y se implementa una aplicación que lleva estas ventajas a la práctica.

Capítulo 8: Conclusiones, líneas futuras, y gestión del proyecto

Se concluye el proyecto realizando un resumen de todos los resultados obtenidos, y de las ventajas que se han podido observar al incluir un dispositivo inteligente en distintos escenarios de nuevas aplicaciones para redes de sensores inalámbricas. También se expondrán líneas futuras de trabajo que busquen aprovechar aún más las capacidades de esta integración. Finalmente se adjuntan los diagramas EDP y Gantt que se emplearon para la organización del proyecto así como el presupuesto del proyecto.

Capítulo 2

Estado del Arte

En este capítulo se va a presentar una introducción a las redes de sensores inalámbricas, se hablará de la evolución de los dispositivos electrónicos inteligentes denominados *wearables* y de su integración dentro de este tipo de redes, y por último del estado actual de los algoritmos de localización para redes de sensores inalámbricas, que son los cuatro temas principales que trata este proyecto.

2.1. Redes de sensores inalámbricas

Las redes de sensores inalámbricas - WSN Wireless Sensor Networks - representan el concepto de computación ubicua de la que hablaba Mark Weiser [3] llevado a la práctica.

“Las tecnologías mas profundas son aquellas que desaparecen. Aquellas que se introducen en el tramo de la vida diaria hasta que son indistinguibles de esta”

Así describía en 1991 su idea de computación ubicua, que consiste en la integración de la tecnología de forma no intrusiva en la vida de las personas, llegando a estar presente en cualquier lugar, a cualquier momento. De forma progresiva, en las últimas décadas, las nuevas tecnologías se han ido incorporando a los distintos hábitos de la existencia humana hasta estar presente en casi cualquier ámbito de esta. De esta forma, una gran mayoría de los objetos que se utilizan en el día a día llevan integrados algún tipo de sistema electrónico o computador: desde nuestros sistemas de transporte a nuestras viviendas, o los accesorios que llevamos encima.

Enmarcadas dentro de este concepto de tecnología ubicua, las redes de sensores inalámbricas están cambiando la manera de recopilar y procesar información de nuestro entorno de forma distribuida, barata, y expandible. Estas redes consisten en despliegues de una gran cantidad de nodos en un área geográfica extensa, para reco-

2.1. Redes de sensores inalámbricas

pilar información sobre parámetros físicos de su entorno, como presión, temperatura, o humedad, y transmitirla a través de la red, de nodo en nodo, hasta un *gateway* o interfaz de salida que hace que estos datos estén disponibles para los usuarios finales. Sus aplicaciones son muy variadas: desde monitorización de variables en entornos de riesgo a sistemas de seguridad, aplicaciones en la industria alimentaria, despliegues en el sector de la agricultura, o sistemas de transporte inteligentes. Las redes de sensores inalámbricas son redes que no necesitan ningún tipo de infraestructura, a diferencia de las redes *Wi-Fi*, por ejemplo, que necesitan un router, o un dispositivo similar para poder funcionar. Este tipo de redes, conocidas como redes *ad-hoc*, son más baratas y sencillas de implementar, al poderse crear *in situ* y no requerir la instalación de equipos especializados que les den soporte.

Las WSN son redes muy inestables debido a que no suele existir ningún tipo de mantenimiento a los dispositivos que las constituyen, por lo que uno de sus requisitos fundamentales es tener algún tipo de dinamismo o adaptabilidad ante cambios en la propia red. Si uno o varios nodos dentro de la misma dejan de estar operativos, la red global tiene que poder seguir funcionando con normalidad. Los dispositivos que forman estas redes son nodos de bajo coste, pequeños y con recursos como capacidades de procesamiento y batería limitados, por lo que desde el nacimiento de esta tecnología se ha investigado mucho tanto en el *hardware* de los dispositivos, como en distintos protocolos de comunicación y rutado para hacer que estos se puedan comunicar de forma eficiente y con el dinamismo mencionado previamente. Es importante resaltar las limitaciones en cuanto a las baterías de los nodos que forman estas redes ya que al desplegarse cientos, quizás miles de nodos, se hace imposible llevar un mantenimiento frecuente de estos, por no decir que no existe mantenimiento alguno. Debido a esto es importante que los nodos tengan un largo tiempo de vida alimentados únicamente por sus baterías, ya que una vez se agoten estas, el nodo dejará de funcionar y generalmente no se reemplazará. Si esto pasa en varios nodos distribuidos a lo largo de la red no hay problema alguno, debido a la capacidad de auto reconfiguración de la red que se ha mencionado con anterioridad. Sin embargo, si esto empezase a ocurrir a gran escala, muchos nodos de la red podrían quedar incomunicados, y la red en sí podría acabar colapsando. Estos nodos constan de cuatro componentes principales:

- Una radio, desde la cual poder modular señales para enviar información y desmodularla para recibirla. La mayor parte del consumo del nodo se debe a estas acciones de enviar y recibir datos inalámbricamente.
- Un procesador, que actuó como cerebro del dispositivo, se comunique con la memoria de este, y gestione todas las acciones del nodo, tanto para el sensado, como para la comunicación y la gestión de la energía. Los procesadores de este tipo de dispositivos no han de ser excesivamente potentes, ya que la mayor parte del procesamiento y gestión de la información no se suele hacer en el nodo, sino en el *gateway*, encargándose el primero solo del preprocesamiento,

recolección y el envío de información.

- Sensores, que pueden recopilar información de dos tipos: analógica o digital. La información analógica tiene una forma de onda continua, como pueden ser temperatura, luz o humedad. La información digital tiene un carácter discreto - on/off o 0 / 1 -, como por ejemplo los dos posibles estados de un interruptor: presionado o no presionado. Independientemente de como sea dicha información, los sensores son los encargados de recopilarla para que se envíe a través de la red, por lo que es de suma importancia su correcto funcionamiento y el adecuado acondicionamiento de ésta información que se recopila para su posterior procesamiento. También es posible que a demás de sensores los nodos incorporen distintos actuadores, como elementos lumínicos o sonoros.
- Una fuente de alimentación que provea al nodo de la energía necesaria para que desarrolle su función. La calidad, el tamaño de esta fuente, y el perfil de funcionamiento del nodo, determinan el tiempo de vida útil del sensor, parámetro que como se ha mencionado anteriormente es de suma importancia en este tipo de redes.

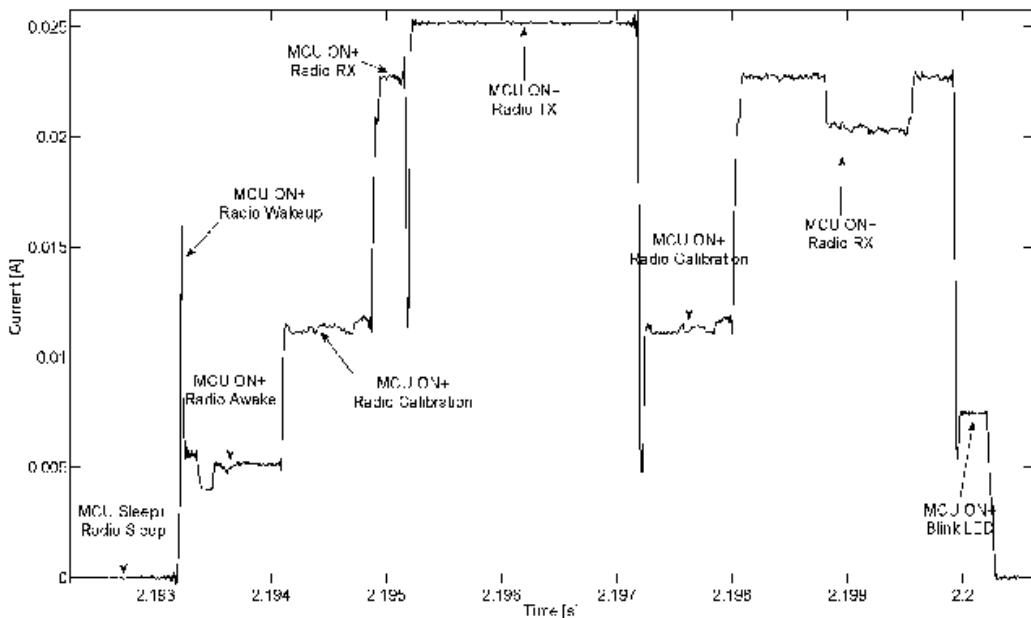


Figura 2.1: Gráfica del consumo asociado a distintas actividades en el funcionamiento de un nodo de una red de sensores inalámbrica

En la Figura 2.1 [4] se muestran los consumos del nodo asociados a distintas acciones. El valle mas alto es debido a la transmisión de datos de forma inalámbrica a otro nodo. El segundo valle mas alto, situado a la derecha del primero se debe

a la recepción de datos a través de la radio. En la parte izquierda de la gráfica se puede observar que cuando el sensor está en modo *Sleep* o de bajo consumo apenas consume potencia de la batería. La idea detrás de este tipo de dispositivos es que estén en este modo la mayoría del tiempo, despertándose periódicamente para tomar una medida y enviarla a través de la red, para acto seguido volver a dormise, alargando así lo máximo posible su vida útil.

La estructura general de un nodo para una red de sensores inalámbrica se muestra en la Figura 2.2.

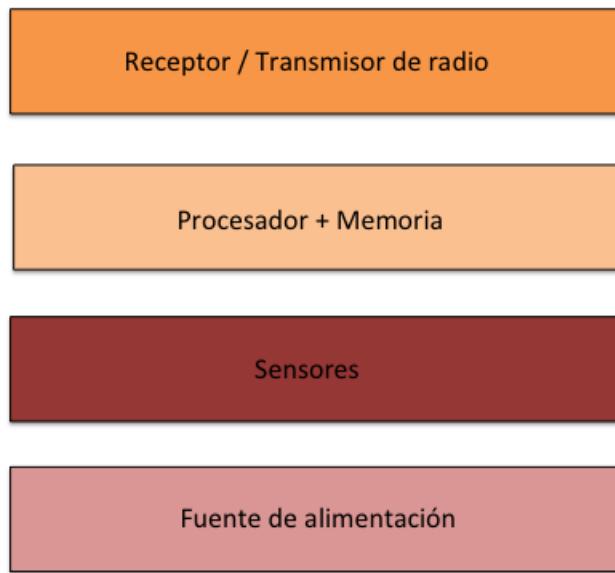


Figura 2.2: Arquitectura general del *hardware* de un nodo para una red de sensores inalámbrica

Desde que se concibió el concepto de una red con estas características, hasta ahora, la miniaturización de los dispositivos electrónicos, así como la mejora de sus capacidades de procesamiento y eficiencia energética, unidas al desarrollo de estándares de comunicación enfocados a sistemas de este tipo, y de mucha investigación en cuanto al software que controla el comportamiento de los nodos, ha hecho que las redes de sensores inalámbricas crezcan en potencial hasta convertirse en una de las tecnologías más prometedoras del siglo XXI.

2.2. La evolución de los *wearables*

Los mismos avances en la electrónica que han hecho que las redes de sensores inalámbricas cobren tanta importancia, han impulsado la evolución de los dispositivos que conocemos hoy en día como *wearables* en las dos últimas décadas. Se trata de dispositivos inteligentes, que generalmente soportan varios tipos de formas de

Capítulo 2. Estado del Arte

comunicación inalámbrica, y que además de ser utilizados como un accesorio para nuestra vestimenta, proveen a su usuario de una gran cantidad de funcionalidades y aplicaciones.

Este tipo de dispositivos se utilizan sobre todo en el campo de la salud y la medicina, como las pulseras *Fitbit*, que miden la frecuencia cardiaca o las calorías quemadas, aunque cada vez se ven más *wearables* con usos más allá de este área, como por ejemplo las *Google Glasses*, gafas de realidad virtual, o ropa fabricada con tejidos inteligentes.

El origen de los *wearables* se remonta a 1810 en Alemania. En el siglo XIX la tecnología ya en si novedosa que hacia funcionar un reloj se miniaturizó hasta conseguirse poder incrustar en una pulsera. De esta forma, los artilleros de la armada Alemana se ahorraban tener que cronometrar con sus relojes de bolsillo mientras manejaban las bombas, algo que a demás de complicado podía ser peligroso. Así, los relojes de pulsera podrían considerarse como los primeros *wearables* de la historia. En el siglo XX las tecnologías *wearable* empiezan a proliferar y a hacerse más populares, al mismo tiempo que se hacen mas sofisticadas, empezando a incorporar componentes electrónicos y a volverse cada vez más inteligentes.

Los primeros relojes digitales empezaron a verse alrededor del año 1956, pero el primer *wearable* moderno, que asentó las bases de lo que son las pulseras inteligentes de hoy en día fue el reloj calculadora de *Casio* que salió al mercado en 1980. Este reloj se muestra en la Figura 2.3. Desde entonces este tipo de dispositivos se han ido haciendo más y más sofisticados y adquiriendo a demás de las funcionalidades mencionadas previamente de monitorización de la salud; capacidades para conectarse con teléfonos móviles y poder realizar todo tipo de tareas como elegir la música que se quiera escuchar, aceptar o rechazar llamadas, acceder a redes sociales, o incluso jugar a juegos.



Figura 2.3: Reloj Calculadora de *Casio* - El predecesor de los *wearables* de hoy en día

Los *wearables* son cada vez mas potentes, baratos y sobre todo inteligentes. En los últimos años los *wearables* han sufrido considerables mejoras en los siguientes aspectos:

2.3. Elementos inteligentes en redes de sensores inalámbricas

- Rendimiento.
- Eficiencia energética y duración de la batería.
- Precio.
- Capacidades de comunicación.
- Aplicaciones.
- Grado de personalización.

En este proyecto se exploran las posibilidades de incluir un dispositivo inteligente de estas características dentro de una red de sensores inalámbrica, siempre respetando los principios de las WSN: se usará un *wearable* de bajo coste, con una alta eficiencia energética, cargado de sensores, y compatible con uno de los principales estándares de comunicaciones para este tipo de redes, el IEEE 802.15.4. Se hablará sobre este dispositivo en el siguiente capítulo.

2.3. Elementos inteligentes en redes de sensores inalámbricas

El desarrollo del *Internet de las cosas*, IoT, o *Internet of Things* en la última década ha posibilitado que se pueda explotar todo el potencial de las redes de sensores inalámbricas, y que las posibles aplicaciones para este tipo de redes se multipliquen. La posibilidad de conectar elementos inteligentes como smartphones, tablets, o los *wearables* mencionados en la anterior sección, a una infraestructura como el internet, y poder comunicarlos de manera sencilla y transparente, ha dado lugar al desarrollo de una enorme cantidad de nuevos protocolos y aplicaciones para dispositivos de estas características. Si a su vez se integran este tipo de dispositivos dentro de una red de sensores inalámbrica, las funcionalidades de la red se ven expandidas gracias a las capacidades de dicho dispositivo, y el propio dispositivo inteligente puede nutrirse de la información que provee la red. Esta integración sin embargo, no es trivial, ya que según las tecnologías que rodean al IoT maduran, la variedad de dispositivos *hardware*, y la heterogeneidad del *software* que dichos dispositivos utilizan crece exponencialmente, haciendo que la integración de todas estas plataformas dentro de un mismo ecosistema sea uno de los mayores desafíos a los que se enfrentan las tecnologías actuales. Para que este proceso pueda realizarse exitosamente, durante los últimos años se han desarrollado protocolos de comunicación enfocados a dar respaldo y facilitar la expansión del IoT a dispositivos de casi cualquier naturaleza, como el IEEE 802.15.4 que se mencionó en la anterior sección. Aunque este estándar está enfocado a cubrir las necesidades de los nodos que forman las redes de sensores inalámbricas, cada vez más dispositivos inteligentes son

compatibles con el, permitiendo la integración de dichos dispositivos dentro de una red de estas características, y su comunicación directa con los nodos que la constituyen. Si dicho dispositivo es además un *wearable*, se rompe la primera barrera de acceso tecnológico, ya que el usuario solo tiene que llevarlo como un accesorio en su día a día. [1] Explora las consecuencias de la integración de un dispositivo de estas características dentro de una red de sensores inalámbrica para la monitorización de la actividad física de un individuo. El *wearable* recolecta parámetros físicos como el ritmo cardíaco, las pulsaciones, o las calorías quemadas. Por su parte la red de sensores inalámbrica recolecta datos ambientales como la presión, temperatura, humedad, o luz ambiente. En función de estos dos sets de datos, la red toma decisiones relacionadas con el desempeño del individuo y le sugiere nuevos ejercicios. En dicha propuesta también se implementa un *middleware* para facilitar la integración del *wearable* con los nodos inalámbricos.

Esta integración de *wearables* con WSN se aleja de las aplicaciones tradicionales de las redes de sensores inalámbricas, buscando propuestas más dinámicas y con una mayor participación por parte del usuario. En [2] se utiliza un nodo *wearable* en una red de sensores inalámbrica para implementar un sistema de detección de riesgo en industrias en las que se llevan a cabo actividades peligrosas, siendo desarrollado dicho nodo específicamente para la aplicación.

El sistema que se propone en el presente trabajo es similar al que se implementa en [2] desde el punto de vista de integración *wearable* / Wireless Sensor Network, pero esta vez el *wearable* no es otro nodo más, sino una plataforma inteligente con las funcionalidades principales de un nodo inalámbrico, un sistema operativo a tiempo real, y una interfaz gráfica de usuario desde la cual monitorizar y gestionar la red. En este trabajo se estudia la integración de este dispositivo con una plataforma HW / SW modular para WSN, con el objetivo de utilizar la estructura global para aplicaciones de localización y detección.

2.4. Introducción a la localización en redes de sensores inalámbricas

Conocer la localización de un nodo determinado en una red de sensores inalámbrica es de suma importancia, ya que la información que estos nodos recopilan y transmiten sobre el entorno es completamente irrelevante si no se conoce cuál es dicho entorno. Debido a ello la localización de los nodos es un amplio campo de investigación dentro del mundo de las redes de sensores inalámbricas. Para esta tarea de localización, existen principalmente dos familias de algoritmos, los llamados *Range-Free algorithms* y los *Range-based algorithms*. Los primeros utilizan la comunicación mediante radio entre los distintos nodos para inferir su posición, y en ningún momento miden la distancia que separa a dos nodos utilizando parámetros de radio,

2.4. Introducción a la localización en redes de sensores inalámbricas

sino que en el caso de necesitarse alguna distancia, esta se obtiene mediante estimaciones o cálculos no relacionados con parámetros inalámbricos. Estos algoritmos no requieren ningún tipo de *hardware* adicional y al no usar estimaciones de distancia a partir de parámetros de radio, eliminan los posibles errores que estas estimaciones suelen acarrear. Sin embargo para su correcto funcionamiento es conveniente que las redes en las que se utilizan dichos algoritmos consten con un elevado número de nodos, y su comportamiento es poco fiable en redes irregulares. En los segundos se utilizan parámetros de radio para estimar la distancia o el ángulo entre los nodos, como la potencia recibida con la señal inalámbrica o el tiempo de viaje de esta entre un nodo y otro. Independientemente de la familia de algoritmos que se utilice, existen una serie de conceptos que son comunes a todos los algoritmos de localización. A continuación se describen una serie de ellos para facilitar la posterior compresión del texto por parte del lector:

- **Nodo ancla o *Anchor node*:** Se trata de nodos con posiciones conocidas que se utilizan como referencias para calcular la posición del resto de los nodos de la red. Estos nodos pueden conocer sus coordenadas ya sea porque utilizan algún tipo de sistema de localización como el GPS, o porque se han depositado en un lugar preciso y conocido al desplegar la red. El uso de sistemas como GPS para algoritmos de localización se utiliza cuando la red cuenta con solo uno o muy pocos nodos ancla, ya que el *hardware* adicional necesario de este sistema incrementa el precio total del despliegue y va en contra de la filosofía de sencillez y bajo coste de las redes de sensores inalámbricas.
- **Nodos estacionarios y nodos móviles:** Los nodos estacionarios son aquellos cuya posición desde que se despliega la red no varía. Los nodos móviles son aquellos que varían su posición con el tiempo, ya sea porque están embebidos en algún tipo de sistema *wearable* o porque son desplazados por algún tipo de agente con un objetivo específico como puede ser descargar de tráfico cierta zona de la red o el envío de mensajes para asistir en la localización de nodos con posición desconocida.
- **Triangulación:** La triangulación se utiliza para determinar de forma unívoca la posición de un objeto. Esto se hace conociendo la distancia a tres puntos, como mínimo. Desde cada uno de estos puntos se traza una circunferencia con radio igual a la distancia desde dicho punto al objeto cuya posición se desea conocer. Al trazar esta circunferencia desde el primer punto, se tiene un lugar geométrico circunferencial donde se puede encontrar nuestro objeto. Al trazar una nueva circunferencia desde un segundo punto con radio igual a la distancia de este segundo punto a el objeto en cuestión, se obtienen dos puntos (que son las intersecciones de esta segunda circunferencia con la circunferencia inicial), donde puede estar nuestro objeto. Al realizar este proceso con un tercer punto determinamos unívocamente la posición de este.

La Figura 2.4 muestra un esquema de la implantación de un algoritmo de triangulación utilizando 3 nodos ancla, y un nodo móvil.

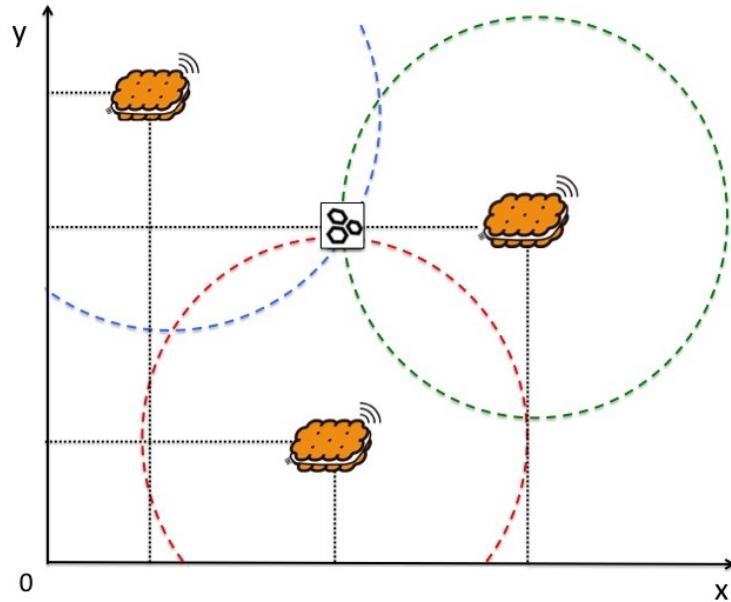


Figura 2.4: Esquema de un algoritmo de triangulación usando 3 nodos ancla y un nodo móvil.

A continuación se van a explicar algunos de los algoritmos mas utilizados para cada una de las familias. En la familia de métodos *Range-Free*, los algoritmos mas importantes son:

- **DV-Hop** [5]: Este algoritmo busca obtener la posición de los nodos de una red desplegada utilizando el número de saltos o *hops* desde cada uno de los nodos con posiciones inciertas a una serie de nodos ancla con posiciones conocidas. Para ello al menos 3 nodos ancla retransmiten sus coordenadas lanzando *broadcast* a través de la red, propagándose dicha información de nodo en nodo a lo largo de esta. Al salir de los nodos ancla estos mensajes tienen una cuenta de salto 0, que se incrementa al llegar al nodo vecino de dicho nodo ancla. Así, al ir propagándose el mensaje desde el nodo ancla hacia los nodos mas lejanos a este, el contador se va incrementando de uno en uno, de forma que cuando este proceso de *broadcast* ha terminado, cada nodo conoce el número de saltos que lo separan de cada uno de los tres nodos ancla. Posteriormente los nodos ancla, conociendo la distancia entre cada uno de ellos y el número de saltos que los separa calculan una distancia media de salto expresada en la siguiente fórmula.

$$distanciaMediaDeSalto = \frac{distanciaEntreNodosAncla}{numeroDeSaltosEntreNodosAncla}$$

2.4. Introducción a la localización en redes de sensores inalámbricas

Una vez hecho esto, cada uno de los nodos de posición desconocida calcula una estimación de la distancia a cada uno de los nodos ancla, multiplicando la distancia media de salto por el número de saltos a cada uno de dichos nodos. De esta forma todos los nodos de la red conocen las distancias a cada uno de los tres nodos ancla de la misma, y utilizando la triangulación que se ha explicado previamente estiman sus coordenadas.

- **Hop terrain** [6]: Este método es exactamente igual que el anterior, pero consta de una segunda fase en la cual los nodos, una vez conocida su posición, mandan mensajes con sus coordenadas a los nodos vecinos, y llevan a cabo un refinamiento de estas coordenadas utilizando el método de mínimos cuadrados.
- **APIT (Approximate point in triangulation)** [7]: Al igual que en los algoritmos descritos previamente, se despliegan una serie de nodos ancla que conocen sus coordenadas debido al uso de sistemas GPS. Los pasos que sigue este algoritmo son los siguientes:
 - 1. Los nodos ancla transmiten mensajes *broadcast* con sus coordenadas.
 - 2. Los nodos desconocidos crean una tabla con las coordenadas, los identificadores de cada uno de los nodos ancla, y la potencia con la que se han recibido los mensajes provenientes de estos.
 - 3. Los nodos desconocidos cogen 3 nodos ancla cualesquiera de la tabla y estiman si es posible que su posición se encuentre dentro del triángulo que forman dichos nodos, utilizando las coordenadas de estos y la potencia de la señal recibida en los mensajes que transmitían dichas coordenadas.
 - 4. Tras realizar dicho test con todas las combinaciones de nodos ancla posibles que permite la red, se superponen las áreas de dichos triángulos y el centroide del área común a todos ellos es el lugar donde se encuentra el nodo desconocido.

Para la familia de métodos *Range based*, los más importantes son:

- **Algoritmos basados en el RSSI (Recieved signal strength indicator)**: El RSSI es un indicador de la potencia con la que se recibe una señal inalámbrica. Se suele utilizar una escala de referencia en relación a $1mW$ para medir este nivel de potencia, que suele variar desde unos $0dBm$ en una señal ideal (en la práctica nunca se consiguen $0dBm$) hasta unos $-95dBm$ para señales muy débiles. Para utilizar este tipo de algoritmos es necesario modelar la forma en la cual la señal inalámbrica se propaga por el espacio, de tal forma que se pueda conocer la relación entre este parámetro y la distancia. Para ello existen distintos modelos en función de si la señal se transmite en exteriores o interiores, de la longitud de las antenas y de la distancia que viaja dicha señal, pero todos ellos se basan en el principio de que cuanto más se desplaza la señal, y por lo tanto mas distancia recorre, menor es la potencia con la cual esta será recibida.

- **Time difference of arrival (TDOA)** [8]: En este tipo de técnicas se calcula la distancia utilizando la diferencia de tiempos entre la señal de radio recibida y una señal proveniente de un altavoz colocado en un nodo con posición conocida. Distintos nodos ancla equipados con altavoces mandan señales de radio al nodo con posición desconocida cuyas coordenadas se quieran estimar. Después de mandar esta señal se espera un tiempo determinado de retraso y se retransmite una nueva señal utilizando los altavoces esta vez y no la radio. Al llegar la señal de radio al nodo desconocido, este activa un micrófono que recogerá la señal emitida por el altavoz del nodo ancla. Terminado este proceso el nodo desconocido calculará la distancia a la que se encuentra del nodo ancla utilizando la siguiente fórmula:

$$d = (s_{radio} - s_{sound}) \times (t_{sound} - t_{radio} - t_{delay})$$

donde las distintas variables representan los tiempos de espera, el tiempo que tarda en llegar la señal de radio, y el tiempo que tarda en llegar la señal de el altavoz.

- **Time of arrival (TOA)** [8]: En este algoritmo, al igual que en el anterior, se mide el tiempo que tarda en llegar la señal de radio de un nodo a otro, pero en vez de utilizarse una señal adicional procedente de un agente externo como un altavoz para medir la diferencia de tiempos y estimar la distancia, todos los nodos de la red están sincronizados temporalmente. De esta forma, al llegar una señal inalámbrica de un nodo ancla a un nodo con posición desconocida, este no tiene más que multiplicar el tiempo que ha tardado dicha señal en su viaje por la velocidad conocida de esta para estimar la distancia a la que se encuentra de dicho nodo ancla. Este algoritmo alcanza unas precisiones muy elevadas pero requiere una alta capacidad de procesamiento, y es la técnica que utilizan los sistemas GPS para la estimación de la distancia.

Las redes de sensores inalámbricas tienen grandes limitaciones en cuanto a tamaño de los nodos, batería, y coste, que han de ser consideradas antes de diseñar cualquier algoritmo de localización, ya que estos algoritmos suelen ser específicos para la aplicación en la que se vayan a utilizar. Algoritmos que consigan una elevada precisión para una aplicación específica no tienen porque conseguir buenos resultados en aplicaciones distintas, ya que cada aplicación presenta peculiaridades que hacen que ciertos algoritmos tengan mejor comportamiento que otros, como el uso de nodos móviles, la densidad de nodos en la red, o la distancia entre estos.

Las figuras 2.5-2.7 muestran distintas comparativas de los aspectos de mayor importancia de las redes de sensores inalámbricas en función de la metodología de localización utilizada.

2.4. Introducción a la localización en redes de sensores inalámbricas

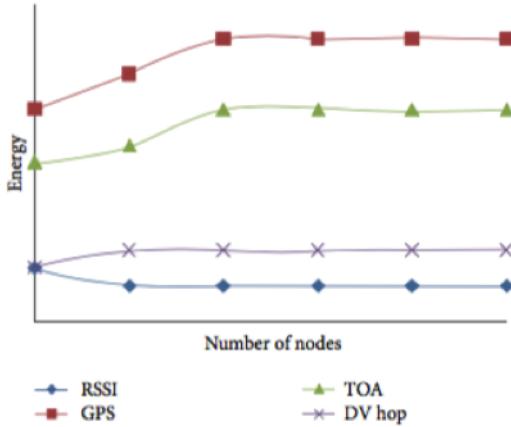


Figura 2.5: Gráfica cualitativa del consumo energético asociado a distintos algoritmos de localización

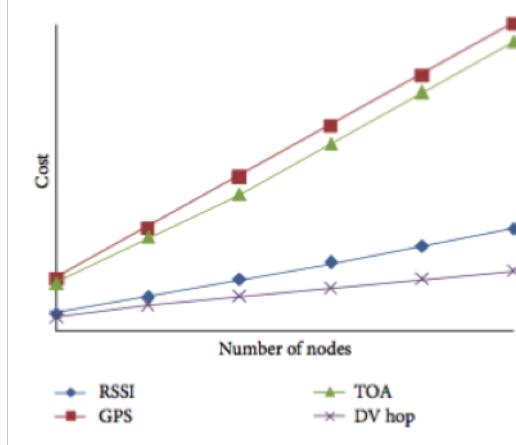


Figura 2.6: Gráfica cualitativa que muestra el coste de implementación de distintos algoritmos de localización para redes de sensores inalámbricas

De estas gráficas se puede observar que el método más caro y con mayor consumo de energía se basa en el uso de GPS, aunque también es el más preciso. Los métodos que se implementarán para la arquitectura propuesta se basan en el uso del RSSI, que como se puede observar tienen el menor consumo energético de los métodos evaluados, y es de los métodos con menor coste. A su vez, el uso de parámetros de radio como este puede dar lugar a errores significativos en las estimaciones, por lo que en el capítulo 5 se propondrá una metodología que reduzca el error de estas estimaciones para distancias reducidas en ambientes *indoor*.

Capítulo 3

Plataformas utilizadas

3.1. Hexiwear

3.1.1. ¿Qué es Hexiwear?

Hexiwear [9] es un kit de desarrollo para aplicaciones de IoT (*Internet of Things*) creado por *MikroElektronika* en colaboración con NXP (compañía desarrollada a partir de la división de semiconductores de la conocida empresa holandesa *Philips*). La idea originalmente se presentó en una campaña de la conocida página web de crowdfunding *Kickstarter*, consiguiendo un 230 % de la cantidad buscada inicialmente, y debido a este éxito empezó su colaboración con NXP.

Este dispositivo, que de apariencia puede parecerse a un clásico *Tamagochi*, está orientado a desarrolladores o *Makers*, ya que es completamente *Open Source*, lo que significa que todos los esquemáticos de los circuitos que lleva dentro, así como los códigos integrados en sus microcontroladores están disponibles para que cualquiera los descargue y modifique, pudiendo dotar al dispositivo de casi cualquier finalidad: desde un *smartwatch* con diversas aplicaciones a un dispositivo desde el que gestionar un sistema de automatización doméstico, o un nodo inteligente para una red de sensores inalámbrica como en el caso de este proyecto. Esta capacidad de personalización, unida a un *hardware* muy bien equipado y fácilmente expandible, a un precio moderadamente reducido, a sus capacidades de conectividad (con Bluetooth Low Energy e IEEE 802.15.4), y un diseño atrevido y elegante, convierte a *Hexiwear* en una plataforma de prototipado que ha despertado mucha expectación desde su lanzamiento. En la Figura 3.1 se muestra dicho dispositivo.



Figura 3.1: Hexiwear

3.1.2. Hardware

Una de las características que hace a *Hexiwear* un producto tan atractivo es la gran cantidad de dispositivos que agrupa en un espacio reducido y por precio muy asequible, lo que se ha conseguido gracias a la asociación con NXP, ya que siendo esta compañía uno de los mayores productores de microcontroladores y sensores del mundo, ha facilitado la posibilidad de que se pueda crear un producto que aporte un rendimiento tan alto por precio tan bajo.

Hexiwear cuenta con dos microcontroladores, que son los cerebros del dispositivo: El Kinetis K64X [10] que hace de microcontrolador principal o maestro, gestionando la pantalla, los sensores, el LED RGB y el motor haptico; y el Kinetis KW40Z [11], que actúa como esclavo del anterior y se encarga de las comunicaciones (Bluetooth Low Energy y IEEE 802.15.4) y de gestionar el input de los botones capacitivos.

La familia de microcontroladores K64 presenta altas densidades de memoria, capacidades de consumo reducido y una integración optimizada en un tamaño muy reducido. Por su parte, el KW40Z es un chip altamente integrado con un consumo ultra-reducido y que permite el uso de Bluetooth Low energy o del estándar IEEE 802.15.4 para el desarrollo de aplicaciones que requieran de un consumo mínimo.

Ambos cuentan con procesadores ARM y se comunican mediante UART. La gestión y personalización de la comunicación entre ambos controladores ha sido una parte esencial del trabajo realizado. Sobre la comunicación entre los dos núcleos del Hexiwear se hablará en el capítulo 4.

Capítulo 3. Plataformas utilizadas

Hexiwear cuenta también con una pantalla OLED, de 1,1 pulgadas (96x96 píxeles), una batería de 190 mAh de polímero de litio, un LED RGB, y un motor haptico, a demás de con los siguientes sensores, fabricados en su gran mayoría por NXP:

- Un acelerómetro y magnetómetro de 3 ejes (NXP FXOS8700CQ)
- Un giroscopio digital de 3 ejes (NXP FXAS21002)
- Un sensor digital de presión absoluta (NXP MPL3115A2R1)
- Un sensor de ritmo cardiaco
- Un sensor de temperatura y humedad
- Un LDR para medir la cantidad de luz ambiente

Todos estos sensores se comunican mediante I2C con el MK64. La Figura 3.2 muestra la interconexión de todos los elementos que constituyen el hardware de Hexiwear [12].

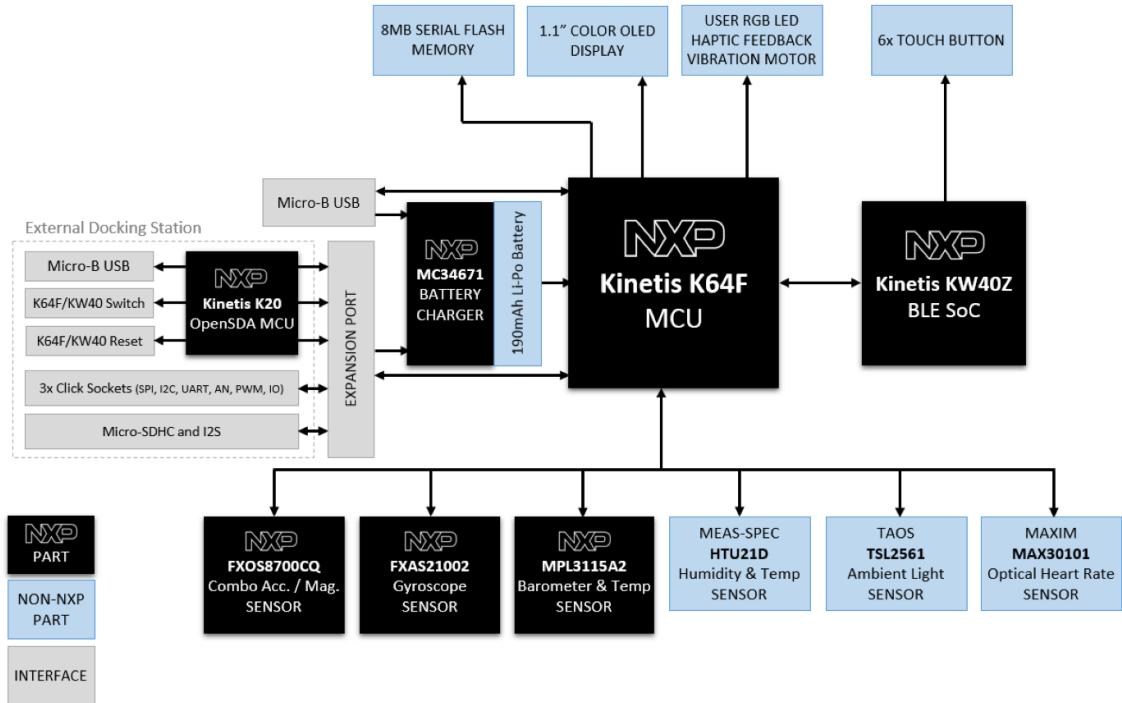


Figura 3.2: Diagrama de la interconexión de los componentes del Hexiwear.

Para programar el dispositivo es necesaria una estación de carga o *Docking Station* (incluida en el *User pack* que vende MikroElektronica). Esta estación cuenta

con una interfaz OpenSDA que permite acceder a ambos microcontroladores conectándolos mediante USB al ordenador y de esta forma modificar su firmware. Dicha interfaz también actúa como un adaptador para poder depurar los programas que corren dentro de los microcontroladores del dispositivo a tiempo real, algo muy importante para poder diagnosticar problemas en el funcionamiento de un sistema de estas características.

Esta estación de carga a parte de para programarlo, también dota al hardware de Hexiwear de la expandibilidad de la que se hablaba antes, ya que cuenta con tres puertos para conectar unas pequeñas tarjetas desarrolladas también por MikroElektronika llamadas *Click Boards*, que pueden tener integrados distintos sensores (como sensores de pH, alcolímetros, o sensores de altitud) módulos de comunicaciones (como Wi-Fi, Zigbee, o 3G) o placas para conectar motores. De esta forma las posibilidades del dispositivo son casi ilimitadas. La Figura 3.3 muestra una imagen de dicha estación de carga y de varios *Click Boards* [13].

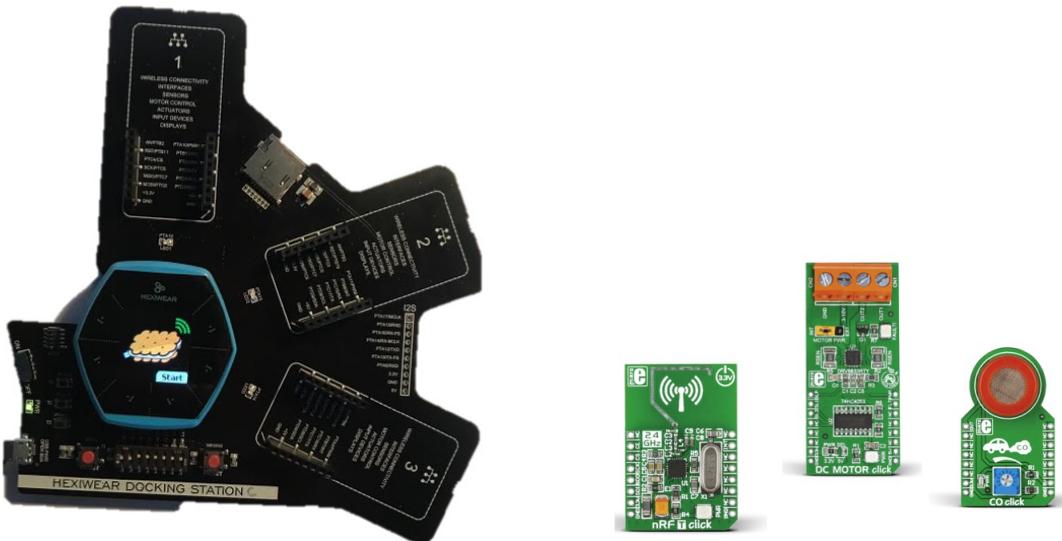


Figura 3.3: Imagen del Hexiwear conectado a la *docking station* y de las *Click Boards*

3.1.3. Software

Como se ha mencionado previamente, el software que utiliza Hexiwear es completamente *Open Source*, por lo que es posible modificarlo según las necesidades del proyecto que se vaya a desarrollar. NXP facilita a aquellos que quieran desarrollar aplicaciones utilizando este dispositivo un Software Development Kit (SDK) con ejemplos de distintos programas, documentación sobre el dispositivo, drivers de los distintos periféricos y un software para programar su microcontrolador principal, el *Kinetis Design Studio* (KDS). Desde este software se puede, una vez descargado,

modificar el firmware de dicho microcontrolador y con la *docking station* cargárselo al dispositivo. De esta forma se pueden incluir nuevas aplicaciones en las que ya trae Hexiwear de por si, o sobrescribir todo el código para desarrollar una aplicación de propósito único. Para este proyecto, aunque la aplicación a desarrollar estaba dentro de un marco bastante concreto, se decidió integrarla dentro del *firmware* principal de Hexiwear, sin sobrescribirlo, lo que ha aumentado la complejidad del proyecto pero también ha posibilitado que el dispositivo conservase sus funcionalidades originales.

Para programar el otro microcontrolador, el KW40Z, encargado de las comunicaciones, ha sido necesaria la instalación de un software adicional, el *IAR Embedded Workbench for ARM*. Desde este entorno de desarrollo los pasos a seguir para modificar el *firmware* del dispositivo son idénticos que desde el KDS. Para este proyecto ha sido necesario modificar el código de ambos microcontroladores, por lo que inicialmente hubo un breve periodo de familiarización con los entornos de desarrollo mencionados previamente. A continuación se explica la estructura del proyecto de Hexiwear para el KDS, que es el software que se utiliza para programar el uC principal, ya que la estructura del proyecto del uC de las comunicaciones es idéntica pero con menor contenido. Dentro de la carpeta *code* de este proyecto se encuentran una serie de carpetas divididas en función de los distintos aspectos o funcionalidades del proyecto que implementan. Dichas carpetas son:

- **apps:** Esta carpeta incluye la lógica de las aplicaciones más complejas del dispositivo, que son el monitor de ritmo cardíaco y el contador de pasos.
- **extern_flash:** Esta carpeta incluye funcionalidades relacionadas con la gestión de la memoria del dispositivo.
- **gui:** Aquí se implementan las funcionalidades que tienen que ver las distintas pantallas que se muestran en el dispositivo, su estructuración en distintos menús, y la navegación de una a otra, así como la funcionalidad del resto de las aplicaciones más sencillas del Hexiwear como la linterna o la estación meteorológica.
- **HEXIWEAR:** Contiene funcionalidades no ligadas a una entidad determinada, sino a aspectos globales del proyecto como la inicialización del *hardware*.
- **intf:** Esta carpeta contiene el código encargado de gestionar la comunicación entre los dos núcleos del Hexiwear.
- **notifications:** Implementa el código encargado de gestionar el intercambio de notificaciones con el dispositivo móvil al que el Hexiwear esté emparejado vía Bluetooth.
- **rtc:** Real time clock, aquí se encuentra el código asociado con el reloj, los timers, y las determinadas acciones que se activan en función del tiempo.

- **sensors**: Código que implementa la funcionalidad de los sensores con los que cuenta el hardware de Hexiwear.
- **OLED**: En esta carpeta se encuentran las funcionalidades asociadas al uso del display en cuanto a añadir imágenes, borrar el contenido de la pantalla, insertar texto, etc.
- **power**: En esta carpeta se incluye el código encargado de gestionar la eficiencia energética del dispositivo.

En el *firmware* de ambos microcontroladores se encuentra embebido el sistema operativo en tiempo real *FreeRTOS* [14]. Normalmente, para aplicaciones sencillas en este tipo de dispositivos se suele ejecutar un programa en bucle que evalúa los distintos periféricos y actúa en consecuencia. De esta manera se pierde un poco de tiempo del procesador comprobando cada condición, para saber de qué manera se debe actuar, pero al ser el programa sencillo, y las condiciones limitadas, el rendimiento general del dispositivo no se ve afectado gravemente. Cuando los proyectos se complican y aumentan la cantidad de condiciones que se han de evaluar, o cuando es muy importante la reacción inmediata del dispositivo ante un evento, se hace inviable proceder de esta manera, ya que se perdería mucho tiempo evaluando todas las condiciones y podría darse un retardo significativo en cada una de las acciones a realizar. En estas ocasiones se incluye en el código en cuestión un sistema operativo en tiempo real, que al igual que un sistema operativo de escritorio se encarga de gestionar los recursos del microcontrolador y abstraer al usuario del hardware, pero dándole mucha más importancia al tiempo de reacción ante un evento. El RTOS (Real Time Operating System) hace esto asignando a cada tarea a ejecutarse una prioridad, de manera que aquellas con una prioridad elevada pasaran a ejecutarse inmediatamente cuando se de la condición necesaria, aunque exista otra tarea ejecutándose en ese momento. Estos sistemas también suelen gestionar la comunicación entre tareas mediante sistemas de colas y eventos. En el siguiente capítulo se profundizará en la importancia de este RTOS y en para qué se utiliza en el presente proyecto.

3.1.4. IEEE 802.15.4

A pesar de todos estos recursos, la característica principal por la cual se ha elegido Hexiwear para este proyecto es por su compatibilidad con el IEEE 802.15.4, un estándar desarrollado para cubrir las necesidades de las comunicaciones entre dispositivos con pocos recursos y en los cuales la eficiencia energética es fundamental. Este estándar define las capas física y de acceso al medio de redes inalámbricas de área personal (WPANs - Wireless Personal Area Networks) con bajas tasas de transmisión de datos, y que tienen un rango máximo de comunicación entre dispositivos de unos diez metros. El objetivo de este estándar es definir los niveles de red fundamentales para dar servicio a redes inalámbricas que busquen comunicar

Capítulo 3. Plataformas utilizadas

dispositivos ubicuos, de bajo coste y velocidad de transmisión de datos reducida. Este tipo de redes no requiere ningún tipo de infraestructura, a diferencia de las redes *Wi-Fi* que requieren algún tipo de dispositivo para poder funcionar, como un *router*, lo que facilita y abarata su implementación. Sobre este estándar se basa el *Zigbee* cuyo objetivo es aportar soluciones para los niveles superiores de la pila de protocolos que el estándar no cubre. La capa MAC cubierta por el IEEE 802.15.4 esta a su vez dividida en dos subcapas con funciones bien diferenciadas. La primera subcapa, MLME (MAC Sublayer Management Entity) se encarga de gestionar los mensajes asociados a comandos dirigidos o provenientes de la capa física, como por ejemplo mensajes de asociación, o de comienzo de la red. La otra subcapa, MCPS (MAC Common Part Sublayer) es la capa encargada de gestionar las primitivas asociadas a los datos que se envían a través de dicha red. La Figura 3.4 muestra la arquitectura por capas del IEEE 802.15.4 y el *Zigbee*.



Figura 3.4: Esquema de la arquitectura por capas del IEEE 802.15.4 y el *Zigbee*

En las redes del IEEE 802.15.4 existen básicamente dos tipos de dispositivos:

- *Full Function Device - FFD*: Son dispositivos que pueden funcionar como coordinador de la PAN (El dispositivo que crea la red. Todas las redes de este tipo necesitan un coordinador) o como un nodo normal.
- *Reduced Function device - RFD*: Dispositivos sencillos con recursos muy limitados, que solo pueden comunicarse con FFDs y nunca pueden ser coordinadores de la PAN.

El IEEE 802.15.4 define dos tipos de topología de red. Ambas topologías utilizan un único coordinador de la PAN:

- Topología en estrella: En esta topología toda la comunicación pasa por el coordinador de la PAN. La comunicación directa entre dispositivos que no sean el coordinador de la PAN no es posible.
- Topología punto a punto: En esta topología, también conocida como *Peer-to-peer* o P2P, la comunicación puede darse entre cualquier dispositivo de la red, siempre que los dispositivos que se comunican se encuentren dentro del alcance que el estándar permite.

La figura 3.5 [15] muestra un diagrama de ambas topologías.

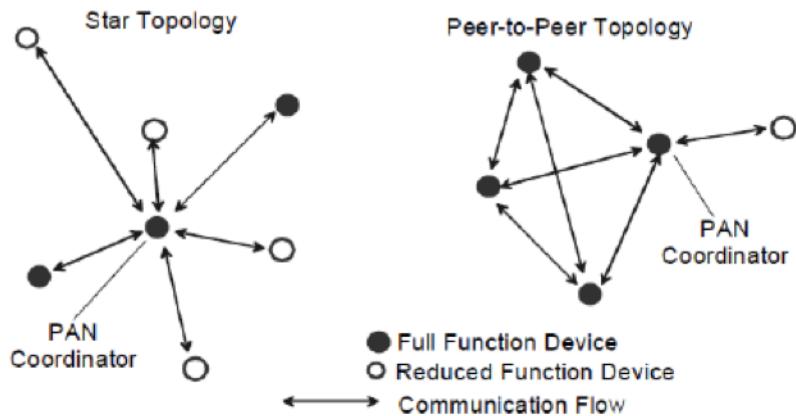


Figura 3.5: Distintas topologías que permiten las redes de IEEE 802.15.4: Estrella (Izq) y P2P (Dcha)

Independientemente de la topología, se necesita al menos un FFD que actué como el coordinador de la red. En este tipo de redes todos los dispositivos cuentan con un identificador único de 64 bits conocido como el *Extended Address* o dirección larga del dispositivo, aunque muchas veces se utilizan identificadores de 16 bits llamados *Short Address* o dirección corta para interactuar dentro de la red, de tal forma que se reduce la cantidad de información que es necesario enviar con cada mensaje, con lo que se mejora la eficiencia de la transmisión de datos. Las redes PAN de este tipo también cuentan con un identificador único de la red llamado PANID.

Este estándar de comunicaciones se diferencia de estándares como el *Wi-Fi* o el Bluetooth en que permite una cantidad enorme de dispositivos en la red, tiene una menor tasa de transmisión de datos y una eficiencia energética mucho mayor. En la Tabla 3.1 se muestra una comparativa de algunos aspectos de estas tres formas de comunicación inalámbrica. Cabe resaltar que aunque previamente se ha mencionado que el alcance que el IEEE 802.15.4 permite es de unos diez metros, esta distancia

Capítulo 3. Plataformas utilizadas

es la máxima a la que se pueden comunicar dos dispositivos vecinos. Añadiendo más dispositivos a la red la distancia que esta cubre puede incrementarse de forma significativa, como se muestra en la tabla.

Característica	Wi-Fi	Bluetooth	IEEE 802.15.4
Batería	Horas	Días	Meses/Años
Dispositivos	32	8	6400
Distancia	100 metros	10 metros	70-300m
Tasa de transferencia	11 Mbps	1 Mbps	250 Kbps

Tabla 3.1: Tabla comparativa de distintos aspectos de los protocolos de comunicación más comunes.

La estructura de las tramas del IEEE 802.15.4 se muestra en la Figura 3.6. Empiezan con la cabecera de la capa física, que contiene un delimitador de comienzo, seguido varios bytes de control y por último un byte que indica la longitud de la trama. A continuación se incluye la trama MAC, que empieza también con unos bytes de control y posteriormente incluye la dirección de destino del mensaje, la dirección de la que proviene dicho mensaje, y el mensaje en si. Los últimos 3 bytes del mensaje contienen los valores del RSSI y LQI de dicho mensaje (parámetros de radio de los que se explicaran en el capítulo 5) y el CRC, valor que se utiliza para comprobar la integridad del mensaje y asegurar que no ha sido modificado.

IEEE 802.15.4 Frame

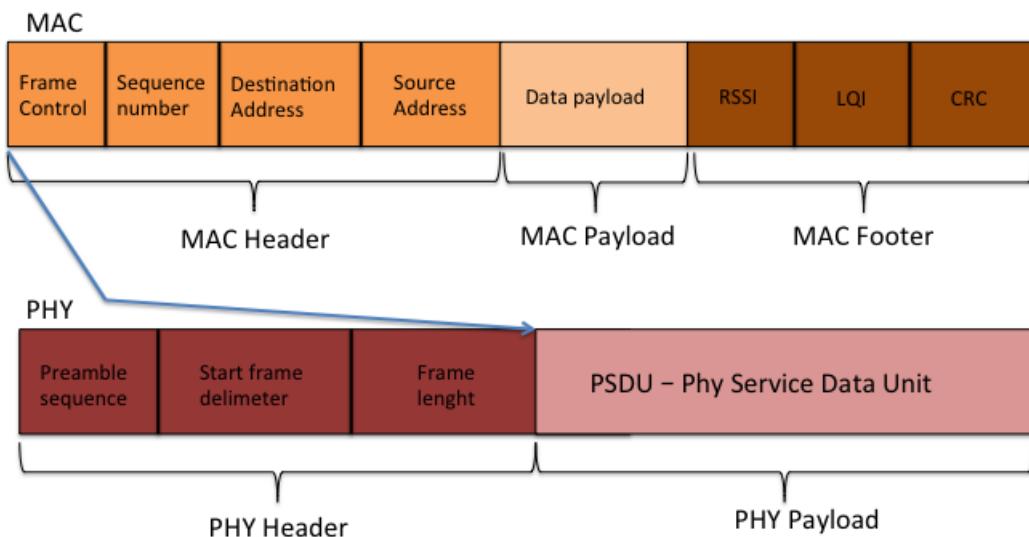


Figura 3.6: Imagen de la estructura de la trama definida por el IEEE 802.15.4

Aunque el Hardware de Hexiwear, y mas específicamente el KW40Z sean compatibles con este estándar, el *firmware* que traía este microcontrolador no hacía uso de él, sino que implementaba distintas aplicaciones con Bluetooth Low Energy. Por ello, una parte muy importante de este proyecto ha consistido en hacer una limpieza del código que existía en dicho microcontrolador para hacer espacio para el código que implementa el IEEE 802.15.4, y posteriormente portar las librerías y distintas funciones de las que hace uso el estándar para su funcionamiento. Una vez hecho esto, se estudiaron ejemplos de códigos de distintas aplicaciones que usan el estándar en dicho MCU y se adaptaron para realizar algunas pruebas que permitiesen establecer una comunicación inicial entre el Hexiwear y las *Cookies*, nodos para redes de sensores inalámbricas desarrollados en el Centro de Electrónica Industrial (CEI) de la Universidad Politécnica de Madrid, de las cuales se hablará en las siguiente sección.

3.2. Plataforma Cookies

3.2.1. ¿Qué son las Cookies?

Las Cookies [16] son una plataforma hardware/software para redes de sensores inalámbricas desarrollada por el Centro de Electrónica Industrial de la Universidad Politécnica de Madrid, con el fin de contar con la capacidad de personalización y control a bajo nivel que conlleva desarrollar una tecnología propia. La característica principal de esta plataforma es la modularidad: está basada en una arquitectura por capas, en la que cada capa realiza una función determinada dentro del nodo, atendiendo a una de las necesidades principales de un dispositivo para una red de sensores inalámbrica. Principalmente se utilizan cuatro capas conectadas entre sí mediante un bus común que les permite interactuar y compartir información. Dicho bus se divide en dos partes, la parte digital y la parte analógica, cada una con un conector situado en uno de los lados de las distintas capas. El bus digital transporta las señales provenientes de sensores digitales, interfaces de comunicación y entradas y salidas de propósito general. El bus analógico, que cuenta con un conector en el extremo opuesto al del bus digital, transporta las señales de sensores analógicos hacia convertidores analógico-digitales, señales de control de actuadores, y también entradas y salidas de propósito general.

Las cuatro capas mencionadas previamente fueron diseñadas teniendo en mente las funcionalidades principales que ha de poseer un nodo inalámbrico: alimentación, procesamiento, sensado y capacidades de comunicación. Es posible agregar más capas a las cuatro fundamentales, como por ejemplo capas adicionales de memoria, de depuración, Bluetooth, o de Ethernet. Este diseño por capas permite a las Cookies adaptarse a una gran cantidad de entornos de aplicación, alternando distintas capas en función de lo que estos entornos demanden. Para pasar de una aplicación en la que se toman medidas de temperatura y humedad a otra en la que sea necesario medir la cantidad de CO₂ en el aire solo se tendría que cambiar la capa de sensado del nodo. De la misma forma solo sería necesario cambiar la capa de comunicaciones si deseásemos cambiar de un protocolo como *Zigbee* a otro como Bluetooth por ejemplo. La filosofía es siempre la misma: modularidad y capas intercambiables para obtener una gran flexibilidad y adaptabilidad para distintos entornos, pero con una misma estructura global para obtener una plataforma robusta y a la vez sencilla. Siguiendo con esta filosofía una serie de bibliotecas han sido desarrolladas para el control a bajo de nivel de los nodos, así como perfiles de aplicación que permiten gestionar la red, tanto desde el punto de vista de la interacción entre usuario y aplicación, como desde la perspectiva de la comunicación entre aplicación y la red. La Figura 3.7 muestra una imagen de un nodo *Cookie* con sus 4 capas interconectadas.



Figura 3.7: Imagen de un nodo Cookie donde se pueden visualizar las cuatro capas principales acopladas

3.2.2. Las distintas capas

En este apartado se va a hablar brevemente sobre la función y estructura de cada una de las capas principales presentes en el nodo inalámbrico.

Capa de procesamiento

Al igual que antes se ha mencionado que Hexiwear cuenta con dos microcontroladores que actúan como los cerebros del dispositivo, las *Cookies* cuentan también con dos elementos de procesamiento: microcontrolador y una FPGA¹ que realizan tareas equivalentes para esta plataforma. Ambos se encuentran en la capa de procesamiento del nodo, que se encarga de procesar toda la información del resto de las capas, recolectar los valores de las magnitudes físicas provenientes de los sensores, gestionar el envío de esta información a la capa de comunicaciones para su transmisión a través de la red, y de tomar decisiones como por ejemplo aquellas que tienen que ver con el ahorro de energía. De estos dos elementos, el microcontrolador funciona como núcleo principal de procesamiento de información, y se encarga de capturar los datos de los sensores analógicos a través del convertidor analógico-digital que incorpora, controlar el transporte de estos datos a la capa de comunicaciones para transmitirlos a través de la red inalámbrica, y de poner en marcha posibles actuadores que estén presentes en el nodo. La FPGA se encarga de obtener datos de los sensores digitales y del envío de esta información al uC, actuando como una unidad de procesamiento secundaria, del procesamiento de las señales digitales y de la implementación de bloques de procesamiento dedicados.

¹Una FPGA (del inglés Field Programmable Gate Array) es un dispositivo programable que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada *in situ* mediante un lenguaje de descripción especializado. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas integrados en un mismo chip

Capítulo 3. Plataformas utilizadas

El microcontrolador es un *ADuC841* de *Analog Devices* [17] con 62KB de memoria flash de programa, 4KB de memoria flash de datos y 2KB de memoria RAM. Cuenta además con una UART, interfaces SPI² e I2C, 4 puertos de entrada/salida de propósito general y un ADC de 12 bits. La FPGA es una Spartan 3 XC3S200 de Xilinx con 200000 puertas equivalentes. La Figura 3.8 muestra una imagen de la capa de procesamiento de un nodo *Cookie*.



Figura 3.8: Capa de procesamiento de un nodo Cookie

Capa de comunicaciones

Esta capa es la encargada de hacer que el nodo sensor pueda comunicarse con otros nodos de forma inalámbrica e intercambiar información dentro de la red. Para que esto ocurra, en esta capa se incluye el módulo que implementa el protocolo de comunicaciones que demande la aplicación. Para redes de sensores inalámbricas, la mayoría de las veces dicho protocolo es el *Zigbee*, por lo que la plataforma Cookies cuenta con una capa de comunicaciones con un módulo que implementa dicha especificación y que provee de una interfaz simple e intuitiva para comunicarse con la capa de procesamiento, mediante comandos AT que se envían a este módulo vía UART, y que permiten obtener información tanto del módulo que implementa el protocolo como de la PAN a la que pertenece. Este módulo se trata de un ETRX2 de *Telegesis* [18], que incluye un uC que implementa la pila del *Zigbee* y la electrónica necesaria para enviar y recibir señales de forma inalámbrica. Como se ha mencionado arriba, este módulo se comunica con la capa de procesamiento mediante UART, y a demás cuenta con diversos registros que permiten la configuración de distintos parámetros de la comunicación.

²SPI o Serial Peripheral interface es un estándar de comunicaciones usado para la transferencia de información entre circuitos electrónicos integrados. Al igual que en el I2C también se definen maestros y esclavos, pero difiere de este en que se utilizan mas líneas para la transmisión de los datos: La línea de reloj (SCLK), que es común a todos los dispositivos, líneas para la comunicación de cada esclavo con el maestro (MISO Y MOSI) y una línea de selección de esclavo (SS)

A parte de esta capa de comunicaciones existen otras capas que implementan otros protocolos de comunicación, como por ejemplo el Bluetooth o *Wi-Fi*, pero para este o proyecto se ha utilizado exclusivamente la capa que implementa el IEEE 802.15.4 (a través de la inclusión del transceptor de radio CC2520 [19]), ya que como se ha mencionado previamente este es el estándar más utilizado a día de hoy en el campo de las redes de sensores inalámbricas, y a su vez es compatible con Hexiwear. Si existiesen perfiles de Zigbee de *Freescale* para el Hexiwear habría sido posible implementarlos por encima de las capas física y MAC aportadas por el IEEE 802.15.4 y se podría haber usado la capa Zigbee de los nodos *Cookies*.

La Figuras 3.9 muestra dos capas de comunicaciones que implementan distintos protocolos : Zigbee en la parte superior de la imagen y Bluetooth en la inferior.



Figura 3.9: Capas de comunicaciones que implementan Zigbee (arriba) y Bluetooth (abajo)

Capa de alimentación

Esta capa es la encargada de alimentar al nodo inalámbrico para que pueda llevar a cabo sus funciones de sensado, procesamiento y comunicación. Para ello se generan tensiones de 3.3V, 2.5V y 1.2V, valores que cubren el rango de tensiones de alimentación de una gran cantidad de los dispositivos incluidos en el *hardware* de la plataforma.

Capítulo 3. Plataformas utilizadas

Existen versiones de esta capa que se alimentan con baterías, mediante conexiones USB que además cuentan con un chip FTDI que permite realizar tareas de depuración del código en caso de que la conexión sea con un PC, o versiones que usan pilas para alimentar el módulo. Esta capa cuenta además con la circuitería asociada a la monitorización del consumo de energía del nodo. Para ello utiliza una resistencia *shunt*³ cuya señal es transmitida hasta el uC, que se encarga de tomar decisiones en función de dicho consumo. La Figura 3.10 muestra una de estas posibles capas de alimentación.

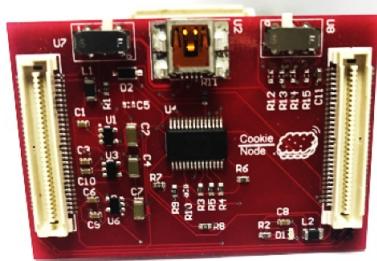


Figura 3.10: Imagen de una capa de alimentación mediante conexión USB

Capa de sensado

Esta capa contiene los sensores que se encargan de recoger los valores de distintos parámetros físicos del entorno como presión, temperatura o humedad para posteriormente enviarlos a la capa de procesamiento, que los trata y envía en a través de la red inalámbrica si así se desea. Dependiendo de si la interfaz del sensor es analógica o digital, los datos recogidos viajan por el bus analógico o digital de la *Cookie*. Esta capa también cuenta con los circuitos de acondicionamiento necesarios para transformar las lecturas de los distintos sensores en información que sea comprensible tanto por el microcontrolador como por la FPGA contenidas en la capa de procesamiento.

La modularidad de la plataforma global, unida a la gran variedad de capas de sensado diseñadas, permite que las Cookies puedan adaptarse a aplicaciones que requieran recolección de magnitudes físicas muy distintas, siendo solo necesario intercambiar la capa de sensado en función de las necesidades de la aplicación. Algunas de las distintas capas de sensado desarrolladas son:

- Una capa con sensores de temperatura y humedad (sensor SHT11) y un LDR⁴ (Light dependent resistor) para medir la cantidad de luz ambiente.

³En electrónica, un shunt es una carga resistiva a través de la cual se deriva una corriente eléctrica. Generalmente la resistencia de un shunt es conocida con precisión y es utilizada para determinar la intensidad de corriente que fluye a través de esta carga, mediante la medición de la diferencia de tensión o voltaje a través de ella utilizando la ley de Ohm ($I = V/R$).

⁴Un LDR (del inglés Light dependent resistor) es un componente electrónico cuya resistencia disminuye con el aumento de la intensidad de la luz ambiente

3.2. Plataforma Cookies

- Una capa con un acelerómetro de dos ejes (ADXL213 de *Analog Devices*) , con interfaz PWM y un sensor de temperatura.
- Una capa que utiliza una galga extensiométrica para la medida de deformaciones, basada en el efecto piezorresistivo⁵ para medir dichas deformaciones.
- Capas con sensores que permiten medir el pH de una sustancia determinada, y los niveles de concentración en el aire ciertos gases como CO o CO2.

En la Figura 3.11 se muestran dos de estas posibles capas de sensado: La capa que contiene sensores de temperatura, humedad y luz ambiente (arriba) y la capa que permite medir deformaciones utilizando una galga extensiométrica (abajo)

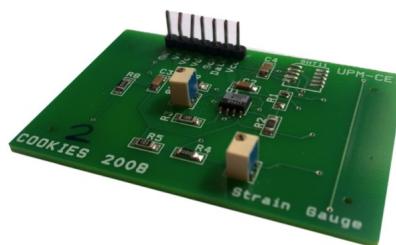


Figura 3.11: Dos posibles capas de sensado para los nodos Cookie: Temperatura, humedad, y luz (arriba) y galga extensiométrica (abajo)

⁵La piezorresistividad es la propiedad de ciertos materiales de cambiar su resistencia eléctrica cuando son sometidos a algún esfuerzo o estrés mecánico.

Capítulo

4

Arquitectura, verificación funcional y interoperabilidad

En los capítulos anteriores se ha hablado por separado de los dos elementos básicos que componen la arquitectura propuesta en este trabajo, el Hexiwear y las *Cookies*. En este capítulo se va a exponer la arquitectura global conseguida con la integración de un dispositivo inteligente como el Hexiwear y unos nodos para redes de sensores inalámbricas con las capacidades de personalización y de adaptabilidad al entorno de aplicación de las *Cookies*. También se detallaran las pruebas realizadas para garantizar que estas dos plataformas son capaces de comunicarse entre sí utilizando el IEEE 802.15.4

4.1. Arquitectura

Hexiwear, por un lado utiliza dos núcleos de procesamiento altamente eficientes, que permiten separar las tareas de gestión del dispositivo en cuanto a interfaz de usuario, gestión energética, y adquisición de datos; de las comunicaciones, implementando el primer lote de tareas en un uC, y la comunicación inalámbrica en otro. Como se ha mencionado previamente, ambos uCs se comunican mediante UART, por lo que a continuación se va a detallar cómo es exactamente esta comunicación, ya que su implementación dentro del código de ambos procesadores está realizada a muy alto nivel, abstrayendo al programador de tener que configurar parámetros específicos de dicha UART o de la necesidad de utilizar *buffers* especializados. Como se mencionó en el capítulo 3 el código que implementa la comunicación entre ambos microcontroladores se encuentra dentro de la carpeta *intf* de los proyectos de Hexiwear para cada uno de sus MCUs. Dichas carpetas cuentan con tres archivos

principales, que implementan la recepción y el envío de paquetes, además de algunas funcionalidades generales de la comunicación entre los dos núcleos. Estos tres archivos son los siguientes:

- **host_mcu_interface.c y .h**: Estos archivos definen como es la estructura de los paquetes que se intercambian entre los dos microcontroladores. También definen los tipos de paquetes que se pueden utilizar. La Figura 4.1 muestra el código que implementa dicha estructura.

```
typedef struct
{
    uint8_t start1;
    uint8_t start2;
    hostInterface_packetType_t type;
    uint8_t length;
    uint8_t data[gHostInterface_dataSize + 1];

} hostInterface_packet_t;
```

Figura 4.1: Código que implementa la estructura de los paquetes de que transmiten por UART entre los dos microcontroladores del Hexiwear.

Como se visualiza en esta imagen, los paquetes cuentan con dos campos de inicio, un campo que define el tipo de paquete, y por último un campo de longitud y otro de datos. La definición del tipo de el paquete es muy importante, ya que es lo primero que se evaluará al llegar dicho paquete a cualquiera de los dos microcontroladores, y en función de dicho tipo se llevarán a cabo unas acciones u otras. El *firmware* inicial de los proyectos de ambos micros implementa distintos tipos de paquetes como paquetes de tipo dato del sensor de ritmo cardiaco, tipo versión, tipo bajo consumo, o tipo notificación, que dan respaldo a las aplicaciones nativas de Hexiwear. Para el desarrollo de las aplicaciones que hacen uso del IEEE 802.15.4 se han creado nuevos tipos de paquetes entre los cuales están tipo *PAN_Init*, tipo *beacon*, tipo *sendZbMessage*, o tipo *ZbData*, tanto en el MK64 como en el KW40Z con el fin de implementar las funcionalidades de estas nuevas aplicaciones.

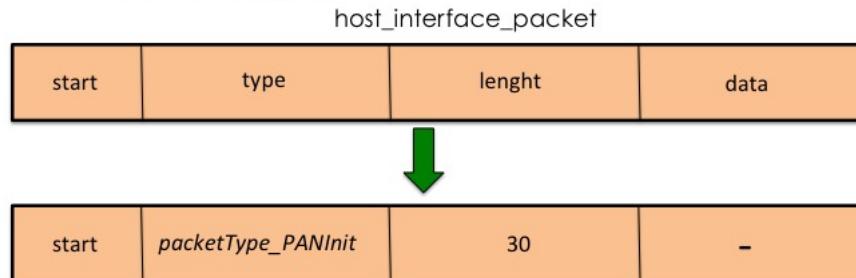
- **host_mcu_interface_rx.c**: Este archivo implementa el código que se encarga de gestionar la recepción de paquetes vía UART para ambos microcontroladores. Esta gestión esta basada en un *packethandler*, que no es más que un *switch* que evalúa el tipo de paquete, y en función de éste lleva a cabo determinadas acciones. En el caso de un paquete tipo notificación por ejemplo,

este *handler* se encarga de despertar la pantalla del dispositivo, encender el motor haptico para producir una pequeña vibración, y dependiendo del tipo de notificación del que se trate incrementar un contador. En este *switch* se han tenido que añadir los casos asociados a los nuevos tipos de paquetes creados para las aplicaciones que hacen uso del IEEE 802.15.4 . El paquete de tipo *PAN_Init* por ejemplo, se envía del MK64 al KW40Z al pulsar un botón dentro de una pantalla de aplicación determinada, y al llegar al *handler* del KW40Z y reconocer este que se trata de un paquete de este tipo, se llama a una función que inicializa la red de área personal. De la misma forma, al iniciarse y configurarse dicha red correctamente el KW40Z devuelve el paquete con tipo *PAN_Init* al MK64, cuyo *handler*, al recibir un paquete de este tipo imprime por pantalla distintos parámetros relacionados con la configuración de la red, como el canal escogido para la misma o los IDs de red y coordinador.

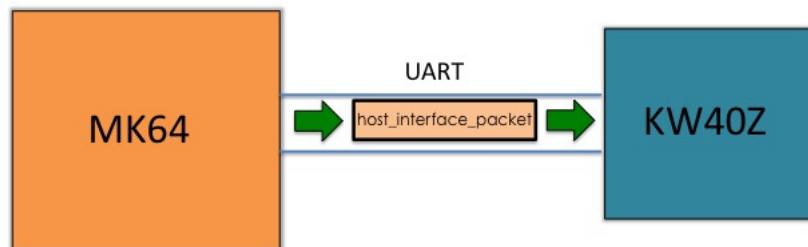
- **host_mcu_interface_tx.c:** Por último, en este archivo se implementa el código encargado de enviar mensajes a través de la UART. Define la función *HostInterface_FlushPacket*, que se encarga de enviar un paquete *host_interface_packet_t* a través de la UART de uno de los dos núcleos del Hexiwear al otro. Esto se hace siempre a un nivel suficientemente alto como para que el desarrollador no tenga que preocuparse por las peculiaridades de la comunicación por UART. El procedimiento para enviar un paquete de cualquiera de las dos entidades de procesamiento a la otra es el siguiente: Primero se crea un tipo de paquete *host_interface_packet_t* y se le asigna un nombre. A continuación se le puede dar una longitud determinada y asignar un valor al campo de datos que contiene, aunque ninguna de estas dos acciones es imprescindible. La única parte estrictamente necesaria del proceso de creación de uno de estos paquetes es la asignación de un tipo determinado al paquete, ya que si este campo se deja vacío, al llegar dicho paquete al microcontrolador de destino, este no sabrá como actuar y por consiguiente el paquete será descartado.

Esta comunicación mediante UART entre los dos uCs de Hexiwear es una parte fundamental de la arquitectura que resulta de su integración con los nodos *Cookie*, por lo que su correcta gestión es de suma importancia. Las siguientes imágenes muestran la secuencia de pasos que constituyen este proceso de comunicación. Primero se define el paquete que se va a enviar desde cualquiera de los dos microcontroladores. En este paso se determina el tipo de paquete a enviar, y se le adjuntan datos en caso de desecharse. Posteriormente se envía el paquete en cuestión, siendo este recibido por el *handler* del microcontrolador al que haya sido enviado. Este *handler* evalúa el tipo de paquete entre los tipos definidos en el archivo *host_mcu_interface.h* y comienza la realización de unas tareas u otras en función de este tipo.

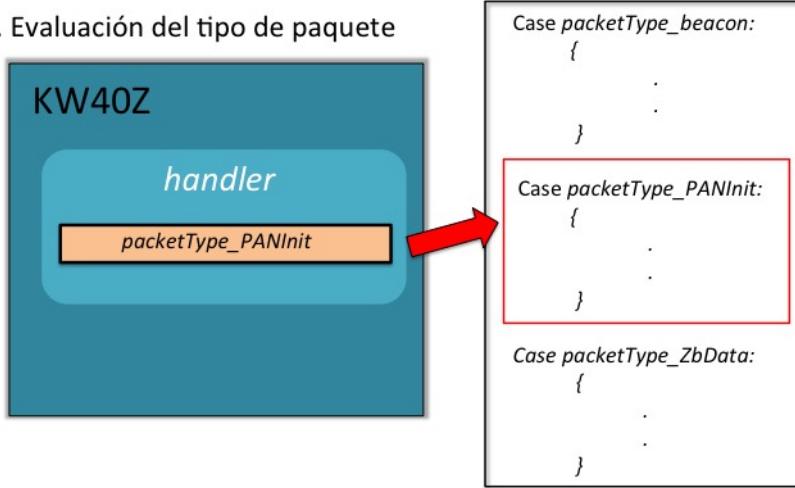
1. Definición del paquete



2. Envío del paquete



3. Evaluación del tipo de paquete



4. Reacción en función del tipo de paquete:

- packetType_PANInit* → Elección del canal de la PAN.
- Elección del PANID.
- Configuración del coordinador de la PAN.

Figura 4.2: Secuencia de pasos de la comunicación via UART de los dos uCs de Hexiwear

Otra de las características a resaltar de la arquitectura propuesta es el uso del sistema operativo en tiempo real *FreeRTOS* en el núcleo principal del Hexiwear. Como se introdujo en el Capítulo 3 este tipo de sistemas operativos contribuyen a optimizar los recursos de los dispositivos que los implementan y a mejorar su reacción a tiempo real ante determinados eventos. Los RTOS se caracterizan por presentar requisitos especiales en las siguientes áreas:

- **Determinismo:** Un sistema operativo es determinista si realiza las acciones en instantes fijos y predeterminados de tiempo. Cuando existen varios procesos que utilizan los mismos recursos la eficiencia global del sistema depende de la rapidez con la que dicho sistema es capaz de gestionar las peticiones de los distintos procesos en cuanto a los recursos a utilizar. En los sistemas operativos a tiempo real este tiempo es considerablemente menor que en un sistema operativo convencional. Esto significa una rápida y eficiente asignación de los recursos con los que cuenta el uC en que el RTOS se implementa.
- **Sensibilidad:** La sensibilidad hace referencia al tiempo que se tarda en dar servicio a una interrupción una vez esta ha sido reconocida. En general la sensibilidad de los sistemas operativos en tiempo real es muy elevada, por lo que no suele existir retardo entre el momento en el que se detecta una interrupción y el comienzo de su tratamiento.
- **Control de usuario:** Los sistemas operativos en tiempo real permiten a los desarrolladores y usuarios asignar prioridades a los distintos procesos que se ejecutan, tomar decisiones sobre el algoritmo de planificación y configurar ciertos aspectos de la memoria, como que procesos deben residir siempre en ella.
- **Fiabilidad:** Los sistemas operativos a tiempo real controlan todos los sucesos que tienen lugar en su entorno y en su propia escala de tiempos, lo que le confiere un mayor conocimiento sobre perdidas o degradaciones en el sistema.
- **Tolerancia a fallos:** Los sistemas operativos a tiempo real están diseñados con una alta tolerancia a fallos en los mismos, y deben estar preparados para responder a estos de forma dinámica, intentando resolverlos o minimizar sus efectos en el caso de no poder corregirlos completamente.

Gracias a este RTOS las distintas tareas de las que se ocupa el MK64 como la adquisición de datos de los sensores, la interacción con la interfaz de usuario, o la gestión de la batería, pueden ser controladas en hilos de procesamiento separados, optimizando el uso del sistema. La comunicación entre el MK64 y el KW40Z que se ha descrito en los párrafos previos también se gestiona en un hilo de procesamiento independiente del resto, lo que facilita la tarea de controlar dicha comunicación y de solucionar posibles problemas que puedan surgir en ella.

4.1. Arquitectura

En las *Cookies*, la capa de procesamiento representa el núcleo de la plataforma modular, ya que se encarga de controlar el resto de las capas, recogiendo datos del entorno mediante los sensores, controlando las conexiones de red y la transmisión de información a través de esta utilizando las capacidades inalámbricas de la capa de comunicaciones, y de configurar el modo de eficiencia energética del nodo. Al igual que el Hexiwear, las *Cookies* cuentan con dos núcleos separados de procesamiento, pero en el caso de estas uno de ellos se trata de un elemento hardware reconfigurable en la forma de una FPGA que se encarga de controlar las interfaces digitales, los protocolos que implementan dichas interfaces, y de el intercambio de información con el uC. A su vez, la capa de comunicaciones se basa en el módulo CC2520 de *Texas Instruments* [19] que implementa las capas física y MAC del IEEE 802.15.4 y se encarga de controlar la recepción y transmisión de datos de forma inalámbrica a través de la red de área personal. En el lado del *software*, continuando con la filosofía modular que caracteriza a las *Cookies*, estas cuentan con un set de bibliotecas desarrolladas [20] como un framework para el control del nodo a bajo nivel. También se han desarrollado una serie de perfiles de aplicación para lograr una interacción transparente a alto nivel, tanto a nivel de aplicación como a nivel de red.

Combinando estas dos plataformas de prototipado la modularidad y flexibilidad de las *Cookies* se combina con las características de dispositivo inteligente del Hexiwear, con su interfaz gráfica de usuario, dos núcleos de procesamiento, y capacidades de personalización, para crear una arquitectura capaz de dar soporte a una gran variedad de entornos de aplicación. La Figura 4.2 muestra un esquema del aspecto global a bajo nivel de la arquitectura propuesta en el presente trabajo, con sus distintos módulos: Los procesadores maestro y esclavo del Hexiwear, el módulo de comunicación del nodo *Cookie*, y ambos componentes de la capa de procesamiento: El microcontrolador de bajo consumo y la FPGA.

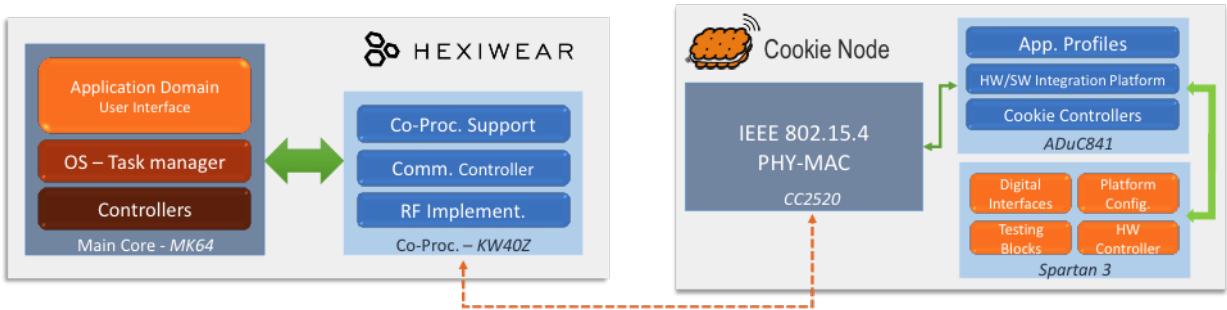


Figura 4.3: Esquema global de la arquitectura propuesta en el presente trabajo, compuesta por Hexiwear y un nodo *Cookie*

A alto nivel la arquitectura propuesta funciona de la siguiente manera: Desde la interfaz de usuario del Hexiwear, distintos aspectos de la red son controlados, como la gestión del comienzo de dicha red o el envío y recepción de mensajes de los nodos

Cookies. El principal núcleo de procesamiento del Hexiwear registra la interacción por parte del usuario y envía el mensaje correspondiente al uC que implementa el IEEE 802.15.4, que lo transmite a través de la red. Ambos microcontroladores se comunican vía UART según lo especificado previamente, y las tareas de control y comunicación asociadas se activan utilizando una estructura de hilos de procesamiento a través del sistema operativo embebido, consiguiendo una eficiente gestión de los recursos *hardware* del sistema. La información enviada por la aplicación de alto nivel viaja desde el Hexiwear de forma inalámbrica a la *Cookie*, donde es recibida por el módulo de IEEE 802.15.4 de la capa de comunicaciones y compartida con el uC de la capa de procesamiento vía UART otra vez, para que este la procese y empiece las tareas colaborativas deseadas. La Figura 4.3 muestra una imagen de los dos elementos en los que se basa esta arquitectura, el Hexiwear y los nodos *Cookies*.



Figura 4.4: Imagen del Hexiwear y dos nodos *Cookies*, elementos que componen la arquitectura descrita en este capítulo.

Esta arquitectura, con el RTOS de Hexiwear por un lado y los SKDs de sus uCs, y las bibliotecas modulares de las Cookies por otro, ha dado lugar a la creación de un *framework* conjunto para el desarrollo transparente de aplicaciones para el IoT, en el que cada plataforma puede programarse de forma sencilla y en función de las necesidades de la aplicación determinada.

4.2. Interoperabilidad y verificación funcional de las comunicaciones

A continuación se detallan las pruebas realizadas para garantizar la capacidad de comunicación de entre los dispositivos que construyen la arquitectura introducida en la anterior sección.

4.2.1. Objetivos

El objetivo de estas primeras pruebas fue conseguir comunicar exitosamente el Hexiwear con las *Cookies*, ya que la comunicación entre estas dos plataformas es la base sobre la cual se sustenta este proyecto. Tanto las *Cookies* como el Hexiwear incorporan la electrónica necesaria para comunicarse utilizando el IEEE 802.15.4, estándar para redes de sensores inalámbricas del que se ha hablado en el anterior capítulo. Sin embargo, por mucho que ambos contengan el hardware necesario y el software que implementa el estándar, muchas veces los fabricantes de estos dispositivos lo implementan a su manera, haciendo que a pesar de que supuestamente dichos dispositivos sean compatibles con él, al final sea imposible conseguir que estos se comuniquen. Estas incompatibilidades suelen deberse a alteraciones a nivel de implementación hardware o customizaciones a nivel software por parte del fabricante que hacen que existan limitaciones a nivel de compatibilidad con dispositivos terceros. En esta primera fase se realizaron pruebas para comprobar si el Hexiwear y las Cookies se podían comunicar, manipulando el *firmware* del uC encargado de las comunicaciones del Hexiwear para programarlo como el coordinador de una PAN de IEEE 802.15.4 y preparando una *Cookie* para enviar mensajes a dicho coordinador con el objetivo de ver si estos se recibían correctamente.

4.2.2. Metodología y configuraciones

Como se ha mencionado en el apartado anterior, la primera parte de estas pruebas consistía en programar el Hexiwear para que crease una red de área personal de IEEE 802.15.4. El *firmware* que este dispositivo lleva por defecto incorpora un código que trabaja conjuntamente con el que trae el otro microcontrolador (el MK64) y que se basa en el uso de Bluetooth para enviar las medidas de los sensores del dispositivo a un Smartphone con el que esté emparejado. Cabía la posibilidad de sobrescribir el *firmware* por defecto de ambos microcontroladores y desarrollar una aplicación específica que cubriese las necesidades de este proyecto, pero se decidió conservar las aplicaciones con las que el Hexiwear ya contaba (reloj, estación meteorológica, linterna, medidor de pasos, aplicación de movimiento, y sensor de ritmo cardíaco), para seguir conservando las funcionalidades de *smartwatch* inteligente que caracterizan al dispositivo. La Figura 4.4 muestra dos de las aplicaciones con las que viene Hexiwear de fábrica y que habrían sido sobrescritas en caso de implementar una aplicación de propósito único.



Figura 4.5: Aplicación de linterna y pantalla principal del Hexiwear

Para poder utilizar el IEEE 802.15.4 en el Hexiwear, se ha modificado principalmente el código del uC de las comunicaciones, el KW40Z, para incorporar la funcionalidad de Coordinador PAN. Este código se ha manipulado usando el entorno de desarrollo *IAR Embedded Studio for ARM*, ya que el proyecto inicial del KW40Z fue desarrollado con él. Utilizando como guía distintos documentos que Freescale facilita para asistir en el desarrollo de aplicaciones que utilicen el IEEE 802.15.4 se fueron incluyendo bibliotecas y funciones necesarias dentro del proyecto original de Hexiwear. Primero se compiló el código con las bibliotecas que incluyen las funciones del IEEE 802.15.4 para verificar que se habían importado correctamente. Al hacer esto, el proyecto inicial de Hexiwear más estas bibliotecas superó en espacio la cantidad de memoria disponible en el KW40Z, por lo que para hacer sitio al nuevo código se eliminaron algunas de las funcionalidades relacionadas con el envío de datos desde el Hexiwear al Smartphone vía Bluetooth, pero conservando el resto de las aplicaciones originales del *wearable*.

Siguiendo uno de los códigos ejemplos que proporciona *Freescale* en el SDK para el KW40Z, *My Wireless App Coordinator*, se programó al Hexiwear para seleccionar un canal determinado de los 16 que permite el estándar, y comenzar una red de área personal en dicho canal con un PAN ID determinado y una dirección corta para si mismo. Con el propósito de comunicarse con el Hexiwear se programó una Cookie para enviar mensajes al Hexiwear dentro de la red de área personal que este había creado. Estos mensajes debían provocar un *acknowledge* por parte del dispositivo que los recibía (Hexiwear) y al llegar dicho *acknowledge* de vuelta la *Cookie* y estar esta conectada a un PC por medio del puerto serie, imprimir una determinada secuencia de caracteres por el terminal previamente configurado del ordenador.

Las configuraciones escogidas fueron:

- Canal 25

4.2. Interoperabilidad y verificación funcional de las comunicaciones

- PAN ID: 0xBEBE
- Short ID del coordinador (Hexiwear): 0x0303
- Short ID de la *Cookie*: 0x0101

La Figura 4.5 expone estas configuraciones, mostradas por la pantalla del Hexiwear cada vez que se comienza la PAN.



Figura 4.6: Configuraciones de la PAN escogidas para la prueba de comunicación, mostradas a través de la pantalla del Hexiwear.

Para gestionar el proceso de creación de la PAN, e ir visualizando mensajes durante este proceso se han tenido que incluir nuevas pantallas en el Hexiwear desde las cuales acceder a las funcionalidades asociadas con el IEEE 802.15.4. Esto se ha hecho modificando el firmware del uC principal, el MK64 desde el IDE *Kinetis Design Studio*, pero manteniendo siempre las funcionalidades con las que venia de fabrica del dispositivo, excepto la comunicación por Bluetooth, que como se ha mencionado antes fue necesaria sacrificar para poder incorporar el código que implementa el IEEE 802.15.4 en el microcontrolador encargado de las comunicaciones. En la comunidad de Hexiwear existen numerosos recursos que explican como importar imágenes al dispositivo y como crear etiquetas de texto e imprimir cadenas de caracteres. La Figura 4.6 muestra dos nuevas pantallas que se han incluido en el proyecto e importado a Hexiwear.



Figura 4.7: Pantallas *custom* de acceso a las funcionalidades del IEEE 802.15.4 (izq) y de comienzo de la red de área personal (dcha)

A parte de la creación de pantallas desde las cuales interactuar con la aplicación se ha tenido que gestionar la comunicación entre ambos uCs para que la información que envía el usuario con los botones para la gestión de los procesos desde las distintas pantallas desencadene las acciones deseadas en el uC de las comunicaciones. De la misma forma, la gestión de esta comunicación entre microcontroladores ha permitido que los paquetes entrantes por la radio del uC de las comunicaciones tengan su debido efecto en el MK64 como por ejemplo pasar de una pantalla a otra o visualizar datos que llegan forma inalámbrica, como se ha mencionado en las secciones previas de este capítulo.

4.2.3. Resultados

Tras programar ambos dispositivos de la manera mencionada en el anterior apartado se procedió a realizar las pruebas que determinarían si ambos se podían comunicar o no. Se inicio la PAN desde una de las pantallas del Hexiwear y se coloco a una distancia de unos dos metros de una *Cookie*. Dicha *Cookie* se conectó a un PC con una instancia de *Telegesis Terminal* (un programa de terminal para Windows) abierta. Tras encender la *Cookie* y configurarla se empezaron a imprimir una serie de mensajes de NO ACK, hasta que los primeros mensajes llegaron al Hexiwear y se visualizaron los primeros indicios de comunicación entre estas dos plataformas heterogéneas. La comunicación se realizó exitosamente y tras estos resultados se pudieron dar los primeros pasos hacia la el desarrollo de una serie de aplicaciones que utilizan la arquitectura que se describe en el siguiente capítulo. La Figura 4.7 muestra los mensajes visualizados al realizar la prueba descrita.

4.2. Interoperabilidad y verificación funcional de las comunicaciones

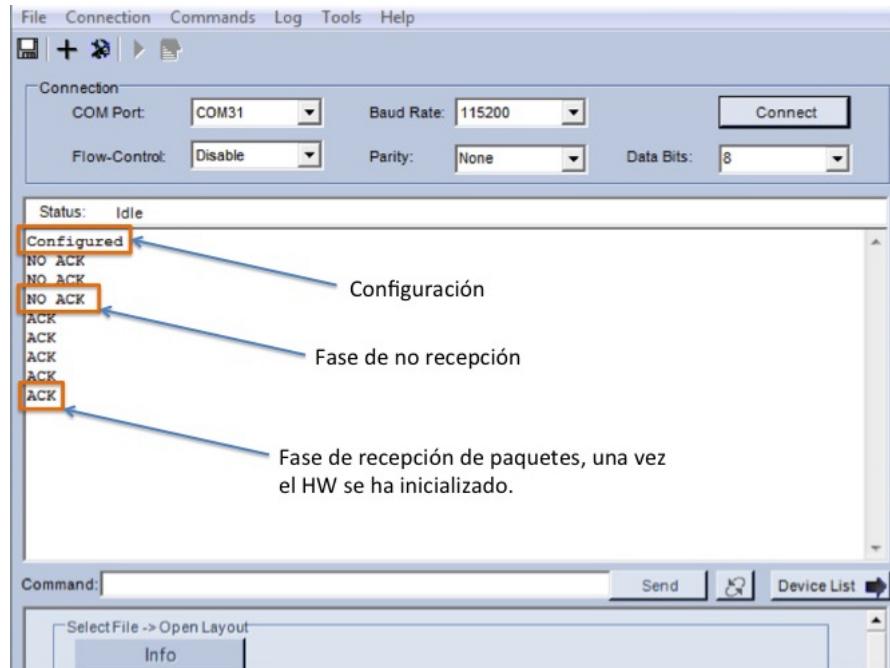


Figura 4.8: Instancia del terminal del PC al realizar la prueba de comunicación entre Hexiwear y las Cookies.

Para comprobar la robustez de esta comunicación se configuraron ambos dispositivos para enviar mensajes con distintas potencias, se personalizaron las tramas del estándar de distintas formas, y los mensajes para producir *acknowledge* y en todos los casos la comunicación fue exitosa. Los paquetes enviados entre el Hexiwear y las *Cookies* son recibidos por estos segundos dispositivos con unas calidades de enlace en su gran mayoría de 233, 234 o 235, para un máximo de 255 que es la calidad para un enlace ideal y nunca se da en la práctica. De todo esto se concluye que la comunicación entre ambos es dispositivos es posible y además que dicha comunicación posee una alta calidad, y posibilitará una interacción eficiente y configurable entre estos. A su vez, para cuantificar la calidad de la conexión entre las dos plataformas, se han tomado una serie de muestras a distintas distancias de la cantidad de paquetes totales recibidos por un nodo *Cookie* al enviar desde Hexiwear 10 tandas de 10 paquetes cada una, lo que supone un total de 100 mensajes.

Distancia (m)	1	2	3	4	5	6	7	8	9	10
Paquetes recibidos	98	100	99	98	99	97	98	97	97	94

Tabla 4.1: Tabla que muestra la cantidad de paquetes recibidos en la comunicación de Hexiwear y un nodo *Cookie* al enviar 100 mensajes desde distintas distancias.

Para estas pruebas hay que tener en cuenta que los paquetes se han enviado en *burst* o grupos de diez, para probar la recepción en condiciones extremas, lo que puede influir en la cantidad de paquetes recibidos exitosamente.

A demás de conseguir comunicar el Hexiwear con los nodos *Cookies*, también se ha conseguido comunicar dos Hexiwear, lo que expande aún mas el abanico de posibles aplicaciones, al ser posible el uso de más de un dispositivo inteligente en los despliegues que se realicen.

Capítulo

5

Localización en redes de sensores inalámbricas

5.1. Formulación del problema

Como ya se introdujo en el capítulo de estado del arte, la localización de un nodo dentro de una red de sensores inalámbrica es una cuestión de suma importancia, ya que la información que dichos nodos recopilan es completamente irrelevante si no se conoce el entorno del que ésta proviene. De la misma manera, aplicaciones mas recientes de este tipo de redes están orientadas a posicionar un objeto móvil dentro de un espacio determinado, para conocer por ejemplo en qué habitación de un edificio se encuentra, acentuando la importancia de resolver de forma precisa el problema de la localización de dichos nodos. En este capítulo se va a plantear el problema de la localización en las redes de sensores inalámbricas, y a presentar una explicación detallada de los dos parámetros de radio que se utilizan en estas redes para el proceso de estimación de la distancia, el RSSI (*Received Signal Strength Indicator*) y el LQI (*Link Quality Indicator*). Se realizará también una propuesta para resolver dicho problema de localización, y se expondrán los resultados experimentales de la correspondencia entre los parámetros mencionados anteriormente y la distancia para la arquitectura propuesta en este trabajo.

En el capítulo de estado del arte se detalló cómo funcionan los distintos algoritmos de las dos familias principales de métodos para resolver el problema de localización: *Range Free* y *Range based*. También se habló de las ventajas los primeros sobre los segundos: No necesitan realizar una conversión de parámetros de radio a distancia, salvando el problema de la poca precisión de dichos parámetros. Sin embargo, también se menciono que los algoritmos *Range Free* necesitan una red muy poblada y estable para su funcionamiento, cosa que no se suele dar en ambientes *indoor*. Es por

esto, que para las aplicaciones de localización propuestas en este trabajo, que están orientadas principalmente a despliegues dentro de edificios, se utilizará la segunda familia de métodos, los *Range based*, aunque ello suponga realizar estimaciones a partir de parámetros de radio para el cálculo de la distancia. Dichos parámetros son el Indicador de fuerza de la señal recibida o RSSI y el Indicador de la calidad de enlace o LQI. Ambos se explican a continuación.

5.2. ¿Qué son el RSSI y el LQI?

5.2.1. Received Signal Strength Indicator – RSSI

El RSSI es un indicador de la potencia con la que la radio recibe una determinada señal inalámbrica, en función a una escala de referencia de 1 mW. Teóricamente este parámetro se define como diez veces el logaritmo de la relación entre la potencia recibida y una potencia de referencia (e.g 1mW)

$$RSSI = 10 \log\left(\frac{Prx}{Pref}\right) \quad (5.1)$$

Donde Prx es la potencia recibida, y $Pref$ es la potencia de referencia de 1mW. En la práctica el RSSI es calculado por la capa física como la media de los ocho primeros símbolos recibidos en el mensaje inalámbrico. Los valores normales del RSSI suelen ir de $0dBm$ en una señal ideal (que nunca se da en la práctica) hasta unos $-95dBm$ en una señal muy débil. Valores intermedios, de unos -40 a unos $-60dBm$ indican una buena potencia recibida en el paquete inalámbrico, y a partir de unos $-80dBm$ la señal llega con la potencia mínima aceptable con la cual se puede establecer una conexión.

Para poder relacionar este parámetro con la distancia es necesario modelar primero la forma en la cual la señal pierde potencia según se propaga por el espacio. A estos modelos se los conoce como modelos de propagación de la señal, y para que se puedan aplicar es necesario que se cumplan una serie de características en la comunicación, como por ejemplo una determinada distancia mínima entre los nodos, que esta sea en interiores o exteriores, o que las antenas de los dispositivos que envían estas señales tengan un determinado tamaño. Los modelos de propagación más utilizados son:

- **Modelo de propagación en espacio libre o *Free space model*:** Este modelo es aplicable cuando: 1) La distancia de transmisión es mucho mayor que el tamaño de la antena y la longitud de la onda que transporta la señal. 2) No hay obstáculos entre emisor y receptor. Estas condiciones suelen implicar que este modelo se utilice cuando la señal se propaga grandes longitudes en exteriores

y sin obstáculos. Si se cumple esto, el modelo relaciona la potencia recibida con la distancia de la siguiente forma:

$$Pr(d) = \frac{Pt.Gt.Gr.\lambda^2}{(4\pi)^2.d^2.L} \quad (5.2)$$

Donde Gt y Gr son las ganancias de las antenas, L es el factor de pérdidas del sistema, Pt es la potencia con la que se transmite la señal inalámbrica, d es la distancia, λ es la longitud de la onda transmitida y Pr es la potencia recibida.

- *Surface bidirectional reflectance model*: Este modelo es aplicable cuando se cumplen las siguientes condiciones: 1) La distancia de transmisión d es de varios kilómetros. 2) La altura de las antenas de transmisor y receptor es de 50 metros o más. Este modelo es muy preciso cuando se usa en entornos urbanos. La relación entre la potencia recibida y la distancia viene representada por la siguiente fórmula:

$$Pr(d) = Pt.Gt.Gr.\frac{hf^2.hr^2}{d^4} \quad (5.3)$$

Donde hf y hr son las alturas de la antena emisora y receptora respectivamente, y Pt , Gt , Gr y d tienen los mismos significados que en el anterior modelo.

- *Log-normal shadowing model* (LNSM): Este es el modelo de propagación de la señal más genérico. Se puede utilizar tanto en interiores como en exteriores, y cuenta con una serie de parámetros configurables en función del entorno en el que se aplique. La relación entre potencia y distancia viene definida por la siguiente fórmula:

$$Pr(d) = Pr(d_0) + 10n\log\left(\frac{d}{d_0}\right) + X_0 \quad (5.4)$$

donde d_0 es la distancia de referencia, que suele ser de 1 metro, $Pr(d_0)$ es la potencia recibida a esta distancia de referencia, n es el índice de pérdida de la señal, que depende del entorno y X_0 es un componente aleatorio de desviamiento de la media Gaussiana del modelo. Si no se tiene en cuenta X_0 y utilizamos la conversión de potencia a RSSI de la fórmula 5.1 llegamos a la siguiente relación entre RSSI y distancia:

$$RSSI = RSSI_{(d_0)} - 10n\log\left(\frac{d}{d_0}\right) \quad (5.5)$$

Donde los parámetros n y d_0 son los descritos anteriormente, y $RSSI_{(d_0)}$ es la medida del RSSI a la distancia de referencia d_0 . Despejando de esta ecuación podemos obtener la distancia en función del RSSI:

$$d = d_0 \times 10^{\left(\frac{RSSI(d_0) - RSSI}{10n} \right)} \quad (5.6)$$

5.2.2. Link Quality Indicator – LQI

El LQI es un parámetro exclusivo de las comunicaciones que utilizan el IEEE 802.15.4 y que indica la calidad del enlace en las transmisiones de radiofrecuencia. Este parámetro, al igual que el RSSI es calculado por la capa física y aporta información de la fuerza de la señal entrante y de la calidad del enlace con el nodo del que proviene dicha señal. Esta relación entre la fuerza de la señal entrante y el LQI no es directa, pero ya que una señal fuerte se ve menos afectada por ruido e interferencias que una señal débil, dichas señales son vistas por el receptor como más 'puras' y por lo tanto resultan en valores de LQI elevados. Sin embargo, una señal débil en un ambiente sin ruido e interferencia también proporciona un alto valor de LQI, por lo que el LQI no se suele utilizar de forma directa para cuantificar la potencia de la señal inalámbrica.

Este parámetro también es un indicador de la facilidad con la que se desmodula la señal entrante. Esto es debido a que la radio lo calcula acumulando el error entre los símbolos entrantes en cada señal de radio y las constelaciones ideales, durante los 64 símbolos siguientes al SYNC. Por último, también se habla del LQI como un indicador de la probabilidad que existe de que un mensaje se reciba correctamente. El rango de valores del LQI depende del módulo que se utilice, siendo los rangos más comunes de 0 a 100 o de 0 a 255, donde el 0 indica una señal de calidad muy baja, y 100 o 255 una señal con calidad ideal.

En la Figura 5.1 se muestran una serie de gráficas que escenifican las relaciones entre la potencia de una señal, el ruido de su entorno, y los valores de RSSI y LQI de dicha señal. La gráfica de la esquina superior izquierda muestra una señal de alta potencia y sin interferencias, que por lo tanto reportará valores elevados tanto de RSSI como de LQI. La gráfica situada a su derecha muestra una señal de elevada potencia y con alguna interferencia, pero al ser la dimensión de estas interferencias mucho menor que la de la señal original, la calidad de esta no se verá casi afectada, por lo que otra vez tendremos valores elevados de RSSI y LQI. La gráfica inferior izquierda muestra una señal de potencia baja, pero sin ruido, por lo que presentará un valor de RSSI bajo y un valor de LQI elevado. Por último la gráfica de la esquina inferior derecha muestra la misma señal pero con ruido. Esta vez, al ser el ruido del mismo orden de magnitud que la señal inicial, interferirá en su proceso de demodulación, alejando la señal entrante de una señal ideal, por lo que dicha señal de presentará valores de RSSI y de LQI bajos.

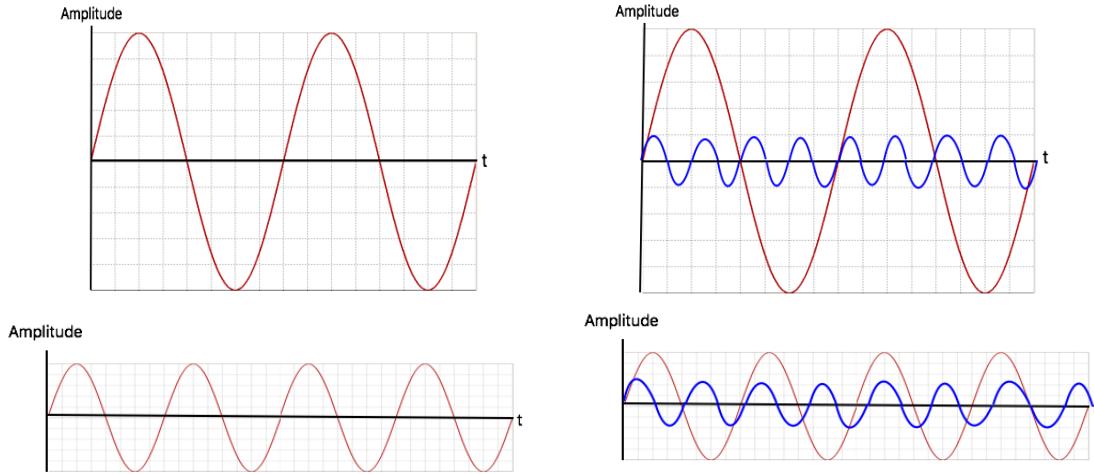


Figura 5.1: Gráficas que muestran distintas señales inalámbricas (rojo) y el ruido en su entorno (azul)

5.3. Correspondencia del RSSI con la distancia

Después de exitosamente comunicar el Hexiwear con las *Cookies*, ya que el objetivo final del sistema es ser usado en aplicaciones de localización y detección, además de en escenarios colaborativos, se llevaron a cabo una serie de experimentos para estudiar la relación entre el RSSI y la distancia en la comunicación entre estas dos plataformas heterogéneas . El procedimiento para la realización de dicha prueba fue el siguiente: Un nodo *Cookie* y un Hexiwear se colocaron a una altura de un metro del suelo, con pleno campo de visibilidad, en un ambiente interior, y al principio del experimento separados por una distancia de un metro. Tras la puesta en marcha de ambos dispositivos y el comienzo de la PAN por parte del Hexiwear, el nodo *Cookie* se programó para enviar mensajes al Hexiwear cada dos segundos, y el Hexiwear para que por cada mensaje recibido, los valores del RSSI y LQI de dicho mensaje se mostraran en la pantalla del dispositivo. Acto seguido, el Hexiwear devolvía un mensaje a la *Cookie* con los valores de RSSI y LQI registrados, para que esta, a través de una conexión a un PC los mostrase y almacenase en un archivo. Se enviaron cincuenta mensajes para cada distancia de uno a diez metros para dos potencias distintas de emisión desde la *Cookie*; -15dBm y 0 dBm. Estas pruebas se escenifican en la Figura 5.2, donde do es igual a 1 metro, $step$ tiene también un valor de 1 metro, y df son 10 metros.

5.3. Correspondencia del RSSI con la distancia

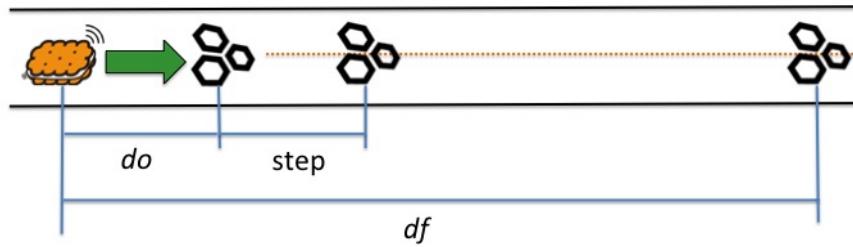
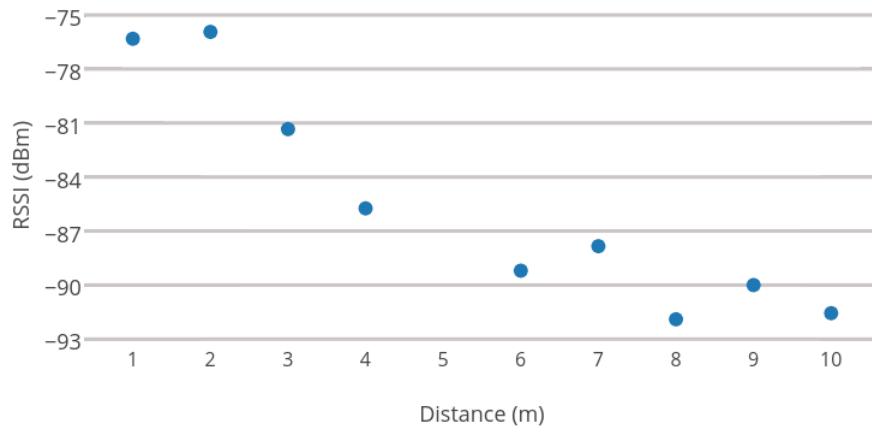
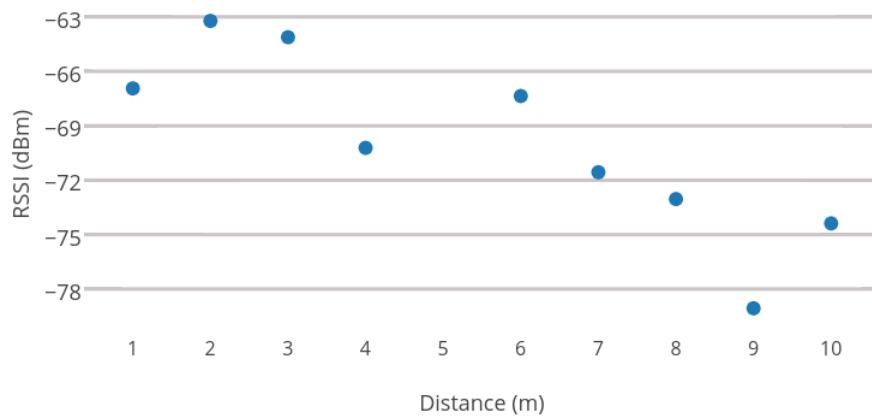


Figura 5.2: Esquema de las pruebas realizadas para la caracterización del RSSI.

Los resultados obtenidos se muestran en las Figura 5.3, donde en el eje y se muestran los valores de la media del RSSI de los cincuenta paquetes enviados para cada distancia y en el eje x la distancia asociada a dicha media.



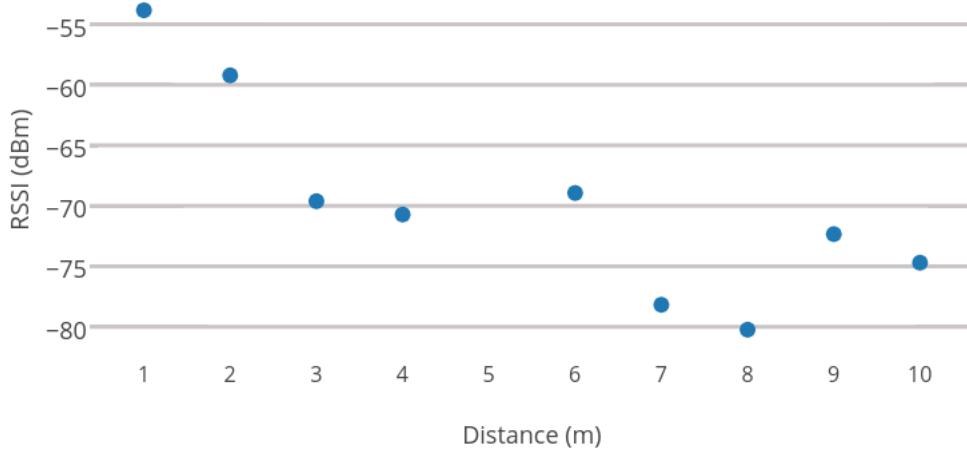
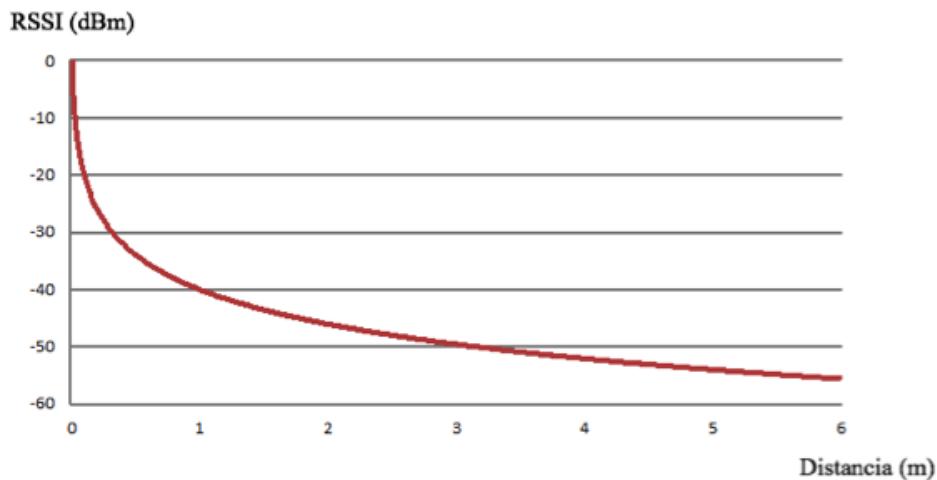


Figura 5.3: Gráficas que muestran el comportamiento del RSSI con la distancia para 0dbm (arriba) y -15dBm (centro) para mensajes enviados de un nodo *Cookie* al Hexiwear. La gráfica inferior muestra la variación del RSSI con la distancia para mensajes enviados de Hexiwear a un nodo *Cookie*.

En las tres gráficas se puede observar que pese a la reducción gradual del RSSI con la distancia, las curvas presentan picos y variaciones abruptas, por lo que de cualquier método de localización que utilizase estos valores de forma pura se vería afectado por estas inconsistencias . La Figura 5.4 muestra el comportamiento ideal del RSSI con la distancia, junto con la gráfica superior de la Figura 5.3 con la línea de tendencia dibujada en naranja.



5.4. Correspondencia del LQI con la distancia

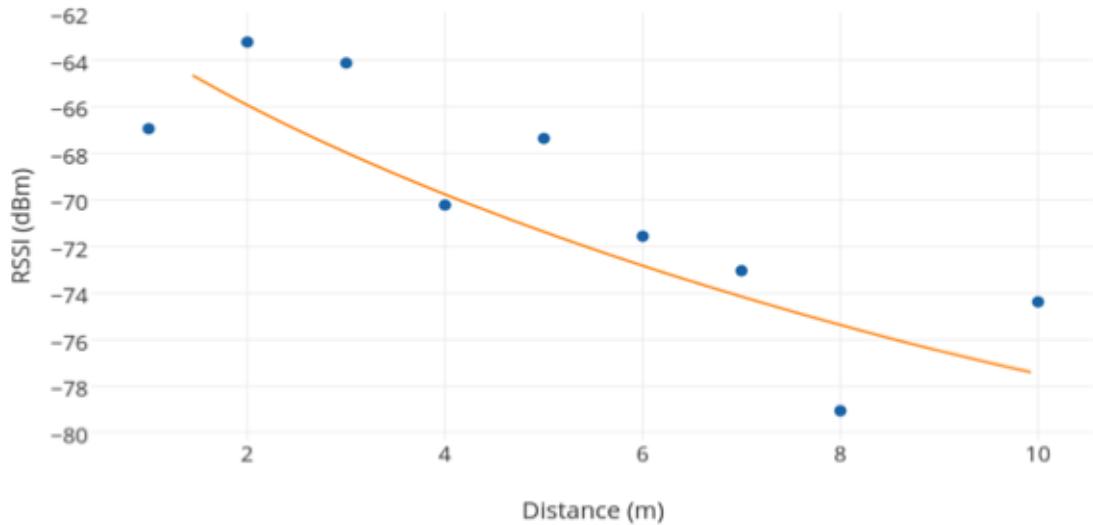


Figura 5.4: Gráficas que muestran el comportamiento ideal del RSSI con la distancia (arriba) y la línea de tendencia para la variación de este parámetro con la distancia para los mensajes enviados de la *Cookie* al Hexiwear (abajo)

Del análisis de estos resultados se concluye que para realizar estimaciones precisas de la distancia utilizando este parámetro es necesario introducir ciertos filtros en la recolección de datos, o mejoras en los modelos que eliminen los picos y espurios que muestran las gráficas expuestas. Para ello, trás la caracterización del Indicador de Calidad de Enlace o LQI con la distancia que se detalla a continuación, se propondrá un modelo que utilice este parámetro para implementar un filtro que mejore el comportamiento de las curvas expuestas en esta sección.

5.4. Correspondencia del LQI con la distancia

De la misma manera que para el RSSI, se procedió a estudiar la variación del LQI con la distancia, pero esta vez enviando los mensajes únicamente del Hexiwear a una *Cookie*, ya que como se explicará posteriormente, Hexiwear calcula el LQI a partir del RSSI, por lo que las gráficas de la variación de ambos con la distancia son idénticas pero con distintas escalas. Al igual que en el anterior experimento, se recogieron 50 muestras a cada distancia, de uno a diez metros, y posteriormente se realizó la media de dichas muestras. Los resultados se muestran en la Figura 5.5 frente al comportamiento ideal del LQI en función de la distancia.

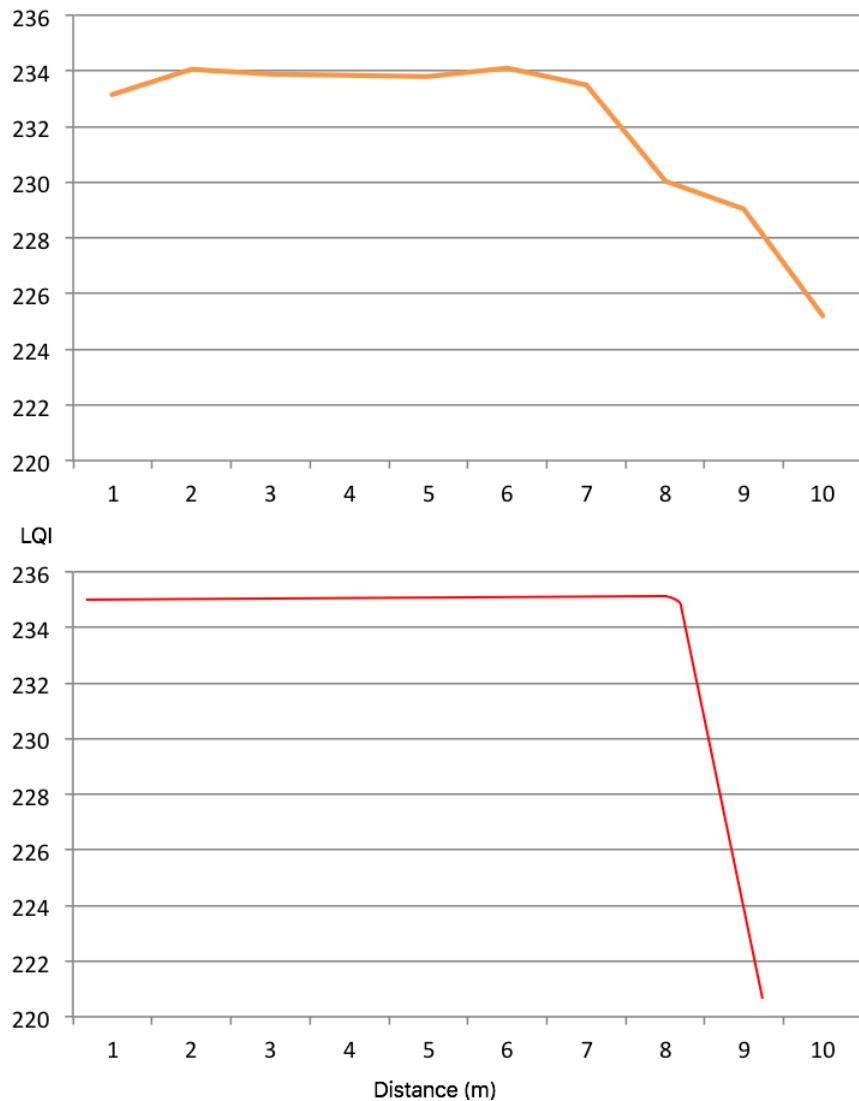


Figura 5.5: Gráficas que muestran la variación del LQI con la distancia de los mensajes enviados del Hexiwear a un nodo *Cookie* (arriba) y el comportamiento ideal de este parámetro con la distancia (abajo)

Del comportamiento ideal del LQI con la distancia se puede observar que este se mantiene estable y constante hasta cierto punto donde cae en picado debido a la pérdida de conexión de los nodos involucrados en el intercambio de mensajes. En la gráfica superior de la imagen 5.5 se puede observar que para el rango de distancias estudiado, las medias de los valores de LQI se mantienen entre 225 y 235, información que posteriormente será de gran utilidad para el diseño de la solución propuesta para mejorar la estimación de distancias utilizando el RSSI y el LQI ,que se detalla a continuación.

5.5. Solución propuesta

Una vez expuestos los resultados experimentales de la caracterización del RSSI y LQI para la arquitectura utilizada en este trabajo, se va a proponer una metodología que pretende mejorar los resultados obtenidos y reducir el error en el cálculo de distancias utilizando estos parámetros. Previamente se introducirán distintos trabajos que utilizan el RSSI, LQI, o ambos para el cálculo de distancias, y las conclusiones a las que estos llegan.

El RSSI es la métrica más utilizada en las redes de sensores inalámbricas para la estimación de la distancia que separa a los nodos que constituyen este tipo de redes, por lo que se han realizado numerosas investigaciones y estudios para determinar si este parámetro conduce a una estimación precisa de la distancia. En [21] se estudian distintos algoritmos que utilizan el RSSI para localización en ambientes interiores y se llega a la conclusión de que soluciones más complejas son necesarias para aumentar la precisión de la localización en dichos entornos. [22] estudia el comportamiento del RSSI en función de la distancia para distintos entornos ideales, y concluye que este parámetro no es fiable para distancias largas y en distintas direcciones. A pesar de ello, son numerosas las propuestas [23] y [24] que utilizan esta métrica para la estimación de distancias, llegando a una solución de compromiso entre complejidad computacional, procesamiento y precisión de los resultados. El modelo utilizado en estas propuestas es el *Log Normal Shadowing Model*, del cual se ha hablado anteriormente, con ligeras modificaciones a su versión más simple para aumentar la precisión de las estimaciones.

El LQI se utiliza menos para la estimación de distancias debido a que en los módulos de radio más populares, la relación entre LQI y potencia, que es la variable con cuyas perdidas se puede determinar la distancia entre transmisor y receptor, es indirecta. Sin embargo existen otros módulos, como el que utiliza el Hexiwear, en los que el LQI se calcula vía software, computando su valor a partir del valor del RSSI, devolviendo un parámetro con relación directa con la distancia pero que no tiene las características del LQI descrito en la anterior sección. En las bibliotecas del KW40Z que implementan el IEEE 802.15.4 la función *PhyConvertLQI* calcula el LQI a partir del RSSI de la siguiente forma:

$$LQI = \frac{RSSI + 95}{6} \times 17$$

En [25] se utiliza dicho parámetro y se llegan a estimaciones de una precisión aceptable tras realizar ciertos procesos de refinamiento al valor de LQI devuelto por la radio.

Recientemente, modelos que implementan el uso de ambos parámetros han hecho ver que fusionando las dos métricas (El RSSI y el LQI de la mayoría de módulos, que

no guarda relación directa con la potencia entrante), se pueden llegar a estimaciones de mayor precisión que con aquellos modelos que utilizan solo una de estas métricas, como en [26]. En el presente trabajo se propone una técnica mixta, que utiliza el RSSI para la estimación de la distancia pero filtrando los paquetes que se utilizan para calcular dicha distancia mediante un filtro de LQI, que descarta ciertos paquetes con alta probabilidad de tener valores de RSSI anómalos. Dicha propuesta se expone en la siguiente sección.

5.5.1. El filtro de LQI

En el proceso de toma de datos para los experimentos de las secciones 5.4 y 5.5 se recogieron mas de 2000 muestras de valores de RSSI y LQI para distintas distancias y para distintos dispositivos. Durante el análisis de este set de datos, se observó que para una distancia específica, ciertos paquetes que llegaban a la *Cookie* con valores de LQI reducidos, presentaban potencias (RSSI) muy por debajo del resto de las potencias previas observadas para esa misma distancia. La Tabla 5.1 muestra algunos ejemplos de este comportamiento a distintas distancias.

Distancia	Media RSSI	Media LQI	RSSI y LQI del paquete anómalo
1 metros	$-53dBm$	234	RSSI: $-92dBm$, LQI: 196
2 metros	$-59dBm$	233	RSSI: $-88dBm$, LQI: 204
3 metros	$-69dBm$	233	RSSI: $-91dBm$, LQI: 199
4 metros	$-70dBm$	233	RSSI: $-91dBm$, LQI: 202
5 metros	$-70dBm$	233	RSSI: $-92dBm$, LQI: 201
6 metros	$-68dBm$	233	RSSI: $-92dBm$, LQI: 195
7 metros	$-78dBm$	230	RSSI: $-89dBm$, LQI: 204
8 metros	$-80dBm$	229	RSSI: $-91dBm$, LQI: 209
9 metros	$-72dBm$	233	RSSI: $-90dBm$, LQI: 208

Tabla 5.1: Tabla que muestra los valores de RSSI y LQI de los paquetes anómalos con respecto a la media para distintas distancias.

Estos paquetes con valores de LQI reducido, se dan en un 4 % de los paquetes enviados para distancias mayores de 5 metros y en un 2 % para distancias de 1 a 5 metros. Es sencillo deducir de esta tabla que un paquete anómalo de estas características, al presentar valores de RSSI muy reducidos (una media de $-90dBm$ para los paquetes que aparecen en la tabla) afectan de forma mas significativa a las medias calculadas para distancias muy cortas (1 y 2 metros) que para distancias mayores, al alejarse más del valor medio del RSSI para dichas proximidades. Al estar esta propuesta orientada a mejorar la precisión de algoritmos de localización para interiores, donde las distancias que separan los nodos no suelen ser muy elevadas, el filtro que se ha diseñado debería reducir el error en las estimaciones de estas dis-

tancias y aumentar la precisión de los algoritmos que las utilizan.

El filtro se implementa de forma embebida en los nodos *Cookies*, que al empezar el proceso de estimación de distancia a Hexiwear envían a este 10 mensajes, cada uno de los cuales genera un mensaje de respuesta. Al llegar cada uno de estos mensajes de respuesta a la *Cookie* se pone en uso el filtro implementado, que evalúa el LQI de cada uno de ellos. Si el valor de LQI se encuentra por encima del valor preestablecido, el paquete se acepta, registrándose el valor del RSSI con el que ha llegado. En caso contrario, el paquete se descarta, enviando un nuevo mensaje a Hexiwear para que dicho paquete sea reenviado. Cuando han llegado 10 paquetes con un valor de LQI por encima del umbral establecido por el filtro, se hace la media del RSSI de dichos paquetes y se utiliza dicho valor para estimar la distancia utilizando el *Log Normal Shadowing Model*. Para cuantificar si mediante el uso de este filtro se produce alguna mejora en las estimaciones de la distancia se han realizado una serie de pruebas, en las que utilizando el *Log Normal Shadowing Model*, del que se habló en la Sección 2 de este capítulo, se han estimado las distancias para un ambiente interior, primero sin el filtro de LQI, y posteriormente utilizando dicho filtro. Estas pruebas y sus resultados se describen en la siguiente sección.

5.5.2. Resultados: Estimación de distancias utilizando el *Log Normal Shadowing Model*

El procedimiento utilizado para la estimación de distancias utilizando el *Log Normal Shadowing Model* ha sido similar al de toma de datos de RSSI y LQI, y se ha realizado en el mismo entorno. La ecuación 5.6 relaciona la potencia con la distancia para dicho modelo, donde es necesario fijar los valores de los parámetros n , $RSSI_{(do)}$, y d_0 . El parámetro n , índice de pérdida de la señal o factor de propagación se ha fijado en 1.8, valor comúnmente utilizado para ambientes interiores y empíricamente contrastado. $RSSI_{(do)}$ se ha fijado a su vez en $-54dBm$, media obtenida para los cincuenta paquetes enviados del Hexiwear a un nodo *Cookie* para una distancia de referencia de 1 metro, valor asignado a la variable d_0 . La siguiente tabla muestra los valores asignados a las variables que se utilizan en el modelo.

Parámetro	d_0	$RSSI_{(do)}$	n
Valor	1 m	$-54dBm$	1.8

Tabla 5.2: Tabla que muestra los valores de los parámetros utilizados en el *Log Normal Shadowing Model*

Una vez escogidos los valores iniciales para los parámetros que se utilizan en el LNSM, una *Cookie* se programó para que por cada diez mensajes recibidos del Hexiwear, se calculase la media del RSSI de dichos mensajes y utilizase este valor

para estimar la distancia mediante la fórmula descrita en la ecuación 5.6. Posteriormente se colocó dicha *Cookie* a un metro del Hexiwear y desde este se enviaron 100 mensajes, resultando en una cantidad total de diez estimaciones de distancia calculadas para dicha separación entre los dos dispositivos, para las cuales posteriormente también se realizó la media. Esta separación se fue incrementando en intervalos de un metro hasta una separación máxima de diez metros. Los resultados obtenidos se muestran en la tabla 5.2

Distancia (m)	1	2	3	4	5	6	7	8	9	10
Estimación (m)	1,18	1,78	2,44	4,78	3,91	7,45	8,65	5,91	11,51	13,2
Error (%)	18	11	18	19	22	24	24	26	28	32

Tabla 5.3: Tabla que muestra las distancias estimadas por el *Log Normal Shadowing Model* junto con el error en la estimación para distancias de uno a diez metros sin el uso del filtro de LQI

Este mismo procedimiento se repitió pero implementando esta vez el filtro propuesto, de tal forma que paquetes que llegasen con un LQI menor de 220 no se utilizarían en la estimación de la distancia. Este valor ha sido elegido debido a que es considerablemente menor que la más reducida de las medias de LQI obtenidas para cada distancia (225), observadas en la Figura 5.6, de tal forma que se descartan paquetes con valores de LQI que estén muy por debajo de estas medias, pero sin ser excesivamente restrictivo, lo cual tampoco interesa ya que por cada paquete que no se utiliza en el proceso de estimación de la distancia, hay que enviar un nuevo paquete que lo reemplace, aumentando la cantidad de tráfico en la red. Las estimaciones obtenidas en el cálculo de distancias utilizando el filtro de LQI propuesto se muestran en la siguiente tabla.

Distancia (m)	1	2	3	4	5	6	7	8	9	10
Estimación (m)	1,08	1,79	2,57	4,44	4,11	7,1	8,35	9,8	6,7	12,8
Error (%)	8	10	14	11	17	18,2	19,3	23,3	25,2	28

Tabla 5.4: Tabla que muestra las distancias estimadas por el *Log Normal Shadowing Model* junto con el error en la estimación para distancias de uno a diez metros utilizando el filtro de LQI

Observando las dos tablas anteriores se puede ver que el error obtenido en la estimación de las distancias es menor utilizando este filtro que sin su uso. A su vez, como se ha mencionado previamente, esta reducción en el error es más notable para distancias reducidas (de 1 a 3 metros) que para el resto de distancias, debido al mayor efecto sobre la media del RSSI que tienen los paquetes anómalos para estas distancias que para distancias mayores. La Figura 5.5 muestra una gráfica comparativa de los

5.5. Solución propuesta

errores obtenidos para la estimación de la distancia sin utilizar el filtro (rojo) y con filtro (azul), donde se puede observar como el filtro propuesto reduce el error en las estimaciones de distancias, sobre todo a distancias reducidas, y como dicho error aumenta con la distancia de separación entre los dos dispositivos. Al estar la arquitectura propuesta intencionada para la localización en entornos interiores donde las distancias son generalmente reducidas, el uso de el filtro propuesto aumentaría la precisión en los algoritmos planteados para dichos entornos.

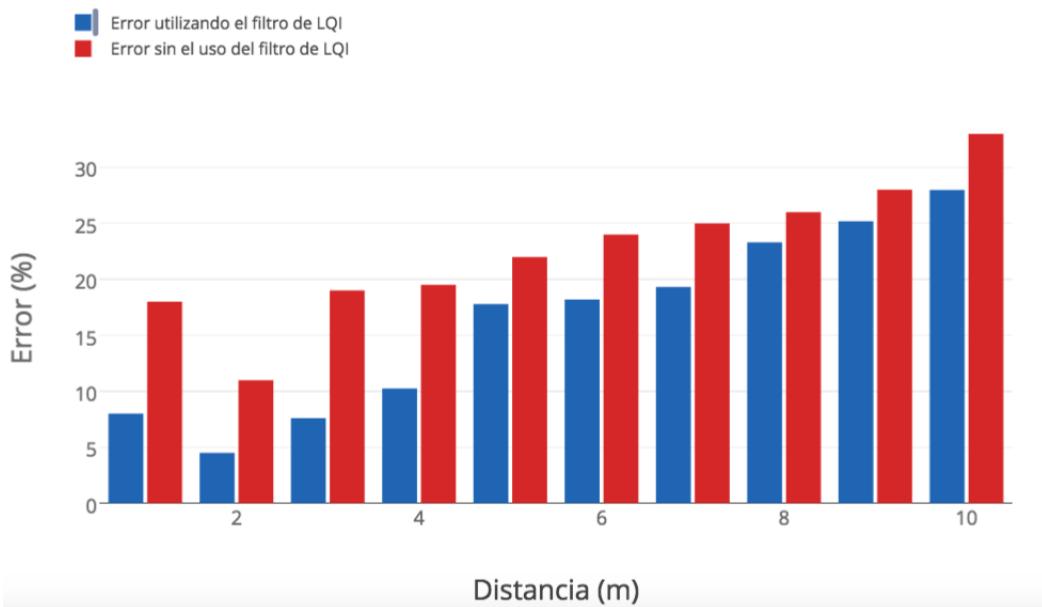


Figura 5.6: Gráfica que muestra el error obtenido en la estimación de distancias sin el uso del filtro de LQI (rojo) y con el (azul)

Ha de tenerse en cuenta que en este proceso de envío masivo de mensajes lo que se busca es caracterizar el entorno en el que se realiza el despliegue para posteriormente tunear el modelo que se utilice en función de los resultados obtenidos en esta caracterización. Esto se hace enviando un elevado número de mensajes y recopilando una gran cantidad de datos, sin embargo cuando se realice verdaderamente el despliegue en cuestión, la cantidad de mensajes enviados para la estimación de la distancia será menor, llegando a un compromiso entre precisión de los resultados y eficiencia de la red. Debido a esto, se hace aún más importante la elección de un valor umbral de LQI adecuado, ya que utilizando un valor de LQI mayor como *threshold* de este filtro es posible reducir aun más el error, pero la cantidad de mensajes que es necesario enviar aumenta según se incrementa el valor de dicho *threshold*. Modificando el valor de dicho filtro es posible llegar a una solución de compromiso entre eficiencia y precisión. Aún así, el error en la estimación de la distancia es imposible de eliminar completamente, aunque se fije el filtro para descartar cualquier paquete que no tenga un valor de LQI igual a 235, que es el máximo valor registrado en el

proceso de toma de datos. La tabla 5.4 muestra las estimaciones que se obtendrían con el LNSM utilizando solo paquetes con valor de LQI de 235.

Distancia (metros)	1	2	3	4
Estimación (metros)	0,92	1,89	3,26	3,59
Error (%)	8	5,5	8,6	10

Tabla 5.5: Tabla que muestra las distancias estimadas por el *Log Normal Shadowing Model* usando solo paquetes con un LQI de 235

Esta tabla muestra que el error se reduce si restringimos aún más los paquetes que se utilizan en el proceso de cálculo de la distancia, sin embargo la cantidad de mensajes que es necesario enviar para realizar dicho cálculo aumenta considerablemente. El valor de 220 utilizado en este proceso de caracterización como umbral de dicho filtro ha sido obtenido experimentalmente llegando a una adecuada relación entre la reducción de error y cantidad de mensajes que es necesario reenviar.

Todo el proceso de obtención de datos, así como los experimentos realizados para la estimación de la distancia se han visto favorecidos por el uso de un dispositivo como Hexiwear, que con su movilidad y su interfaz de usuario, ha proporcionado un control total sobre la posición de dicho nodo y el momento de transmisión de los mensajes al resto de nodos. De la misma forma, el uso de un nodo móvil y el problema de localización en las redes de sensores inalámbricas están estrechamente relacionados, ya que la utilización de nodos con capacidad de desplazamiento acentúa la necesidad de resolver este problema de forma precisa, para poder conocer la posición de dichos nodos según esta varía.

Capítulo 6

Redes Colaborativas y Cooperación en Redes de Sensores Inalámbricas

6.1. Redes Colaborativas

En las redes de sensores inalámbricas muchas veces los sensores recopilan información sobre parámetros físicos de su entorno de forma independiente para después transmitir la información recopilada a una interfaz de salida que se encarga de procesarla e interpretarla. De esta forma cada nodo actúa sin tener en cuenta el comportamiento del resto de nodos de la red, lo que puede restar eficiencia a la red global, sobre todo si la red se ha planificado de tal forma que los nodos cercanos se encargan de medir un mismo parámetro físico, lo que hace que la información que transmiten sea redundante. Este tipo de redes, en las cuales los nodos se comunican directamente con el *gateway*, y no tienen acceso a la información que recolecta el resto de los nodos, se están viendo desplazadas por las denominadas redes colaborativas, en las cuales los nodos comparten información entre ellos para mejorar la eficiencia y productividad de la red. En las redes colaborativas los nodos trabajan de forma coordinada para obtener una mejor perspectiva global de su entorno, mejorando el comportamiento de la red de muchas formas. A continuación se describen algunas de estas posibles mejoras:

- Como se ha mencionado en el párrafo anterior, si los nodos conocen el tipo de información que recolectan y transmiten los nodos vecinos, dichos nodos pueden ponerse de acuerdo para en el caso de recopilar el mismo tipo de información, o de que la información que recopilan tenga algún tipo de correlación, transmitir de forma intermitente, turnándose, reduciendo de esta forma el tráfico en la red, la cantidad de información que es necesario procesar, y mejorando

la eficiencia energética global de esta.

- Muchas veces un único nodo no cuenta con la capacidad para detectar un evento por si solo, ya que estos nodos suelen contar con recursos, como ya se ha mencionado, muy limitados. La colaboración entre estos dispositivos proporciona la forma de hacer frente a estas limitaciones. Si nodos con distintas capacidades de sensado colaboran entre si, compartiendo información, distintos eventos pueden detectarse sin necesidad de que la información sea procesada en el *gateway*, reduciendo el tiempo de respuesta de la red ante estos eventos y mejorando la eficiencia global. Un ejemplo de esto podría ser en el caso de una red desplegada con la intención de monitorizar un terreno y detectar posibles incendios. En esta red existen tres tipos de nodos: Nodos que miden la temperatura, nodos que miden la cantidad de luz ambiente, y nodos que miden la cantidad de CO₂ en el aire. Los incendios serían detectados cuando ciertos nodos de la red midiesen una temperatura, y un nivel de luz ambiente por encima de un determinado *threshold*, y una cantidad de CO₂ en el aire determinada. De forma independiente una temperatura alta no tiene porque significar fuego, al igual que una cantidad de luz ambiente elevada o un nivel de CO₂ determinado. El evento debería solo detectarse si las tres medidas superan un cierto límite. Si los nodos en cuestión comparten las medidas de su sensor determinado con sus vecinos, y en cualquier momento algún nodo detecta que las tres medidas están por encima de los límites establecidos, se detectaría el evento y se enviaría un mensaje directo al *gateway*, o incluso a un nodo con alguna capacidad de actuación determinada si existe dicho nodo, para llevar a cabo las acciones necesarias de forma automática, rápida y eficiente. En este segundo caso el tiempo de reacción ante la aparición de estos eventos es menor que si el *gateway* se encarga de procesar la información y detectar el evento, ya que son los propios nodos los que se encargan de cuantificar la aparición de este sin la necesidad de centralizar la información en un único punto, lo que puede llevar asociados retardos y fallos.
- Las redes colaborativas pueden tener un gran impacto sobre la eficiencia energética de la red global, tanto por la reducción del tráfico que se ha mencionado previamente como por la posibilidad de implementar algoritmos colaborativos enfocados a reducir el consumo. En [27] se utiliza un algoritmo de cooperación entre dos tipos de nodos en una red de sensores inalámbrica para aumentar la eficiencia energética de esta: Nodos con un consumo reducido se encargan de detectar determinados eventos, y una vez detectados dichos eventos despiertan a nodos con un mayor consumo que toman las medidas y las envían, para volver a modo de bajo consumo inmediatamente después. De esta forma los nodos con mayor consumo no tienen que despertarse periódicamente para comprobar si los eventos que les involucran se han producido, sino que esperan a que nodos con un consumo mucho menor los detecten y se lo comuniquen. Existen también algoritmos de sincronización, en los cuales los nodos,

operando bajo un tiempo de reloj común (que ha de estar precisamente calibrado) se despiertan en determinados intervalos de interés, por ejemplo para que un nodo que envía información, y el nodo al que dicha información va dirigida abandonen el modo *sleep* simultáneamente y tras el intercambio de datos vuelvan a dormirse.

- La colaboración entre los nodos también puede dar lugar a algoritmos y mecanismos de rutado más eficientes para redes colaborativas que los algoritmos utilizados en las redes de sensores tradicionales, como en [28]. Donde utilizando algoritmos de rutado *multi-path* se reenvían los mensajes cooperativamente entre los nodos para mejorar la fiabilidad de la transmisión y distribuir la carga de trabajo equitativamente. Los nodos de la red pueden comunicarse entre sí para notificar a sus vecinos más cercanos caminos con alta probabilidad de error en el envío de mensajes o con una calidad de enlace entre nodos pobre, para evitar que se envíen mensajes a través de dichas rutas, reduciendo así el tráfico innecesario de datos en la red y mejorando la eficiencia energética de ésta.

Todas estas virtudes de las redes colaborativas hacen que el campo de la cooperación en las redes de sensores inalámbricas esté despertando un gran interés, dando lugar a una amplia investigación con el objetivo de maximizar sus ventajas. La integración de un dispositivo inteligente como Hexiwear dentro de una red de sensores inalámbrica potencia la capacidad colaborativa de la misma, otorgándole capacidades de interacción, respuesta y dinamismo muy por encima de las que tendría en caso de no contar con tal dispositivo. Debido a esto, se evalúa la arquitectura propuesta mediante una serie de aplicaciones colaborativas que se describen a continuación.

6.2. Implementación de colaboración para la detección de incendios.

Con el objetivo de demostrar el amplio abanico de posibles aplicaciones a las que la arquitectura propuesta puede dar soporte, se ha decidido implementar una aplicación colaborativa de detección de fuegos como la que se introdujo en el anterior apartado, en la que se pondrá a prueba el tiempo de respuesta del sistema formado por Hexiwear y las *Cookies* y se resaltarán algunas de las ventajas resultantes de su integración.

Esta aplicación se basa en el despliegue de 3 nodos *Cookies* fijos, y un nodo móvil Hexiwear que acompaña al usuario en todo momento. Los nodos de las redes de sensores inalámbricas muchas veces tienen capacidades de sensado limitadas, contando con uno o dos sensores como mucho, que les impiden detectar la aparición de un evento que requiera varios tipos de medidas para poderse cuantificar. De la misma

6.2. Implementación de colaboración para la detección de incendios.

forma, aunque un nodo determinado tenga la capacidad necesaria para detectar dicho evento, la aparición de las condiciones necesarias para la detección de este en un único nodo pueden deberse a algún agente externo, evento puntual, o a un fallo del propio nodo, por lo que siempre conviene contrastar la información obtenida con algún nodo cercano que tenga las mismas capacidades de sensado. La aplicación implementada tiene en cuenta ambos escenarios, usándose dos nodos con capacidades de sensado limitadas (Nodos 1 y 3; de los cuales el nodo 1 solo mide temperatura y el nodo 3 luz ambiente), y un nodo 2, situado entre los dos anteriores, que mide temperatura y luz ambiente, y que contrasta la información que el mismo recopila con la de los dos nodos anteriores para ver si el evento realmente ha ocurrido. La Figura 6.1 muestra un esquema de estos nodos con las medidas que recopila cada uno.



Figura 6.1: Esquema que muestra el despliegue de nodos *Cookie* con su identificador y las medidas que recopilan: Ti para temperaturas y Li para luz ambiente

Estos nodos *Cookie* toman datos de su entorno cada segundo, y si la medida que evalúan está por encima de un umbral determinado, la envían a los otros nodos de la red. De la misma forma, si pasan 5 segundos sin que un nodo haya transmitido su medida a los nodos restantes, se envía la información igualmente. Por lo tanto, existen dos *thresholds*: uno establecido por un umbral de luz o temperatura, necesario para la detección del evento de riesgo, y otro límite de tiempo, en el cual los nodos transmiten información y cuya utilidad es determinar que cada uno de los nodos de la red funciona correctamente. De esta forma, cada nodo guarda un registro de las medidas que el mismo recopila, y de las últimas medidas recibidas de los nodos restantes, aunque dichas medidas no sobrepasen los umbrales para la detección del riesgo de incendio. Así, cada nodo conoce en todo momento T1, T2, L2 y L3. La condición definida para que se detecte un evento es que para ambas temperaturas T1 y T2, se supere un cierto umbral, y igual para las cantidades de luz ambiente L2 y L3. En el momento en el que cualquiera de los tres nodos detecta que todas las variables de temperatura y luz superan los umbrales establecidos, dicho nodo envía un mensaje al Hexiwear para alertar de la presencia de fuego, provocando la

vibración de este y la aparición en la pantalla del dispositivo de una imagen orientativa. Hexiwear, desde el momento de su encendido, implementa mediante un hilo del sistema operativo a tiempo real del microcontrolador principal, el proceso que recibe mensajes del KW40Z, uC encargado de recibir las tramas del IEEE 802.15.4 enviadas por los nodos *Cookie*. De esta forma, al encontrarse Hexiwear dentro del rango de distancias que el estándar permite para sus comunicaciones, y recibir un mensaje de alerta a través de la radio, el microcontrolador encargado de las comunicaciones envía un paquete con un nuevo tipo predefinido *Firealert*, que al ser recibido por el *handler* del uC principal, provoca las acciones de vibración y visualización de la alerta mencionadas previamente. Así usuarios que trabajen en las proximidades del entorno donde se prevea riesgo de incendio son notificados de la posible aparición de este de forma rápida y sin ser necesario que la información se envie a una estación central que posteriormente comunique a los trabajadores la aparición de este evento. Este sistema a su vez, no depende de equipos centralizados como ordenadores, evitando posibles fallos en el sistema que se puedan dar por errores en estos equipos, y el elevado coste que suele acarrear su uso.

6.2.1. Metodología

Para la implementación de esta aplicación colaborativa se han ido siguiendo una serie de pasos para verificar y depurar el funcionamiento global, empezando con un sistema sencillo con dos nodos *Cookie*, hasta llegar a el sistema global formado por 3 *Cookies* y el Hexiwear. Estos pasos han sido los siguientes:

- **Paso 1:** Primero se ha realizado una implementación sencilla con dos *Cookies* en la cual cada nodo toma medidas de luz ambiente y las envía al otro nodo cada 5 segundos o cuando alguna de las dos medidas sobrepasa el umbral determinado. Así los dos nodos conocen ambas medidas de luz ambiente, de tal forma que si se supera dicho umbral de luz preestablecido en ambas, se dispara una bandera que al activarse hace que ambos nodos dejen de medir e impriman por la pantalla de un terminal de ordenador un mensaje de alerta. Previo al mensaje de alerta cada nodo imprime por pantalla las medidas conocidas de las cantidades de luz ambiente en intervalos de 5 segundos. La Figura 6.2 muestra el registro de las terminales del PC para cada una de las dos *Cookies* que intervienen en este proceso. Los valores de luz ambiente que devuelve el ADC de las *Cookies* van de 0 (oscuridad completa) a 4095 para una elevada cantidad de luz ambiente. Para la realización de estas pruebas ha sido más sencillo trabajar con umbrales de luz ambiente bajos, por lo que las *Cookies* mandarán el mensaje de alerta a Hexiwear cuando ambas medidas provenientes de los ADC de las *Cookies* estén por debajo de 1000.

6.2. Implementación de colaboración para la detección de incendios.

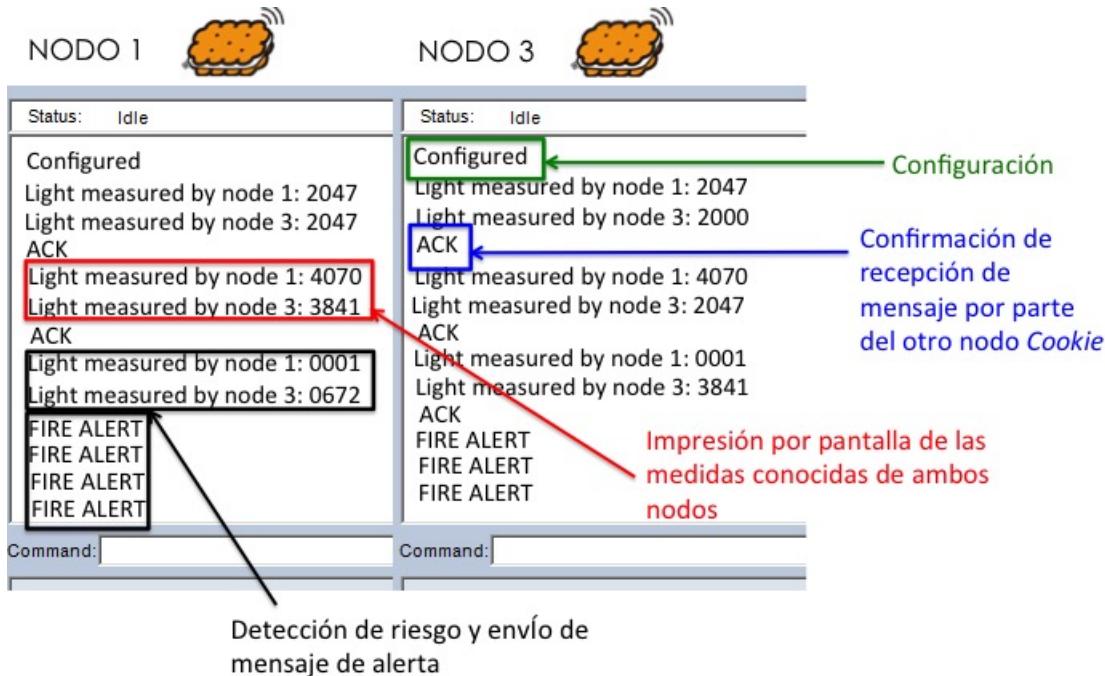


Figura 6.2: Captura de pantalla de los terminales de ordenador asociados a los nodos *Cookie* utilizados en la implementación anterior

- **Paso 2:** En este segundo paso se ha incluido al Hexiwear en el despliegue. Esta vez además de imprimir por pantalla un mensaje de alerta, ambos nodos envián un mensaje mediante la radio de IEEE 802.15.4 al Hexiwear. Cuando este mensaje llega al uC de las comunicaciones del Hexiwear, este envía al núcleo principal un paquete con un nuevo tipo predefinido *packetType_Firealert*, que al llegar al *handler* del uC principal provoca la vibración del dispositivo y la redirección a una nueva pantalla creada con un mensaje de alerta y un botón habilitado para que al pulsarlo se envíe un mensaje a ambos nodos *Cookie* para que dejen de enviar mensajes de alerta y vuelvan a tomar medidas de luz ambiente, reiniciando así el funcionamiento del sistema.

Para esta prueba se han tomado una serie de medidas del tiempo de reacción del sistema; es decir el tiempo desde que se da el evento hasta que llega el mensaje al Hexiwear. Se ha utilizado este valor, ya que al tratarse de una aplicación crítica, la velocidad de la respuesta ante el evento de interés es de suma importancia. Para diez muestras la media es de 3,6 segundos, siendo la reacción más rápida de 1,7s y la más lenta de 5,2s. La tabla 6.1 muestra estos resultados.

N Muestra	1	2	3	4	5	6	7	8	9	10
Tiempo de reacción (s)	2,68	4,2	5,1	2,8	3,4	4,3	5,2	1,7	3,3	4,2

Tabla 6.1: Tiempos de reacción de la aplicación colaborativa de detección de riesgos ante la detección del evento usando dos nodos *Cookie*.

La figura 7.3 muestra una imagen de la pantalla de alerta de Hexiwear.



Figura 6.3: Imagen de la pantalla de alerta del Hexiwear

- **Paso 3:** En este paso se ha incluido el tercer nodo *Cookie* en el despliegue y se han modificado las medidas de los parámetros físicos que mide cada uno para adaptarlas a lo descrito en la anterior sección. Los nodos *Cookies*, al igual que en el Paso 1 se intercambian mensajes con los valores de las distintas medidas que conocen, y cuando todas estas medidas superan un cierto umbral, imprimen por pantalla un mensaje de alerta. La Figura 6.4 muestra los terminales de el PC asociados a cada uno de los nodos Cookies en el momento de la alerta. De la misma forma que en el Paso 1, se ha trabajado con umbrales de luz ambiente para la detección de riesgo por debajo de 1000, y se ha incorporado un umbral de temperatura de 29 grados para que se de la alerta.

6.2. Implementación de colaboración para la detección de incendios.

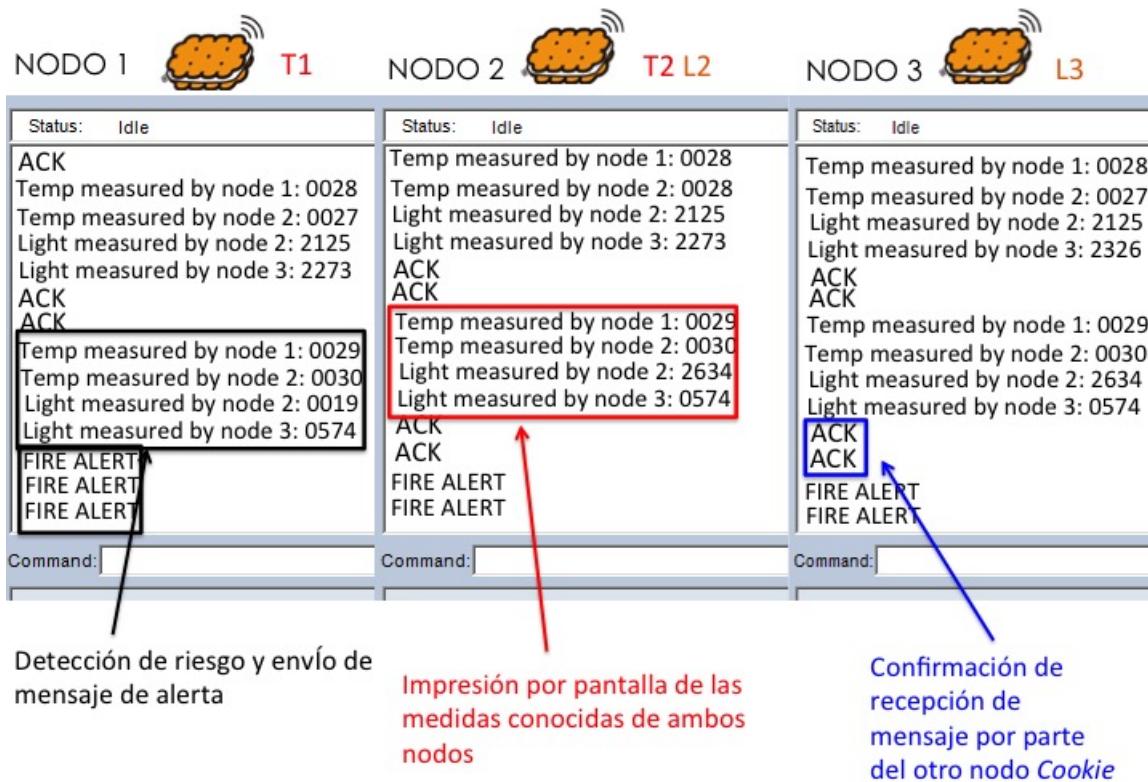


Figura 6.4: Captura de pantalla de los terminales de ordenador asociados a los nodos *Cookies* utilizados en la implementación de la prueba descrita en el Paso 3

- **Paso 4:** Por último se ha vuelto a incluir el Hexiwear en el despligue, y al igual que en el Paso 2 se han configurado los nodos *Cookies* para que al superarse los umbrales establecidos en cada medida de luz y temperatura envíen un mensaje al Hexiwear para que este vibre y imprima por pantalla la alerta descrita anteriormente. De la misma forma, si se pulsa el botón de *Stop* los nodos *Cookies* dejan de mandar mensajes de alerta y vuelven a tomar medidas de luz y/o temperatura. Al igual que ya se hizo anteriormente, se ha medido el tiempo que tardan las *Cookies* en enviar el mensaje al Hexiwear una vez el evento ha ocurrido. Para diez muestras la media fue de 4,15 segundos, siendo el menor tiempo de reacción de 1,9 y el mayor de 6,86 segundos. La Tabla 6.2 muestra estos resultados.

Capítulo 6. Redes Colaborativas y Cooperación en Redes de Sensores Inalámbricas

N Muestra	1	2	3	4	5	6	7	8	9	10
Tiempo de reacción (s)	4,37	3,9	2,84	1,9	6,86	5,12	3,84	3,6	2,98	6,1

Tabla 6.2: Tiempos de reacción de la aplicación colaborativa de detección de riesgos ante la detección del evento usando tres nodos *Cookie*.

Comparando los resultados expuestos por las tablas 6.1 y 6.2 se puede observar que al aumentar la cantidad de nodos que comparten información, también aumenta el tiempo de reacción del sistema, ya que es necesario evaluar más condiciones y cada nodo ha de esperar a recibir las medidas del resto de nodos. Aún así, este tiempo de respuesta es significativamente inferior que si el mensaje tuviese que viajar a un nodo específico, procesarse y posteriormente poner en marcha algún tipo de sistema de alerta. El tiempo que el mensaje tardaría en llegar a dicho nodo *gateway* sería similar al tiempo que tarda en llegar al Hexiwear, pero la posterior puesta en marcha de un sistema de alerta específico, ya sea visual, auditivo, o mediante algún tipo de servicio web que comunique al usuario la aparición del evento de riesgo, haría que el tiempo de reacción global incrementase considerablemente. El sistema de alerta implementado no depende de infraestructuras centralizadas como ordenadores, etc, y al necesitar de la intervención de una reducida cantidad de dispositivos para su funcionamiento no existe prácticamente tiempo de retardo en el sistema, lo que da lugar a una infraestructura robusta y rápida de detección de riesgos.

6.2. Implementación de colaboración para la detección de incendios.

La Figura 6.5 muestra un flujograma del comportamiento de la aplicación.

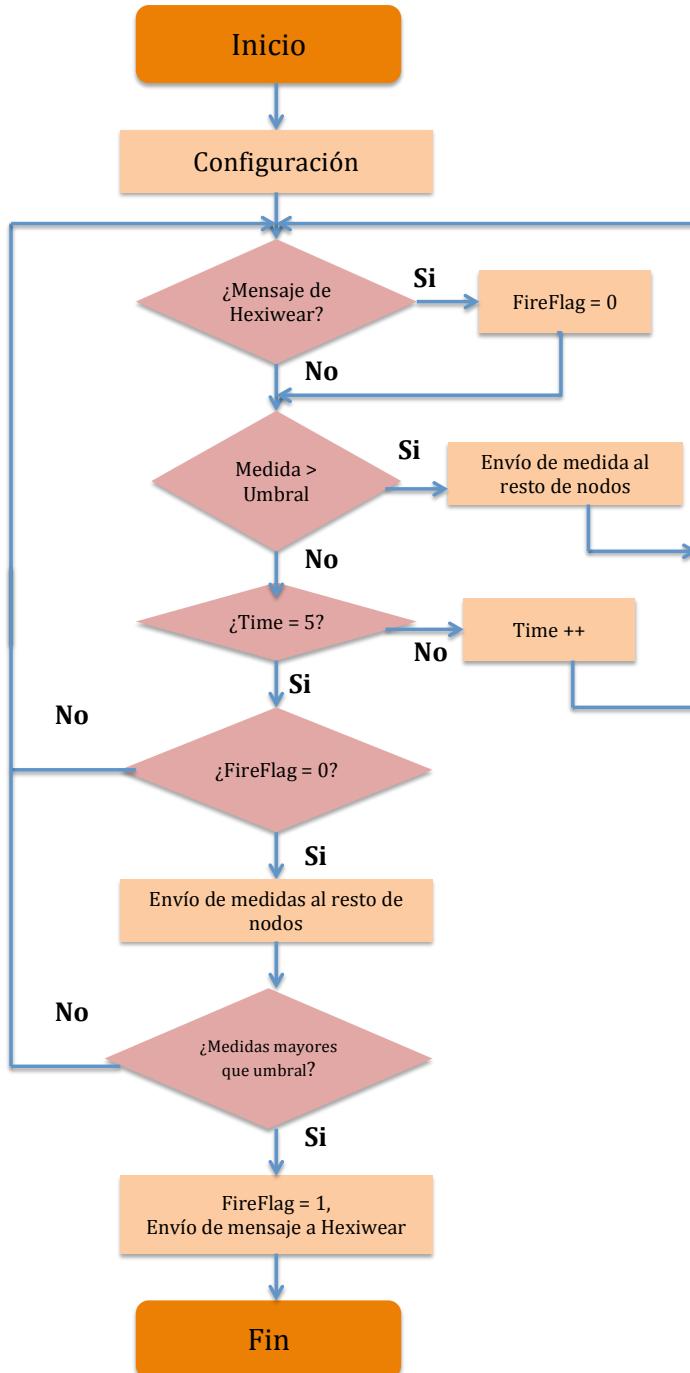


Figura 6.5: Flujo de trabajo del comportamiento de cualquiera de los tres nodos *Cookie* para la implementación de la aplicación colaborativa

6.3. Implementación de aplicación colaborativa para la detección de incendios con localización

Tras el desarrollo de esta aplicación colaborativa y la depuración de su funcionamiento, se decidió incorporar a esta un elemento de localización que pusiese en uso el filtro descrito en el anterior capítulo. Para ello se ha incorporado al despliegue un cuarto nodo *Cookie*, que estaría posicionado en una salida de emergencia, extintor, o algún lugar de utilidad en el caso de que se diese una situación de riesgo. Tras ser detectado el evento, y enviado el mensaje de alerta al Hexiwear, este nuevo nodo *Cookie* y Hexiwear se intercambian una serie de paquetes que mediante el uso del filtro de LQI descrito en el Capítulo 5, y el *Log Normal Shadowing Model* darían lugar a una estimación de la distancia a la que se encuentra dicho nodo, y consecuentemente el elemento de utilidad en caso de incendio.

El funcionamiento a bajo nivel de esta aplicación es el siguiente: Una vez se ha detectado el evento de riesgo y nos encontramos en la pantalla de alerta Hexiwear que se mostró en la Figura 6.3, al pulsar el botón de *Stop*, los nodos que toman medidas, al igual que en la aplicación anterior, vuelven a su funcionamiento normal, mientras que el nodo colocado en la salida de emergencia, al recibir este mensaje del Hexiwear, comenzará un proceso de intercambio de paquetes con este hasta que hayan llegado al nodo *Cookie* diez mensajes con LQI superior al umbral establecido por el filtro. Con los valores de RSSI de estos paquetes útiles se hace una media y se estima la distancia del nodo *Cookie* al dispositivo móvil. Esta medida se envía al Hexiwear a través del IEEE 802.15.4, y este al recibirla en el uC de las comunicaciones, se la transmite al uC principal que la muestra por pantalla. La Figura 6.6 muestra las dos pantallas del Hexiwear que intervienen en esta aplicación.

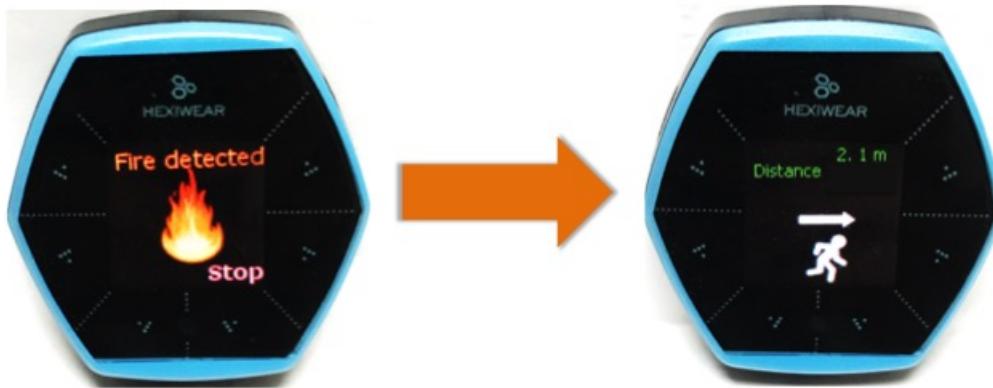


Figura 6.6: Pantallas de Hexiwear utilizadas en la aplicación de detección de riesgo con distancia.

6.3. Implementación de aplicación colaborativa para la detección de incendios con localización

Para esta aplicación se han realizado mediciones del error entre la distancia estimada por el LNSM y la distancia real a la que se encuentra el nodo de emergencia y los resultados obtenidos para distancias menores de diez metros son muy similares a los mostrados en la Tabla 5.3 del capítulo anterior. Los resultados de diez estimaciones de distancia, situando el nodo de emergencia a dos metros de Hexiwear se muestran en la tabla 6.3.

N Muestra	1	2	3	4	5	6	7	8	9	10
Estimación(m)	2,21	1,31	1,3	2,63	1,72	2,1	1,48	2,36	1,15	1,42

Tabla 6.3: Distancias estimadas al nodo de emergencia usando el *Log normal shadowing model* con el filtro de LQI para una distancia de referencia de dos metros.

La media de estas estimaciones es de 1,77 metros, siendo el error de un 11,5 %, que se encuentra dentro de un rango de precisión aceptable. Para distancias mayores a 10 metros el error aumenta considerablemente, y la cantidad de paquetes que es necesario enviar para realizar una estimación de distancia aumenta debido al uso del filtro. La tabla 6.4 muestra las estimaciones para diez muestras realizadas a 12 metros de distancia entre el Hexiwear y el nodo de emergencia.

N Muestra	1	2	3	4	5	6	7	8	9	10
Estimación(m)	17,28	16,5	15	14,6	15,2	11,12	18,7	12,4	14,3	17,8

Tabla 6.4: Distancias estimadas al nodo de emergencia usando el *Log normal shadowing model* con el filtro de LQI para una distancia de referencia de doce metros.

La media de estas estimaciones es de 15,29 metros, siendo el error de un 27,4 %, que es una magnitud demasiado elevada como para considerarse aceptable. La figura 6.7 muestra el despliegue realizado en el Centro de Electrónica Industrial de la Escuela superior técnica de Ingenieros Industriales de la UPM para la realización de estas pruebas.

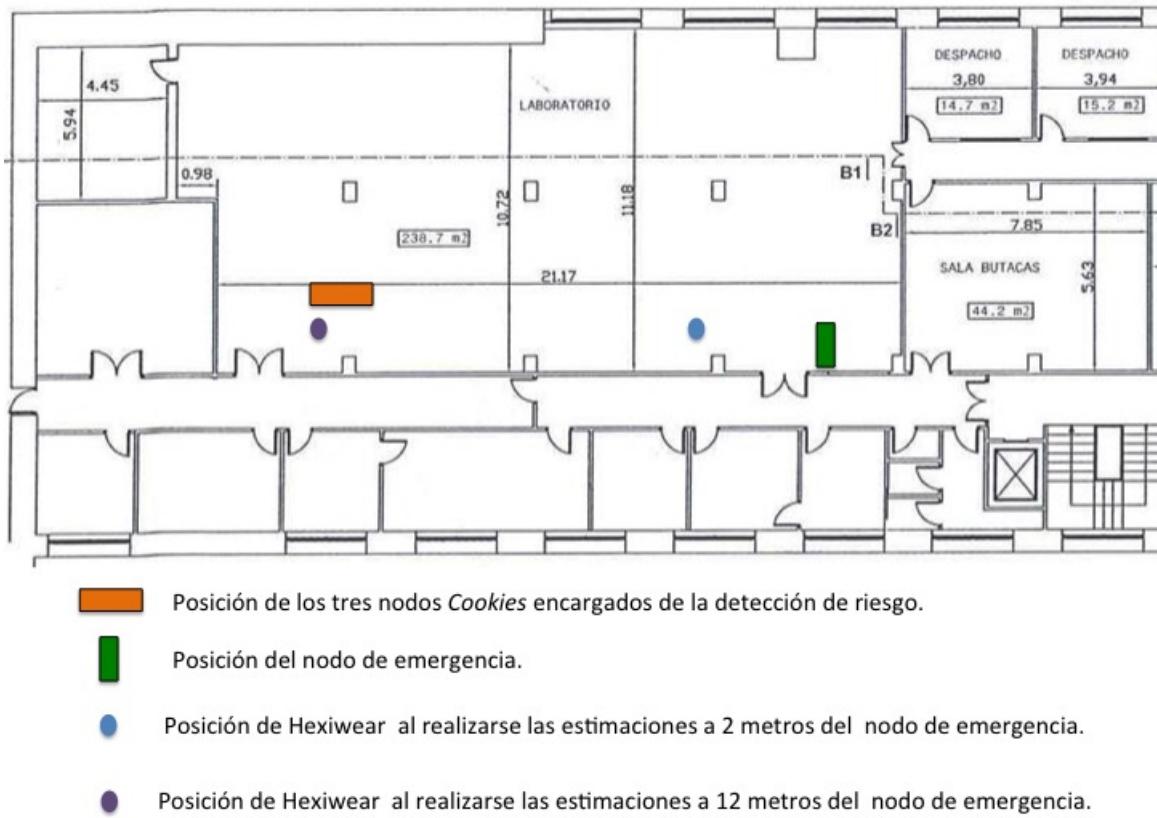


Figura 6.7: Despliegue realizado en el CEI para la estimación de distancias en la aplicación colaborativa de localización.

Para reducir este error en las estimaciones, y aumentar la precisión a distancias superiores a 10 metros entre Hexiwear y el nodo de emergencia, se podrían incluir más nodos *Cookie* distribuidos por la habitación en la que se realice el despliegue de tal forma que Hexiwear siempre se encuentre a una distancia máxima de 10 metros de alguno de estos nodos, que asistirían a este en la estimación de la distancia al nodo de importancia utilizando un sencillo algoritmo de triangulación en el que participarían también los nodos que toman las medidas de luz y temperatura. Este sería otro escenario donde la colaboración entre los nodos de una red aporta soluciones para problemas existentes en esta. Considerese un despliegue como el de la Figura 6.8, realizado en el mismo entorno que el anterior.

6.3. Implementación de aplicación colaborativa para la detección de incendios con localización

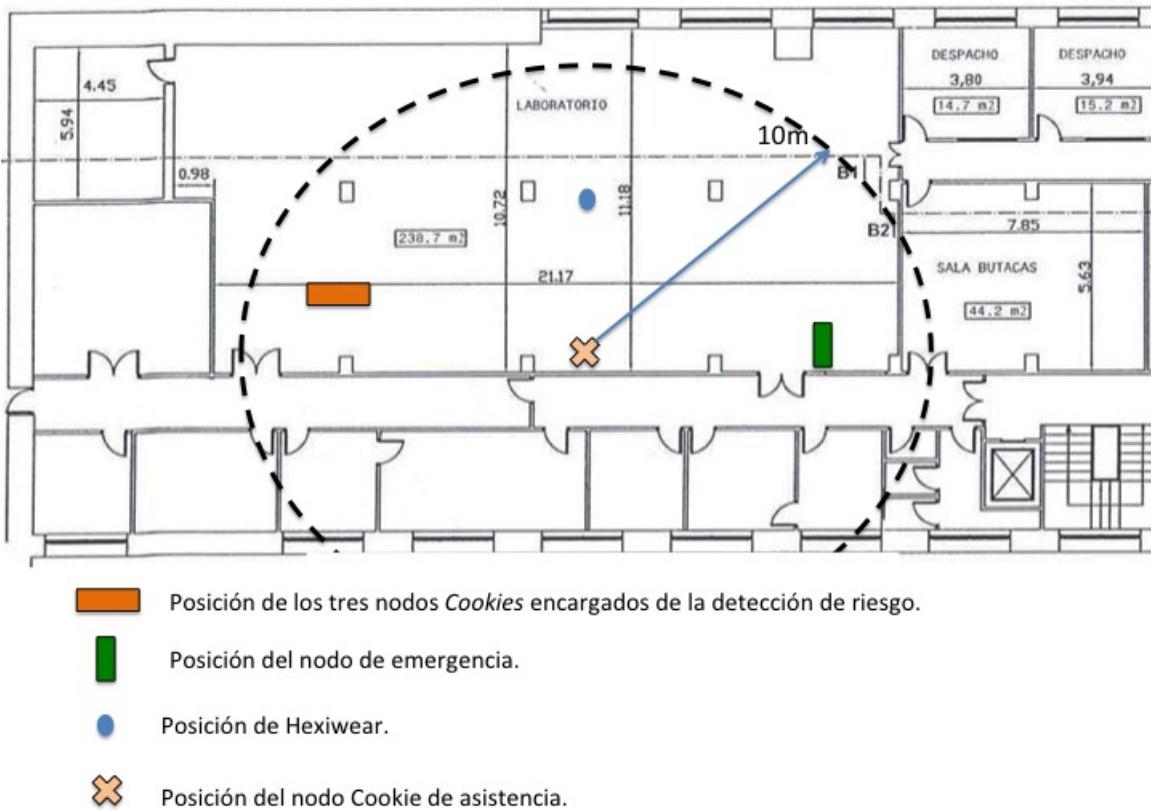


Figura 6.8: Despliegue de la aplicación de detección de riesgos más localización utilizando un nodo de asistencia.

Al encontrarse las *Cookies* que toman las medidas a unos doce metros del lugar en el que se encuentra el nodo de la salida de emergencia, el uso exclusivo de las primeras para asistir en la estimación de la distancia no tendría sentido, ya que es muy posible que no se puedan comunicar de forma eficiente con dicho nodo de emergencia debido a la distancia y los obstáculos existentes. Lo verdaderamente útil sería la utilización de un quinto nodo en el lugar marcado con una X, posicionado a una distancia intermedia entre los nodos que se encargan de tomar los datos y el nodo de emergencia, de tal forma que el usuario del Hexiwear siempre esté a una distancia aproximada de dicho nodo de unos diez metros, y este nodo a su vez a una distancia inferior de diez metros del nodo de emergencia. Si Hexiwear se encuentra en la parte derecha de la habitación donde se realiza el despliegue, y por lo tanto a menos de diez metros del nodo de emergencia, se calcularía la distancia a dicho nodo de forma directa. Si por el contrario Hexiwear se encuentra en la parte izquierda de la habitación, a más de diez metros del nodo de emergencia, se triangularia entre los nodos que toman las medidas y el nodo auxiliar para estimar la posición del Hexiwear, y de esta forma se calcularía la distancia al nodo de emergencia.

Este algoritmo de triangulación se ha puesto en práctica en el escenario que se muestra en la Figura 6.9 para intentar reducir el error en las estimaciones de distancia cuando Hexiwear se encuentra a distancias cercanas o superiores a diez metros del nodo de emergencia.

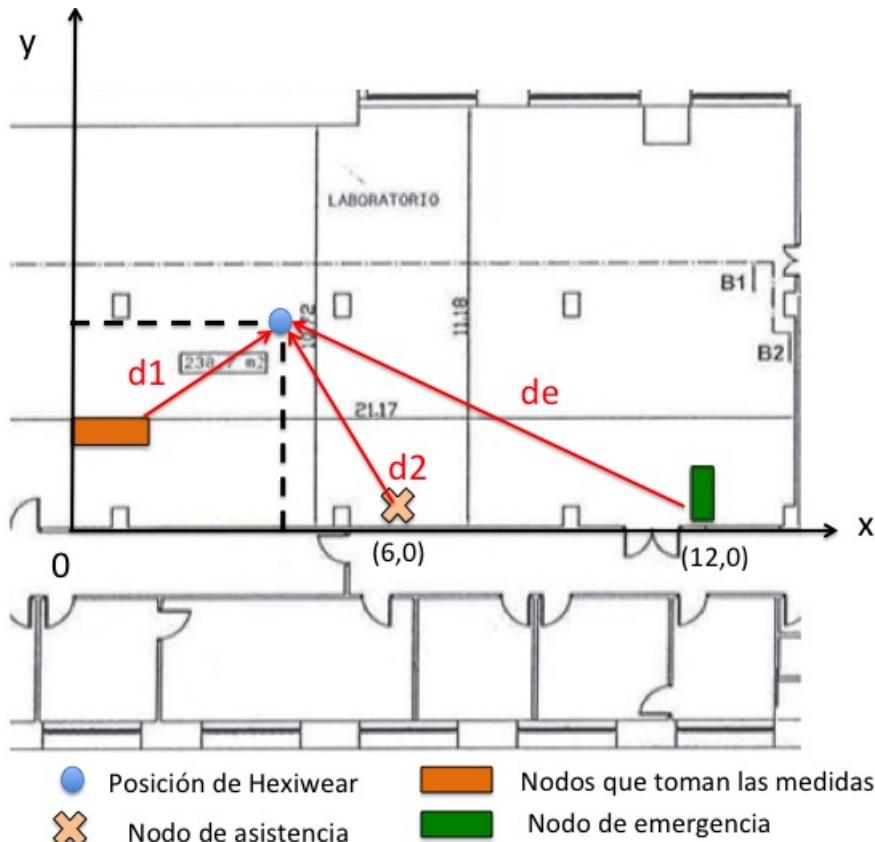


Figura 6.9: Esquema del algoritmo de triangulación utilizado para asistir a Hexiwear en la estimación de la distancia al nodo de emergencia

El algoritmo funciona de la siguiente manera: Primero se escoge un origen de coordenadas, que en el caso de este despliegue es una de las esquinas del laboratorio. Posteriormente se posicionan los nodos que toman las medidas, el nodo de asistencia, y el nodo de emergencia en lugares de coordenadas conocidas. En este caso los nodos que toman las medidas se encuentran en el punto de coordenadas (0,1), el nodo de asistencia en el punto (6,0) y el nodo de emergencia en el punto (12,0). Al recibir el Hexiwear el mensaje de alerta, y pulsar el botón de *Stop*, se calcula la distancia al nodo de emergencia (de) como se ha descrito en las páginas anteriores. Si dicha distancia resulta ser mayor de doce metros (para dar un poco de margen de error a distancias próximas a diez metros del nodo de emergencia), la estimación se descarta, y se estiman las distancias d_1 (distancia a los nodos que toman las medidas) y d_2 (distancia al nodo de asistencia).

6.3. Implementación de aplicación colaborativa para la detección de incendios con localización

Una vez estas distancias han sido calculadas, la distancia $d1$ es enviada de los nodos que toman las medidas al nodo de asistencia, que mediante la resolución del siguiente sistema calcula las coordenadas (x,y) de Hexiwear.

$$x^2 + (y - 1)^2 = d1^2$$

$$(x - 6)^2 + y^2 = d2^2$$

Conocidos x e y , coordenadas de Hexiwear, el nodo de asistencia calcula la distancia de este al nodo de emergencia utilizando el Teorema de Pitágoras:

$$(12 - x)^2 + y^2 = de^2$$

Este algoritmo se ha probado para una distancia al nodo de emergencia superior a diez metros, situando a Hexiwear en el punto de coordenadas (2,1). Se han realizado 50 estimaciones de distancia para las cuales la distancia media ha resultado ser de 9,4 metros. La Tabla 6.5 muestra algunos de estos resultados:

$d1$	$d2$	de	x	y
2,7	3,5	9,5	2,5	0,9
2,4	4	9,9	2,2	1,5
4	2,1	8	3,9	0,6
3,1	3,1	9	2,9	0,8
5,2	5,9	10,8	2,2	4,6
4,6	2,1	7,7	4,3	1,4
2,1	3,5	9,5	2,5	0,9
3,1	5,2	10,8	1,4	2,7
3,5	5,2	10,7	1,7	3
1,8	5,2	11,1	0,9	1,5

Tabla 6.5: Tabla que muestra los resultados para las distintas medidas utilizadas en el algoritmo de triangulación.

De la media obtenida en estas pruebas (9,4m) para una distancia real al nodo de emergencia de unos 11 metros, puede observarse que el error es menor que si el cálculo de la distancia se hubiese hecho de forma directa, ya que el error resultante de esta prueba es de un 15 %, mientras que de forma directa el error en la estimación de la distancia para 10 metros era de un 28 % aún usando el filtro de LQI. La distancia a la que se encuentra Hexiwear de los nodos que toman las medidas y del nodo auxiliar para esta prueba es de 4 y 2 metros respectivamente, cuyos errores individuales en la estimación de la distancia reflejados en la tabla 6.3 son de un 10 % y un 11 %. Al realizar el algoritmo la estimación de distancia con respecto a estos dos nodos, el error total debería ser superior a los errores individuales para cada una de estas

distancias, debido a que se usan dos estimaciones, cada una con su error, que se acumulan entre sí, pero sin llegar a la magnitud del error que se da a distancias elevadas.

Han de tenerse en cuenta varios factores al utilizar el algoritmo descrito. El primero es que el algoritmo se realiza en un mapa de coordenadas de dos dimensiones, que no considera la altura en el eje z, magnitud que si interviene en las distancias d_i calculadas. Otro factor a tener en cuenta es que para ciertas combinaciones de d_1 y d_2 , al intervenir en el proceso de cálculo de x e y (coordenadas de Hexiwear) una gran cantidad de operaciones con raíces cuadradas, si el error en las distancias iniciales estimadas es grande, el sistema puede no tener solución real, dando lugar a un fallo en el algoritmo. En el caso de esta implementación, dichas soluciones se descartan, reiniciándose el proceso de estimación de la distancia. Por último ha de considerarse que aunque se utilice un quinto nodo auxiliar, siguen existiendo zonas de la habitación en las cuales Hexiwear se encuentra a más de diez metros de dos de los tres puntos de triangulación posibles. Para cubrirse dichas zonas y contar con una elevada precisión en todo el laboratorio se podrían incluir aún más nodos que asistan en la localización de Hexiwear.

De esta forma, combinando el uso del filtro de LQI propuesto y la colaboración entre los nodos de una red de sensores inalámbrica se consigue reducir el error en la estimación para distancias superiores a diez metros, confiriendo a las aplicaciones que hagan uso de estas estimaciones mayor robustez y precisión.

Capítulo

7

Provisión de servicios en una red de sensores inalámbrica mediante el uso de un *smart wearable*

El fin último de las redes de sensores inalámbricas es proveer a sus usuarios de una serie de servicios que estos puedan aprovechar para mejorar su calidad de vida. En las redes tradicionales, para que estos usuarios puedan beneficiarse de la información que provee la red, es necesario que se encuentren cerca de algún tipo de ordenador o un equipo de esas características, desde el cual se gestiona y monitoriza la red. Al utilizar un dispositivo móvil e inteligente como Hexiwear en combinación con una WSN la necesidad de equipos especializados desaparece, pudiendo realizar las tareas de gestión de la red, y visualización de la información que esta recopila desde este dispositivo inteligente. Esta utilidad se ha puesto en práctica mediante el desarrollo de una última aplicación, basada en la provisión de distintos servicios por parte de los nodos de una red de sensores inalámbrica a un usuario que utiliza un dispositivo Hexiwear como sumidero de la información que proporciona dicha red. De esta forma se muestran distintos servicios que puede ofrecer una red de sensores inalámbrica: desde los mas conocidos de recolección de parámetros físicos del entorno, hasta servicios más innovadores de detección de presencia; y se resalta la forma en la cual un dispositivo de las características de Hexiwear puede recibir estos servicios de forma dinámica, descentralizada y de bajo coste.

7.1. Implementación de la aplicación de provisión de servicios

Para esta aplicación se ha utilizado un despliegue de tres nodos *Cookie* en distintas habitaciones de un mismo edificio, y un nodo móvil Hexiwear. Según el nodo

7.1. Implementación de la aplicación de provisión de servicios

móvil entra en las distintas áreas de servicio de cada uno de los nodos *Cookie*, se muestra por pantalla el servicio provisto por dicho nodo. Los servicios provistos son los siguientes:

- **Servicio 1 - Estación meteorológica:** Este servicio provee al usuario del Hexiwear de la humedad y temperatura en el laboratorio del Centro de Electrónica Industrial de la ETSII, utilizando una *Cookie* con la ya conocida capa de sensor de temperatura, luz y humedad. Este nodo toma dichas medidas periódicamente y se las comunica al usuario del Hexiwear una vez este entra en dicho laboratorio. La Figura 7.1 muestra una imagen de la pantalla de Hexiwear que aparece al entrar en el área de servicio de este nodo. La medida de presión que aparece en esta pantalla es la que mide el propio Hexiwear.



Figura 7.1: Pantalla de estación meteorológica de Hexiwear

- **Servicio 2 - Detección de presencia:** En este servicio, un nodo *Cookie* se encuentra en un recinto en el que se quiere monitorizar la presencia de un segundo usuario con otro nodo Hexiwear, que puede ser un anciano, un bebe, o en general un individuo de necesidades especiales. Esta alternativa se propone frente a un sistema de detección como podría ser el uso de un sensor PIR¹ o ultrasonidos, debido a que dichos sensores no detectan presencia en el caso de que no exista movimiento, llegando a la conclusión de que el recinto en el que se encuentran está vacío. Esta aplicación se aleja también de sistemas de monitorización inalámbrica que usan cámaras y micrófonos, en el precio y la facilidad de uso, si bien es cierto que dichos sistemas aportan una mayor cantidad de información.

¹Un sensor PIR (*Passive Infrared Sensor*) o Sensor Infrarrojo Pasivo, es un sensor electrónico que mide las variaciones de luz ambiente infrarroja, y que se suelen utilizar como detectores de movimiento.

Para el correcto funcionamiento de este servicio se utiliza un segundo nodo *Cookie*, situado inmediatamente a la salida de la habitación que se quiere monitorizar. De esta forma, en este servicio intervienen dos Hexiwears y dos *Cookies*. Para evitar confusiones llamaremos Hexiwear 1 al Hexiwear que lleva el usuario que quiere monitorizar la presencia en la habitación, y Hexiwear 2 al Hexiwear que lleva el usuario cuya presencia se quiere monitorizar. De la misma forma se llamará *Cookie 1* al nodo que se encuentra en las inmediaciones de la habitación y *Cookie 2* al nodo que se encuentra dentro de esta. La figura 7.2 muestra un esquema de los elementos que intervienen en este servicio.

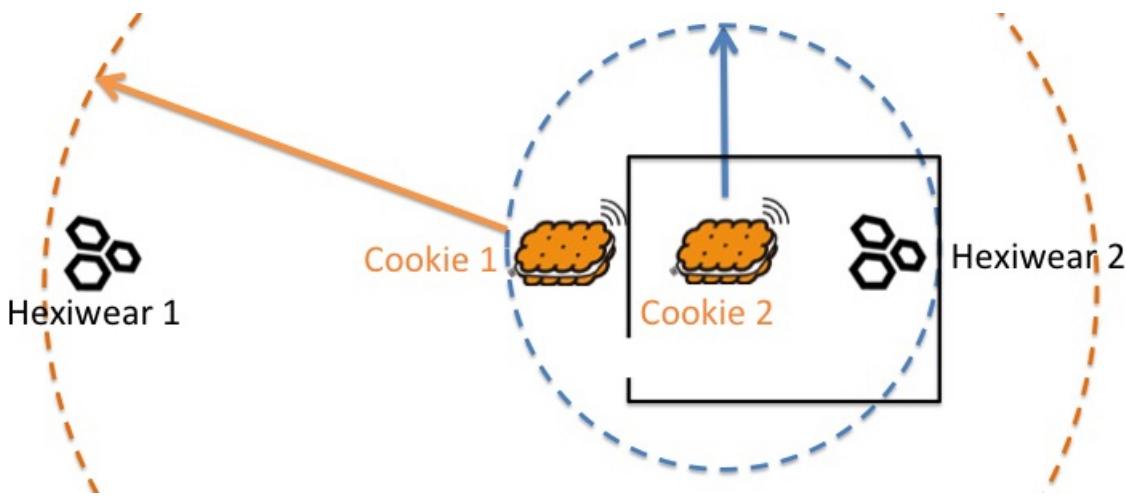


Figura 7.2: Imagen de los elementos que intervienen en el servicio de detección de presencia

La *Cookie 2*, situada dentro de la habitación, envía mensajes de baja potencia en un intervalo de tiempo programable al Hexiwear 2 que se encuentra (o no) dentro de la habitación, y al nodo *Cookie 1*. Si el Hexiwear 2 se encuentra en el recinto, y recibe correctamente el mensaje de la *Cookie 2*, responde a esta con un mensaje de vuelta, que cuando es recibido por dicha *Cookie* provoca la activación de una bandera. El valor de esta bandera se resetea en el intervalo de tiempo programado para el envío de mensajes, tras haberse transmitido como *payload* del mensaje al nodo *Cookie 1* que se encuentra en las inmediaciones de el recinto. Este segundo nodo *Cookie*, al recibir un mensaje del nodo *Cookie 2* en el intervalo de tiempo programado, lo retransmite a una potencia mucho mayor, de tal forma que el área de servicio provista por este despliegue abarque una zona mucho más extensa que la habitación que se monitoriza. El usuario del Hexiwear 1, que se nutre de los distintos servicios provistos por el despliegue global, al entrar en el área cubierta por este tercer servicio, dependiendo del valor de la bandera que le llegue con el mensaje transmitido por el nodo *Cookie 1*.

7.1. Implementación de la aplicación de provisión de servicios

1, mostrará por pantalla si existe o no presencia en la habitación de interés. Estas pantallas se muestran en la Figura 7.3.

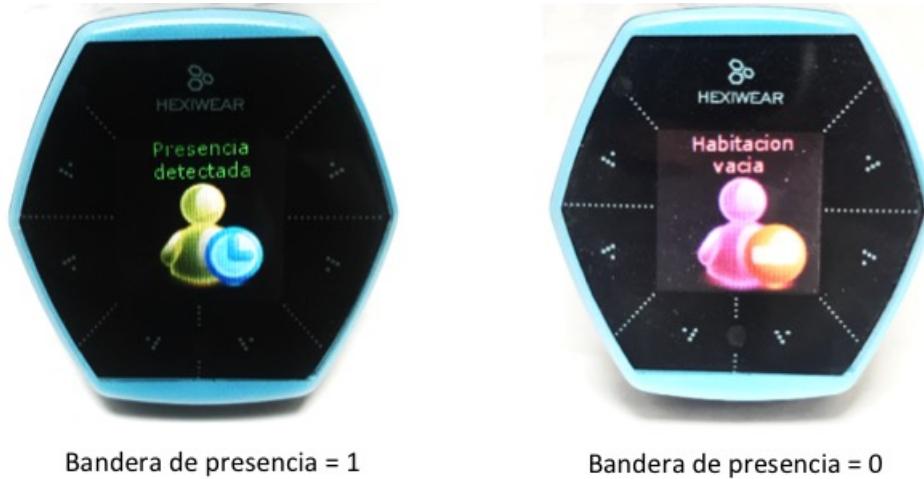


Figura 7.3: Pantallas de Hexiwear utilizadas en el servicio de detección de presencia.

Para que este servicio funcione adecuadamente se han tenido que hacer pruebas de caracterización en el recinto que se quiere monitorizar, de tal forma que los mensajes enviados por el nodo *Cookie 2* que se encuentra dentro de este tengan la suficiente potencia para llegar a todos los puntos de la habitación y al otro nodo *Cookie 1*, situado justo a la salida de esta, pero sin sobrepasarse y abarcar un área mucho más amplia que la habitación y su salida. Esto es debido a que en el caso de que el Hexiwear 2 se encuentre fuera de la habitación, si recibe un mensaje de dicha *Cookie 2*, la respondería, activándose la bandera, y por lo tanto se llegaría a la incorrecta conclusión de que dicho nodo se encuentra en el interior del recinto de interés. Para esto se ha realizado un despliegue en la habitación del CEI que se desea monitorizar, y se ha ido midiendo el rango de alcance de los mensajes que manda el nodo *Cookie 2*. En las primeras pruebas este rango fue mayor que el deseado, abarcando la habitación entera, su entrada, y una amplia porción del pasillo que se encuentra inmediatamente al salir de esta. Debido a ello se redujo la potencia de emisión del nodo, y se volvió a realizar el despliegue. Esta segunda vez los mensajes enviados por esta *Cookie* abarcaban toda la habitación, pero no eran siempre recibidos por el nodo *Cookie 1*, que se encuentra a la salida de esta. Para solucionar este problema, se desplazó el nodo *Cookie 2* a una zona cercana a la entrada de la habitación, de tal forma que los mensajes que envía siguen abarcando todo el recinto, y a su vez, al estar los dos nodos *Cookies* situados más cerca el uno del otro, los mensajes entre ambos son recibidos sin ningún tipo de problema. La siguiente figura muestra el despliegue realizado para la caracterización de este entorno, así como el flujo hipotético de mensajes para este servicio.

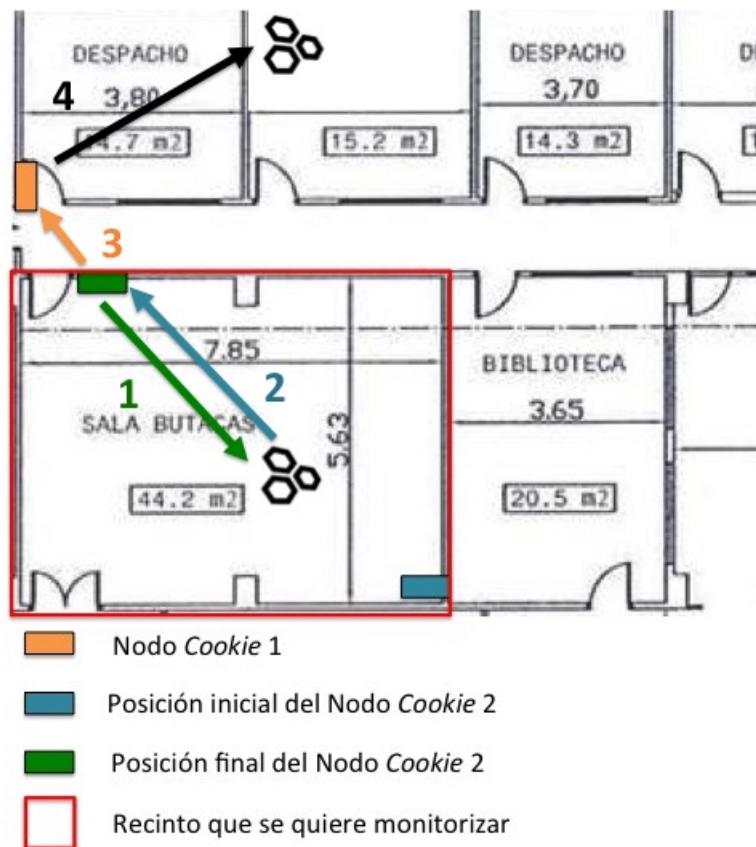


Figura 7.4: Despliegue realizado para la implementación del servicio de detección de presencia.

Este flujo consta de los siguientes pasos, suponiendo que el Hexiwear cuya presencia se quiere monitorizar se encuentra dentro de la habitación.

- **Paso 1:** El nodo *Cookie* 2 envía un mensaje de baja potencia para detectar si Hexiwear 2 se encuentra en la habitación.
- **Paso 2:** Si Hexiwear 2 se encuentra dentro de esta habitación responde a la *Cookie* 2 con un mensaje que indica su presencia.
- **Paso 3:** El nodo *Cookie* 2 retransmite la presencia de Hexiwear 2 al nodo *Cookie* 1.
- **Paso 4:** El nodo *Cookie* 1 notifica al usuario del Hexiwear 1 de la presencia de Hexiwear 2 en la habitación.

En las siguientes imágenes se muestran los flujogramas del comportamiento de los distintos elementos que intervienen en este servicio: Los dos nodos *Cookies* y los dos Hexiwears.

7.1. Implementación de la aplicación de provisión de servicios

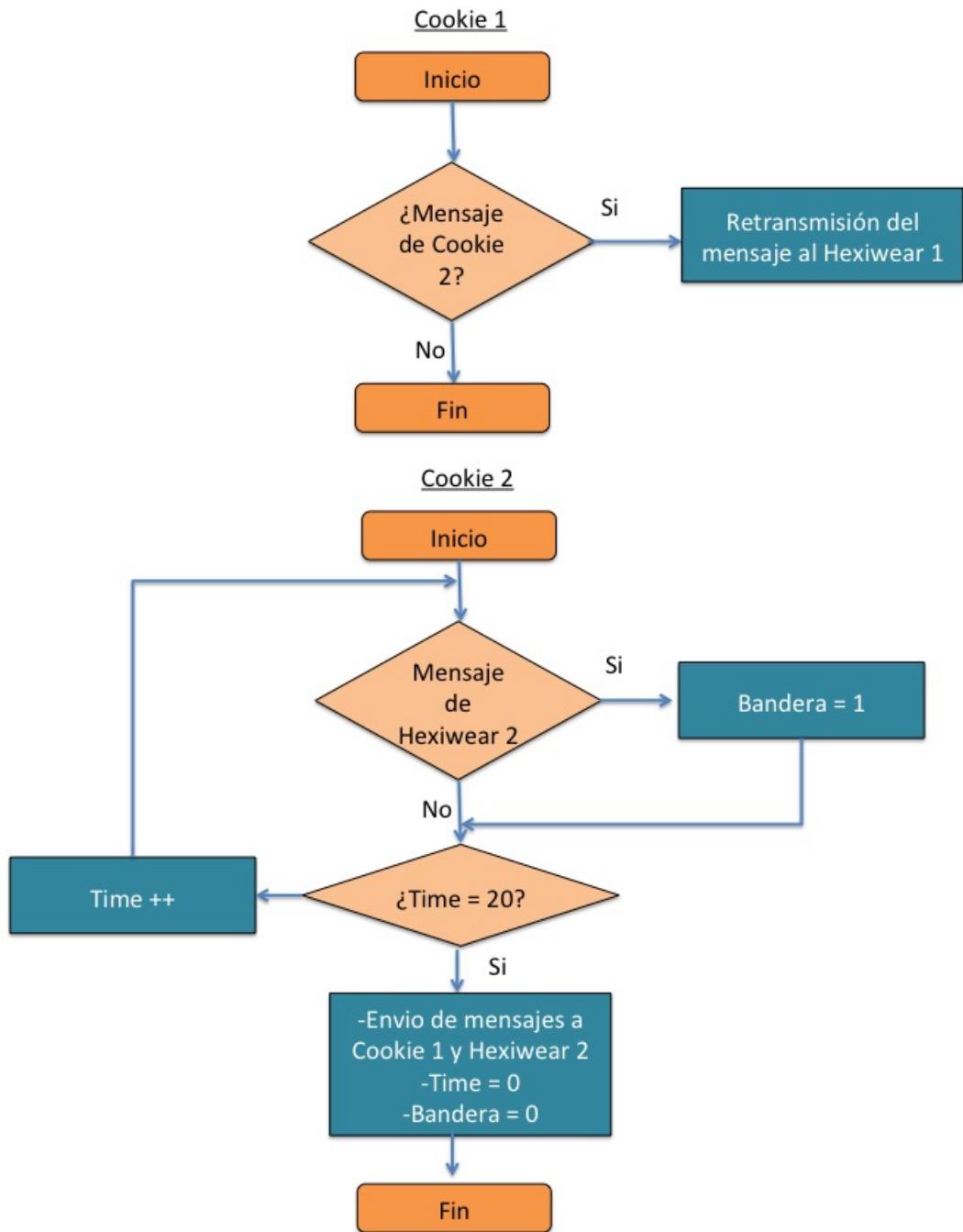


Figura 7.5: Flujograma del funcionamiento de los dos nodos *Cookie* utilizados para el servicio de detección de presencia.

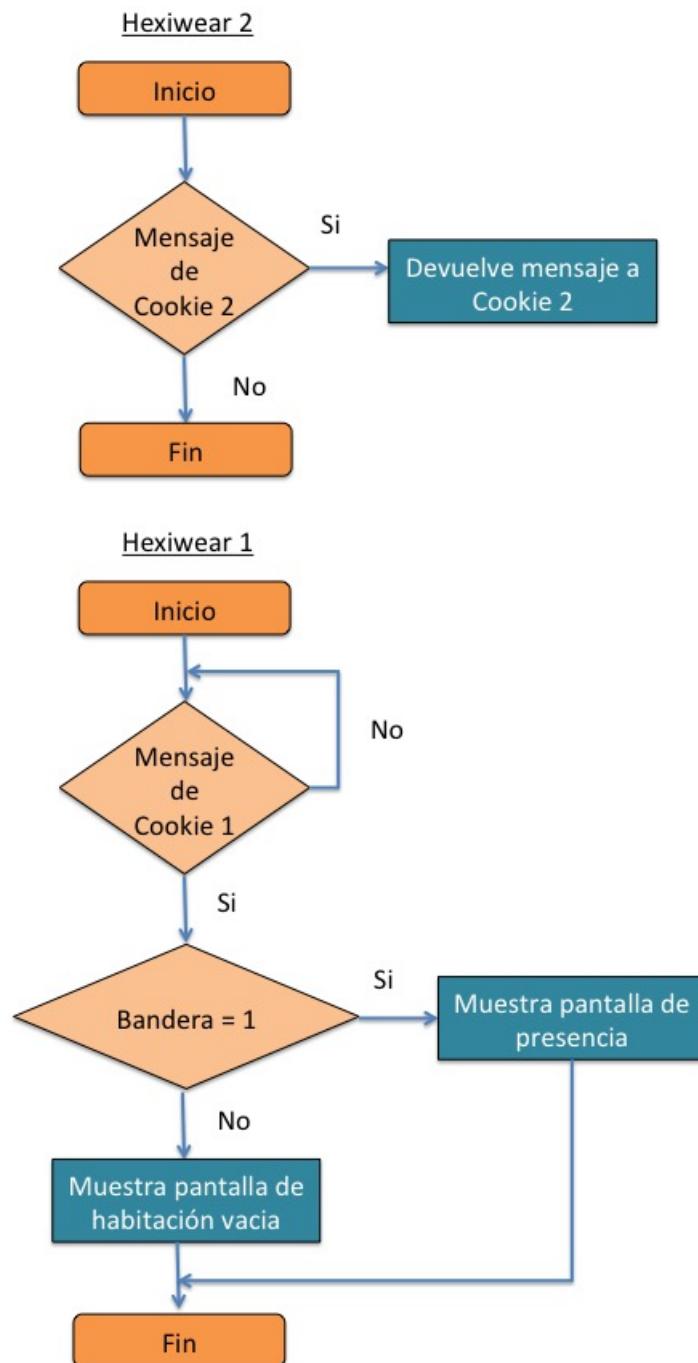


Figura 7.6: Flujograma del funcionamiento de los dos nodos Hexiwear utilizados para el servicio de detección de presencia.

7.1. Implementación de la aplicación de provisión de servicios

- **Servicio 3 - Reporte de estado de puerta:** Este servicio utiliza un nodo *Cookie* que incorpora una capa de sensado que cuenta con un acelerómetro. Este sensor se utiliza para cuantificar los eventos de apertura y cierre de una puerta de emergencia, de tal forma que se pueda conocer en todo momento si dicha puerta se encuentra abierta o cerrada. Para hacer esto se han tenido que caracterizar los eventos de apertura y cierre de esta puerta, registrando y analizando los valores del acelerómetro en dichos instantes. El acelerómetro que incorpora la capa de sensado de este nodo *Cookie* es un acelerómetro de dos ejes, con un eje normal a esta capa y otro eje paralelo a uno de los lados. La figura 7.6 muestra una imagen de dicha capa con el posicionamiento de sus ejes y una de las pantallas de Hexiwear implementadas en este servicio.

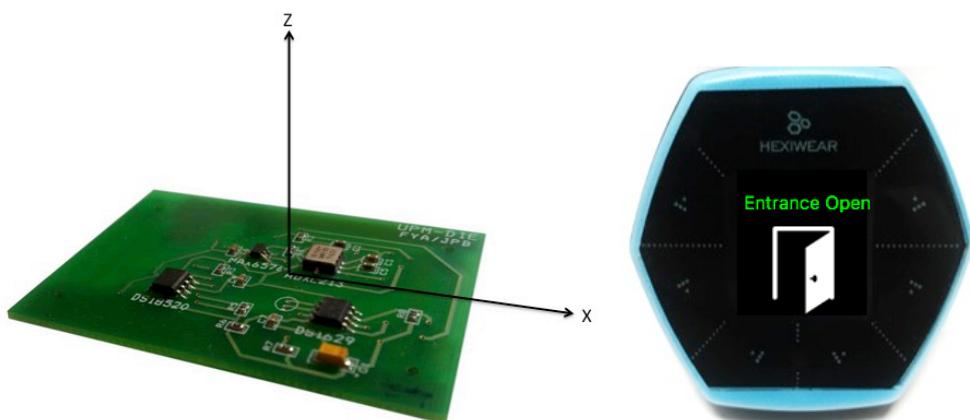
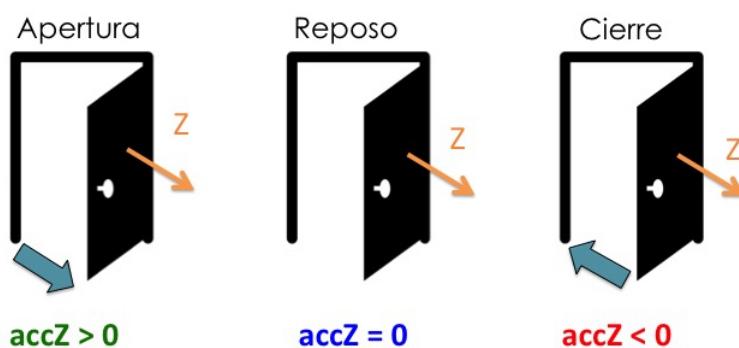


Figura 7.7: Imagen de la distribución de los ejes de la capa de sensado que usa el acelerómetro y de una de las pantallas de Hexiwear implementadas para este servicio

Cuando la puerta se abre, el acelerómetro registra una variación de aceleración positiva en el eje Z, orientado en dirección normal a esta. Si la puerta está detenida la aceleración en este eje es nula, y si la puerta se cierra la variación de aceleración que experimenta en este eje es negativa. La siguiente figura muestra este comportamiento.



Para poder cuantificar los eventos de apertura y cierre de la puerta, han de tenerse en cuenta dos consideraciones: el valor de la variación de la aceleración y su signo. El evento de apertura de la puerta es sencillo de detectar: la aceleración experimenta un incremento positivo mayor que umbral determinado experimentalmente, que sirve para descartar posibles variaciones positivas de aceleración que puedan ser originadas por golpes en la puerta en cuestión. En el momento en que la puerta se abre, en la *Cookie* se activa una bandera, que se mantendrá activa hasta que esta se haya cerrado. El cierre de la puerta es mas complejo de detectar. Si el acelerómetro mide aceleraciones negativas en el eje Z, significa que la puerta se esta cerrando, sin embargo medir aceleraciones nulas significa únicamente que la puerta está en reposo, no que se haya cerrado completamente. Por ello, para poder detectar el evento de cierre correctamente, han de utilizarse las oscilaciones en el signo de la aceleración que se producen en el momento en el que la puerta se cierra definitivamente. En ese instante se ha detectado que se producen unos rebotes durante los cuales el signo de la aceleración cambia varias veces. Para el cierre mas suave posible, se ha detectado que el mínimo número de cambios de signo es de dos. Para el cierre más brusco se han detectado hasta cinco. Por lo tanto, para la detección del evento de cierre evaluamos que se produzcan más de dos cambios de signo en la aceleración medida por el nodo *Cookie* en menos de un segundo. Una vez esto ha ocurrido, se esperan tres segundos, a partir del momento en el que la puerta está en reposo, para poner la bandera a cero y determinar que la puerta está cerrada. El nodo *Cookie*, cada cinco segundos envía un paquete a Hexiwear con el valor de esta bandera, de tal forma que si el mensaje llega a este con la bandera puesta a uno, se muestra por pantalla un mensaje en el que se puede leer que la puerta está abierta. En el caso de que el mensaje llegue con la bandera puesta a cero, Hexiwear muestra que la puerta está cerrada. Las siguientes gráficas muestran las medidas del acelerómetro en el proceso de cierre de la puerta. En el eje vertical se puede visualizar el valor de la aceleración en g^2 . El eje horizontal muestra la evolución del tiempo en intervalos de una décima de segundo.

²La fuerza g es una medida de la aceleración en función de la aceleración de la gravedad ($9,86m/s^2$)

7.1. Implementación de la aplicación de provisión de servicios

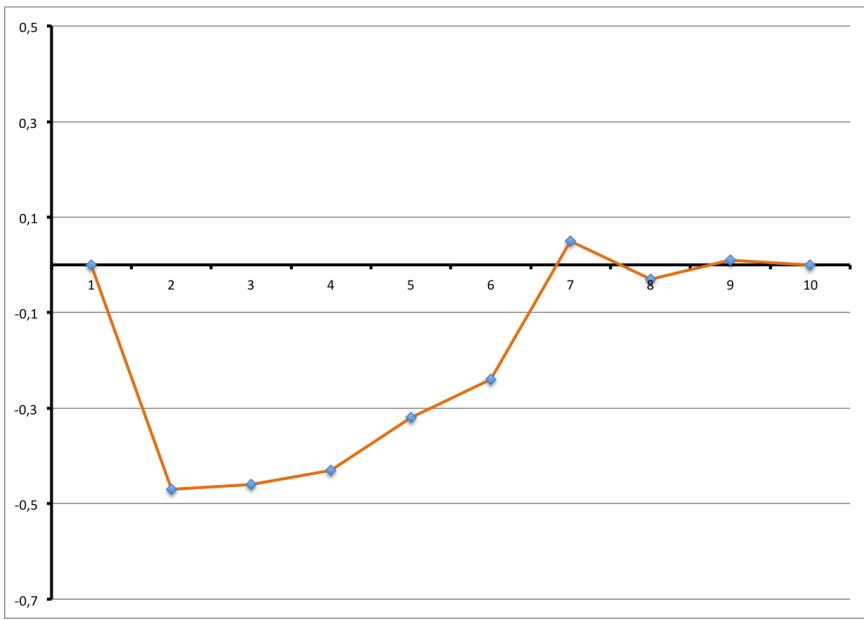


Figura 7.8: Gráfica que muestra la evolución de la aceleración medida por el nodo *Cookie* en el proceso de cierre de la puerta.

En esta gráfica se puede observar como la aceleración comienza siendo negativa, pasa por cero y oscila varias veces en magnitudes cercanas a cero entre valores positivos y negativos. La figura 7.8 muestra la zona con estas oscilaciones.

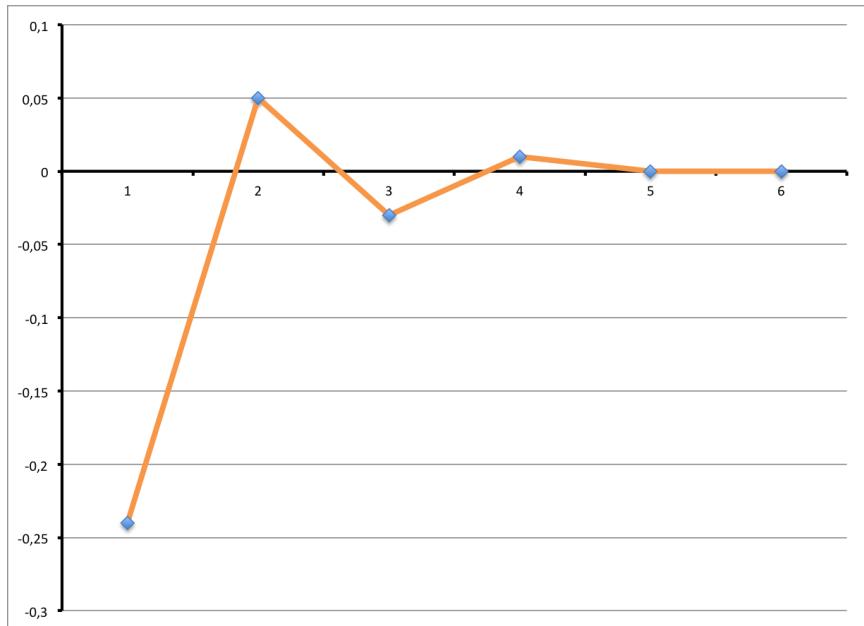


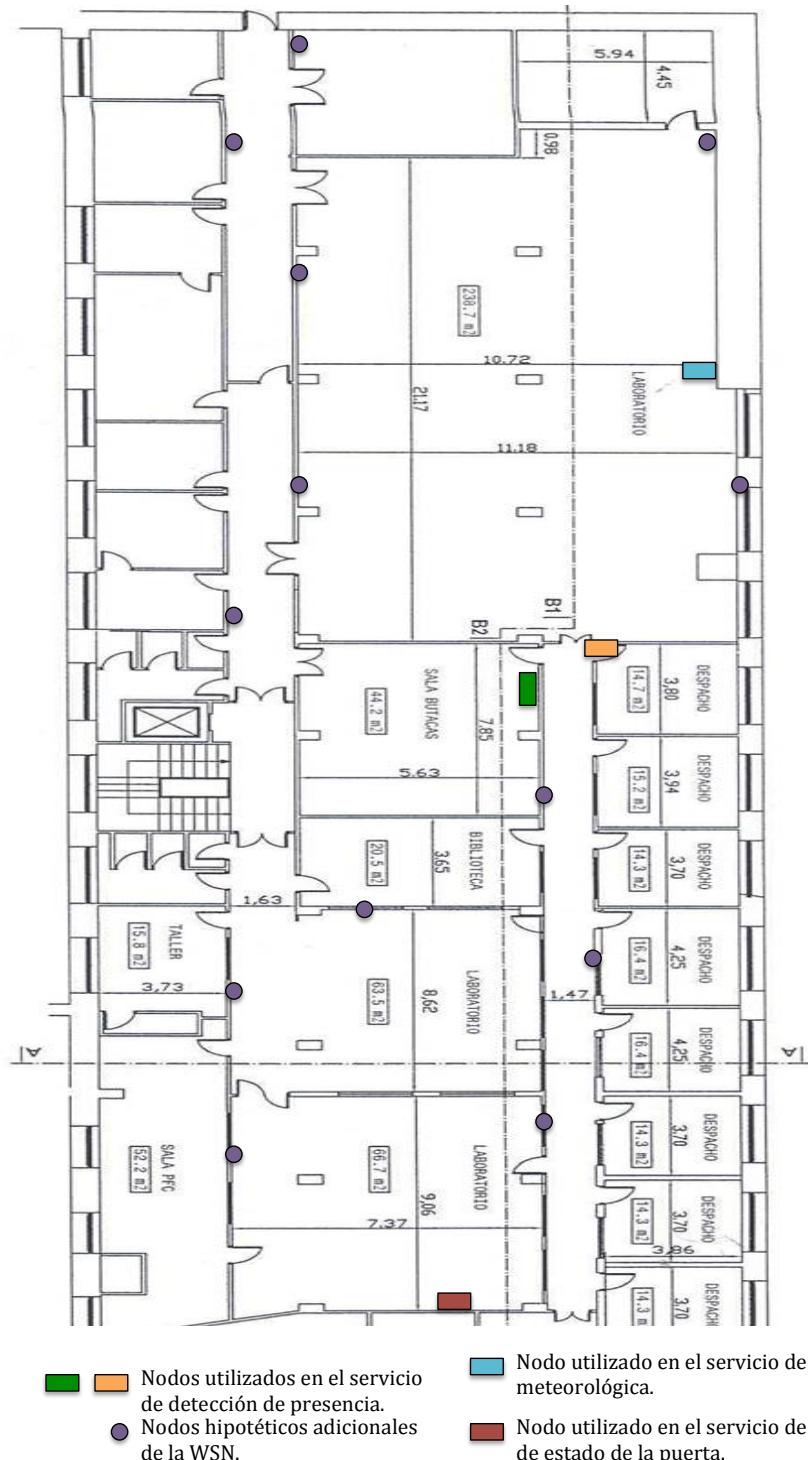
Figura 7.9: Gráfica que muestra las oscilaciones en la aceleración medida por el nodo *Cookie* en el momento de cierre de la puerta.

La aplicación resultante funciona correctamente, con un elevado grado de acierto en la determinación del estado de la puerta en todos los casos menos el siguiente: Si la puerta se encuentra abierta, y alguien procede a cerrarla, y en el instante justo anterior a que llegue a cerrarse esta se frena manualmente de forma brusca, las oscilaciones en el signo de la aceleración a partir de las cuales cuantificamos el evento de cierre de la puerta se producen también, llegándose a la incorrecta conclusión de que se encuentra cerrada.

Con la implementación de estos tres servicios, concretamente con el segundo y el tercero, se han querido mostrar usos para los nodos de las WSN que se alejen un poco de las aplicaciones tradicionales de recolección de parámetros físicos. Se han buscado aplicaciones sencillas de implementar pero a la vez con gran utilidad y que puedan proveer información relevante.

7.1. Implementación de la aplicación de provisión de servicios

En la siguiente imagen se muestra el despliegue realizado, con los nodos que implementan cada uno de los servicios.



Capítulo 8

Conclusiones, líneas futuras y gestión del proyecto.

En este capítulo se expondrá un resumen de las conclusiones derivadas de los resultados obtenidos en este proyecto, así como de las ventajas que se han observado durante la realización del mismo, obtenidas de la integración de un dispositivo inteligente dentro de una red de sensores inalámbrica. También se explorarán líneas futuras de investigación, orientadas a mejorar esta integración y maximizar sus ventajas.

8.1. Conclusiones generales

En esta sección se exponen las conclusiones generales obtenidas al realizar la integración de un dispositivo inteligente como Hexiwear dentro de una red de sensores inalámbrica.

La primera conclusión es que para que esta integración pueda aprovecharse al máximo, el dispositivo inteligente ha de ser compatible con el estándar *de facto* para las redes de sensores inalámbricas; el IEEE 802.15.4. De esta forma este dispositivo puede comunicarse de forma directa con los nodos de la red, con los requisitos de bajo consumo y baja transmisión de datos que caracterizan a dichos nodos. Si esto no ocurre, existen dos alternativas para que el dispositivo inteligente pueda conocer la información que los nodos de la red envían: La primera es que los nodos se comuniquen utilizando *Bluetooth* o *Wi-Fi*. Esto permitiría que el dispositivo inteligente, utilizando uno de estos dos protocolos, pudiese comunicarse de forma directa con los nodos de la red, aunque esta se encontraría lejos de su comportamiento ideal, ya que ninguno de estos dos protocolos está optimizado para WSN. La segunda alternativa es que los nodos se comuniquen utilizando el IEEE 802.15.4, y que un nodo *gateway*

8.1. Conclusiones generales

haga que la información que el resto de nodos de la red recopila esté disponible en la nube, de tal forma que el dispositivo inteligente pueda acceder a ella. Con esta alternativa, a pesar de que la red opera de manera óptima, existiría un mayor retraso en el tiempo que tarda en llegar la información al dispositivo inteligente que si este se comunica de forma directa con los nodos utilizando el IEEE 802.15.4. A su vez, las capacidades de interacción que proporciona la comunicación directa del dispositivo inteligente con el resto de nodos de la red se perderían o se verían muy limitadas y se perdería la ventaja de no necesitar equipos centralizados que aporta el uso de un *smart wearable*.

En segundo lugar se ha estudiado la estructura a bajo nivel de la integración realizada entre los nodos *Cookie* y el Hexiwear, analizando sus puntos fuertes y posibles mejoras. Entre estos puntos fuertes están el sistema operativo a tiempo real, la interfaz gráfica de usuario a nivel de aplicación, la separación de las tareas más significativas de cada dispositivo en distintos módulos y la modularidad aportada por las *Cookies*. Como conclusión, la plataforma global resultante es robusta, flexible y dinámica, siendo posible su uso en una gran variedad de entornos de aplicación.

La tercera conclusión es que es posible mejorar las estimaciones de distancia de los algoritmos que utilizan únicamente el RSSI, mediante el uso mixto de RSSI y LQI. En la propuesta que se hace en este trabajo se ha implementado un filtro sencillo de LQI para el cálculo de distancias utilizando el *Log Normal Shadowing Model*. Este filtro, descartando paquetes con LQI inferior a un umbral determinado, logra reducir el error medio en las estimaciones para distancias de 1 a 10 metros en un 5% (De 22% sin el uso del filtro, el error se reduce a un 17%).

Se han estudiado también las peculiaridades de las redes colaborativas llegando a la conclusión de que la cooperación entre los nodos de una red de sensores permite la realización de tareas de mayor complejidad que la operación independiente de cada nodo. A su vez, se ha observado que las virtudes de esta cooperación se ven amplificadas por el uso de un dispositivo inteligente que trabaje conjuntamente con el resto de nodos, aumentando de manera significativa la cantidad de aplicaciones posibles para las redes de sensores inalámbricas cooperativas. Con el fin de llevar esta percepción a la práctica se han implementado una serie de aplicaciones de detección de riesgos, en las cuales los nodos comparten información sobre los parámetros que miden entre sí, para cuantificar la aparición de un evento que de otra forma les sería imposible detectar debido a su limitada capacidad de sensado. En una de estas aplicaciones se ha hecho uso del filtro de LQI para la estimación de distancias que se utilizan en un algoritmo de triangulación, implementado con el objetivo de reducir el error en el cálculo de distancias mayores de diez metros. Con este algoritmo, se ha logrado reducir este error en más de un 10%, acentuando aún más las ventajas de la colaboración de los nodos de las WSN. En todos los casos (con el elemento de localización y sin él) se han conseguido desarrollar aplicaciones sencillas pero a la

vez robustas, y con una gran utilidad real.

Por último se ha querido resaltar la forma en la cual un dispositivo inteligente, con una interfaz de usuario sencilla y elegante, puede utilizarse para recibir los servicios que provee una red de sensores inalámbrica de forma dinámica, barata y descentralizada. Para ello se ha implementado una sencilla aplicación en la cual el usuario a cargo del dispositivo inteligente recibe distintos servicios según se va desplazando por un edificio en el que hay una red de sensores inalámbrica desplegada. Esta aplicación, al igual que las descritas anteriormente ha sido un éxito debido a su facilidad de implementación y por su gran posibilidad de aplicación en entornos reales, resaltando la gran variedad de servicios que puede ofrecer una red de sensores inalámbrica.

De forma global, de este trabajo se concluye que la integración de un dispositivo inteligente y además *wearable* dentro de una red sensores inalámbrica es un paso gigante hacia la integración de este tipo de redes dentro del ecosistema del *Internet de las cosas*. Integraciones como esta multiplican la cantidad de aplicaciones posibles para las redes de sensores inalámbricas, potencian sus virtudes, y ponen al alcance de los usuarios de estas una serie de servicios y utilidades sin precedentes.

8.2. Consecución de objetivos

A continuación se muestran los objetivos que se definieron en la primera parte del proyecto, para evaluar si se han alcanzado y como se han alcanzado.

El objetivo último del proyecto, que consistía en la integración de un dispositivo inteligente dentro de una red de sensores inalámbrica y el estudio de las posibilidades de esta integración, se ha alcanzado, llegando a las conclusiones expuestas en la sección anterior . El desglose de objetivos iniciales y la evaluación de la consecución de dichos se expone a continuación:

- **Objetivo 1: Comunicación efectiva entre el dispositivo inteligente y los nodos *Cookies* utilizando el IEEE 802.15.4**

Este objetivo marcaba el futuro del presente proyecto, ya que en el caso de no haberse podido establecerse la comunicación entre las dos plataformas utilizadas, se tendría que haber considerado un cambio de rumbo en el objetivo final. Una parte muy extensa de este proyecto ha consistido en verificar esta comunicación. Como se redactó en el Capítulo 4 la comunicación fue exitosa, pudiéndose realizar el resto del proyecto según lo planeado, por lo que este objetivo se ha realizado exitosamente.

- **Objetivo 2: Mejorar las estimaciones de distancia entre dos nodos usando una propuesta mixta de RSSI y LQI**

Para la realización de este objetivo se implementó el filtro de LQI en los paquetes que se utilizan en la estimación de distancias con el *Log Normal Shadowing Model*. Este objetivo también fue un éxito, ya que se comprobó que los errores en las estimaciones de distancias se reducen con el uso dicho filtro, siendo esta propuesta publicada en un artículo científico de investigación. A su vez, en una de las aplicaciones desarrolladas, se ha implementado un algoritmo de triangulación que reduce el error en las estimaciones a distancias mayores de diez metros mediante la colaboración de distintos nodos de una red y el uso de dicho filtro.

- **Objetivo 3: Desarrollo de una serie de aplicaciones colaborativas que resalten las posibilidades de la integración wearable/WSN**

Al igual que para los dos objetivos previos, los resultados para el objetivo 3 han sido positivos al ser las aplicaciones implementadas sencillas, robustas y de gran utilidad. La aplicación colaborativa desarrollada, con el elemento de localización incluido, resalta las virtudes que ofrece la cooperación de los nodos de las redes de sensores inalámbricas, y las ventajas que aporta a esta colaboración la inclusión de un nodo inteligente en dicha red. Por otro lado, la aplicación de provisión de servicios muestra tres usos distintos que se le pueden dar a los nodos de las redes de sensores inalámbricas, y pone de manifiesto la forma en la cual un dispositivo inteligente puede integrarse dentro de una red de este tipo y recibir la información que esta recopila de manera dinámica y sin la necesidad de infraestructuras centralizadas.

8.3. Lineas futuras de investigación

- **Utilización de Zigbee en el dispositivo inteligente:** Una primera linea futura podría ser la implementación del *Zigbee* en el dispositivo inteligente que se utilice en la integración, de forma que adquiera las capacidades de rutado y las funcionalidades de las capas superiores con las que el IEEE 802.15.4 no cuenta. Esto ampliaría aun más el abanico de posibilidades de la integración y daría lugar a una mayor cantidad aun de posibles aplicaciones en las que se podrían implementar algoritmos de elevada complejidad.

.

- **Uso de dos estándares de comunicaciones en el dispositivo inteligente:** En el caso de la integración propuesta en este trabajo, la compatibilidad Bluetooth del dispositivo inteligente tuvo que ser eliminada para poder incorporar la pila de comunicaciones del IEEE 802.15.4. Si se contase con un dispositivo inteligente que implementase el IEEE 802.15.4 o el *Zigbee* y otro

estándar de comunicaciones como Bluetooth o *Wi-Fi*, dicho dispositivo podría usarse como *gateway* de la red, transmitiendo la información que los nodos recopilan a algún otro dispositivo, como un *smartphone* o *tablet*, o directamente a la nube. Encontrar o diseñar un dispositivo de estas características, y estudiar las posibilidades del uso de dos protocolos de comunicaciones distintos en este dispositivo inteligente es un campo futuro de investigación del cual se esperan prometedores resultados.

- **Desarrollo de aplicaciones que aprovechen al máximo el potencial de la integración wearable/WSN:** Siguiendo en esta línea, aplicaciones más complejas que resalten aun más las fortalezas y capacidades de la integración de un dispositivo inteligente con los nodos *Cookie* serán diseñadas y implementadas, usando la arquitectura al máximo de sus capacidades y explorando nuevas metas para las redes de sensores inalámbricas. El éxito de integraciones de este estilo abre un nuevo mundo de aplicaciones para las redes de sensores inalámbricas, que gradualmente van destapándose como una de las tecnologías mas prometedoras de nuestro tiempo.
- **Mejora de los algoritmos de localización para redes de sensores inalámbricas:** Por último, aún existe una gran cantidad de trabajo pendiente en cuanto a los algoritmos de localización para redes de sensores inalámbricas, ya que para la familia de algoritmos que se ha utilizado en este trabajo (*Los Range based algorythms*) la inestabilidad de los parámetros de radio a partir de los cuales se estima la distancia, hace que estas estimaciones estén aun lejos de ser precisas. En este trabajo se ha desarrollado un filtro que mejora la precisión de las estimaciones de distancia, y se ha implementado en aplicaciones mediante el uso de algoritmos de triangulación. Aunque se ha reducido el error en las estimaciones de distancia aun existen muchos factores que hacen que estas sean muy oscilantes, como la orientación de los dispositivos que se comunican, los obstáculos en el medio, y la distancia. Futuras líneas de investigación buscarán nuevos modelos que hagan frente a estos problemas.

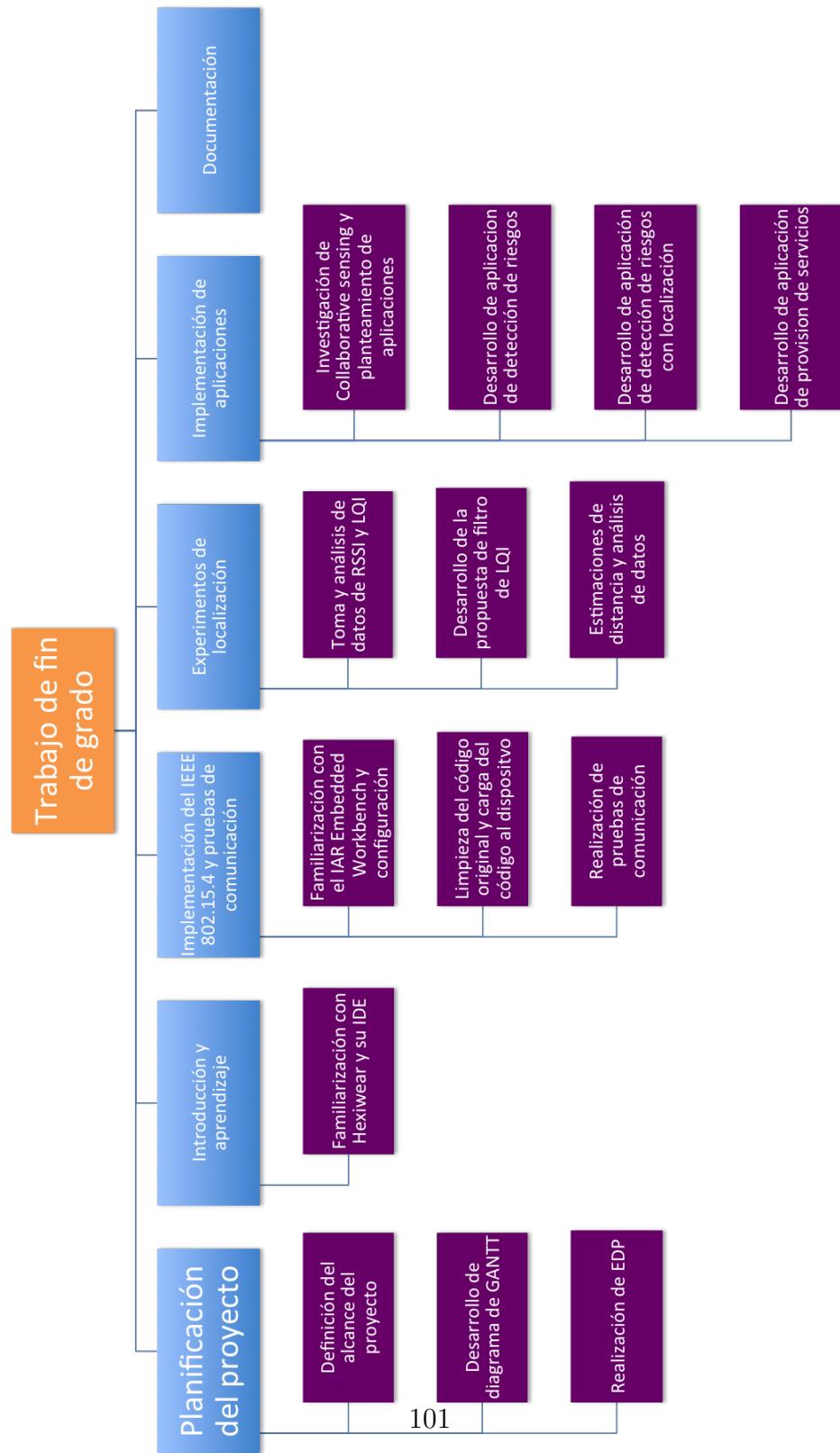
8.4. Gestión del proyecto

8.4.1. Planificación temporal

El proyecto inicialmente se dividió en distintos paquetes de trabajo, de tal forma que durante la realización de este se ha ido buscando alcanzar diversos hitos en determinados plazos. Los paquetes de trabajo han sido los siguientes:

- **Planificación del proyecto:** En esta fase se estudiaron las distintas tareas a realizar para la consecución de los objetivos finales del proyecto, y se definieron los plazos aproximados para la realización de dichas tareas.
- **Introducción y aprendizaje:** Esta fase del proyecto consistió en la familiarización con Hexiwear, los entornos de desarrollo necesarios para programarlo, y se aprendió a gestionar distintos periféricos del dispositivo como la pantalla o los sensores, así como el sistema operativo a tiempo real.
- **Implementación del IEEE 802.15.4 en Hexiwear y pruebas de comunicación:** En esta fase del proyecto se implementó el estándar IEEE 802.15.4 en Hexiwear, teniendo que hacer una limpieza previa del código que este traía de fábrica. Tras haber conseguido esto, se establecieron las primeras comunicaciones entre Hexiwear y las *Cookies*.
- **Experimentos de localización:** En esta fase del proyecto se realizaron las pruebas orientadas a la toma de datos y implementación de un nuevo modelo de estimación de distancias para las redes de sensores inalámbricas.
- **Implementación de aplicaciones:** En esta fase se desarrollaron las distintas aplicaciones colaborativas y de provisión de servicios.
- **Documentación:** Por último, se procedió a la redacción de este documento, en el que se detalla el proyecto que se ha llevado a cabo.

A continuación se adjunta el diagrama EDP del proyecto, donde se pueden visualizar las distintas fases de este, así como sus subtareas asociadas.



8.4. Gestión del proyecto

A continuación se muestra el diagrama de Gantt. El proyecto comenzó a finales de Octubre de 2016. El diagrama se establece entonces desde esta fecha hasta Junio de 2017, fecha de entrega del proyecto. En las primeras fases de este se llevaron a cabo la planificación y el aprendizaje de las distintas plataformas a utilizar, de las redes de sensores inalámbricas, y la familiarización con los distintos dispositivos que se han utilizado a lo largo del proyecto. Tras esta fase introductoria vino el periodo más extenso del proyecto, que consistió en conseguir implementar correctamente el IEEE 802.15.4 en Hexiwear. Una vez este estándar se consiguió implementar correctamente se realizaron las pruebas de comunicación para garantizar que Hexiwear y las *Cookies* podían comunicarse utilizando dicho estándar.

Una vez garantizada la comunicación se comenzó a utilizar la arquitectura conseguida de con esta integración para los experimentos de localización y la propuesta de un nuevo modelo que mejore las estimaciones de distancia. Por último se procedió al desarrollo de las aplicaciones que evalúasen el potencial de la arquitectura propuesta, solapándose esta fase con la fase final de redacción de este documento.

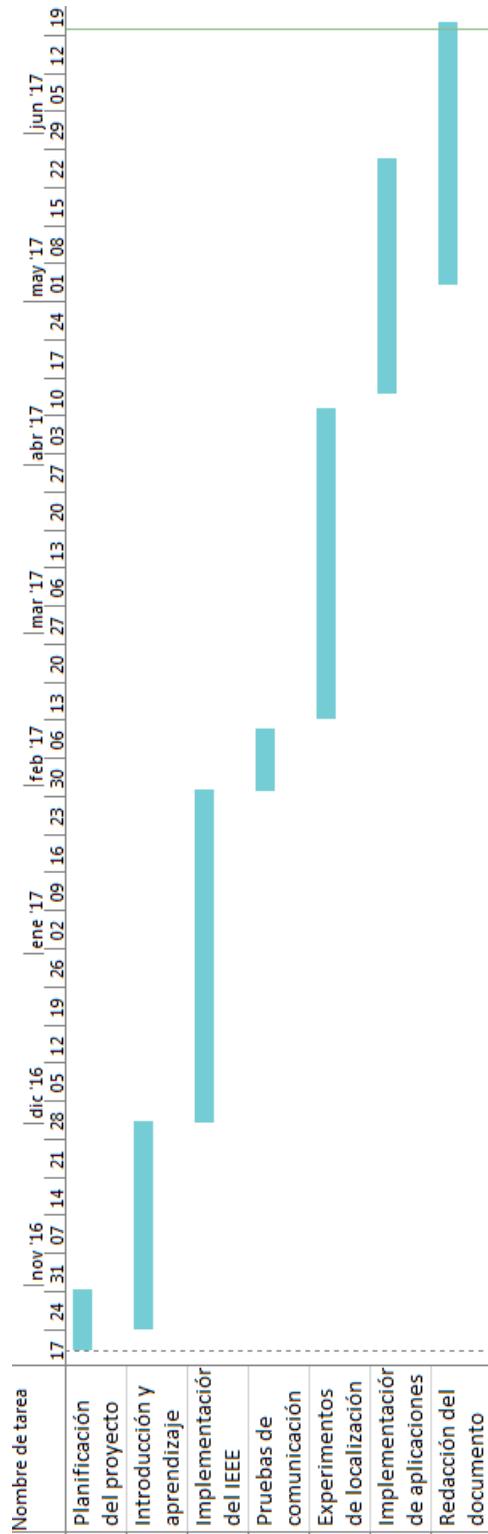


Figura 8.1: Diagrama de Gantt del presente proyecto.

8.4. Gestión del proyecto

Para mayor claridad, la siguiente tabla muestra las distintas fases del proyecto con sus fechas de inicio y final.

Fase del proyecto	Fecha de inicio	Fecha de fin
Planificación del proyecto	20/oct/16	30/oct/16
Introducción y aprendizaje	31/oct/16	30/nov/16
Implementación del IEEE 802.15.4	1/dic/16	30/ene/17
Pruebas de comunicación	31/ene/16	10/feb/17
Experimentos de localización	13/feb/17	10/abr/17
Implementación de aplicaciones	14/abr/17	26/may/17
Documentación	16/may/17	20/jun/17

Tabla 8.1: Tabla que muestra el desglose de las distintas fases del proyecto

8.4.2. Presupuesto

Este proyecto se ha realizado con un ordenador de sobremesa DELL Vostro 200. El precio de este equipo se expone en la tabla 8.1. Para la realización del proyecto se han utilizado también distintos *softwares*, ya incluidos en el PC utilizado, por lo que el coste de sus licencias no se tendrá en cuenta. Por último, se han utilizado dos kits *Hexiwear user's pack* y un total de 6 nodos *Cookies* en los despliegues realizados.

Los costes se van a evaluar en amortizaciones de los elementos mencionados en el primer párrafo de esta sección, considerándose la amortización del ordenador, de Hexiwear, y de los nodos *Cookies* a 4 años.

Elemento	Vida útil (años)	Precio (€)	Tiempo de uso en el proyecto (años)	Amortización (€)
Ordenador DELL Vostro 200	4	840	0.5	105
Microsoft Office	–	–	0.5	–
IAR Embedded Studio	–	–	0.5	–
Telegesis terminal	–	–	0.5	–
Keil µVision 3	–	–	0.5	–
Kinetis Design Studio	–	–	0.5	–
Hexiwear user pack (2 unidades)	4	200	0.5	25
Nodos Cookies (6 unidades)	4	360	0.5	45
TOTAL				175

Figura 8.2: Presupuesto del presente proyecto

Al coste total de los elementos utilizados en el proyecto es de 175€. A esto hay que sumarle el precio de mano de obra de un estudiante sin graduar, que se considera de 10euros/hora. Teniendo en cuenta que la duración del proyecto ha sido aproximadamente de unas 380h, el precio de mano de obra es:

$$\frac{10\text{€}}{h} \times 380h = 3800\text{€}$$

Por lo tanto el precio total del proyecto ha sido de:

$$3800\text{€} + 175\text{€} = \mathbf{3975\text{€}}$$

Bibliografía

- [1] L. L. P Castillejo, J-F Martínez, “LIBSVM: An internet of things approach for managing smart services provided by wearable devices,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 2, 2013.
- [2] B. C. D Antolín, N Medrano, “LIBSVM: A wearable wireless sensor network for indoor smart environment monitoring in safety applications,” *Sensors Journal*, vol. 17, no. 2, 2017.
- [3] M. Weiser, “The computer for the 21st century,” *Scientific American*, vol. 256, no. 3, 1991.
- [4] “Grafica que muestra el consumo de un nodo de una red de sensores inalambrica.” <https://www.intechopen.com/books/ict-energy-concepts-towards-zero-power-information-and-communication-technology-power-consumption-assessment-in-wireless-sensor-networks>.
- [5] R. P. Anil A. Agashe, Amrita A. Agashe, “Evaluation of DV hop localization algorithm in wireless sensor networks,” *Advances in Mobile Network, Communication and its Applications (MNCAPPS), 2012 International Conference*, 2012.
- [6] S. O. Amin Karbasi, “Distributed sensor network localization from local connectivity: Performance analysis for the HOP-TERRAIN algorithm,”
- [7] H. L. WenHua Cheng, Jia Li, “An improved APIT location algorithm for wireless sensor networks,”
- [8] M. R. Gholami, “Positioning algorithms for wireless sensor networks,” 2011.

- [9] “Hexiwear official web page.” <http://www.hexiwear.com/>.
- [10] “Hexiwear’s main mcu datasheet: Mk64.” <http://www.nxp.com/assets/documents/data/en/data-sheets/K64P142M120SF5.pdf>.
- [11] “Hexiwear’s ieee 802.15.4 mcu datasheet: Kw40z.” <http://www.nxp.com/assets/documents/data/en/data-sheets/MKW40Z160.pdf>.
- [12] “Esquema que muestra la interconexion de los elementos del hardware de hexiwear.” <http://www.hexiwear.com/hardware/>.
- [13] “Pagina web oficial de venta de las clickboards de mikroelektronica.” <https://shop.mikroe.com/click>.
- [14] “Hexiwear’s real time operating system: Freertos.” <http://www.freertos.org/>.
- [15] “IEEE 802.15.4 topologies.” http://ecee.colorado.edu/~liue/teaching/comm_standards/2010F_802.15/home.html.
- [16] J. Portilla, “Plataforma modular e interfaces genericas de transductores para redes de sensores inalambricas,” 2010.
- [17] “Cookie’s microcontroller: Aduc841 datasheet.” http://www.analog.com/media/en/technical-documentation/data-sheets/ADUC841_842_843.pdf.
- [18] “Datasheet del modulo zigbee de las cookies.” http://www.farnell.com/datasheets/1513914.pdf?_ga=2.1733937.1831669618.1497942567-729185089.1494255101.
- [19] “CC2520 radio module datsheet.” <http://www.ti.com/lit/ds/symlink/cc2520.pdf>.
- [20] G. Mujica, “Plataforma de integracion hardware-software para testbed de redes de sensores inalambricas,” 2012.
- [21] R. M. E. Elnahrawy, Xiaoyan Li, “The limits of localization using signal strength: a comparative study,” *Sensor and Ad Hoc Communications and Networks*, 2004.
- [22] S. U. A T Parameswaran, Mohammad Iftekhar Husain, “Is RSSI a reliable parameter in sensor localization algorithms: an experimental study,” 2009.
- [23] F. L. Jiuqiang Xu, Wei Liu, “Distance measurement model based on RSSI in WSN,” *Wireless Sensor Network*, vol. 2, no. 8, 2010.
- [24] M. S. Miroslav Botta, “Adaptive distance estimation based on RSSI in 802.15.4 network,” 2013.

Bibliografía

- [25] Q. Y. Ting Yang, Lihua Cheng, “Experimental study: a LQI-based ranging technique in zigbee sensor networks,” *International Journal of Sensor Networks*, vol. 19, no. 2.
- [26] W. K. Sharly Joana Halder, “A fusion approach of RSSI and LQI for indoor localization system using adaptive smoothers,” *Journal of Computer Networks and Communications*, vol. 2012, 2012.
- [27] X. L. Rui Tan, Guoliang Xing, “Adaptive calibration for fusion-based wireless sensor networks,” *INFOCOM Proceedings IEEE*, 2010.
- [28] H. H. Sergio K. Sobarzo, Sergio N. Torres, “Multipath routing for eVLBI,” *Computers and Geosciences*, 2010.

Bibliografía