

UNIVERSIDAD DE MENDOZA  
FACULTAD DE INGENIERÍA

TESIS DOCTORAL  
DOCTORADO EN INGENIERIA

Plataforma de Desarrollo de Laboratorios  
Remotos de Redes de Sensores Inalámbricos  
basados en Cloud Computing

Autor: Ing. Pablo Daniel Godoy

Director: Dr. Carlos García Garino

Co-Director: Dr. Ricardo Cayssials

Mendoza, Mayo de 2016



## Resumen

Las redes de sensores inalámbricos, o WSN por sus siglas en inglés, son sistemas formados por un gran número de dispositivos denominados nodos. Estos nodos poseen capacidad de procesamiento de datos, memoria, capacidad de comunicación inalámbrica y sensores de diferentes tipos. En algunos casos, estos nodos tienen la capacidad de actuar sobre el medio ambiente que los rodea. El desarrollo de estos dispositivos ha sido posible gracias a los avances en el campo de los MEMS (Sistemas Micro Electromecánicos), que han permitido el desarrollo de componentes electrónicos de pequeñas dimensiones y bajo costo.

Las WSN poseen actualmente un gran número de aplicaciones, por ejemplo: monitoreo del medio ambiente, agricultura, monitoreo de la salud, automatización de procesos y fábricas, automatización de edificios, aplicaciones militares, etc. Se espera que el número de aplicaciones crezca en los próximos años, como también el número de WSN y tipos de nodos. Frente a los métodos tradicionales de medición empleados en estas aplicaciones (instrumentos y estaciones meteorológicas), las WSN permiten una mayor densidad espacial de puntos de medición. Además permiten el acceso en tiempo real a los datos y monitoreos de larga duración.

Los laboratorios remotos son sistemas que permiten el acceso de manera remota a laboratorios de distinta naturaleza. Generalmente el acceso es a través de Internet, aunque puede ser a través de redes LAN de organizaciones o universidades. Existen en la actualidad laboratorios remotos de distinto tipo, por ejemplo, laboratorios remotos de física, de circuitos analógicos, de circuitos digitales, de robótica, de redes de computadoras, de WSN, etc (Marianetti, 2006; Steyn and Hancke, 2011). En general, son sistemas complejos, que incluyen un número considerable de componentes y de recursos tecnológicos.

Los laboratorios remotos constituyen una valiosa herramienta en los campos de la investigación científica, desarrollo de aplicaciones, formación y entrenamiento de usuarios y tareas de enseñanza. Estos permiten a los usuarios acceder a laboratorios con un gran número de equipos o instrumentos, o con equipamiento de costo elevado, que pueden estar ubicados en distintos lugares del mundo, al cual el acceso sería difícil o imposible por otros medios. Por ejemplo, existen laboratorios remotos de redes de computadoras con miles de nodos, que los usuarios difícilmente podrían montar localmente, pero que pueden ser accedidos de manera remota.

Los laboratorios remotos pueden clasificarse en dos grandes grupos: orientados a tareas de investigación científica o desarrollo de aplicaciones y orientados a tareas de formación y entrenamiento de usuarios. En el caso de laboratorios remotos orientados a investigación científica, el usuario posee un nivel de conocimientos muy elevado, y

puede llevar a cabo los experimentos sin ninguna restricción. Un laboratorio remoto orientado a tareas de formación y entrenamiento de usuarios está destinado a usuarios con variados niveles de conocimientos, e incluye componentes que facilitan el acceso y la configuración de experimentos por parte de los usuarios. En el caso de un laboratorio remoto destinado a formación y entrenamiento de usuarios sin conocimientos elevados, los experimentos que los usuarios pueden realizar están predefinidos y restringidos, de modo que no tengan que configurar completamente el experimento. Además, para el caso de tareas de docencia, los profesores o entrenadores necesitan poder realizar el seguimiento de los estudiantes, con lo cual surge la necesidad de componentes adicionales que se encarguen de esta tarea (Jona and Uttal, 2013).

Existe numerosos laboratorios remotos de WSN. La gran mayoría de ellos están destinados a tareas de investigación científica. Éstos permiten a los usuarios el acceso casi sin restricciones, pudiendo los usuarios programar los nodos con el sistema operativo y las aplicaciones que deseen. Las aplicaciones más importantes son testeos de nuevos protocolos de comunicación, aplicaciones, algoritmos, etc. Los usuarios pueden acceder de forma remota a la red, configurar y programar los nodos, recuperar los resultados de los experimentos y, en algunos casos, interactuar en tiempo real con los nodos de la WSN. Sin embargo, no existen en la literatura trabajos sobre laboratorios remotos de WSN orientados a educación o formación y entrenamiento de nuevos usuarios con pocos conocimientos sobre WSN o programación.

En los últimos dos años, algunos autores han propuesto la implementación de laboratorios remotos empleando servicios de Cloud Computing. Como se muestra en esta tesis, la utilización de servicios de Cloud Computing pueden brindar características útiles a los laboratorios remotos cuando éstos están orientados a educación.

**Palabras Clave:** Laboratorios Remotos, Redes de Sensores Inalámbricos, Laboratorios Remotos de Redes de Sensores

## Agradecimientos

Al Dr. Carlos García Garino y al Dr. Ricardo Cayssials, director y co-director de esta tesis doctoral, por toda su ayuda y sabios consejos para llevar adelante el trabajo de investigación cuyo resultado es esta tesis y cada una de las publicaciones que de la misma surgieron.

Al Dr. Carlos García Garino le agradezco que me haya permitido ser parte del ITIC, instituto en el cual desarrollé esta tesis, y por su ayuda a la hora de buscar el financiamiento de las becas sin las cuales yo no hubiera podido dedicar el 100 % de mi tiempo y esfuerzo a llevar adelante esta tesis.

A mis compañeros del ITIC, por transmitirme su experiencia, la cual ha sido de gran valor para mí, y por sus consejos y a veces sus necesarias y valiosas críticas a mi trabajo, que contribuyeron significativamente al resultado final presentado en este documento.

A las instituciones que financiaron esta tesis de doctorado a través de dos becas. La primera, financiada en conjunto por la Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT) y la Universidad Nacional de Cuyo, a través del Programa PRH-UNCuyo, desde abril de 2010 hasta marzo de 2014. La segunda beca, financiada por CONICET, desde abril de 2014 hasta marzo de 2016.

A la Universidad Nacional de Cuyo y el Instituto Tecnológico Universitario (ITU), carrera de Redes y Telecomunicaciones, por el espacio físico en el cual se encuentra el LAPIC, laboratorio en el cual ha sido desarrollada esta tesis.

A la Universidad de Mendoza, por permitirme cursar mis estudios de doctorado, y al jurado que en 2013 evaluó mi memoria y defensa de aptitud investigadora, integrado por la Dra. Cristina Párraga, el Dr. Rubén Santos y el Dr. Carlos Puliaffito, cuyos consejos y sugerencias fueron de gran utilidad para completar el trabajo en aquel entonces propuesto.

A mi familia, por el apoyo recibido durante todos mis estudios, y por enseñarme con el ejemplo a que el esfuerzo y el trabajo honrado conducen a lograr los objetivos que uno se propone en la vida.

A mi amada Roxana, mi compañera en esta vida, mi futura esposa. Por estar siempre a mi lado, por darme palabras de aliento y apoyo cuando fueron necesarias, por hacerme sentir que siempre tendré su apoyo sin importar lo que pase en la vida.

Por último, quiero agradecer a Dios, porque se que siempre estuvo a mi lado todo este tiempo, y por haber puesto en mi camino a las personas que de alguna u otra manera me han permitido llevar adelante esta tesis doctoral.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. La Tesis . . . . .	2
1.3. Contribuciones . . . . .	4
1.4. Organización de la tesis . . . . .	5
<b>2. Antecedentes</b>	<b>7</b>
2.1. Redes de sensores inalámbricos (WSN) . . . . .	7
2.1.1. Características generales de las WSN . . . . .	7
2.1.2. Aplicaciones de las WSN . . . . .	8
2.1.2.1. Monitoreo del medioambiente . . . . .	9
2.1.2.2. Cuidado de la salud . . . . .	9
2.1.2.3. Aplicaciones militares . . . . .	10
2.1.2.4. Aplicaciones en agricultura. . . . .	10
2.1.2.5. Aplicaciones industriales . . . . .	10
2.1.2.6. Domótica . . . . .	11
2.1.3. Nodos . . . . .	11
2.1.4. Requisitos de diseño de las WSN . . . . .	11
2.1.5. Topologías . . . . .	13
2.1.6. Protocolos de comunicación de datos . . . . .	13
2.1.6.1. Protocolos de la capa física y MAC . . . . .	14

2.1.6.2.	Protocolos de ruteo de datos . . . . .	15
2.1.6.3.	ZigBee . . . . .	16
2.1.6.4.	6LoWPAN . . . . .	16
2.1.7.	Técnicas para reducir el consumo de energía . . . . .	17
2.1.8.	Interconexión de WSN con redes TCP/IP . . . . .	17
2.2.	Herramientas utilizadas para investigación científica y enseñanza sobre las WSN . . . . .	18
2.2.1.	Modelos analíticos . . . . .	19
2.2.2.	Simuladores . . . . .	19
2.2.3.	Emuladores . . . . .	20
2.2.4.	Laboratorios remotos de WSN . . . . .	20
2.2.5.	Comparación entre simuladores y emuladores vs WSN reales . . . . .	20
2.2.5.1.	Comparaciones Experimentales . . . . .	21
2.3.	Laboratorios remotos . . . . .	22
2.3.1.	Introducción . . . . .	22
2.3.2.	Laboratorios remotos orientados a formación y entrenamiento y orientados a investigación . . . . .	23
2.3.3.	Caso de estudio: Laboratorios remotos de redes de computadoras . . . . .	26
2.3.3.1.	Emulab . . . . .	27
2.3.3.2.	ORBIT . . . . .	28
2.3.3.3.	PlanetLab . . . . .	29
2.3.3.4.	Características generales de los laboratorios remotos de redes de computadoras . . . . .	30
2.3.4.	Caso de estudio 2: Laboratorio remoto de programación de PLD . . . . .	31
2.3.5.	Un modelo de capas para laboratorios remotos orientados a educación . . . . .	33
2.4.	Cloud Computing . . . . .	34
2.4.1.	Modelo de servicios Cloud . . . . .	35
2.4.1.1.	Servicios de aplicación . . . . .	36
2.4.1.2.	Servicios de plataforma . . . . .	36

<i>ÍNDICE GENERAL</i>	IX
2.4.1.3. Servicios de infraestructura . . . . .	36
2.4.2. Modelos de despliegue . . . . .	37
2.5. Conclusiones . . . . .	37
<b>3. Trabajos Relacionados</b>	<b>39</b>
3.1. Arquitectura de los laboratorios remotos de WSN . . . . .	39
3.1.1. Nodos de un laboratorio remoto de WSN . . . . .	40
3.1.2. Sistema de interconexión . . . . .	42
3.1.2.1. Ethernet . . . . .	42
3.1.2.2. Puerto USB . . . . .	43
3.1.2.3. IEEE 802.11 . . . . .	43
3.1.2.4. IEEE 802.15.4 . . . . .	44
3.1.3. Operación en modo por lotes y en tiempo real . . . . .	45
3.1.4. Inyección de datos predefinidos . . . . .	46
3.1.5. Almacenamiento de datos . . . . .	47
3.1.6. Plataformas replicables . . . . .	48
3.1.7. Interface con los usuarios . . . . .	48
3.1.8. Sistema de reserva de turnos . . . . .	48
3.1.9. Nodos móviles y sistemas de localización . . . . .	49
3.1.10. Movilidad virtual . . . . .	51
3.1.11. Laboratorios remotos de WSN de tamaño reducido y de tamaño real	52
3.1.12. Herramientas de análisis in situ . . . . .	54
3.1.12.1. MoteLab Connectivity Daemon . . . . .	54
3.1.12.2. TestbedProfiler . . . . .	54
3.1.13. Laboratorios remotos de WSN geográficamente distribuidos . . . . .	56
3.1.14. Resumen . . . . .	57
3.2. Clasificaciones de los laboratorios remotos de WSN . . . . .	59
3.2.1. Clasificación de acuerdo a la implementación del sistema de gestión . . . . .	59
3.2.2. Clasificación de acuerdo a su tipo de objetivos . . . . .	59

3.2.3.	Clasificación de acuerdo a su estructura . . . . .	60
3.2.4.	Clasificación de acuerdo a características específicas . . . . .	60
3.3.	Análisis, problemas no resueltos y posibles mejoras en el campo de los laboratorios remotos de WSN . . . . .	61
3.3.1.	Diferentes resultados para un mismo experimento . . . . .	62
3.3.2.	Variabilidad de resultados . . . . .	63
3.3.3.	Empleo de un número reducido de plataformas de hardware . . . . .	64
3.3.4.	Necesidad de un mayor número de laboratorios remotos de WSN . . . . .	64
3.3.5.	Falta de laboratorios remotos para WSN desplegados al aire libre o industriales . . . . .	65
3.3.6.	Nodos que no son alimentados por baterías . . . . .	65
3.3.7.	No es posible realizar experimentos de larga duración . . . . .	66
3.3.8.	Necesidad de conocimientos avanzados de programación y de WSN . . . . .	66
3.3.9.	Experimentos centrados en los nodos de la WSN . . . . .	66
3.4.	Aplicabilidad de los laboratorios remotos de WSN actuales a formación y entrenamiento de usuarios . . . . .	67
3.4.1.	Componentes o características que afectan la aplicabilidad de los laboratorios remotos de WSN a formación y entrenamiento de usuarios . . . . .	67
3.4.2.	Análisis de la aplicabilidad de los laboratorios remotos de WSN actuales a formación y entrenamiento de usuarios . . . . .	69
3.5.	Conclusiones . . . . .	70
<b>4.</b>	<b>Laboratorios Remotos de WSN Orientados a Entrenamiento, Experimentación y Desarrollo</b> . . . . .	<b>71</b>
4.1.	Introducción . . . . .	71
4.2.	Hipótesis de trabajo . . . . .	72
4.3.	Modelo de capas propuesto . . . . .	72
4.3.1.	Requisitos de diseño de los módulos de gestión . . . . .	72
4.3.1.1.	Interface web didáctica, de fácil uso e intuitiva. . . . .	73
4.3.1.2.	Almacenamiento seguro de la actividad realizada por los usuarios . . . . .	73

<i>ÍNDICE GENERAL</i>	XI
4.3.1.3. Experimentos predefinidos . . . . .	74
4.3.1.4. Distinción entre distintos tipos de usuarios . . . . .	74
4.3.1.5. Posibilidad de acceso desde cualquier dispositivo con conexión a Internet . . . . .	74
4.3.1.6. Fácil adaptación a cualquier tipo de nodos . . . . .	75
4.3.1.7. Alta disponibilidad y robustez . . . . .	75
4.3.2. Modelo de capas de Marianetti . . . . .	75
4.3.3. Adaptación del Modelo de Marianetti para implementar un labo- ratorio remoto de WSN . . . . .	76
4.3.4. Modelo de capas para implementar el laboratorio remoto de WSN en el Cloud . . . . .	78
4.3.4.1. Interacción con aplicaciones externas . . . . .	79
4.4. Descripción e Implementación de los Componentes del Laboratorio Remoto	80
4.4.1. Proveedor de servicios de Cloud Computing . . . . .	80
4.4.2. Usuarios . . . . .	81
4.4.3. Interface Web . . . . .	81
4.4.4. Módulo de interacción con el usuario remoto . . . . .	83
4.4.5. Control de acceso . . . . .	83
4.4.6. Sistema de reserva de turnos . . . . .	84
4.4.7. Registro de actividad de los usuarios . . . . .	85
4.4.8. Base de datos . . . . .	85
4.4.9. Material de ayuda . . . . .	86
4.4.10. Interacción en tiempo real . . . . .	86
4.4.11. Laboratorio físico . . . . .	87
4.4.12. Sistema de interconexión . . . . .	88
4.5. Implementación basada en el Cloud vs implementación tradicional . . . . .	90
4.6. Conclusiones . . . . .	92

<b>5. Experimentos</b>	<b>95</b>
5.1. Experimentos de prueba de concepto . . . . .	95
5.1.1. Tiempo de respuesta de la WSN . . . . .	95
5.1.2. Potencia de RF recibida . . . . .	98
5.1.3. Tiempo de respuesta en función de la ocupación del canal de comunicaciones . . . . .	99
5.1.4. Formación de dos redes . . . . .	101
5.2. Experimentos de evaluación de desempeño . . . . .	101
5.2.1. Influencia de la latencia sobre los experimentos . . . . .	101
5.2.1.1. Influencia de la latencia sobre los resultados experimentales	102
5.2.1.2. Influencia de la latencia en la experiencia del usuario . . .	102
5.2.2. Medición de las latencias introducidas por los módulos de gestión del laboratorio remoto . . . . .	103
5.2.2.1. Intervalos de tiempos y latencias involucradas . . . . .	103
5.2.2.2. Metodología experimental . . . . .	104
5.2.2.3. Resultados experimentales . . . . .	105
5.2.3. Latencias introducidas por implementar los módulos de gestión en el Cloud respecto a un laboratorio remoto tradicional . . . . .	107
5.2.3.1. Metodología experimental . . . . .	107
5.2.3.2. Resultados experimentales . . . . .	108
5.3. Conclusiones . . . . .	108
<b>6. Conclusiones y Trabajos Futuros</b>	<b>111</b>
6.1. Contribuciones . . . . .	111
6.1.1. Una arquitectura para laboratorios remotos de WSN orientados a experimentación y generación de casos de estudio . . . . .	111
6.1.2. Una propuesta para el despliegue de laboratorios remotos em- pleando servicios de Cloud Computing . . . . .	111
6.1.3. Una propuesta para integrar dispositivos electrónicos a Cloud Computing . . . . .	112
6.1.4. Construcción de una interface web didáctica . . . . .	112

6.1.5.	Una nueva clasificación de los laboratorios remotos . . . . .	112
6.1.6.	Construcción de un prototipo que emplea nodos disponibles en el mercado local . . . . .	113
6.2.	Limitaciones . . . . .	113
6.2.1.	Reducido número de nodos . . . . .	113
6.2.2.	Latencia . . . . .	114
6.2.3.	Necesidad de probar la adaptabilidad a otro tipo de equipo bajo prueba . . . . .	114
6.2.4.	Necesidad de estudiar su aplicación a docencia desde el punto de vista pedagógico . . . . .	114
6.3.	Trabajos futuros . . . . .	114
6.3.1.	Posibles mejoras . . . . .	114
6.3.1.1.	Estudiar formas para disminuir la latencia introducida por el laboratorio remoto . . . . .	115
6.3.1.2.	Mejorar el sistema de seguimiento de usuarios . . . . .	115
6.3.1.3.	Incrementar el número de nodos y el tamaño del despliegue	115
6.3.1.4.	Estudiar la integración a plataformas de aprendizaje . . . . .	115
6.3.1.5.	Más opciones para la realización de los experimentos . . . . .	116
6.3.2.	Profundizar la investigación en cuanto a la integración de labora- torios remotos al Cloud . . . . .	116
6.3.3.	Estudio de los aspectos pedagógicos para su aplicación a tareas de docencia . . . . .	116
6.3.4.	Aplicación del laboratorio remoto a tareas de docencia . . . . .	117
6.3.5.	Implementación de un modelo simulado para interactuar con el laboratorio remoto . . . . .	117
6.4.	Observaciones finales . . . . .	119
<b>A.</b>	<b>Publicaciones Realizadas</b>	<b>121</b>
A.1.	Artículos publicados . . . . .	121
A.2.	Presentaciones en Conferencias . . . . .	122
A.3.	Artículos en estado de evaluación . . . . .	123
A.4.	Artículos en escritura . . . . .	123



# Índice de figuras

1.1. Arquitectura . . . . .	4
2.1. Arquitectura general de una WSN . . . . .	8
2.2. Arquitectura general de un nodo WSN . . . . .	12
2.3. Topologías de una WSN. . . . .	14
2.4. Cluster de Emulab. . . . .	28
2.5. Nodos de ORBIT, suspendidos desde el techo. . . . .	29
2.6. Esquema del laboratorio remoto propuesto por Marianetti. . . . .	33
2.7. Modelo de capas para laboratorios remotos destinados a educación propuesto por Marianetti. Figura obtenida de (Marianetti, 2006) . . . . .	34
3.1. Nodo de MoteLab . . . . .	41
3.2. Nodo de Kansei . . . . .	41
3.3. Arquitectura de Indriya . . . . .	44
3.4. Arquitectura de Sensei . . . . .	45
3.5. Robot y pista que siguen los nodos del laboratorio de WSN con nodos móviles Sensei-uu. Figuras obtenidas de (Rensfelt et al., 2010) . . . . .	50
3.6. Laboratorio de WSN de Lambrou <i>et al.</i> . . . . .	51
3.7. Nodos y marcas de posición de Emulab . . . . .	51
3.8. Kansei, ejemplo de laboratorio remoto de WSN de tamaño reducido. . . . .	53
3.9. Porción del mapa de conectividad de MoteLab. . . . .	55
3.10. Ejemplo de resultados producidos por TestbedProfiler. . . . .	56

3.11. Goodput producido por tres laboratorios remotos de WSN para dos protocolos de ruteo. Figuras obtenidas de (Langendoen, 2006) . . . . .	62
4.1. Modelo de capas para laboratorios remotos destinados a educación propuesto por Marianetti. Figura obtenida de (Marianetti, 2006) . . . . .	76
4.2. Modelo de capas de para laboratorios remotos de WSN . . . . .	77
4.3. Laboratorio remoto implementado mediante servicios de Cloud Computing	79
4.4. Configuración de parámetros en la interface web . . . . .	83
4.5. Comunicación de comandos y datos durante la realización de un experimento . . . . .	87
4.6. Nodos Coordinador y End-Device . . . . .	88
4.7. Sistema de interconexión . . . . .	89
4.8. Sistema de interconexión para laboratorio remoto tradicional y para laboratorio remoto basado en Cloud Computing . . . . .	91
5.1. Configuración del ciclo de trabajo de un nodo XBee . . . . .	96
5.2. Latencias y tiempos de respuesta . . . . .	103
6.1. Modelo de integración de laboratorio remoto con modelos simulados . . .	118

# Índice de tablas

2.1. Diferencias entre laboratorios remotos orientados a formación y entrenamiento y laboratorios remotos orientados a investigación . . . . .	25
3.1. Resumen de características de los laboratorios remotos de WSN . . . . .	57
3.2. Influencia de los diferentes componentes o características de los laboratorios remotos de WSN sobre su aplicabilidad en formación y entrenamiento de usuarios. . . . .	67
3.3. Análisis de la aplicabilidad de los laboratorios remotos de WSN actuales a formación y entrenamiento de usuarios . . . . .	69
5.1. Tiempos de respuesta del nodo en milisegundos . . . . .	97
5.2. Potencia transmitida por el nodo según el valor del parámetro PL . . . . .	98
5.3. RSSI recibido por los nodos para distintas potencias de transmisión del nodo coordinador (en dBm). . . . .	99
5.4. Tiempo de respuesta (en segundos) en función de la ocupación del canal de comunicaciones . . . . .	100
5.5. Tiempos de respuesta y latencias en milisegundos . . . . .	106
5.6. Latencia introducida por los módulos de gestión del laboratorio remoto para diferentes días en milisegundos . . . . .	106
5.7. Latencia de los módulos de gestión con y sin Cloud en milisegundos . . . . .	108



# Glosario

**CAPTCHA** Completely Automated Public Turing test to tell Computers and Humans Apart (prueba de Turing completamente automática y pública para diferenciar computadoras de humanos)

**CSMA/CA** Carrier Sense Multiple Access with Collision Avoidance (Acceso Múltiple por Detección de portadora con Anulación de Colisiones)

**html** HyperText Markup Language (lenguaje de marcas de hipertexto)

**IaaS** Infrastructure as a Service (Infraestructura como servicio)

**PaaS** Platform as a Service (Plataforma como Servicio)

**PHP** Lenguaje de programación para el desarrollo de contenido web dinámico del lado del servidor

**PLD** Programmable Logic Device (Dispositivo lógico programable)

**RPC** Remote Procedure Call (Llamadas a procedimientos remotos)

**RSSI** Received Signal Strength Indicator o Indicador de Fuerza de la Señal Recibida

**SaaS** Software as a Service (Software como Servicio)

**SAR** Sistema de Acceso Remoto

**TCP** Transmission Control Protocol (Protocolo de control de transmisión)

**WSN** Wireless Sensor Network (Redes de sensores inalámbricos)

**WSN** Wireless Sensor Networks (Redes de Sensores Inalámbricos)



# Introducción

## 1.1. Motivación

Existe un número considerable de publicaciones que reportan aplicaciones de las WSN en el campo de la agricultura. Por este motivo, en la provincia de Mendoza y la región, las WSN pueden llegar a convertirse en una valiosa herramienta en este campo. Pueden ser de utilidad en el estudio de diferentes fenómenos meteorológicos que afectan los cultivos, como ser heladas o eventos peligrosos para los cultivos, por ejemplo plagas. Prueba de ello es la existencia de una tesis de doctorado recientemente aprobada que emplea las WSN como herramienta en el estudio y predicción de heladas (Iacono, 2015). Las WSN pueden ser también de utilidad para el estudio del efecto de diferentes factores ambientales sobre la calidad de las frutas obtenidas, como también del efecto de los diferentes procesos de traslado, almacenamiento, procesamiento, etc. Además, ingenieros agrónomos o agricultores necesitan de datos espacial y temporalmente distribuidos para desarrollar sus tareas. Estos datos permiten aplicar de manera correcta técnicas de cultivo (deshoje, estrés hídrico, etc) y maximizar sus beneficios.

Para potenciar el uso de las WSN, de manera de permitir que éstas sean aplicadas ampliamente en la provincia y el país, nace la necesidad de contar con personal capacitado en el uso de las mismas. Por tanto, surge la necesidad de contar con herramientas que asistan en el proceso de formación y entrenamiento de los usuarios en el uso de las WSN.

El estudio del estado del arte de los laboratorios remotos de WSN permite ver que todos éstos han sido diseñados con propósitos de investigación científica. Éstos permiten a los usuarios acceder sin restricciones a los nodos de las WSN. Los usuarios pueden configurar el sistema operativo, los protocolos de comunicación, cargar componentes de software en los nodos, y realizar cualquier tipo de experimentos. Para realizar

estas tareas, los usuarios necesitan contar con conocimientos elevados sobre WSN y programación.

Sin embargo, un usuario sin conocimientos avanzados necesitará realizar inicialmente experimentos simples, y luego, a medida que adquiera mayores conocimientos, podrá realizar experimentos más complejos. Por lo tanto, para este usuario será muy difícil realizar sus primeros experimentos y prácticas con WSN empleando los laboratorios de WSN disponibles actualmente. Se concluye que es de gran utilidad el estudio y desarrollo de laboratorios remotos de WSN destinado a usuarios que deseen formarse y entrenarse en el uso de las WSN, de fácil uso, destinados a usuarios sin conocimientos avanzados de WSN ni de programación. Por este motivo, esta tesis propone un modelo conceptual para laboratorios remotos de WSN que cumple estas características, y una implementación de dicho modelo.

Los mecanismos utilizados para permitir el acceso remoto a los equipos bajo prueba empleados por distintos laboratorios remotos son similares, diferenciándose fundamentalmente por el sistema bajo prueba. Por este motivo, las conclusiones y aportes que puedan realizarse en la implementación de un laboratorio remoto de WSN, pueden ser útiles también para otro tipo de laboratorios remotos.

Pensando en los trabajos futuros que continuarán el trabajo presentado en esta tesis, se espera que el trabajo realizado sea el punto de partida de una línea de investigación más amplia y de largo plazo, destinada a estudiar la aplicación de laboratorios remotos y TIC en educación y desarrollo de aplicaciones. Un laboratorio remoto destinado a tareas de educación requerirá de herramientas especiales que permitan al docente realizar tareas de seguimiento y evaluación, no presentes en los laboratorios remotos de WSN existentes (Jona and Uttal, 2013). Esta tesis propone la implementación de módulos que sean de ayuda cuando este laboratorio remoto sea aplicado a tareas de docencia.

Como motivación adicional, se menciona el hecho de que los laboratorios remotos de WSN existentes actualmente emplean nodos sensores no disponibles en el mercado local. Por lo tanto, es de interés el desarrollo de un laboratorio remoto que emplee nodos sensores disponibles en el mercado local.

## **1.2. La Tesis**

En esta tesis se propone un enfoque para la implementación de laboratorios remotos de WSN para ser empleados por usuarios que deseen formarse, entrenarse o experimentar en programación y funcionamiento de las WSN, que pueda ser empleado por usuarios con pocos conocimientos sobre WSN o programación, como también para la realización de experimentos o desarrollos. La hipótesis principal de esta tesis doctoral es:

*“Se pueden desarrollar laboratorios remotos de WSN, que puedan ser utilizados por usuarios que deseen formarse, entrenarse o experimentar en programación y funcionamiento de las WSN, o en desarrollo de aplicaciones que utilicen WSN, mediante el adecuado diseño e implementación de sus módulos de gestión”.*

Para probar la hipótesis, se ha propuesto y desarrollado un modelo para laboratorios remotos de WSN que cumpla los requisitos planteados en la hipótesis. Como componentes distintivos del modelo para laboratorios remotos de WSN propuesto respecto a los existentes, este modelo emplea una interface web que permite realizar experimentos de manera fácil y rápida, de manera que pueda ser empleado para poner a prueba conceptos básicos de las WSN, sin necesidad de poseer conocimientos avanzados en WSN o programación. Se ha realizado también una implementación y puesta en funcionamiento de un prototipo, empleando nodos de WSN disponibles en el mercado local.

Pensando en futuras aplicaciones del laboratorio remoto propuesto a tareas de educación, se ha planteado la incorporación de un módulo para seguimiento de los usuarios, basado en el almacenamiento de la actividad que realizan. Esta información debe almacenarse de manera segura y confiable, de modo que pueda ser utilizada para seguimiento y evaluación de la actividad que los usuarios realizan.

La arquitectura propuesta se divide en tres sistemas:

- El sistema de gestión, compuesto por varios módulos de software que gestionan el funcionamiento del laboratorio remoto. Este sistema se despliega en la infraestructura provista por un proveedor de servicios de Cloud Computing. Dicho proveedor debe proveer servicios de almacenamiento de datos, autenticación de usuario y creación de agendas de turnos. Este sistema ejecuta la mayoría de las tareas que permiten que el laboratorio pueda ser accedido remotamente
- El sistema o equipo bajo prueba, que es el sistema sobre el cual se realizarán experimentos remotamente. Para este laboratorio remoto el sistema bajo prueba es una WSN.
- El sistema de interconexión, que conecta el sistema bajo prueba (en este caso una WSN) con el resto del laboratorio remoto.

Los distintos módulos que componen los tres sistemas mencionados anteriormente se encuentran desplegados en tres sitios diferentes, que son:

- Un laboratorio físico, en el cual se encuentra desplegada el equipo bajo prueba (la WSN).
- La infraestructura del proveedor de servicios de Cloud Computing.

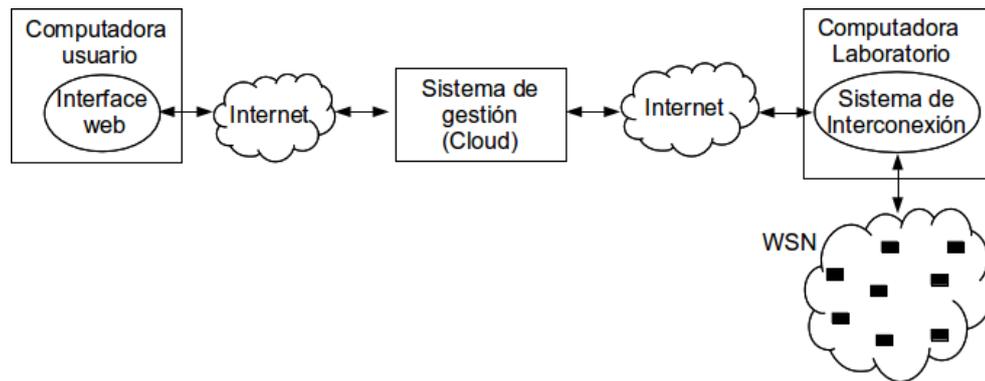


Figura 1.1: Arquitectura

- La computadora del usuario remoto.

La figura 1.1 muestra la arquitectura propuesta.

La interface web debe ser simple y de bajo costo de procesamiento, para permitir que los usuarios puedan acceder desde cualquier computadora, incluso teléfonos celulares. Debe ser didáctica, intuitiva y de fácil uso, de manera que puedan utilizarla usuarios con pocos conocimientos de WSN.

### 1.3. Contribuciones

Esta tesis presenta seis aportes en el campo de los laboratorios remotos de WSN y de laboratorios remotos en general, que se listan a continuación. Los dos primeros aportes son los aportes principales de esta tesis relacionados con la hipótesis enunciada en la sección 1.2. Los cuatro restantes, si bien no surgen directamente de la hipótesis planteada, también constituyen contribuciones al campo de los laboratorios remotos.

**Una arquitectura para laboratorios remotos de WSN orientados a usuarios que deseen formarse, entrenarse o experimentar en programación y funcionamiento de las WSN, o en desarrollo de aplicaciones que utilicen WSN:** Se han documentado en la bibliografía laboratorios remotos de WSN de diferentes tipos, pero ningún laboratorio remoto de WSN con la arquitectura propuesta en este trabajo. Esta arquitectura es el aporte principal de la tesis.

**Una propuesta para el despliegue de laboratorios remotos empleando servicios de Cloud Computing:** Se propone un nuevo modelo de implementación de laboratorios remotos no documentado en la bibliografía, el despliegue de varios componentes propios

de un laboratorio remoto empleando servicios de Cloud Computing. Como se muestra en esta tesis, el empleo de Cloud Computing brinda ventajas propias de esta tecnología a los laboratorios remotos.

**Una propuesta para interconectar dispositivos o equipos electrónicos al Cloud:** Se ha propuesto un sistema de interconexión entre los equipos del laboratorio físico y los módulos de gestión desplegados en el Cloud no documentado en la bibliografía, basado en una aplicación cliente-servidor. Dicho sistema podría ser utilizado para conectar cualquier dispositivo o equipo electrónico a componentes de software desplegados en el Cloud.

**Construcción de una interface web didáctica:** Esta interface permite a los usuarios programar la WSN de manera fácil y didáctica, de modo que los usuarios no necesiten conocimientos avanzados de programación ni WSN.

**Una nueva clasificación de los laboratorios remotos:** Se ha esbozado en la literatura la existencia de laboratorios remotos destinados a tareas de investigación científica, desarrollo o educación, pero ningún autor ha propuesto una clasificación formal ni ha realizado una comparación exhaustiva. Por este motivo, se propone en esta tesis una clasificación de los laboratorios remotos, distinguiendo entre laboratorios remotos destinados a investigación científica o desarrollo, y destinados a formación, educación y entrenamiento de usuarios.

**Construcción de un prototipo que emplea nodos disponibles en el mercado local:** Existe un número grande de nodos sensores comerciales disponibles en el mercado. Los laboratorios remotos de WSN construidos típicamente emplean nodos de la familia TelosB (Crossbow Technology, Inc., 2010), que permiten a los usuarios programar completamente los nodos. Sin embargo, estos nodos no están disponibles en el mercado local, ni son los más adecuados para usuarios que aún no poseen los conocimientos de programación necesarios para programar este tipo de nodo. Por ello, el prototipo emplea nodos de fácil programación y disponibles en el mercado local.

## 1.4. Organización de la tesis

El resto de la tesis se organiza como se describe a continuación. El capítulo 2 describe los conceptos teóricos abarcados en esta tesis. Incluye conceptos sobre WSN, laboratorios remotos y Cloud Computing. El capítulo 3 describe los trabajos relacionados con esta

tesis. Describe los módulos que forman un laboratorio remoto de WSN, una clasificación de los mismos, y un listado de problemas no resueltos en el campo de los laboratorios remotos de WSN. El capítulo 4 presenta la propuesta de esta tesis, que consiste en un modelo conceptual para laboratorios remotos de WSN destinados a usuarios que deseen formarse, entrenarse o experimentar en programación y funcionamiento de las WSN, o en desarrollo de aplicaciones que utilicen WSN, y una implementación del mismo. El capítulo 5 describe los experimentos realizados, que incluyen experimentos de pruebas de concepto y de performance. Por último, el capítulo 6 concluye esta tesis, indicando las contribuciones de la misma, las limitaciones y los trabajos futuros a realizar.

## Antecedentes

En esta sección se introducen conceptos teóricos sobre WSN, laboratorios remotos y Cloud Computing necesarios para el desarrollo de esta tesis.

### 2.1. Redes de sensores inalámbricos (WSN)

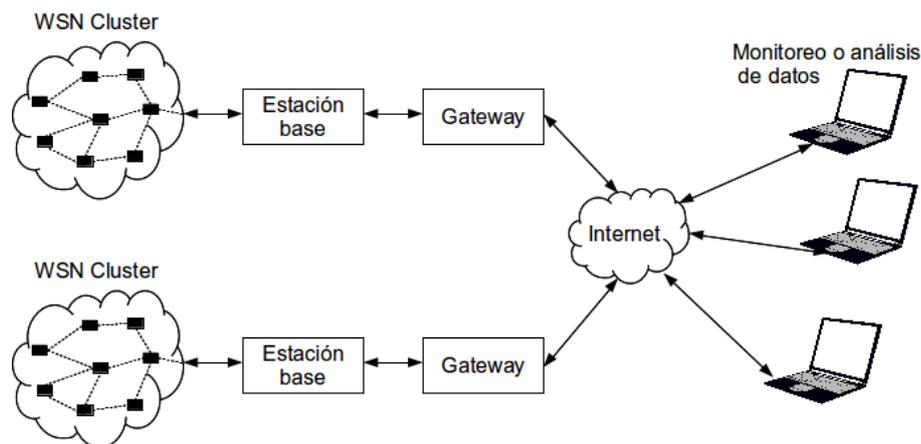
#### 2.1.1. Características generales de las WSN

Una WSN es un sistema integrado por dispositivos conocidos como nodos o motes, los cuales poseen capacidad de sensado, procesamiento de datos, almacenamiento de datos y comunicación inalámbrica (Yick et al., 2008)(Potdar et al., 2009)(Akyildiz et al., 2002). El desarrollo de estos dispositivos ha sido posible gracias a los avances en el campo de los MEMS (Sistemas Micro electromecánicos), que han permitido la integración de componentes electrónicos de pequeñas dimensiones y bajo costo. Al igual que los métodos tradicionales de medición (instrumentos y estaciones meteorológicas), las WSN permiten acceso en tiempo real a los datos y monitoreos de larga duración. Pero frente a los métodos tradicionales de medición mencionados anteriormente, las WSN permiten una mayor densidad espacial de puntos de medición.

La figura 2.1 muestra la arquitectura general de una WSN. Los nodos pueden comunicarse de forma inalámbrica con otros nodos y con la estación base. Los datos sensados pueden ser transmitidos a sistemas de computadoras para su análisis.

En general, todas las WSN tienen los siguientes tipos de nodos:

- Estación Base o Coordinador: Sus funciones más importantes son: controlar y gestionar la WSN, recolectar información sobre los nodos (nodos en funcionamiento, nodos con fallas, etc), e inicializar la WSN.



**Figura 2.1:** Arquitectura general de una WSN

- **Nodo sumidero:** Es el encargado de recolectar los datos generados por los demás nodos. Generalmente la estación base cumple esta función, aunque puede ser un nodo diferente.
- **Nodos router o simplemente nodos:** Tienen la función de retransmitir datos generados por otros nodos para que alcancen el destino final. También pueden sensor variables.
- **Nodos finales o End Devices:** Su función es recolectar datos y transmitirlos hacia nodos routers, para que puedan alcanzar el nodo sumidero. No pueden rutear datos, solo pueden comunicarse con el nodo ruteador. Este tipo de nodo puede ser de menor costo, ya que su protocolo de ruteo es muy simple: transmitir el dato al router más cercano. También puede tener un consumo de energía mucho menor al resto de los tipos de nodos, ya que solo se enciende para transmitir su dato y recibir cualquier mensaje destinado a él, y puede permanecer apagado el resto del tiempo, independientemente de las comunicaciones entre los demás nodos.

En algunas WSN, todos los nodos pueden cumplir la función de rutear datos. En cambio, en otras, puede haber nodos específicos con capacidad de rutear datos, mientras que otros sólo sensan variables y transmiten los valores sensados al nodo router más cercano.

### 2.1.2. Aplicaciones de las WSN

Existen en la literatura muchas publicaciones que reportan distintos tipos de aplicaciones que emplean WSN. Los trabajos de Gilbert (Gilbert, 2012) y Oppermann *et al.* (Oppermann et al., 2014) presentan estudios del estado del arte sobre aplicaciones de las WSN en general. También hay en la bibliografía estudios del estado del arte que presentan

y analizan los distintos tipos de aplicaciones de las WSN en áreas específicas, como se menciona a continuación.

#### **2.1.2.1. Monitoreo del medioambiente**

Abarca los diferentes despliegues que se realizan para recolectar datos con fines científicos. Oliveira *et al.* (Oliveira and Rodrigues, 2011) y Oppermann *et al.* (Oppermann et al., 2014) presenta un resumen de varios despliegues de este tipo de WSN realizados desde 2002. A continuación se mencionan, a modo de ejemplo, tres de los despliegues mencionados por Oliveira *et al.* y Oppermann *et al.* Se escogen estos despliegues por ser los más citados en la bibliografía:

- Estudio de patrones de anidamiento de aves en la isla Great Duck. Este despliegue de una WSN es mencionada por muchos autores como la primer WSN empleada para recolectar datos como parte de una investigación científica. El objetivo fue estudiar la influencia de la temperatura y la humedad en los patrones de anidamiento de aves. La WSN usada empleó 120 nodos para medir humedad y temperatura, y un sensor de radiación para detectar la presencia de aves.
- Monitoreo de variables ambientales en Sonoma Dust. Este despliegue, formado por 120 nodos sensores, se realizó para monitorear variables ambientales en un bosque de Secoyas, en California.
- Volcán Tungurahua, esta WSN se utilizó para monitorear la actividad del volcán Tungurahua en Ecuador. Los nodos sensores fueron equipados con micrófonos capaces de sensar señales infrasónicas.

#### **2.1.2.2. Cuidado de la salud**

Alemdar *et al.* (Alemdar and Ersoy, 2010) y Kumar *et al.* (Kumar and Lee, 2011) presentan estudios del estado del arte de aplicaciones de las WSN en el cuidado de la salud. El primer autor presenta un detallado resumen de 29 aplicaciones de las WSN en el cuidado de la salud. Entre las aplicaciones mencionadas por los autores, se encuentran:

- Sistemas para reconocimiento de patrones de actividad. Destinado a estudiar patrones de actividad que pueden ser síntomas de alguna enfermedad, como epilepsia.
- Sistemas para detección de caídas.
- Sistema de localización y seguimiento para ayudar a personas con discapacidades.

- Sistemas para el monitoreo de la ingesta de medicación.
- Monitoreo de signos vitales.

### **2.1.2.3. Aplicaciones militares**

Estudios del estado del arte sobre aplicaciones militares de las WSN pueden encontrarse en (Durisic et al., 2012; Bekmezci and Alagöz, 2009).

### **2.1.2.4. Aplicaciones en agricultura.**

Ruiz-Garcia *et al.* (Ruiz-Garcia et al., 2009) presenta un estudio del estado del arte sobre aplicaciones de las WSN en agricultura. Las aplicaciones de las WSN en agricultura pueden ser de dos tipos:

- Sistemas de medición de variables, entre ellas: irrigación, radiación solar que llega a los frutos, temperatura, humedad del suelo y del aire, etc. El monitoreo en el tiempo de estas variables ayuda a los agrónomos y agricultores a tomar decisiones para la aplicación de técnicas de cultivo en función de los objetivos que se desean realizar.
- Detección de eventos peligrosos, como plagas, alteraciones químicas del suelo, heladas, etc.

### **2.1.2.5. Aplicaciones industriales**

La aplicación de las WSN en ambientes industriales requiere de cuidados especiales, ya que en estos ambientes existen numerosas fuentes de interferencia electromagnética (motores, relés, contactores, etc), las cuales son perjudiciales para las comunicaciones inalámbricas. Aún así, las WSN ofrecen ventajas sobre las redes cableadas, por lo cual se han desarrollado variadas aplicaciones, como también adaptaciones de las WSN para que puedan ser utilizadas en ambientes industriales.

El trabajo de Mikhaylov *et al.* (Mikhaylov et al., 2012) presenta un estudio del estado del arte sobre distintas aplicaciones de las WSN en ambientes industriales, y distintos métodos para adaptar el uso de las WSN a este tipo de ambientes. Entre las adaptaciones mencionadas por el autor, se encuentra el desarrollo de protocolos de comunicaciones aptos para aplicaciones industriales, como WirelessHART y ISA100.11a. El trabajo de Mikhaylov *et al.* también menciona varias empresas y los productos que ofrecen para la aplicación de las WSN a ambientes industriales.

### **2.1.2.6. Domótica**

Entre las aplicaciones propuestas de las WSN en domótica o automatización de edificios pueden mencionarse: control de temperatura, humedad, luminosidad, detección de eventos peligrosos o intrusos, sistemas de reconocimiento de sonidos, etc. A pesar de haberse desarrollado un número importante de aplicaciones de las WSN en domótica, al día de la fecha, ningún autor ha realizado un estudio del estado del arte que liste las posibles aplicaciones de las WSN en domótica.

A modo de ejemplo se citan a continuación dos trabajos. El trabajo de Kelly *et al.* (Kelly et al., 2013) propone un sistema para monitorear a través de Internet variables ambientales en una vivienda, y para controlar el funcionamiento de artefactos domésticos (luces, calentadores, etc.) en función de los valores de esas variables. Wang *et al.* (Wang et al., 2014) presenta un sistema para reconocimiento de eventos sonoros basado en WSN para su aplicación en domótica.

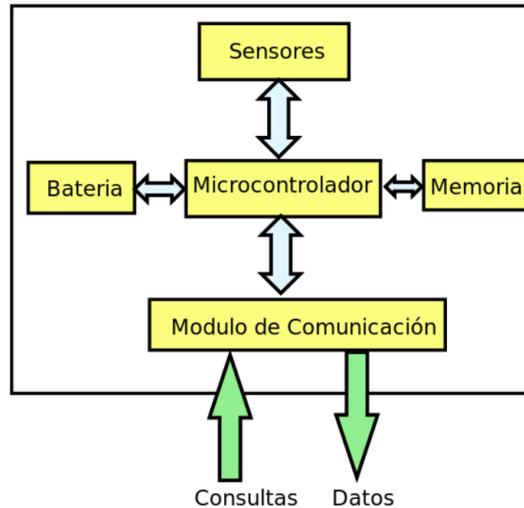
### **2.1.3. Nodos**

Los nodos están formados por un microcontrolador, memoria, fuente de alimentación (generalmente compuesta por baterías) y un módulo de comunicación inalámbrica, como se muestra en la figura 2.2. El procesador y la memoria son de reducida capacidad para mantener el costo y el consumo bajos. Lewis Girod *et al.* (Girod et al., 2004, 2007) distinguen dos tipos de nodos: los “motes” o “nodos”, con capacidad de procesamiento muy reducida (arquitecturas por lo general de 8 o 16 bits) y los “microservers” con mayor capacidad de procesamiento (por lo general arquitecturas de 32 bits). En comparación, los microservers pueden realizar tareas más complejas y pueden soportar sensores más sofisticados (por ejemplo, sensores con capacidad de transmitir datos utilizando el protocolo I2C), pero son más costosos y consumen más energía.

En cuanto al módulo de comunicaciones, en la gran mayoría de los nodos es un transceptor que implementa las capas físicas y la subcapa MAC del estándar IEEE 802.15.4 (IEEE, Institute of Electrical and Electronic Engineers, 2003). Éste, por lo general, permite modificar la potencia de transmisión, haciéndolo adaptable a la distancia a la que de los nodos deben comunicarse. De esta manera, puede adaptarse el consumo de energía al valor mínimo necesario y, por lo tanto, maximizar la vida útil de las baterías, y por tanto, la autonomía de los nodos.

### **2.1.4. Requisitos de diseño de las WSN**

Los nodos que componen una WSN deben cumplir una serie de requisitos de diseño. A continuación se mencionan los más citados en la bibliografía (Potdar et al., 2009):



**Figura 2.2:** Arquitectura general de un nodo WSN

- Capacidad para elevado número de nodos, para garantizar la cobertura espacial requerida por la aplicación.
- Costo reducido. La necesidad de un gran número de nodos impone el requisito de nodos de bajo costo, y por lo tanto, con reducidas capacidades de procesamiento y memoria.
- Vida útil suficiente, sin o con mínimo mantenimiento.
- Consumo de energía bajo. Teniendo en cuenta que los nodos son alimentados por baterías, y el mantenimiento debe ser bajo o inexistente, para maximizar la autonomía, el consumo de energía debe ser bajo.
- Robustez adecuada para soportar las condiciones ambientales bajo las cuales los nodos deben operar.
- Los nodos deben consumir toda la energía de las baterías. Con el fin de cumplir este requisito, la tensión mínima de funcionamiento de los nodos debe ser igual o menor que la tensión de corte de las baterías (Polastre et al., 2005).
- Fiabilidad alta. Para preservar la integridad de los datos en condiciones ambientales adversas.
- Autonomía adecuada según la aplicación. Esto significa que una WSN debe ser capaz de adaptarse a cambios en la topología debido a la reubicación o daño de los nodos, solucionar fallas, etc.

Estos requisitos de diseño y el número cada vez mayor de aplicaciones, han motivado a investigadores y empresas al desarrollo de plataformas de hardware y software específicas para WSN. Por otro lado, estos requisitos de diseño imponen al programador de aplicaciones dificultades extras a la hora de diseñar una aplicación.

### **2.1.5. Topologías**

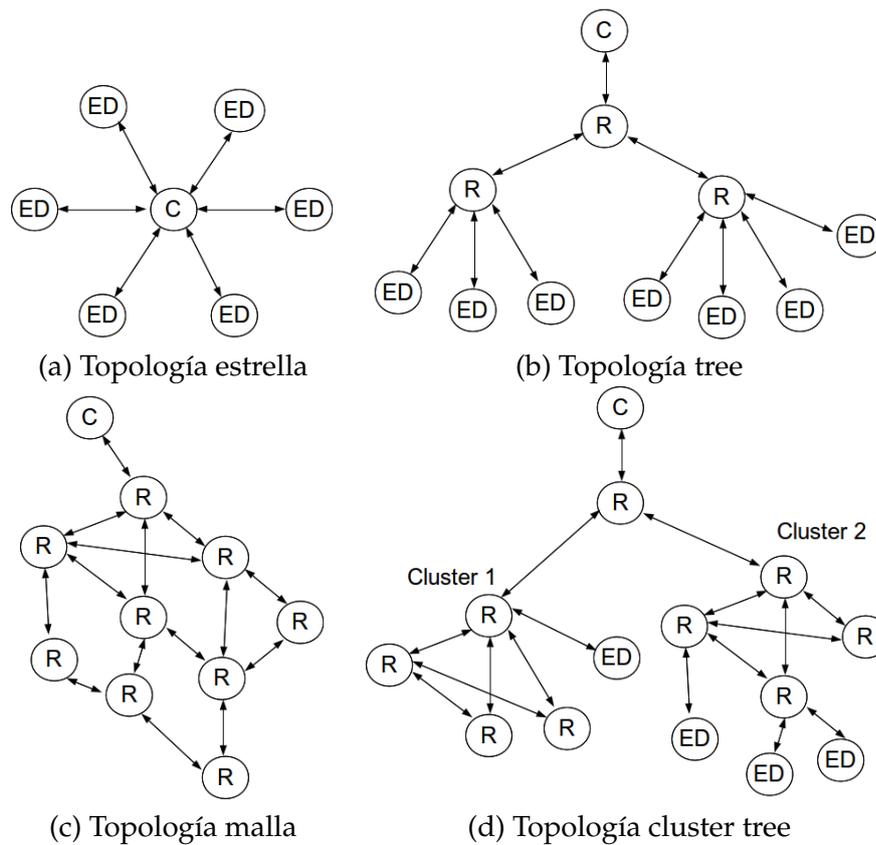
Las WSN pueden estar organizadas de acuerdo a cinco tipos de topologías, las cuales se describen a continuación.

- Topología punto a punto (peer-to-peer): permite la comunicación solo entre dos nodos.
- Topología estrella (star): Esta topología contempla varios nodos y una estación base. La comunicación es solo entre la estación base y los nodos, es decir, no existe comunicación entre los nodos. La estación base se encarga de iniciar la red y gestionarla.
- Topología árbol (tree): La comunicación se da de forma jerárquica, atravesando distintos niveles, en los cuáles cada nodo es hijo del nodo que le precede.
- Topología malla (mesh): Todos los dispositivos pueden comunicarse entre sí y participan en el ruteo de datos hacia y desde la estación base. En caso de que un nodo salga de servicio, el mensaje puede encontrar otro camino para llegar a destino.
- Topología cluster-tree: consiste de varias redes tipo estrella o maya, denominadas cluster, conectadas con la estación base vía nodos routers.

Cada una de las topologías posee ventajas y desventajas. Por ejemplo, la topología mesh o malla es la más robusta, ya que si un nodo sale de servicio, los mensajes tendrán mayores posibilidades de encontrar otro camino hacia el destino, pero requiere que todos los nodos posean capacidad de ruteo de datos, lo cual implica mayor costo y consumo de energía.

### **2.1.6. Protocolos de comunicación de datos**

Existe un gran número de protocolos de comunicación de datos para WSN (Baronti et al., 2007)(Singh et al., 2010), aptos para diferentes tipos de aplicaciones. Al igual que en las redes de computadoras, los protocolos de ruteo de datos para WSN se organizan en capas.



**Figura 2.3:** Topologías de una WSN.  
 Nota: C: Coordinador o estación base; R: router; ED: End Device o Nodo final

### 2.1.6.1. Protocolos de la capa física y MAC

En cuanto a la capa física y subcapa de control de acceso al medio (MAC), el estándar IEEE 802.15.4 (IEEE, Institute of Electrical and Electronic Engineers, 2003) se ha impuesto, siendo empleado por casi todos los nodos sensores para WSN disponibles en el mercado.

El estándar IEEE 802.15.4 define las especificaciones de la capa física (PHY) y la subcapa MAC. Permite topologías peer-to-peer y estrella. El protocolo define dos tipos de dispositivos, llamados FFD (Full Function Device) y RFD (Reduced Function Device). Los FFD son dispositivos capaces de operar como estación base o router, y deben estar conectados a una fuente de energía que se recargue periódicamente o a la red eléctrica. Están permanentemente activos y listos para recibir datos.

Los RFD son dispositivos que no pueden rutear datos, sólo pueden sensar variables y enviar los valores sensados al router o estación base más próximos. Generalmente permanecen la mayor parte del tiempo en modo bajo consumo, y se encienden para realizar tareas de sensado de datos y transmitir los resultados, o recibir un dato o comando desde un router o estación base.

### 2.1.6.2. Protocolos de ruteo de datos

Los protocolos de ruteo para WSN son muy diferentes a los protocolos de ruteo para redes de computadoras. En una WSN, los nodos pueden fallar, los enlaces no son confiables y los nodos poseen fuertes restricciones en cuanto al consumo de energía. Esto, sumado a que existe un gran y creciente número de aplicaciones, con objetivos y características diferentes, ha llevado al desarrollo de un gran número de protocolos de ruteo. Completos estudios sobre protocolos de ruteo para WSN pueden encontrarse en (Tyagi and Jain, 2012; Singh et al., 2010).

En general, todos los protocolos de ruteo asignan a cada nodo una dirección para identificarlos. La forma de asignar esas direcciones y la forma de rutear los datos definen los diferentes tipos de protocolos de ruteo. Los autores citados anteriormente clasifican los protocolos de ruteo en:

- Protocolos basados en localización: La dirección de los nodos está dada por su localización. Las decisiones de ruteo pueden tomarse de acuerdo a diferentes criterios, por ejemplo, un nodo que recibe un mensaje y debe tomar una decisión de ruteo, puede reenviar ese mensaje al nodo más cercano, al nodo vecino con mayor nivel de energía, al nodo más cercano al destino final, etc. Es necesario que los nodos puedan establecer su posición.
- Protocolos centrados en los datos: En estos protocolos, el nodo sumidero solicita por un determinado dato, por ejemplo, la temperatura promedio en determinada región, y los nodos que poseen ese dato, lo transmiten.
- Protocolos jerárquicos: Se basan en topologías cluster tree. Cada cluster posee un nodo cluster head o nodo cabecera. Para rutear los datos, los nodos envían sus datos al nodo cabecera de su grupo, y el nodo cabecera lo rutea a través de otros nodos cabeceras hacia el destino final, usualmente el nodo sumidero. Los grupos de nodos y el nodo cabecera cambian dinámicamente de diferentes maneras, y las diversas opciones sugeridas por diferentes autores ha dado origen a varios protocolos de este tipo.
- Protocolos basados en movilidad: Las WSN con nodos móviles plantean nuevos desafíos no presentes en WSN con nodos estáticos, que han dado lugar a diferentes propuestas de protocolos para WSN con estas características.
- Protocolos basados en multipath: En estos protocolos, cuando se necesita enviar un dato, se computan varios caminos entre el origen y el destino. Existen varios protocolos de este tipo, en algunos, los caminos comparten nodos y en otros, los caminos son totalmente independientes. El mensaje a enviar puede transmitirse de

forma paralela por todos los caminos hallados, o puede dividirse para que cada parte sea enviada por un camino diferente. Las ventajas de estos protocolos son mayor robustez y mayor balanceo de carga.

- Protocolos basados en heterogeneidad: Combinan nodos alimentados con fuentes de energía finita (baterías), con nodos alimentados con fuentes de energía no finita (por ejemplo, alimentación de la red eléctrica).
- Protocolos basados en calidad de servicio: Están pensados para aplicaciones donde los requisitos de calidad de servicio (por ejemplo, latencia, confiabilidad en la entrega de datos, tolerancia a fallas, etc), son tan importantes, o más importantes que las restricciones en consumo de energía.

Existen algunos protocolos de ruteo ampliamente usados en la práctica, los cuales se describen a continuación.

### **2.1.6.3. ZigBee**

Fue desarrollado por la Zigbee Alliance (ZigBee Alliance, 2008a). Define la capa de ruteo de datos y la capa de aplicación. Estas se definen sobre la capa MAC especificada por el estándar IEEE 802.15.4. La capa de aplicación está dividida en dos capas, la capa de soporte de aplicación (APS), y un conjunto de aplicaciones definidas por el propio estándar. El usuario puede escribir sus propias aplicaciones.

Permite topologías de redes malla (mesh), estrella y árbol (tree). Para poder implementar ZigBee, la plataforma de hardware elegida deberá implementar al menos la capa física definida por el estándar IEEE 802.15.4. A su vez, el microcontrolador deberá poseer capacidad de memoria y procesamiento suficiente para poder soportar y ejecutar el código que implementa el protocolo. Esto puede comprobarse si el fabricante aplica a su plataforma la etiqueta "Zigbee Compliant Platforms".

Algunas de sus aplicaciones son domótica, monitoreo de variables, juguetes, etc.

### **2.1.6.4. 6LoWPAN**

Su objetivo es proveer conectividad con redes IPv6, y la asignación a cada nodo de una dirección IPv6. Adopta como capas físicas y MAC las capas definidas por el estándar IEEE 802.15.4. Debido a que la longitud de una trama IPv6 es mayor que la longitud de una trama MAC del protocolo IEEE 802.15.4, 6LoWPAN agrega una capa llamada "capa de adaptación" entre la capa MAC y la capa de red. Esta capa de adaptación realiza tareas de compresión, fragmentación y reensamble de paquetes (Ma and Luo, 2008), (Yibo et al., 2011).

### 2.1.7. Técnicas para reducir el consumo de energía

En muchas aplicaciones los nodos sensores se alimentan con baterías. En estas aplicaciones, el consumo de energía del nodo sensor afecta su autonomía. El consumo de energía de un nodo en una WSN depende del consumo de energía del microcontrolador, de los sensores que pueda tener conectados y del transceptor, siendo este último usualmente el componente que mayor consumo de energía posee (Gholamzadeh and Nabovati, 2008). El consumo de energía de un microcontrolador o un transceptor depende de la frecuencia de funcionamiento, la tensión de alimentación y del número de periféricos internos encendidos.

Los nodos implementan varios mecanismos para reducir el consumo de energía:

- Detener la ejecución de instrucciones por parte del microprocesador, pero sin quitarle alimentación. Este modo de bajo consumo es usualmente denominado Standby.
- Detener el funcionamiento del transceptor. En este modo de bajo consumo, el transceptor apaga sus módulos de transmisión y recepción, de modo que no envía ni espera datos, pero no apaga su procesador interno.
- Apagar el microprocesador o transceptor (quitarles la energía eléctrica).
- Disminuir la frecuencia de trabajo o la tensión de alimentación del microprocesador o transceptor (Hempstead et al., 2008).
- Apagar periféricos internos.

Generalmente los microcontroladores y transceptores empleados para construir nodos de WSN poseen varios modos de bajo consumo, con diferentes niveles de consumo de energía cada uno. Mientras menor sea el consumo del microcontrolador, mayor será el número de funciones que se restringirán.

Existen diferentes mecanismos que permiten “despertar” a un nodo que está en modo bajo consumo. Por ejemplo: un temporizador, detección de una señal en un pin, que un dispositivo de hardware termine o comience una tarea (un conversor analógico digital, un módulo de comunicaciones, etc.). Utilizar los modos de bajo consumo puede provocar variaciones bruscas de corriente, las cuales son perjudiciales para las baterías.

### 2.1.8. Interconexión de WSN con redes TCP/IP

Existen trabajos que presentan esquemas para integrar WSN a redes TCP/IP, y de esta manera compartir los datos a través de Internet. Iacono *et al.* (Iacono et al., 2012) presenta un estudio del estado del arte sobre integración de WSN a redes TCP/IP.

## 2.2. Herramientas utilizadas para investigación científica y enseñanza sobre las WSN

Desplegar una WSN posee un costo que puede ser elevado, dependiendo del número y tipo de nodos utilizado. El nodo más económico que se comercializa en Argentina posee un costo de 78 dólares sin considerar las baterías ni cajas de protección (junio de 2015). Por este motivo, desplegar una WSN con 6 u 8 nodos puede ser inaccesible para la mayoría de los usuarios con fines de entrenamiento o formación. Algunos experimentos científicos o desarrollo de aplicaciones pueden requerir de WSN con un gran número de nodos, pudiendo ser también inaccesible para muchos investigadores.

Por este motivo se han desarrollado herramientas de diferentes tipos que permiten a los usuarios poner a prueba conceptos sobre WSN, llevar a cabo experimentos científicos o poner a prueba desarrollos de aplicaciones, sin necesitar desplegar una WSN real. Entre estas herramientas se encuentran los modelos analíticos, simuladores y emuladores. Uno de los cursos de doctorado tomados por el autor de esta tesis, denominado “Redes de Sensores Inalámbricos”, utilizó un simulador, Castalia (NICTA, 2010), para que los estudiantes realizaran todas las prácticas.

Estas herramientas no trabajan sobre un sistema real, sino sobre un modelo del mismo. Los modelos sobre los cuales se construyen estas herramientas son siempre una aproximación a la realidad. Por consiguiente, los resultados obtenidos serán siempre una aproximación a los resultados reales.

En el caso de las comunicaciones inalámbricas, existen varios factores que hacen que los modelos utilizados se alejen de la realidad. De acuerdo a Raychaudhuri *et al.* (Raychaudhuri et al., 2005), algunos de estos factores son:

- Las propiedades del canal de radio inalámbrico dependen del lugar específico donde se encuentre el nodo y sus alrededores. Por lo que cada despliegue tendrá características que afecten de manera distinta al canal de comunicaciones.
- La tasa de bits y la tasa de error de la capa física varían de manera muy compleja en el tiempo.
- Las interacciones entre diferentes capas de la pila de protocolos inalámbricos no pueden ser modeladas fácilmente.
- La movilidad de los usuarios, generalmente aleatoria, afectan el comportamiento del enlace.

### 2.2.1. Modelos analíticos

Ejemplos del empleo de modelos analíticos para estudiar WSN pueden encontrarse en los primeros trabajos sobre WSN, por ejemplo, en el trabajo de Liu *et al.* (Liu and Towsley, 2004). En dicho trabajo, los autores analizan la cobertura de una WSN (el área total cubierta), mediante el empleo de modelos analíticos. Se dice que un punto está cubierto si está dentro del área de sensado de al menos un sensor. Para este propósito, los autores utilizan dos modelos a los que denominan: 1) *modelo de sensado booleano* y 2) *modelo de sensado general*. Éstos se distinguen de acuerdo a la forma en que modelan el funcionamiento de un sensor.

Los autores consideran que se necesita detectar un fenómeno que está dado por una señal que se propaga (por ejemplo, señales sonoras u ondas sísmicas). En el *modelo de sensado booleano*, los sensores tienen un alcance  $r$ , dentro del cual el sensor puede detectar el fenómeno o evento. Fuera de ese alcance, el sensor no puede detectar eventos. El *modelo de sensado general* tiene en cuenta la degradación de la capacidad de detección del sensor, en función de la distancia entre el sensor y el punto en el cual se produce el fenómeno que se desea detectar. El nivel de la señal detectada por un nodo está dado por una expresión matemática que depende de la distancia.

Imran *et al.* (Imran et al., 2010) desaconseja el uso del modelado analítico por dos motivos: por un lado, debido a que requiere grandes simplificaciones que hacen que los modelos se aparten de la realidad, y por otro lado, debido a que la complejidad inherente de las WSN conducen a modelos muy complejos.

### 2.2.2. Simuladores

Son herramientas de software que modelan el comportamiento de un sistema real o ciertas partes del mismo, en este caso, una WSN. Su objetivo es llevar a cabo experimentos sobre WSN sin necesidad de soporte de hardware. Constituyen el método de evaluación de desempeño de WSN más empleado para diseñar y probar protocolos de redes para WSN.

Poseen como ventajas: bajo costo, facilidad de uso, ahorro de tiempo y escalabilidad de los experimentos. Como desventaja, su exactitud depende de la calidad del modelo empleado, el cual nunca es exacto. Cada simulador posee ventajas y desventajas frente a los demás simuladores. Por ejemplo, algunos dan resultados más precisos cuando el tamaño de los experimentos escalan, otros dan resultados más precisos en experimentos de larga duración, etc., por lo que para cada experimento, deberá elegirse el simulador que más se adapte a los objetivos del mismo. Estudios del estado del arte sobre simuladores para WSN pueden encontrarse en (Imran et al., 2010; Singh et al., 2008; Sundani et al., 2011; Zvkovic et al., 2014).

### **2.2.3. Emuladores**

Un emulador es una combinación de simulación y hardware real. Muchos autores consideran que un emulador es un tipo de simulador, en el cual se simula parte del comportamiento de una red, con algunos de sus componentes implementados en hardware real y otros simulados. Una ventaja importante de los mismos es que permiten ejecutar el mismo código que se ejecutaría en el sistema real (Imran et al., 2010; Singh et al., 2008).

### **2.2.4. Laboratorios remotos de WSN**

Un laboratorio remoto de WSN consiste en una WSN desplegada en un ambiente controlado, que los usuarios pueden utilizar para verificar sus desarrollos, como ser protocolos de comunicación, aplicaciones, etc. Los usuarios pueden acceder remotamente, configurar o programar los nodos, recuperar resultados experimentales, y en algunos casos, interactuar en tiempo real con los nodos de la WSN.

Es una herramienta intermedia entre un simulador y una WSN real. A diferencia de un simulador o un emulador, los usuarios pueden probar sus aplicaciones o realizar experimentos sobre hardware real. A diferencia de un despliegue real, los usuarios no necesitan comprar nodos sensores, desplegarlos en el campo, implementar herramientas para el seguimiento y análisis de datos, etc., debido a que un laboratorio remoto de WSN es un despliegue real que por lo general posee estos elementos.

La sección 2.3 describe los laboratorios remotos, mientras que el capítulo 3 está dedicado a los laboratorios remotos de WSN.

### **2.2.5. Comparación entre simuladores y emuladores vs WSN reales**

Las WSN reales producen resultados más precisos y confiables a la hora de realizar experimentos. Sin embargo, el despliegue de una WSN real con fines experimentales o para formación y entrenamiento de usuarios requiere mayores recursos financieros y tiempo respecto a los necesarios para el uso de un simulador o emulador, ya que requiere programar y reprogramar los nodos, desplegarlos en campo (o laboratorio), implementar herramientas para la extracción de datos de rendimiento, etc. (Singh et al., 2008).

Por este motivo, los experimentos realizados en un gran número de trabajos publicados en la bibliografía sobre WSN se basan en simulación (Singh et al., 2008). Muchos cursos que incluyen las WSN utilizan simuladores para realizar las prácticas. Sin embargo, un simulador y una WSN real pueden producir resultados diferentes, por diferentes motivos, algunos de ellos se enumeran a continuación:

- Simplificaciones y supuestos poco realistas en los modelos utilizados para el diseño de simuladores. Estas simplificaciones son especialmente importantes en la capa física y en los modelos de propagación de radiofrecuencias (Bouckaert et al., 2010; Imran et al., 2010).
- Comportamientos muy difíciles de modelar, tales como: interacciones no previstas entre capas del protocolo de comunicación, interferencias externas, comportamiento del medio ambiente, errores causados por el hardware o los controladores, propagación de las señales de radiofrecuencia, sincronización precisa de paquetes, interrupciones de hardware, etc. (Raychaudhuri et al., 2005; Bouckaert et al., 2010; Sheu et al., 2008).

### 2.2.5.1. Comparaciones Experimentales

Hay pocos trabajos que realizan comparaciones experimentales entre WSN reales y simuladores para WSN. A continuación se citan dos trabajos que comparan WSN reales contra simuladores y emuladores: El trabajo de Elhadi Shakshuki M. *et al.* (Shakshuki et al., 2011) y el trabajo de Langendoen (Langendoen, 2006).

El trabajo de Elhadi Shakshuki M. *et al.* (Shakshuki et al., 2011) muestra las diferencias entre los resultados producidos por un simulador y una WSN real para varios experimentos. Los experimentos consistieron en ejecutar un protocolo MAC para WSN en dos simuladores y en una WSN real. Los simuladores empleados por Shakshuki *et al.* fueron:

- NS-2 simulator (NS-2, 2009).
- Un simulador personalizado, implementado en Java, desarrollado por los autores.

Algunos de los resultados obtenidos por los autores son los siguientes:

- Los simuladores mostraron estimaciones de consumo de energía menores a las obtenidas por la implementación real. Los autores afirman que la estimación de consumo de energía producida por el simulador NS-2 fue 10% menor que la energía consumida por la WSN real. Los autores atribuyen estas diferencias a que el simulador NS-2 no tiene en cuenta la pérdida de energía producida por el calor y la pérdida de energía cuando el sensor pasa del estado bajo consumo a activo. En cuanto al simulador de Java personalizado, los autores no proporcionan datos numéricos sobre el consumo de energía estimado por el mismo, pero declaran que las diferencias con la implementación real son considerables.

- El experimento que analiza el consumo de energía de la WSN en función del tiempo muestra que la implementación real y las simulaciones tienen un comportamiento diferente.

Langendoen (Langendoen, 2006) llevó a cabo un experimento similar empleando dos laboratorios remotos de WSN y un simulador. Los nodos empleados por Langendoen emplean el sistema operativo TinyOS, y el simulador fue TOSSIM, el simulador del sistema operativo TinyOS (Levis et al., 2005).

Los resultados obtenidos en el simulador, en comparación con los obtenidos a partir de las WSN reales, muestran diferencias significativas. Langendoen atribuye estas diferencias al modelo de radio deficiente de TOSSIM. Estas diferencias llevan a Langendoen a declarar: "Los resultados producidos por el simulador difieren dramáticamente de los producidos por los laboratorios remotos".

Estos estudios muestran que un desarrollo o aplicación puede ser probado en un simulador para obtener resultados aproximados. Sin embargo, para conocer el efecto de las condiciones ambientales reales o el comportamiento del hardware, los experimentos deben ser llevados a cabo en WSN reales.

## **2.3. Laboratorios remotos**

### **2.3.1. Introducción**

Los laboratorios remotos son plataformas que permiten el acceso, de manera remota y a través de Internet, a laboratorios de distinta naturaleza. Existen laboratorios remotos dedicados a física, electrónica digital, redes de computadoras, WSN, etc. Algunos están orientados a tareas de investigación científica, mientras que otros están orientados a tareas de formación y entrenamiento o docencia. En general, son plataformas complejas, que incluyen un número importante de recursos tecnológicos (Marianetti, 2006; Steyn and Hancke, 2011).

Su arquitectura puede dividirse en dos partes diferentes. Por un lado, el sistema bajo prueba, que puede ser un laboratorio de física, de circuitos electrónicos digitales, de dispositivos lógicos programables (PLD) junto con sus módulos o herramientas de programación y testeo, de redes de computadoras, de WSN, etc. Esta parte puede considerarse un laboratorio en sí mismo, sobre el cual pueden llevarse a cabo experiencias de laboratorio de manera presencial.

La otra parte es el sistema que permite llevar a cabo las experiencias de manera remota. A través de la bibliografía, los autores proponen diferentes nombres para este sistema. En

este trabajo lo llamaremos “sistema de acceso remoto” o SAR. Este sistema incluye varios subsistemas o módulos con funciones específicas. En general, su arquitectura es similar en todos los laboratorios remotos, con algunas diferencias entre laboratorios remotos orientados a formación y entrenamiento de usuarios y laboratorios remotos orientados a investigación científica.

Uno de los componentes del SAR incluye las interfaces necesarias para interactuar con el sistema bajo prueba. Estas interfaces son diferentes según la naturaleza del laboratorio remoto. Por ejemplo:

- Un laboratorio remoto de física incluirá dispositivos que permitan manipular, mover, trasladar, etc., los dispositivos del laboratorio, como también instrumentos de medición e instrumentos para recuperar los datos que el experimento genere, pudiendo incluir cámaras.
- Un laboratorio remoto de PLD incluirá el kit de desarrollo y programación, interfaces para poder controlar dicho kit desde Internet e instrumentos para registrar los resultados del experimento realizado.
- Un laboratorio remoto de redes de computadoras por lo general incluye una interface que permite el acceso a través de Internet.

Los demás componentes de un laboratorio remoto por lo general son similares para todos los tipos de laboratorios remotos. Estos componentes incluyen:

- Bases de datos para almacenar resultados o configuraciones de los experimentos.
- Interface web que permite a los usuarios el acceso remoto al laboratorio.
- Software de gestión, que configura el sistema cuando un experimento debe comenzar, reconfigura e inicializa el sistema cuando un experimento finaliza para que pueda comenzar el siguiente, y controla el buen funcionamiento del sistema.
- Módulos de control y autenticación de usuarios.
- Material de apoyo.
- En el caso de laboratorios remotos orientados a docencia, se pueden incluir módulos para el seguimiento de estudiantes, evaluaciones, etc.

### **2.3.2. Laboratorios remotos orientados a formación y entrenamiento y orientados a investigación**

El estudio del estado del arte realizado en esta tesis ha permitido diferenciar dos tipos de laboratorios remotos:

- Laboratorios remotos destinados a investigación o desarrollo.
- Laboratorios remotos destinados a formación y entrenamiento de usuarios (y dentro de este grupo, laboratorios remotos destinados a docencia).

La diferencia más importante entre laboratorios remotos orientados a formación y entrenamiento de usuarios y orientados a investigación, es el nivel de conocimiento de los usuarios. En el caso de laboratorios remotos orientados a investigación, el usuario posee un nivel de conocimientos muy elevado, pudiendo llevar a cabo los experimentos sin la ayuda de ninguna otra persona. Los laboratorios remotos orientados a investigación permiten que el usuario acceda a los recursos del laboratorio sin ninguna restricción. Por ejemplo, un laboratorio remoto de WSN orientado a investigación, tal como MoteLab (Werner-Allen et al., 2005a), permite a los usuarios cargar cualquier programa o sistema operativo en los nodos, sin ninguna restricción.

Los usuarios de un laboratorio remoto destinado a formación y entrenamiento poseen un nivel de conocimiento muy variado, pudiendo ser alto, como también bajo. En el caso de un laboratorio remoto destinado a formación de usuarios sin conocimientos elevados, como el laboratorio remoto de PLD propuesto por Marianetti (Marianetti, 2006), los experimentos que los usuarios pueden realizar (en este caso estudiantes) están predefinidos y restringidos, de modo que los usuarios no tengan que configurar completamente el experimento. Por lo general el usuario no puede cargar cualquier programa al laboratorio remoto. Esto plantea la necesidad de nuevos componentes, tanto de software como de hardware, por ejemplo, programas pre-definidos, a los cuales los usuarios pueden agregar bloques de código o modificar variables. Por otro lado, el menor nivel de conocimientos de los usuarios hace que sea mayor la posibilidad de que cometan errores, con lo cual surge la necesidad de diseñar sistemas de protección más robustos. Además, si los laboratorios remotos se aplican a tareas de docencia, necesitan del seguimiento de los estudiantes, por ejemplo, para verificar que éstos realicen trabajos prácticos y aprueben exámenes, con lo cual surge la necesidad de componentes adicionales que se encarguen del seguimiento de las actividades realizadas por los usuarios (Jona and Uttal, 2013).

Un laboratorio remoto orientado a investigación generalmente no requiere que parte del experimento esté predefinido. Por ejemplo, los usuarios de un laboratorio remoto de redes de computadoras orientado a investigación (sección 2.3.3), usualmente desean cargar sus propios programas, sistemas operativos e incluso drivers. En cambio, un laboratorio remoto orientado a formación y entrenamiento requiere que parte de los experimentos esté predefinida, de modo que los usuarios no tengan que configurar el experimento desde cero, facilitándole así la realización de los mismos.

La duración de los experimentos también será diferente. Los experimentos realizados en un laboratorio remoto orientado a investigación pueden ser de corta o muy larga duración, días, semanas o más. Por ejemplo, un usuario que haya creado un nuevo protocolo de comunicación para WSN, necesitará probar el mismo durante varios días o meses, para incluir en el experimento el efecto de la descarga de las baterías o de diferentes condiciones climáticas. En cambio, en un laboratorio remoto orientado a formación y entrenamiento, los experimentos serán usualmente de corta duración.

Otra diferencia puede encontrarse en cuanto a la exactitud necesaria de los resultados. Si bien en el caso de un laboratorio remoto orientado a formación y entrenamiento los resultados deben ser cercanos a la realidad, puede no ser necesaria una gran exactitud, ya que el fin de los mismos es didáctico. En cambio, un laboratorio remoto orientado a investigación científica requiere la mayor exactitud posible en cuanto a los resultados.

En el caso de laboratorios multiusuario, también habrá diferencia en cuanto a la interacción de los usuarios. En un laboratorio remoto orientado a investigación científica, que es accedido por varios usuarios, en general los experimentos serán diferentes y no habrá interacción entre estos usuarios. En cambio, en el caso de un laboratorio remoto con fines de formación y entrenamiento, los experimentos llevados a cabo por los usuarios serán similares, o los mismos, y puede requerirse mucha interacción entre los usuarios, con el fin de que intercambien apreciaciones y conocimientos adquiridos durante los experimentos.

La tabla 2.1 resume estas diferencias.

**Tabla 2.1:** Diferencias entre laboratorios remotos orientados a formación y entrenamiento y laboratorios remotos orientados a investigación

	<b>Laboratorios remotos orientados a investigación</b>	<b>Laboratorios remotos orientados a formación y entrenamiento</b>
Usuarios	Investigadores y desarrolladores	Usuarios que desean formarse o entrenarse en el uso de una tecnología o estudiantes
Nivel de conocimientos de la tecnología	Elevado	Intermedio o bajo
Restricciones de acceso	Pocas o ninguna, el usuario puede realizar cualquier experimento	Restricciones importantes, experimentos predefinidos

Necesidad de predefinir parte de los experimentos	No. Los usuarios necesitan cargar sus propios experimentos y configuraciones, usualmente desde cero.	Si. Los usuarios interactúan con experimentos predefinidos.
Robustez de sistemas de protección	Baja, los usuarios poseen conocimientos elevados.	Elevada, usuarios con pocos conocimientos pueden cometer errores o hacer uso inadecuado.
Duración de los experimentos	Muy variable. Experimentos de corta o muy larga duración	Los experimentos son de corta duración, salvo excepciones
Exactitud necesaria de resultados	Muy elevada. Fines científicos.	No elevada. Fines didácticos.
Interacción entre usuarios	Poca o ninguna	Variable. Puede haber mucha o poca interacción.
Interfaces web	Diseño poco elaborado o incluso no existe interface web.	Diseño cuidadoso. El diseño de la interface web debe ser tal que el uso del laboratorio remoto sea simple, intuitivo y didáctico.
Datos que se almacenan en las bases de datos	Resultados de los experimentos.	Resultados de los experimentos. Si se usa en tareas de docencia, se necesita almacenar la actividad realizada por los usuarios para permitir el seguimiento de los mismos.
Distintos tipos de usuarios.	No. Generalmente todos los usuarios son investigadores.	Si se usa en tareas de docencia, se distingue entre estudiantes y docentes.

### 2.3.3. Caso de estudio: Laboratorios remotos de redes de computadoras

A modo de ejemplo, se presentará una breve introducción a los laboratorios remotos de redes de computadoras. Existen un número importante de laboratorios remotos de

redes de computadoras. Los primeros laboratorios remotos de redes de computadoras fueron diseñados hace alrededor de 20 años, y desde entonces diferentes autores han solucionado problemas reportados en la literatura y han introducido diferentes características y mejoras a los mismos. Un gran número de usuarios han utilizado estos laboratorios remotos de redes de computadoras para realizar sus experimentos. Esto ha generado una gran cantidad de bibliografía e información que se resume en esta sección.

Los laboratorios remotos de redes de computadoras están destinados fundamentalmente a permitir que los usuarios puedan desarrollar o modificar aplicaciones o protocolos de comunicaciones, tanto para redes de computadoras o sistemas distribuidos. El general, permiten acceso sin restricciones, es decir, no existen experimentos predefinidos, sino que el usuario puede programar completamente las computadoras que forman parte del laboratorio remoto, cargando en las mismas las aplicaciones que el usuario desee, y en algunos casos, el sistema operativo que el usuario desee.

Existen laboratorios remotos de redes de computadoras con miles de computadoras disponibles, con arquitectura distribuida, con cluster de computadoras dispuestos en diferentes partes del mundo, que permiten a los usuarios sumar sus clusters de computadoras a los mismos. Algunos laboratorios remotos de redes de computadoras permiten experimentos en redes cableadas, otros permiten realizar experimentos sobre redes inalámbricas, otros sobre redes mixtas. Algunos de los más importantes son ORBIT (Open Access Research Testbed for Next-Generation Wireless Networks) (Raychaudhuri et al., 2005; Rutgers University, 2005), PlanetLab (PlanetLab PlanetLab Consortium, 2003) y Emulab (Emulab, University of Utah, 2006). Se describirán brevemente las características de estos tres laboratorios remotos de redes de computadoras, para mostrar las características fundamentales de este tipo de laboratorios remotos.

#### **2.3.3.1. Emulab**

Emulab (Emulab, University of Utah, 2006) posee instalaciones en Estados Unidos, Canadá, Europa, Brasil, Sudáfrica y China entre otros países. Algunas de las instalaciones permiten realizar experimentos sobre redes cableadas y otras sobre redes inalámbricas. Posee dos instalaciones principales, una de 168 nodos localizada en la Universidad de Utah, y otra con 50 nodos localizada en la Universidad de Kentucky. Las computadoras de estas instalaciones poseen 5 interfaces Ethernet, una de las cuales es usada para propósitos de control, y las otras cuatro para realizar experimentos. Los nodos de Emulab son 100 % configurables, pudiendo el usuario utilizar el sistema operativo que desee.

El acceso a todos los nodos es a través de una interfaz de usuario. Esta interfaz se implementa por medio de un framework. Este framework proporciona abstracciones,



**Figura 2.4:** Cluster de Emulab.  
Figura obtenida de (University of Utah, 2006)

servicios, posibilidad de referenciar nodos y enlaces, etc. Este framework tiene, entre sus funciones, la de ocultar gran parte de la heterogeneidad de los diferentes recursos.

Los nodos inalámbricos poseen placas del proyecto GNU radio (GNU Radio project, 2006). Estos sistemas permiten configurar los parámetros de la capa física del protocolo de comunicaciones. La figura 2.4 muestra uno de los cluster de computadoras de Emulab instalados en la Universidad de Utah.

#### **2.3.3.2. ORBIT**

ORBIT (Open Access Research Testbed for Next-Generation Wireless Networks) (Raychaudhuri et al., 2005; Rutgers University, 2005) es un laboratorio remoto de redes inalámbricas de computadoras. Posee dos instalaciones. Una de las instalaciones posee 400 nodos, formando una grilla de 20x20, con un metro de separación entre ellos, desplegados en el interior de un edificio, que pueden comunicarse a través de IEEE802.11. La otra instalación, ubicada al aire libre, posee nodos que permiten la comunicación tanto a través de IEEE802.11 o 3G. Los nodos pueden ser dinámicamente interconectados.

Cada nodo es una PC, con un procesador de 1 GHz, 512 MB de memoria RAM y 20 GB de capacidad en su disco duro. Cada nodo posee dos placas de comunicación inalámbrica 802.11a/b/g con drivers Linux. El usuario puede configurar ciertos parámetros que controlan la comunicación inalámbrica, tales como potencia de transmisión o velocidad de transmisión. Varios parámetros pueden ser medidos, tales como: RSSI, potencia de transmisión, ruido, velocidad de entrega de datos, etc. Los nodos están conectados a través de una red Ethernet de alta velocidad, la cual es empleada con propósitos de programación y control. Además cuenta con instrumentos de medición de señales de radio y generadores de interferencia. Algunos nodos poseen interface Bluetooth. Los

nodos corren el sistema operativo Debian GNU/Linux, versión 2.6. Sin embargo, el usuario puede utilizar el sistema operativo que desee. Los usuarios tienen total control sobre los nodos, pudiendo instalar en los mismos cualquier sistema operativo y software, como también reiniciarlos.

Para realizar un experimento, el usuario debe cargar en los nodos el sistema operativo y software necesario, además de un script de configuración del experimento. Este script debe ser escrito en lenguaje Ruby. Cada nodo posee un programa denominado NodeAgent, que al comienzo del experimento, ejecuta el script de configuración cargado por el usuario. Este script configura las interfaces de comunicación Ethernet y las inalámbricas (de acuerdo a lo indicado por el usuario), y comienza el experimento. Los pasos a seguir por el experimento están indicados en dicho script.

La figura 2.5 muestra el laboratorio remoto ORBIT. Pueden verse los nodos suspendidos desde el techo, en el arreglo de 20x20 mencionado arriba.



**Figura 2.5:** Nodos de ORBIT, suspendidos desde el techo.  
Figura obtenida de (Rutgers University, 2005)

### 2.3.3.3. PlanetLab

PlanetLab (PlanetLab Consortium, 2003) es un laboratorio remoto de redes de computadoras creado en 2003 por PlanetLab Consortium. Posee una arquitectura distribuida, con la mayoría de sus nodos aportados por diferentes instituciones de investigación de diferentes partes del mundo.

Cualquier institución puede aportar nodos a PlanetLab, siempre y cuando los mismos cumplan las características mínimas requeridas (procesadores de 2.4Ghz, 4 GB de memoria RAM, 500 GB de disco duro, capacidad de arrancar desde USB o CD-ROM, y disponer de una IP pública por nodo).

Todos los nodos que forman parte de planetLab ejecutan un paquete de software ofrecido por PlanetLab Consortium, que incluye un sistema operativo Linux, mecanismos de inicialización, mecanismos de distribución de actualizaciones de software y herramientas de gestión. Además este software permite la capacidad de asignar una porción de los recursos de hardware de PlanetLab, o todos, a una aplicación, y que varias aplicaciones puedan correr simultáneamente en PlanetLab. Este paquete de software es ofrecido libremente, de modo que cualquier usuario puede desplegar un laboratorio remoto privado usando el software de PlanetLab.

PlanetLab cuenta actualmente con 1077 nodos distribuidos en 571 lugares del mundo. La página web de PlanetLab indica que el mismo ha sido usado por más de 1000 investigadores pertenecientes tanto al sector académico como al sector industrial. Algunas investigaciones para las cuales ha sido usado son: desarrollo de tecnologías para almacenamiento distribuido de datos, mapeo de redes, sistemas peer-to-peer, etc. Actualmente hay 600 proyectos empleando PlanetLab (datos publicados el 11/1/2013).

Todos los nodos están conectados a Internet. Al usuario se le asigna una porción (slice) de PlanetLab, que consiste en un conjunto de computadoras de este laboratorio remoto, durante un periodo de tiempo definido. A cada una de estas máquinas se le carga un sistema operativo de mínimo tamaño (Fedora Core 8 Linux). Una vez creado el slice, el usuario puede acceder a los nodos mediante comandos SSH. Mediante los mismos, el usuario puede cargar en los nodos sus propias aplicaciones y ponerlas a prueba. El usuario posee privilegios de superusuario, es decir, puede ejecutar cualquier tipo de operaciones, aun operaciones que podrían dañar el software de las máquinas, pero no puede realizar operaciones sobre el hardware.

Una de las ventajas principales de PlanetLab, es que provee un gran conjunto de máquinas distribuidas geográficamente, que experimentan congestión, fallas y comportamientos similares a la Internet real. Esta característica hace que PlanetLab sea un entorno bastante similar a la Internet real para la cual los usuarios crean aplicaciones.

Sólo usuarios cuyas instituciones son miembros de PlanetLab Consortium pueden acceder a PlanetLab. Una institución académica que desee ser miembro de PlanetLab Consortium debe contribuir con al menos un sitio con dos nodos, y no necesita pagar membresía. En cambio, empresas con fines de lucro deben pagar cuotas de membresía.

#### **2.3.3.4. Características generales de los laboratorios remotos de redes de computadoras**

El análisis de los laboratorios remotos de redes de computadoras realizado, nos permite obtener las siguientes conclusiones generales:

- No poseen experimentos predefinidos, el usuario debe cargar las aplicaciones desde cero, y en algunos casos, el sistema operativo.
- Los usuarios necesitan conocimientos avanzados de redes de computadoras.
- El usuario posee acceso casi sin restricciones. Tiene control casi total sobre el software, sistema operativo, y en algunos casos, parámetros del hardware.
- En el caso de laboratorios remotos de redes de computadoras cableadas, existen implementaciones distribuidas, donde todas las computadoras del laboratorio remoto ejecutan algún paquete de software común, que permite configurar y controlar los experimentos.
- Están orientados a investigación y desarrollo de aplicaciones, no a formación y entrenamiento de usuarios.

#### **2.3.4. Caso de estudio 2: Laboratorio remoto de programación de PLD**

Marianetti (Marianetti, 2006) presenta una implementación de laboratorio remoto para prácticas de programación de PLD orientado a tareas de docencia.

Un PLD, o dispositivo lógico programable, es un dispositivo que permite programar circuitos digitales a nivel de hardware. Consiste de un gran número de componentes de hardware elementales, como compuertas lógicas, temporizadores, etc., que no están conectados entre sí de fábrica. Es el usuario quien debe conectarlos, creando sus propios circuitos digitales. Estos dispositivos son reprogramables. Permiten crear circuitos digitales complejos, como microprocesadores. La programación se realiza mediante lenguajes denominados lenguajes de descripción de hardware, como por ejemplo, VHDL y Verilog HDL.

El laboratorio remoto implementado por Marianetti permite controlar de forma remota un kit de desarrollo de PLD, la programación de un dispositivo PLD conectado a este kit de desarrollo, y la verificación del funcionamiento del programa implementado en dicho PLD.

Los componentes que conforman el laboratorio remoto son:

- Un dispositivo PLD.
- Un kit de desarrollo conectado al dispositivo PLD.
- Una computadora, con un software de desarrollo de PLDs que puede ser comandado mediante la línea de comandos, característica clave para permitir el acceso remoto, y una aplicación servidor.

- Una aplicación embebida en un microcontrolador que permite asignar valores a los pines de entrada del PLD y leer los valores de salida producidos por el mismo.
- Un entorno de aula virtual al cual se delegan las tareas de autenticación de usuarios.

El entorno de aula virtual permite el acceso de los estudiantes a información de las diferentes materias que deben cursar. Cada estudiante posee una página con sus datos e información de las materias que cursa. Para poder ingresar a este entorno de aula virtual, los estudiantes deben ingresar con un usuario y contraseña. Una vez autenticado el estudiante, el mismo puede encontrar un link de acceso al laboratorio remoto.

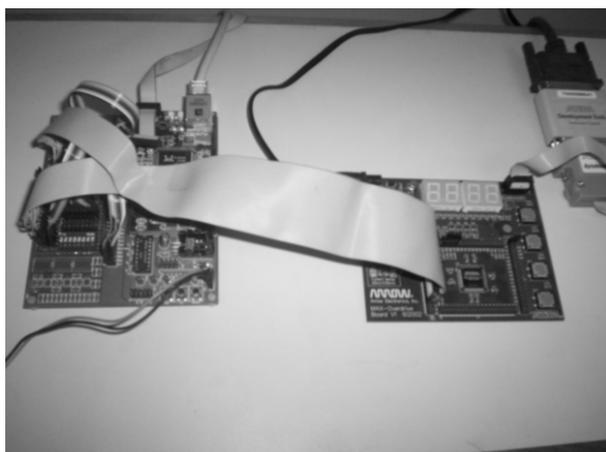
La interacción con los usuarios se realiza mediante una aplicación servidor, escrita en lenguaje PHP, instalada en la computadora mencionada anteriormente. En dicha computadora también se encuentra instalado el entorno de desarrollo de PLD.

En este laboratorio remoto, los experimentos a llevar a cabo están predefinidos, es decir, el usuario debe elegir entre un conjunto de experimentos que el sistema permite. Esto significa que cada experimento posee una parte del código que programa el PLD ya definido, y por lo tanto el usuario no lo puede modificar. Éste es el código que configura los pines del PLD.

El usuario puede escribir la otra parte del programa que se desea probar en el PLD. Luego de escribir el código en lenguaje VHDL, el usuario solicita la compilación del mismo a través de la interface web. Esta acción invoca a la aplicación servidor, la cual recibe el código enviado por el usuario y lo almacena en un archivo en el disco duro, y luego envía al entorno de desarrollo los comandos necesarios para que el mismo realice la compilación del código escrito en ese archivo. El entorno de desarrollo produce como salida un informe con los resultados de la compilación, que es enviado al usuario a través de la interface web.

El usuario puede verificar el código escrito asignando valores a los pines de entrada del PLD, y verificando el valor que toman las salidas. Para esto, la interface web permite al usuario elegir los valores a asignar a los pines de entrada del PLD. Esos valores son enviados a una aplicación servidor embebida en un microcontrolador. Esta aplicación genera en los pines de salida del microcontrolador los valores indicados por el usuario, los cuales se aplican a los pines de entrada del PLD. La aplicación en el microcontrolador también toma los valores producidos por los pines de salida del PLD (conectados a los pines de entrada del microcontrolador) y los envía al usuario, para que el mismo pueda verificar los resultados del experimento.

La figura 2.6 muestra los componentes de este laboratorio remoto. La placa que puede verse al centro de la imagen es el kit de desarrollo de PLD, con su correspondiente



**Figura 2.6:** Esquema del laboratorio remoto propuesto por Marianetti.  
Figura obtenida de (Marianetti, 2006)

PLD. La placa que puede verse a la izquierda es el microcontrolador que permite asignar valores a los pines de entrada del PLD y leer las salidas producidas por el mismo. A la derecha, puede observarse un cable que se conecta al puerto paralelo de la computadora donde se implementa el servidor php. Dicho cable constituye la interface con dicha computadora.

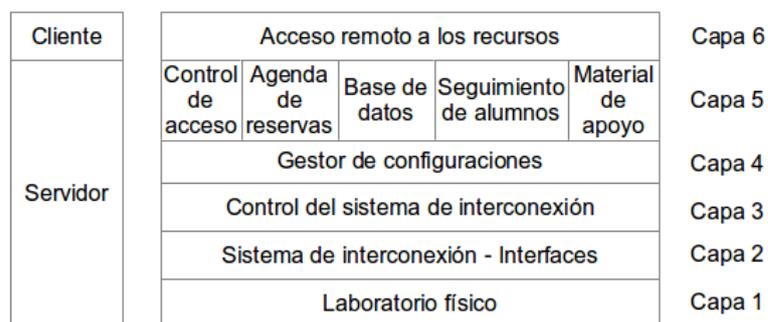
### **2.3.5. Un modelo de capas para laboratorios remotos orientados a educación**

La figura 2.7 muestra un modelo de capas para laboratorios remotos orientados a educación propuesto por Marianetti (Marianetti, 2006). Este modelo fue utilizado por el autor para implementar el laboratorio remoto de PLD descrito en la sección 2.3.4. Se describe a continuación su funcionamiento.

La capa 6, o capa de acceso remoto a los recursos, permite la interacción de los usuarios con el laboratorio remoto. En la mayoría de los laboratorios remotos, esta capa está implementada por una interface web. En el laboratorio remoto propuesto por Marianetti la autenticación de los usuarios se lleva a cabo mediante un entorno de aula virtual.

La capa 5 incluye varios bloques con diferentes funciones. Estos bloques incluyen: el sistema que controla el acceso a los recursos, la agenda de reservas que controla los turnos que posee cada usuario para utilizar el laboratorio remoto, una base de datos donde se almacenan los distintos experimentos predefinidos que el usuario puede realizar y resultados de los mismos, un bloque que permite el seguimiento de los estudiantes por parte del docente y material de apoyo.

La capa 4 o gestor de configuraciones se encarga de buscar la configuración del laboratorio remoto que se debe utilizar para la práctica seleccionada por el usuario



**Figura 2.7:** Modelo de capas para laboratorios remotos destinados a educación propuesto por Marianetti. Figura obtenida de (Marianetti, 2006)

y entregar dicha configuración a la capa inferior. La capa 3, o control del sistema de interconexión, se encarga de controlar el funcionamiento del sistema de interconexión en función de las configuraciones recibidas desde la capa superior.

La capa 2 es el sistema de interconexión o interfaces que conectan con el laboratorio. Está compuesto por interfaces que permiten actuar sobre los recursos físicos del laboratorio y recuperar la información producida por el mismo. Es común el uso de dispositivos embebidos en esta capa, por el bajo costo y la flexibilidad de los mismos. Estos pueden recibir comandos de las capas superiores y controlar relés (interruptores electrónicos), controlar motores, medir diferentes magnitudes físicas (mediante el uso de sensores y conversores analógico a digital), controlar cámaras de video, etc.

En el caso de que el laboratorio remoto sea de dispositivos embebidos, computadoras o dispositivos programables, la capa de interconexión puede estar formada por cables de conexión, más los kits de desarrollo de los dispositivos bajo prueba. En el caso de emplear los kits de desarrollo, éstos deben poder controlarse remotamente desde una PC.

Por último, la capa 1 está formada por el laboratorio físico sobre el cual se desean realizar los experimentos.

## 2.4. Cloud Computing

Cloud Computing es un modelo para permitir acceso bajo demanda y ubicuo a un conjunto compartido de recursos computacionales configurables virtualizados (por ejemplo redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente provistos y liberados con mínimo esfuerzo de administración o interacción con el proveedor de servicios (Mell and Grance, 2011). Estos recursos son ofrecidos por proveedores de servicios Cloud a clientes externos a través de Internet como servicios

Web, en base a acuerdos negociados entre los proveedores de servicios Cloud y los clientes (Buyya et al., 2009).

De acuerdo a (Mell and Grance, 2011) (Zhang et al., 2010) y (Buyya et al., 2009), algunas características distintivas de Cloud Computing son:

- Servicios provistos bajo demanda. Cada cliente puede ser provisto con las capacidades de computación que necesite, pudiendo ser muy diferentes los servicios proporcionados a cada cliente.
- Autoservicio. El proceso de proveer servicios al cliente es automático, sin requerir interacción humana con el proveedor de servicios.
- Acceso ubicuo: Los servicios Cloud generalmente son accesibles a través de Internet y hacen uso de Internet como una red de prestación de servicios. Por ello, cualquier dispositivo con conexión a Internet, ya sea un teléfono móvil, una tableta, una computadora portátil o una PC, es capaz de acceder a servicios Cloud.
- Agrupación de recursos. Los distintos tipos de recursos computacionales, que pueden ser muy variados y estar ubicados en diferentes sitios, son agrupados y ofrecidos al cliente como un conjunto de servicios, de los cuales el cliente puede escoger los que necesite.
- Asignación dinámica de recursos. Los recursos físicos y virtuales son asignados y liberados dinámicamente de acuerdo a la demanda de los clientes.
- Elasticidad y escalabilidad. Las capacidades computacionales pueden ser rápidamente modificadas, en algunos casos de forma automática, para escalar rápidamente en consonancia con la demanda. Esto hace que para el consumidor, las capacidades computacionales disponibles parecen ser ilimitadas y pueden ser solicitadas en cualquier cantidad y en cualquier momento.
- Virtualización: Los recursos computacionales (hardware y software) son provistos de manera virtualizada.
- Robustez: Los proveedores de servicios se encargan de la instalación, mantenimiento y actualización de software, respaldo y seguridad de los datos almacenados y calidad de servicio de acuerdo con los requerimientos del cliente.

#### **2.4.1. Modelo de servicios Cloud**

Un proveedor de servicios Cloud puede ofrecer tres tipos de servicios: Infraestructura, plataforma y aplicación.

#### **2.4.1.1. Servicios de aplicación**

Estos servicios se identifican en la bibliografía con la sigla SaaS (Software as a Service). Los recursos computacionales provistos al usuario son aplicaciones que corren en una infraestructura Cloud. Las aplicaciones pueden ser accedidas por múltiples clientes a través de una interface web, como un navegador web, o una interface incluida en el programa del cliente. El consumidor no gestiona ni controla la infraestructura subyacente, con la posible excepción de los ajustes de configuración de aplicaciones específicas.

Algunos ejemplos de SaaS son Google Docs (Google Inc., 2013), que permite crear en línea y multiusuario distintos tipos de documentos de texto, o Dropbox (Drago et al., 2012), que ofrece espacio de almacenamiento de archivos a través de Internet.

#### **2.4.1.2. Servicios de plataforma**

Estos servicios se identifican en la bibliografía con la sigla PaaS (Platform as a Service). El proveedor de estos servicios provee al cliente la capacidad de desplegar y ejecutar en la infraestructura Cloud aplicaciones creadas o adquiridas por el cliente, como también provee al cliente con entornos de desarrollo de aplicaciones, interfaces para la escritura de programas, librerías, servicios, sistemas operativos, herramientas de testeo, almacenamiento de datos, etc.

El cliente puede hacer uso de la infraestructura Cloud subyacente para ejecutar sus aplicaciones, pero no tiene el control ni puede administrar esta infraestructura. La infraestructura Cloud subyacente puede incluir servidores, redes, sistemas operativos o de almacenamiento de datos, etc.

Un ejemplo de PaaS es Google App Engine (Google Google Inc., 2013). Google App Engine permite el desarrollo de aplicaciones web en distintos lenguajes de programación, entre ellos Java, PHP, Python, etc. Además, provee interfaces para que el usuario pueda hacer uso de un conjunto grande de herramientas u otros servicios provistos por Google, como correo electrónico, base de datos, autenticación de usuarios, etc.

#### **2.4.1.3. Servicios de infraestructura**

Se refieren en la bibliografía con la sigla IaaS (Infrastructure as a Service). Los recursos computacionales ofrecidos al cliente son capacidad de procesamiento, cantidad de memoria RAM, tamaño de espacio de almacenamiento, componentes de redes, etc. El cliente puede desplegar y ejecutar software arbitrario, que puede incluir sistemas operativos y aplicaciones. El cliente no puede gestionar ni controlar la infraestructura

Cloud subyacente, pero tiene el control de los sistemas operativos, sistemas de almacenamiento y aplicaciones implementadas, y control limitado de componentes de red (por ejemplo firewalls).

Por ejemplo Amazon EC2 (Amazon Web Services, 2012) que permite que el cliente desarrolle aplicaciones para ser procesadas en máquinas virtuales especificadas por los clientes, denominadas instancias, provistas por Amazon.

## 2.4.2. Modelos de despliegue

De acuerdo con (Mell and Grance, 2011), hay cuatro modelos para el despliegue de infraestructuras Cloud:

- Cloud privada. La infraestructura Cloud es provista para el uso exclusivo de una sola organización que comprende múltiples consumidores (por ejemplo, unidades de negocio). Puede ser de propiedad, administrado y operado por la organización, un tercero, o alguna combinación de ellos, y puede existir dentro o fuera de las instalaciones de la organización.
- Cloud comunitaria. La infraestructura Cloud es provista para el uso exclusivo de una comunidad específica de organizaciones que comparten características (por ejemplo, la misión, los requisitos de seguridad, políticas, objetivos, etc.). Puede ser de propiedad, administrado y operado por una o más de las organizaciones de la comunidad, un tercero, o alguna combinación de ellos, y puede existir dentro o fuera de las instalaciones.
- Cloud pública. La infraestructura Cloud es provista para el uso abierto al público en general. Puede ser de propiedad, administrado y operado por una organización académica, una organización gubernamental, una empresa, o una combinación de ellos. Existe en las instalaciones del proveedor del Cloud.
- Cloud híbrido. La infraestructura Cloud es una composición de dos o más infraestructuras Cloud diferentes (privadas, comunitarias o públicas) que están unidas por tecnología estandarizada o propietaria que permite la portabilidad de datos y aplicaciones.

## 2.5. Conclusiones

En este capítulo se presentaron los conceptos teóricos en los cuales se ha basado esta tesis doctoral, y que son necesarios para la comprensión de la misma. En primer lugar

se presentaron las WSN, luego se presentaron los laboratorios remotos, haciendo una distinción entre laboratorios remotos destinados a tareas de formación y entrenamiento de usuarios, y destinados a tareas de investigación científica. Se presentaron dos casos particulares de laboratorios remotos. Primero, se presentaron los laboratorios remotos de redes de computadoras, ya que por su cercanía a las WSN, pueden aportar conceptos útiles para el desarrollo de un laboratorio remoto de WSN. En segundo lugar, se presentó un laboratorio remoto de PLD destinado a docencia, ya que este trabajo es la base del trabajo presentado en esta tesis doctoral. Por último se presentaron conceptos sobre Cloud Computing, ya que el modelo para laboratorios remotos de WSN presentado en esta tesis hace uso de servicios de Cloud Computing.

Puede verse que las WSN poseen un gran número e aplicaciones. El estado del arte sobre las WSN es extenso, existiendo numerosas publicaciones sobre las mismas durante los últimos 15 años, y siendo el número de publicaciones por año creciente. Además, puede verse que son sistemas complejos, y que el dominio de las mismas requiere conocimientos de varios campos, fundamentalmente electrónica e informática.

Los laboratorios remotos permiten acceder a laboratorios físicos de manera remota, usualmente a través de Internet. Existen laboratorios remotos de diferentes tipos. En este capítulo se enfocó el estudio en los laboratorios remotos de redes de computadoras y dispositivos electrónicos. Éstos permiten a los usuarios realizar experimentos científicos, probar nuevos desarrollos de software, protocolos de comunicación, etc., formarse y entrenarse en el uso de una tecnología y realizar tareas de docencia.

Las tecnologías de Cloud Computing ofrecen características que pueden ser útiles para los laboratorios remotos. Entre estas características se encuentran el almacenamiento seguro de información, la autenticación de usuarios y la disponibilidad permanente de los servicios a través de Internet.

## Trabajos Relacionados

Este capítulo comienza presentando y analizando trabajos relacionados con la construcción de laboratorios remotos de WSN, con la finalidad de identificar los distintos componentes y características propios de los mismos, y los diferentes métodos que varios autores han empleado para construirlos. Se presenta también una clasificación de los laboratorios remotos de WSN, y se listan y analizan problemas no resueltos. Posteriormente, teniendo en cuenta las características que deben cumplir los laboratorios remotos destinados a formación, entrenamiento o educación, presentadas en el capítulo 2, se analiza si los componentes de los laboratorios remotos de WSN presentados en este capítulo cumplen estas características.

### 3.1. Arquitectura de los laboratorios remotos de WSN

Los laboratorios remotos de WSN actuales poseen diferentes arquitecturas. Hay componentes y características comunes a todos ellos, pero también hay diferencias importantes. Esta sección analiza las diferentes arquitecturas para la implementación de laboratorios remotos de WSN. Para este propósito, se analizan diferentes enfoques documentados en la bibliografía sobre los siguientes elementos:

- Nodos (sistema bajo prueba)
- Sistema de interconexión
- Operación en modo por lotes o tiempo real
- Inyección de datos
- Almacenamiento de datos

- Plataformas replicables
- Interface con el usuario remoto
- Sistema de reserva de turnos
- Nodos móviles y sistemas de localización
- Movilidad virtual para laboratorios remotos de WSN estáticos
- Laboratorios remotos de WSN de escala real y de tamaño reducido
- Herramientas de análisis para laboratorios de WSN
- Laboratorios remotos de WSN geográficamente distribuidos

### **3.1.1. Nodos de un laboratorio remoto de WSN**

Los nodos de la WSN y los programas que los usuarios carguen en ellos constituyen el sistema bajo prueba. Para poder integrar los nodos a un laboratorio remoto de WSN es necesario agregar algunos componentes, como interfaces que conecten los nodos al resto del sistema, o sensores e instrumentos de medición que permitan medir variables de interés. Estos componentes son:

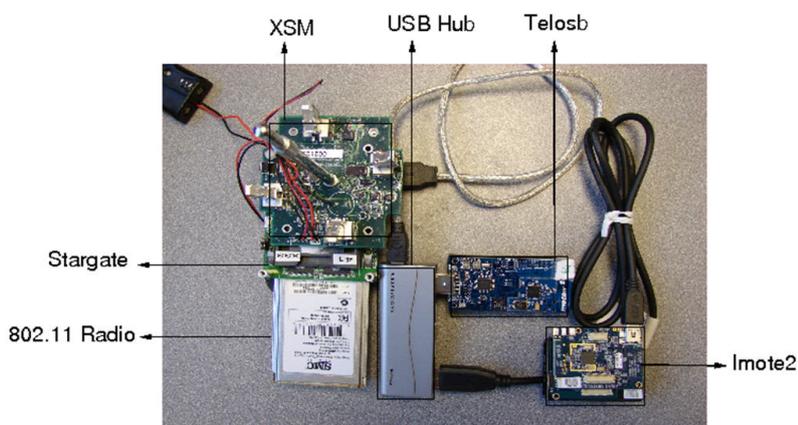
- Interfaces entre los nodos y el laboratorio remoto. La forma de implementar estas interfaces depende de cómo esté construido el sistema de interconexión. Por ejemplo, los autores de MoteLab (Werner-Allen et al., 2005a) utilizan placas Ethernet como interfaces, y los autores de WSNTB (Sheu et al., 2008), utilizan conversores serial a Ethernet, ya que ambos laboratorios remotos de WSN emplean un sistema de interconexión basado en redes Ethernet. Las placas Ethernet, o los conversores seriales a Ethernet, se comunican con los nodos, y a su vez se conectan a una red Ethernet, a través de la cual se conectan con el servidor central. En cambio, los autores de Neteye (Ju et al., 2012)(Wayne Wayne state University, 2012) e Indriya (M. Doddavenkatappa and Ananda, 2011), emplean conectores USB, ya que los nodos se comunican por cables USB con el resto del sistema.
- Sensores e instrumentos de medición. Por ejemplo, uno de los nodos de MoteLab (Werner-Allen et al., 2005a) posee un amperímetro que toma muestras de corriente eléctrica y las almacena en un archivo, que es entregado al usuario al finalizar el experimento. Por otro lado, cada nodo posee sensores de luz, temperatura y humedad. Como segundo ejemplo citamos a X-Sensor (Kanzaki et al., 2009), cuyos nodos están compuestos por un nodo MicaZ (Crossbow Technology, Inc., 2010), y están equipados con una placa de desarrollo, la cual posee sensores de luz, temperatura, humedad, presión atmosférica y un acelerómetro.



**Figura 3.1:** Nodo de MoteLab  
 figura obtenida de (Werner-Allen et al., 2005b)

La figura 3.1 muestra un nodo del laboratorio remoto de WSN MoteLab. Puede verse el nodo de WSN propiamente dicho, y al lado del mismo, una Interfaz Ethernet.

Los nodos de algunos laboratorios remotos de WSN son complejos, incluyen nodos de varios fabricantes, computadoras on-board, sensores, etc. Por ejemplo, cada nodo del laboratorio remoto de WSN Kansei (Ertin et al., 2006) está compuesto de 4 componentes distintos: (1) un nodo TelosB (Polastre et al., 2005), (2) un nodo Imote2 (Crossbow Technology Inc., 2015), (3) una computadora (la cual posee un microprocesador Intel de 400 MHz, sistema operativo Linux y varias interfaces, entre ellas, Ethernet, 802.11a/b, Bluetooth, etc.) y (4) sensores. La figura 3.2 muestra un nodo del laboratorio remoto de WSN Kansei. Pueden verse los diferentes componentes que forman este nodo.



**Figura 3.2:** Nodo de Kansei  
 figura obtenida de (KanseiGenie, 2006)

En general, los nodos utilizados por los laboratorios remotos de WSN actuales son similares a los anteriores. Estos nodos son de gran utilidad para realizar experimentos científicos o desarrollo de componentes de software, ya que permiten que el usuario pueda programarlos totalmente.

### **3.1.2. Sistema de interconexión**

Consiste en el sistema que conecta los nodos con el resto del laboratorio remoto. Es denominado back-end por la mayoría de los autores. El mismo permite la comunicación entre los nodos y el laboratorio remoto. Tiene diferentes funciones, entre ellas:

- Transferir los programas o configuraciones que los usuarios necesitan verificar cuando llega el momento de ejecutar un experimento. Estos programas deben ser enviados por los usuarios al laboratorio remoto mediante diferentes mecanismos, y pueden ser almacenados en las bases de datos del mismo.
- Transmitir los datos generados por los nodos durante el experimento para que sean almacenados en bases de datos o transmitidos al usuario. Estos datos pueden ser indicadores de performance propios del laboratorio remoto, datos generados por los nodos, los valores medidos por los sensores, etc.
- Permitir la interacción en tiempo real con los usuarios, en los laboratorios remotos que soporten este tipo de interacción.

El sistema de interconexión puede ser de dos tipos:

- Fuera de banda: Se utiliza como sistema de interconexión un canal de comunicaciones diferente al canal de comunicaciones propio del sistema bajo prueba, es decir, un canal de comunicaciones diferente al canal IEEE802.15.4.
- En banda: Se utiliza el propio canal de comunicaciones IEEE802.15.4 del sistema bajo prueba como sistema de interconexión.

Existen cuatro enfoques para la implementación del sistema de interconexión que se explican a continuación.

#### **3.1.2.1. Ethernet**

Algunos laboratorios remotos de WSN emplean una red Ethernet como sistema de interconexión entre los nodos y el resto del sistema. Tiene la desventaja de que los nodos sensores típicos normalmente no tienen soporte para redes Ethernet, por lo tanto, es

necesario hardware adicional que cumpla la función de interface, como tarjetas Ethernet conectadas a los nodos, o convertidores de serie a Ethernet o USB a Ethernet. Ejemplos de laboratorios remotos de WSN que emplean Ethernet como Back-End son: MoteLab (Werner-Allen et al., 2005a), WSNTB (Sheu et al., 2008) y w-iLab.T (Bouckaert et al., 2010).

### **3.1.2.2. Puerto USB**

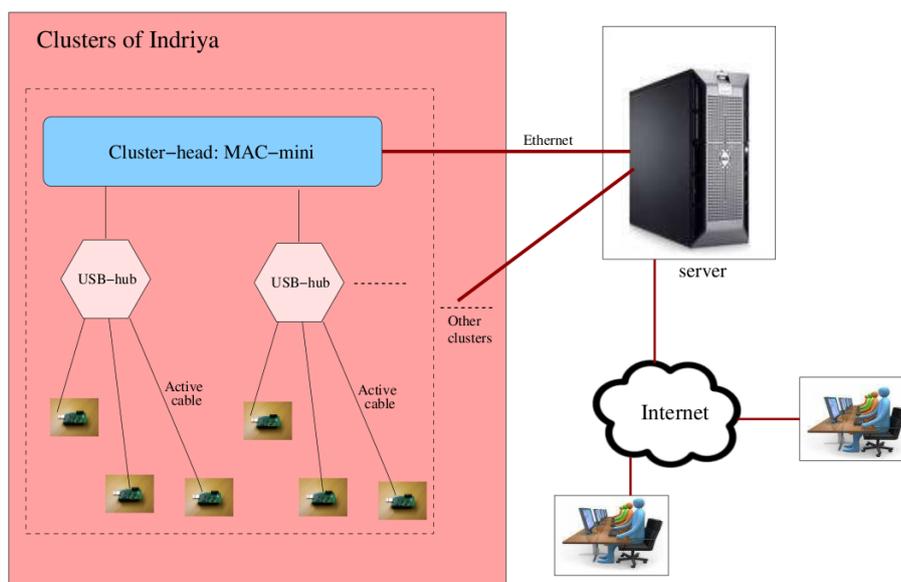
Algunos laboratorios remotos de WSN emplean cables USB para conectar los nodos con el resto del laboratorio remoto. La mayoría de los nodos comercialmente disponibles tienen conectores USB, por lo que no se necesita ninguna interface adicional para conectarlos al sistema. Sin embargo, la longitud máxima de un cable USB está limitada a 5 metros (M. Doddavenkatappa and Ananda, 2011). Como resultado, para lograr longitudes mayores, son necesarios componentes adicionales, tales como concentradores USB o computadoras. Estos laboratorios remotos de WSN están formados por grupos o clusters de nodos. Dentro de cada cluster, los nodos se comunican a través de cables USB, y los clusters se comunican con el resto del sistema por diferentes medios, tales como redes Ethernet o IEEE802.11.

Ejemplos: NetEye (Ju et al., 2012) posee clusters de nodos conectados mediante cables USB a computadoras portátiles, y las mismas se conectan con el resto del sistema mediante una red Ethernet (Wayne Wayne state University, 2012). Los nodos de Indriya (M. Doddavenkatappa and Ananda, 2011) se organizan en clusters, y se conectan con el resto del sistema mediante un sistema de interconexión de varios niveles. Los nodos se conectan a concentradores USB a través de cables USB activos (pueden acoplarse hasta formar cables de 25 metros de longitud), y estos concentradores USB se conectan a computadoras que cumplen la función de coordinadores de clusters. Por último, los clusters se conectan con un servidor central a través de una red Ethernet. La figura 3.3 muestra la arquitectura de Indriya, en la cual pueden verse los distintos componentes que forman parte de su sistema de interconexión.

### **3.1.2.3. IEEE 802.11**

En este caso, se utiliza comunicación inalámbrica IEEE802.11 para conectar los nodos con el resto del laboratorio remoto. La ventaja respecto a IEEE802.15.4 es mayor velocidad de comunicación con los nodos.

Sensei (Rensfelt et al., 2010) es un ejemplo de un laboratorio remoto de WSN que emplea comunicación inalámbrica IEEE802.11 para conectar los nodos con el resto del sistema. La figura 3.4 muestra la arquitectura de este laboratorio remoto de WSN. Como puede verse, los nodos se conectan a través de cables USB a dispositivos llamados “sensor



**Figura 3.3:** Arquitectura de Indriya  
 Figura obtenida de (M. Doddavenkatappa and Ananda, 2011)

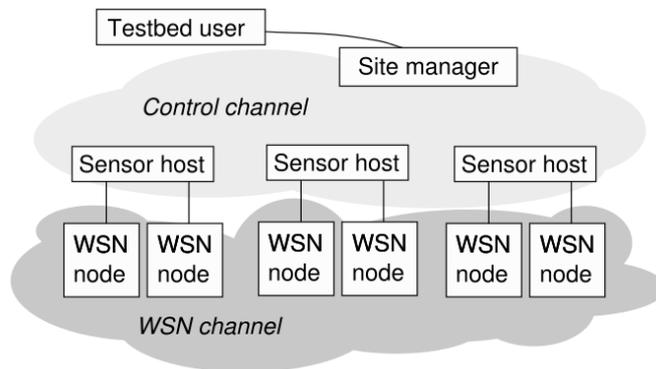
hosts”, que poseen capacidad de comunicación inalámbrica IEEE802.11. Los sensor host se comunican con el resto del sistema (site manager) a través del canal inalámbrico llamado “control channel”, mediante IEEE802.11.

Debido a que las normas IEEE802.11 y IEEE802.15.4 emplean la misma banda de frecuencias para transmitir señales de radiofrecuencia, es necesario conocer las posibles interferencias que puedan surgir, y su efecto sobre la WSN bajo prueba. Para estudiar esta interferencia, los autores de Sensei llevaron a cabo experimentos, donde evaluaron la interferencia entre señales IEEE 802.11 y 802.15.4. Los autores concluyen que es necesario tomar algunas precauciones para evitar interferencias (ver (Lambrou and Panayiotou, 2011) para más detalles).

#### 3.1.2.4. IEEE 802.15.4

En este caso, el canal de comunicación IEEE802.15.4 es compartido entre la comunicación normal dentro de una WSN y las tareas de programación de los nodos. Este método es el más económico y simple. Sin embargo, se emplea el canal IEEE802.15.4, que es uno de los elementos bajo prueba, para tareas de configuración del laboratorio remoto. Para evitar afectar el sistema bajo prueba, las tareas de ejecución del experimento y de configuración de los nodos no deberían ser simultáneas.

Ejemplo: X-Sensor (Kanzaki et al., 2009). Este laboratorio remoto de WSN posee una arquitectura distribuida en varias universidades de Japón. Está construido con



**Figura 3.4:** Arquitectura de Sensei  
figura obtenida de Rensfelt et al. (2010)

nodos MicaZ (Crossbow Technology, Inc., 2010). La programación a través del canal IEEE802.15.4 se realiza utilizando el protocolo XOTAP (over-the-air executable uploading protocol), que permite programar los nodos de la familia MicaZ a través del canal IEEE802.15.4.

Pocos de los trabajos analizados en el estado del arte hacen referencia a los componentes de software que utiliza el sistema de interconexión. Esto es debido probablemente a que conectar los nodos sensores a una computadora es una tarea trivial, que requiere los drivers adecuados y algún componente de software provisto por el fabricante de los nodos.

### 3.1.3. Operación en modo por lotes y en tiempo real

Durante el intervalo de tiempo asignado a un usuario, un laboratorio remoto puede funcionar en modo por lotes (batch) o en modo tiempo real, o ambos (esta terminología es utilizada por los autores de MoteLab, pero varía para otros laboratorios remotos de WSN).

En el modo por lotes, cada usuario tiene un intervalo de tiempo para la ejecución de sus experimentos. Al comienzo del experimento, el laboratorio remoto programa los nodos y el experimento se inicia automáticamente. Los datos generados durante el experimento se almacenan en una base de datos. El usuario no puede interactuar con su experimento hasta que el mismo haya terminado. En general, todos los laboratorios remotos de WSN utilizan el modo por lotes.

En el modo tiempo real, el usuario puede interactuar con los nodos mientras que el experimento se está ejecutando. Esta interacción consiste en el envío de comandos hacia los nodos y la recepción de datos generados por los nodos en tiempo real. Algunos

laboratorios remotos de WSN que emplean el modo en tiempo real son: MoteLab (Werner-Allen et al., 2005a) (que le da el nombre de "acceso en tiempo real") y WSNTB (Sheu et al., 2008) (que le da el nombre de "modo local"). A fin de lograr interacción en tiempo real, el método elegido por estos laboratorios remotos de WSN es una conexión TCP. A través de estas conexiones, los datos se transmiten en ambas direcciones.

Tanto MoteLab como WSNTB emplean mecanismos muy similares para lograr la interacción en tiempo real. En el caso de WSNTB, el software de gestión del laboratorio remoto crea  $n$  puertos TCP con una IP pública cada uno. El usuario debe crear  $n$  puertos TCP en su computadora, y conectarlos con los puertos TCP creados por el laboratorio remoto. De este modo, habrá  $n$  conexiones TCP entre el laboratorio remoto y el usuario. A cada conexión TCP se le asigna un nodo. Los autores proveen un software que permite al usuario crear puertos serie virtuales en su computadora, y conectarlos a las conexiones TCP. Los datos que dichos nodos envíen a través de sus puertos serie, se transmitirán a través de las conexiones TCP correspondientes, y el usuario los recibirá a través de los puertos seriales virtuales correspondientes. De este modo, el usuario ve los nodos como si estuvieran conectados a los puertos serie de su propia computadora.

Este enfoque, empleando puertos TCP tiene como ventaja la velocidad, ya que hay una conexión TCP dedicada entre cada nodo y la computadora del usuario. Sin embargo puede tener dos inconvenientes. Primero, requiere que se realicen operaciones a nivel del sistema operativo en la computadora del usuario, y algunas computadoras o dispositivos móviles podrían no tener capacidad para realizar estas operaciones. Segundo, algunas redes locales, que utilicen servidores proxy o firewalls para acceder a Internet, podrían no permitir un esquema como este, en el cual hay múltiples conexiones TCP entre la computadora del usuario y el laboratorio remoto.

### **3.1.4. Inyección de datos predefinidos**

Algunos laboratorios remotos permiten que los usuarios puedan inyectar datos predefinidos a los nodos. Esto tiene dos funciones:

- Permite analizar el comportamiento de los nodos cuando reciben mensajes desde el exterior.
- Servir como herramienta para tareas de depuración.

Estos datos deben estar predefinidos en un archivo. Este archivo debe ser enviado al laboratorio remoto cuando el usuario crea el experimento. NetEye (Ju et al., 2012; Wayne state University, 2012) es un laboratorios remotos de WSN que utiliza inyeccion de datos predefinidos. Es importante notar que la inyeccion de datos predefinidos no es interacción en tiempo real.

En el caso particular de NetEye, en el archivo deben indicarse el nodo al que se inyectan datos. Neteye posee una arquitectura de clusters conectados a través de una red Ethernet, siendo computadoras los dispositivos que cumplen la función de coordinadores de clusters. Para especificar el nodo que recibirá los datos, el usuario debe indicar la dirección IP de la computadora coordinadora de cluster, el número de nodo, el intervalo de tiempo entre eventos de inyección de datos, y los datos a inyectar.

Al momento de ejecutar el experimento, estos archivos se enviarán a las computadoras correspondientes, y las mismas se encargarán de inyectar los datos.

Este enfoque puede no ser el más útil en laboratorios remotos destinados a formación y entrenamiento de usuarios con pocos conocimientos, en los cuales se desea que los usuarios puedan interactuar en tiempo real con los nodos.

### **3.1.5. Almacenamiento de datos**

Con el fin de almacenar los datos, la mayoría de los laboratorios remotos citados en este trabajo usan bases de datos MySQL. En general, la información almacenada en estas bases de datos incluye:

- La información del usuario: Todos los laboratorios remotos de WSN analizados en este trabajo requieren que el usuario cree una cuenta, asignándole un nombre de usuario y contraseña. Además, algunos laboratorios remotos mantienen copias de los experimentos realizados por los usuarios y los resultados producidos, para que los usuarios puedan accederlos en el futuro. Algunos laboratorios remotos permiten que los usuarios consulten experimentos de otros usuarios (por ejemplo, MoteLab), en cambio, otros laboratorios remotos no permiten esta característica, siendo los experimentos de los usuarios de carácter privado (por ejemplo, NetEye).
- El estado de los nodos: Si el nodo está listo para ser programado, si está fuera de servicio, si está corriendo un experimento, etc.
- El estado del laboratorio remoto en general: Cantidad de nodos ocupados, cantidad de nodos libres, etc.
- Información sobre los experimentos pendientes (programas que se cargarán en cada nodo, archivos de configuración de los experimentos, etc).
- Los resultados producidos por los experimentos, los cuales pueden ser recuperados cuando el experimento finalice.

Los usuarios pueden acceder a estas bases de datos, pero generalmente con derechos de acceso restringidos.

Una excepción a este esquema es el laboratorio remoto de WSN Kansei (Ertin et al., 2006), que almacena los resultados de los experimentos en el sistema de archivos del servidor.

### **3.1.6. Plataformas replicables**

No todos los laboratorios de WSN permiten a los usuarios acceso remoto. Algunos laboratorios de WSN han sido concebidos como plataformas replicables, no como laboratorios remotos. El objetivo de estas plataformas replicables es que los usuarios puedan realizar experimentos en diferentes lugares del mundo utilizando una plataforma común. Estas plataformas están diseñadas con hardware y software comercialmente accesible. Algunos ejemplos son (Günes et al., 2008)(Lambrou and Panayiotou, 2011). (Rensfelt et al., 2010)

Estas plataformas replicables poseen arquitecturas similares a los laboratorios remotos de WSN, con la diferencia de que no poseen los componentes que permiten el acceso remoto, como ser interface web, sistema de reservas, base de datos, etc.

### **3.1.7. Interface con los usuarios**

Todos los laboratorios remotos de WSN, como también los laboratorios de WSN sin acceso remoto, utilizan algún mecanismo que permite a los usuarios interactuar con el mismo. De este modo, los usuarios pueden programar experimentos, cargar los programas con los cuales se programarán los nodos, recuperar resultados producidos por los experimentos, interactuar con el laboratorio remoto en tiempo real, etc.

La mayoría de los laboratorios remotos de WSN citados en este trabajo usan una interfaz web para la interacción con los usuarios. La mayoría de estas interfaces web se han programado empleando lenguajes de programación PHP y Javascript.

Una excepción es TWIST (Handziski et al., 2006), que no tiene una interfaz web. En TWIST, los nodos deben ser programados a través de scripts de Python, que se invocan remotamente a través de protocolo SSH.

### **3.1.8. Sistema de reserva de turnos**

La mayoría de los laboratorios remotos existentes implementan sistemas de reserva de turnos. Esto permite el acceso de manera ordenada a un número grande de usuarios. El sistema de reserva de turnos se encarga de permitir a los usuarios propietarios de un turno el acceso al laboratorio remoto en un determinado momento, y de configurar el sistema bajo prueba antes del comienzo de cada experimento.

### 3.1.9. Nodos móviles y sistemas de localización

Muy pocos laboratorios remotos de WSN utilizan nodos móviles. Algunos de éstos son (Rensfelt et al., 2010)(Lambrou and Panayiotou, 2011), sin embargo, los mismos no poseen acceso remoto. Mobile Emulab (University University of Utah, 2008) fue un laboratorio remoto de WSN accesible remotamente con nodos móviles, pero su sitio web declara que ha sido dado de baja. En todos los casos, estos laboratorios remotos de WSN utilizan robots para el transporte de los nodos.

Los autores han empleado dos mecanismos para lograr la movilidad de los nodos:

- Siguiendo caminos totalmente libres, como en (Lambrou and Panayiotou, 2011) y (University University of Utah, 2008).
- Siguiendo caminos predefinidos, como en (Rensfelt et al., 2010).

Es necesario que los laboratorios remotos de WSN implementen sistemas de localización que permitan conocer la ubicación de los nodos. Hay dos enfoques para implementar el sistema de localización:

- Uso de cámaras, por ejemplo (Lambrou and Panayiotou, 2011) y (University University of Utah, 2008).
- A través de marcas en el piso por las cuales los nodos pasan, y algún parámetro que indique la distancia recorrida, por ejemplo, el conteo de las revoluciones dadas por las ruedas, por ejemplo (Rensfelt et al., 2010).

El uso de cámaras es más preciso y no limita la movilidad de los nodos (excepto a la zona de cobertura de las cámaras), pero requiere el uso de algoritmos de reconocimiento de imágenes y cámaras. El segundo enfoque está sujeto al error de los sistemas de conteo de revoluciones y los nodos tienen que pasar a través de las marcas, lo que puede limitar la movilidad de los nodos.

Sensei-UU (Rensfelt et al., 2010) es un laboratorio para WSN que tiene como objetivo permitir la reproducibilidad de los experimentos en WSN con nodos sumideros móviles. Este laboratorio de WSN no posee acceso remoto, su objetivo es ser una plataforma replicable de experimentación. Los nodos móviles de este laboratorio remoto están implementados mediante robots seguidores de línea, los cuales portan un nodo sensor TelosB (Polastre et al., 2005) y un smartphone. Las trayectorias que los nodos móviles pueden seguir son fijas, y están implementadas mediante cintas en el suelo. El sistema de localización de los nodos se implementa mediante marcas en el suelo, también implementadas con cintas, y el conteo de las revoluciones de las ruedas de los robots. El



(a) Robot



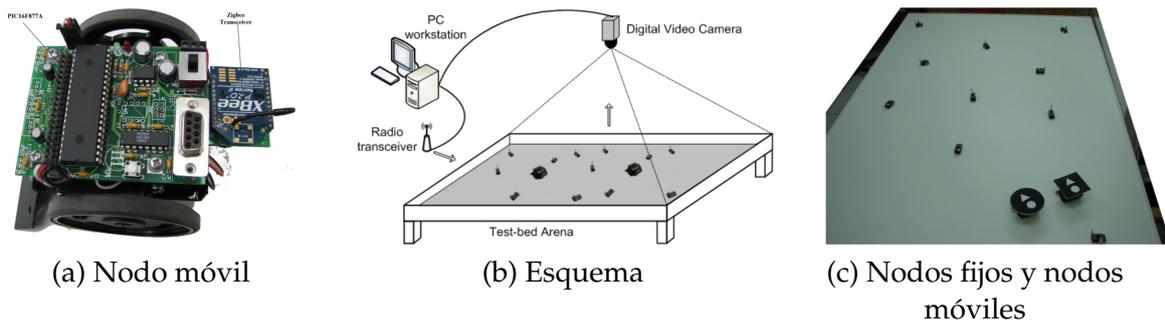
(b) Pista y marcas de posición implementadas con cintas

**Figura 3.5:** Robot y pista que siguen los nodos del laboratorio de WSN con nodos móviles Sensei-uu. Figuras obtenidas de (Rensfelt et al., 2010)

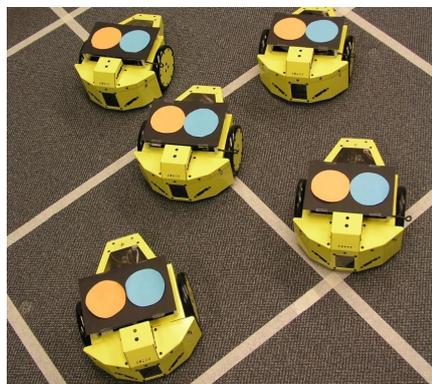
sistema de interconexión está implementado inalámbricamente (IEEE802.11). La figura 3.5 muestra un robot de Sensei y una imagen de la pista que los robots siguen. Pueden verse la pista y las marcas de posición implementadas con cintas.

Lambrou *et al.* (Lambrou and Panayiotou, 2011) desarrollaron un laboratorio para WSN con nodos móviles y nodos fijos. Dicho laboratorio no posee acceso remoto, ya que es una plataforma replicable de experimentación. Los nodos fijos se implementan mediante nodos MICAz (Crossbow Technology, Inc., 2010), y los nodos móviles se implementan utilizando un microcontrolador PIC16F877A, un robot Mark III, y una radio XBee Pro (Digi International Inc., 2008). La figura 3.6.a muestra una foto de un nodo móvil. La posición de los nodos fijos es conocida. Los nodos móviles no siguen caminos predefinidos, sino que pueden moverse libremente dentro de un área definida.

El sistema de localización de este laboratorio de WSN se basa en el empleo de una cámara. Cada nodo sensor posee dos marcas que permiten que puedan ser fácilmente detectables con una cámara de baja resolución. A partir de un algoritmo de procesamiento de imágenes, el laboratorio de WSN puede calcular la posición de los nodos móviles y entregar dicha información al usuario. La figura 3.6.b muestra un esquema de este laboratorio de WSN, donde puede verse la posición de la cámara utilizada para la localización de los nodos. La figura 3.6.c muestra una foto del laboratorio de WSN, donde pueden verse 10 nodos fijos, de menor tamaño, y 2 nodos móviles, con las marcas que facilitan su localización. Los autores mencionan como problema de este sistema la elevada cantidad de información a procesar para obtener la posición de los nodos,



**Figura 3.6:** Laboratorio de WSN de Lambrou *et al.*  
Figuras obtenidas de (Lambrou and Panayiotou, 2011)



**Figura 3.7:** Nodos y marcas de posición de Emulab  
figura obtenida de (University University of Utah, 2008)

que puede resultar en demoras inadmisibles, dependiendo de la velocidad a la cual se mueven los nodos.

Mobile Emulab (University University of Utah, 2008) utilizó una combinación de ambos sistemas, es decir, el empleo de una cámara y nodos con marcas que facilitan su localización (dos puntos de diferentes colores), y marcas de posición en el suelo (una cuadrícula), como puede verse en la figura 3.7.

### 3.1.10. Movilidad virtual

Una de las principales deficiencias de los laboratorios remotos de WSN actuales, mencionada por varios autores, es la falta de movilidad (Lambrou and Panayiotou, 2011)(Rensfelt et al., 2010). Puccinelli y Giordano (Puccinelli and Giordano, 2009) proponen una solución basada en movilidad virtual llamada ViMobiO (Virtual Mobility Overlay). Los autores proponen la utilización de nodos lógicos, que pueden moverse a través de la red, siendo los nodos reales (es decir, los nodos físicos) posibles ubicaciones de los nodos lógicos. Esta movilidad virtual es implementada en la WSN como una

transferencia de estados entre nodos reales. Los autores utilizan el canal inalámbrico para realizar las operaciones que permiten el movimiento de los nodos lógicos.

ViMobiO funciona dividiendo el tiempo en periodos, y cada período se divide en dos. En una parte del período, los nodos trabajan de acuerdo con su funcionamiento normal, y en la otra parte del período, los nodos realizan las tareas relacionadas con la movilidad virtual.

Necesita que las capas inferiores del laboratorio remoto provean tres servicios: sincronización de tiempo, descubrimiento de conexiones, y estimador de la calidad del enlace. La sincronización de tiempo es necesaria para asegurar que todos los nodos están en el mismo estado en todo momento (funcionando normalmente o realizando las tareas de la movilidad virtual). No es necesario que la sincronización sea precisa. El servicio de descubrimiento de conexiones se emplea para encontrar potenciales candidatos a los cuales el nodo lógico puede pasar. Se implementa mediante el envío de mensajes beacon periódicos. El servicio de estimador de la calidad del enlace se emplea para evitar que el algoritmo pretenda utilizar enlaces de mala calidad, evitando así complicaciones. Un componente denominado gestor de transferencia de estado, el cual es la pieza central de ViMobiO, es el encargado de la implementación distribuida del mapeo de nodos lógicos a nodos físicos.

La regla mediante la cual se selecciona el nodo real al cual pasará el nodo lógico puede seleccionarse por el usuario, y dependerá del tipo de movimiento que se pretenda simular (movimiento aleatorio, movimiento en una determinada dirección, etc). Las tareas necesarias para implementar la movilidad virtual la realizan los nodos, lo cual significa sobrecarga de tareas para los mismos.

Los autores realizaron una implementación de ViMobiO empleando el sistema operativo TinyOS, y se puso a prueba en el laboratorio remoto de WSN MoteLab.

### **3.1.11. Laboratorios remotos de WSN de tamaño reducido y de tamaño real**

De acuerdo con la clasificación mencionada en (Slipp et al., 2008), los laboratorios remotos de WSN pueden ser de tamaño real o de tamaño reducido. En un laboratorio remoto de WSN de tamaño real, como su nombre lo dice, los nodos están a distancias semejantes a las que estarían en una aplicación real. En un laboratorio remoto de tamaño reducido, las distancias entre nodos son menores respecto a una WSN real, pudiendo ser pocos metros o centímetros.

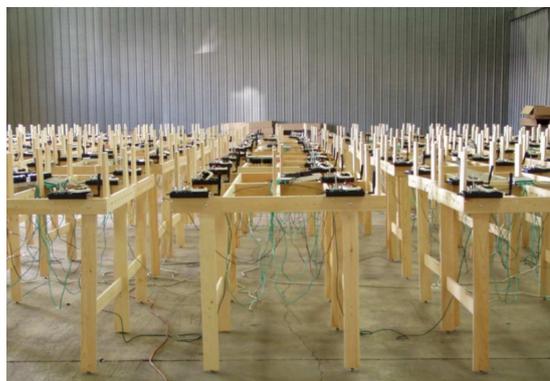
Los laboratorios remotos de tamaño real están más cerca de las condiciones de funcionamiento reales de una WSN de campo, pero requieren de instalaciones de gran tamaño para el despliegue de los nodos. Ejemplos de laboratorios remotos de WSN de tamaño real son MoteLab (Werner-Allen et al., 2005a) y TWIST (Handziski et al., 2006).

Los laboratorios remotos de tamaño reducido tienen como ventaja que pueden desplegarse en instalaciones de tamaño reducido. Sin embargo, los nodos no estarán a la misma distancia a la que estarían en una WSN real. Para simular nodos situados a distancias reales, los laboratorios remotos de WSN de tamaño reducido utilizan dos enfoques diferentes (o combinaciones de ellos):

- Empleo de atenuadores (Ertin et al., 2006)(Slipp et al., 2008)(Handziski et al., 2006).
- Eliminación de paquetes de datos procedentes de los nodos situados a una distancia mayor que un umbral específico (Lambrou and Panayiotou, 2011).

Ambos tipos de laboratorios remotos de WSN tienen ventajas y desventajas, que conducen a que algunas aplicaciones requieran laboratorios remotos de tamaño reducido, y otras aplicaciones requieran laboratorios remotos de tamaño real. Por ejemplo, aplicaciones donde sea de interés conocer el comportamiento de parámetros relacionados con la propagación de las señales de radiofrecuencia, requerirán laboratorios remotos de tamaño real, ya que el comportamiento de estas señales depende de la distancia entre los nodos. En cambio, para una aplicación en la cual el usuario necesite realizar control de topología, un laboratorio remoto de tamaño reducido será más adecuado, ya que en un laboratorio remoto de tamaño real será más difícil controlar la topología de la red (Slipp et al., 2008).

La figura 3.8 muestra una foto del laboratorio remoto de WSN Kansei (Ertin et al., 2006), como ejemplo de laboratorio remoto de WSN de tamaño reducido. Puede verse que los nodos están a muy corta distancia entre ellos.



**Figura 3.8:** Kansei, ejemplo de laboratorio remoto de WSN de tamaño reducido.  
Figura obtenida de (Ertin et al., 2006)

### **3.1.12. Herramientas de análisis in situ**

Existen herramientas para analizar el funcionamiento de un laboratorio remoto de WSN, o una WSN en general, a través de la evaluación de diferentes parámetros, tales como la conectividad entre nodos, la tasa de entrega de paquetes, indicador de intensidad de señal recibida (RSSI), indicador de la calidad del enlace (LQI), etc. En esta sección se presentan algunas de las herramientas existentes.

#### **3.1.12.1. MoteLab Connectivity Daemon**

MoteLab Conectividad Daemon es una aplicación propia del laboratorio remoto de WSN MoteLab, que se utiliza para evaluar la conectividad entre los nodos del mismo (Werner-Allen et al., 2005a)Werner-Allen et al. (2005b). Se ejecuta periódicamente cuando el laboratorio remoto no está realizando ninguna tarea, y se ejecuta como cualquier otro experimento. Su funcionamiento es el siguiente: se divide el tiempo en ciclos, y en cada ciclo, sólo un nodo transmite paquetes de datos, mientras que los demás nodos escuchan para recibir estos paquetes de datos. Cuando un nodo recibe uno de estos paquetes, registra el nodo emisor y el valor del RSSI. Estos datos son almacenados por los nodos hasta el final de la ejecución del programa. Cuando la ejecución finaliza, estos datos son transmitidos y almacenados en la base de datos.

Un script, escrito en lenguaje Perl, utiliza los datos obtenidos para calcular las tasas de pérdida de paquetes entre cada par de nodos. Esta información se utiliza para ilustrar gráficamente la conectividad entre los nodos a través de la interface web. Cuando un usuario desea visualizar el mapa de conectividad, un programa en PHP recoge la información almacenada en la base de datos y genera la representación gráfica. Los datos de conectividad pueden ser recuperados por los usuarios.

La figura 3.9 muestra una parte del mapa de conectividad generado por MoteLab. Las líneas representan los enlaces entre nodos, y el número sobre cada línea, el valor de la tasa de pérdida de paquetes en ese enlace.

#### **3.1.12.2. TestbedProfiler**

TestbedProfiler es una herramienta de análisis para laboratorios remotos de WSN y WSN en general. Se puede utilizar en WSN cuyos nodos tienen la radio CC2420 de Texas Instruments, por ejemplo WSN con nodos MicaZ o TelosB (Metcalf et al., 2010). El laboratorio remoto de WSN tiene que tener un canal de retorno (canal de comunicaciones distinto al canal IEEE802.15.4, como la mayoría de los laboratorios remotos de WSN) para el envío de datos al servidor central. TestbedProfiler analiza los siguientes parámetros:



Power Level	Average Neighbor Count	Power Level	Average Neighbor Count	Power Level	Average Neighbor Count
01	1.909	12	49.124	22	49.832
02	3.922	13	49.144	23	49.915
03	33.295	14	49.152	24	49.978
04	42.982	15	49.396	25	49.970
05	43.140	16	49.564	26	50.007
06	43.364	17	49.592	27	49.331
07	46.196	18	49.622	28	47.133
08	47.892	19	49.750	29	46.870
09	47.807	20	49.840	30	46.901
10	47.855	21	49.829	31	45.421
11	48.601				

**Figura 3.10:** Ejemplo de resultados producidos por TestbedProfiler.  
Figura obtenida de (Metcalf et al., 2010)

de un nodo determinado, para todos los niveles de potencia permitidos por el nodo. Puede verse que cuando se supera el cuarto nivel de potencia, el nodo puede alcanzar prácticamente todos los nodos de la red.

### 3.1.13. Laboratorios remotos de WSN geográficamente distribuidos

Algunos laboratorios remotos de WSN poseen arquitecturas distribuidas, con instalaciones desplegadas en lugares geográficamente separados. Algunos ejemplos son: X-Sensor (Kanzaki et al., 2009), WISEBED (Hellbruck et al., 2011) y SensLAB (SensLAB Consortium, 2010).

Una ventaja de un laboratorio remoto de WSN distribuido es que los usuarios pueden llevar a cabo experimentos en distintos medio ambientes (Kanzaki et al., 2009). Otra ventaja es que los usuarios pueden cargar el mismo experimento en distintos laboratorios remotos y comparar los resultados (Hellbruck et al., 2011).

X-Sensor es un laboratorio remoto distribuido en 8 sitios de 4 universidades de Japón. Posee una arquitectura tipo cluster-tree, es decir, está formado por clusters, cada uno con un gateway. Los gateways se conectan a un servidor central vía Internet. Cuando el laboratorio remoto no está en uso, los sensores toman datos que transmiten a sus gateways, los cuales los almacenan, y luego pueden ser accedidos por los usuarios.

WISEBED es un laboratorio remoto de WSN distribuido. Los laboratorios remotos que forman parte de WISEBED se encuentran distribuidos de la siguiente manera: 3 en Alemania, 2 en Suiza, 1 en Holanda, 1 en Grecia, 1 en Reino Unido y 1 en España. Estos laboratorios remotos de WSN utilizan nodos de distintos fabricantes, entre estos: Tmote Sky, MicaZ, iMote, TelosB, SunSPOT, etc. Los usuarios pueden

construir sus propios laboratorios remotos de WSN y asociarlos a WISEBED. Para poder asociar un laboratorio remoto de WSN a WISEBED, el mismo debe cumplir ciertas características. Los componentes fundamentales de WISEBED son una serie de APIs (interfaces de programación de aplicaciones), que el usuario puede descargar e instalar en sus computadoras, y así construir un laboratorio remoto compatible con WISEBED.

WISEBED permite conectar laboratorios remotos geográficamente alejados mediante el concepto de “enlaces virtuales” (Virtual Links en la bibliografía). Esto se logra mediante componentes de software, alojados en los gateways, que controlan el flujo de datos entre dos laboratorios remotos físicamente separados.

### 3.1.14. Resumen

La tabla 3.1 resume los diferentes aspectos de los laboratorios remotos de WSN mencionados en esta sección. Se resaltan las principales características de cada uno de estos aspectos, como las diferentes formas de implementar algunos componentes, o diferentes opciones empleadas para implementar alguna funcionalidad, como ser movilidad. Para una tabla que liste los laboratorios remotos de WSN existentes y las características de cada uno de ellos, ver el trabajo de Steyn *et al.* (Steyn and Hancke, 2011).

**Tabla 3.1:** Resumen de características de los laboratorios remotos de WSN

<b>Componente o característica</b>	<b>Descripción</b>
Nodos	Constituyen el sistema bajo prueba. Algunos autores les agregan componentes adicionales como medidores, sensores, interfaces Ethernet, etc.
Sistema de interconexión o back-end	Sistema que permite la comunicación entre los nodos y el resto del laboratorio remoto. Permite programar los nodos, intercambiar datos con los mismos, o la interacción en tiempo real con los usuarios. Puede ser del tipo “fuera de banda” (Ethernet, cables USB, IEEE802.11), o “en banda” (IEEE802.15.4).
Interacción con los usuarios	Puede ser en modo por lotes o en tiempo real. Algunos laboratorios remotos de WSN permiten la inyección de datos predefinidos (los cuales deben estar definidos en un archivo).

Almacenamiento de datos	Se almacenan los datos producidos por los experimentos, configuraciones de los experimentos pendientes, programas que se deben cargar en los nodos, información de los usuarios, e información sobre el estado de los nodos y del laboratorio remoto en general. Se emplean bases de datos MySQL o el sistema de archivos del servidor.
Plataformas replicables	Son plataformas de experimentación que pueden ser replicadas fácilmente. No poseen acceso remoto. Tienen como objetivo permitir que los usuarios realicen experimentos sobre plataformas estándares.
Herramientas de análisis <i>in situ</i>	Permiten evaluar el estado del laboratorio remoto y sus nodos, y brindan información útil a los usuarios, tal como conectividad entre pares de nodos, tasa de entrega de paquetes entre pares de nodos, etc. Ejemplos: MoteLab connectivity daemon (Werner-Allen et al., 2005a) y TestbedProfiler (Metcalf et al., 2010).
Movilidad	Pocos laboratorios remotos de WSN poseen nodos móviles. Los nodos pueden moverse sin restricciones o seguir caminos predefinidos. Se utilizan robots que transportan los nodos. ViMobiO es un mecanismo que permite implementar movilidad virtual en laboratorios remotos con nodos fijos.
Sistemas de localización	Se implementan mediante marcas de posición o cámaras junto con algoritmos de procesamiento de imágenes.
Laboratorios remotos de tamaño real y de tamaño reducido	Los laboratorios remotos de WSN pueden ser de tamaño reducido o de tamaño real. Los laboratorios remotos de tamaño reducido pueden desplegarse en instalaciones de menor tamaño, pero los laboratorios remotos de tamaño real simulan mejor las condiciones de trabajo de una WSN en una aplicación real. Cada uno se adapta mejor a diferentes tipos de experimentos.
Despliegue	Pueden estar distribuidos en sitios geográficamente separados o centralizados en un solo sitio. Algunos laboratorios remotos de WSN distribuidos permiten que los usuarios asocien sus propios laboratorios remotos.
Interface con el usuario	Permite a los usuarios interactuar con el laboratorio remoto. Por lo general son interfaces web programadas utilizando los lenguajes PHP y Javascript.
Sistema de reserva de turnos	Permite el acceso de manera ordenada a un gran número de usuarios.

## 3.2. Clasificaciones de los laboratorios remotos de WSN

En esta sección se presentan las clasificaciones de los laboratorios remotos más citadas en la bibliografía. Los autores realizan diferentes clasificaciones de acuerdo a diferentes criterios.

### 3.2.1. Clasificación de acuerdo a la implementación del sistema de gestión

Steyn y Hancke (Steyn and Hancke, 2011) clasifican a los laboratorios remotos, de acuerdo a la implementación del sistema de gestión, en tres tipos:

- Laboratorios remotos de WSN de servidor central: Este tipo de laboratorios remotos de WSN poseen un servidor central, usualmente formado por varias computadoras, que realiza las tareas de gestión y base de datos. Son típicamente de arquitectura fija, con back-end centralizado. Emplean interfaces Ethernet o WiFi para conectarse con los nodos. La mayoría de ellos emplea interface web y sofisticados programas de reserva de recursos y control. Tienen como ventaja que por lo general poseen muchos recursos y permiten experimentos a gran escala. La desventaja es que el costo y el mantenimiento son elevados. Ejemplos: MoteLab (Werner-Allen et al., 2005a) y Kansei (Ertin et al., 2006).
- Laboratorios remotos de WSN de una única PC: Utilizan una única PC como punto central del laboratorio remoto, la cual incorpora todas las funciones de gestión y almacenamiento de datos. Generalmente utilizan interfaces seriales o USB para conectar los nodos con el resto del sistema. Por lo general poseen menos recursos que los anteriores. El costo y mantenimiento no son tan elevados como en los laboratorios remotos de servidor central. Ejemplo: Sensei (Rensfelt et al., 2009).
- Laboratorios remotos de WSN híbridos: están formados por varios clusters de laboratorios remotos de una única PC, que se comunican con un servidor central, constituyendo una arquitectura de varios niveles. Cada cluster está formado por nodos sensores que se conectan a una PC que funciona como gateway, y realiza tareas de gestión. Estas PC suelen comunicarse a través de redes Ethernet o Internet con un servidor central. El costo es alto. Son fácilmente escalables mediante la adición de nuevos gateways. Ejemplos: TWIST (Handziski et al., 2006) y WSNTB (Sheu et al., 2008).

### 3.2.2. Clasificación de acuerdo a su tipo de objetivos

El-Darymli *et al* (El-Darymli and Ahmed, 2012) clasifican los laboratorios remotos de WSN de acuerdo a su tipo de objetivos en dos grupos:

- Laboratorios remotos de WSN multi-usuario.
- Laboratorios remotos de WSN para prueba de concepto.

La diferencia es que los laboratorios remotos de WSN para prueba de concepto son contruidos para un propósito específico, por ejemplo, permitir a los usuarios realizar experimentos en ambientes industriales, mientras que los laboratorios remotos multiusuario no poseen un propósito específico.

### 3.2.3. Clasificación de acuerdo a su estructura

El-Darymli *et al.* (El-Darymli and Ahmed, 2012) clasifica los laboratorios remotos de WSN de acuerdo a su estructura en 4 tipos:

- Kits de desarrollo: Son provistos por vendedores, para permitir a los usuarios instalar y usar un laboratorio de WSN rápidamente. Generalmente están compuestas por algunos nodos, algunas estaciones base y herramientas de software para programar los nodos y analizar la operación de la WSN.
- Laboratorios remotos tipo Cluster: Formados por una WSN y un sistema que permite el acceso remoto a la misma. De acuerdo a El-Darymli *et al.* (El-Darymli and Ahmed, 2012), la mayoría de los laboratorios remotos de WSN pertenecen a esta categoría.
- Laboratorios remotos de WSN Overlay: Consisten en una WSN instalada y funcionando, construida para una función específica diferente a un laboratorio remoto, a la que se le agrega un sistema que permite el acceso remoto. Estructuralmente es muy similar a un laboratorio remoto de WSN tipo cluster. La diferencia es que un laboratorio remoto tipo cluster se construye con el propósito de funcionar como laboratorio remoto. En cambio, un laboratorio remoto de WSN overlay se construye con otro propósito, y luego se le agrega el sistema de acceso remoto que permite usarlo como un laboratorio remoto.
- Laboratorios remotos de WSN federados: Están formados por varios laboratorios remotos de WSN independientes. Estos son interconectados para permitir realizar experimentos de gran escala.

### 3.2.4. Clasificación de acuerdo a características específicas

- Laboratorios remotos de WSN de hardware especializado: Estos laboratorios remotos de WSN son contruidos con hardware especializado, el cual permite

funciones específicas no brindadas por otros laboratorios remotos de WSN. Por ejemplo, pueden estar formados por nodos que poseen motes de distintos fabricantes, o sistemas que permiten acceder a parámetros de la capa física. Por ejemplo: HINT (Huangfu et al., 2010).

- Laboratorios remotos de WSN multi-sitio: Son laboratorios remotos de arquitectura híbrida, con clusters desplegados en lugares geográficamente distintos. Como ventaja puede mencionarse que poseen clusters instalados en ambientes diferentes, permitiendo al usuario realizar experimentos en distintos ambientes. Además, permite que los clusters sean provistos por distintas instituciones, permitiendo que puedan formar laboratorios remotos de WSN de gran tamaño, aportando cada institución un cluster de bajo costo. Ejemplos: X-Sensor (Kanzaki et al., 2009) y SensLAB (SensLAB Consortium, 2010).
- Laboratorios remotos de WSN que no poseen un canal secundario para realizar las tareas de gestión. Estos laboratorios remotos de WSN comparten un solo canal para la comunicación normal dentro de la WSN y para las tareas de gestión, es decir, no poseen un sistema de back-end. Por lo general, un componente de software instalado en los nodos permite la comunicación con el laboratorio remoto para tareas de programación y gestión. Ejemplo: Sensenet (Dimitriou et al., 2007).
- Laboratorios remotos de WSN para testear aplicaciones industriales: En ambientes industriales existen numerosas fuentes generadoras de señales de radiofrecuencias como motores, relés, etc., que pueden interferir de manera muy intensa con las comunicaciones inalámbricas. Existe un laboratorio remoto de WSN específico para el estudio del comportamiento de las WSN en este tipo de ambientes: WINTeR (Slipp et al., 2008). La características fundamentales del mismo es que incluye generadores de señales de radiofrecuencias para generar interferencias y estructuras metálicas como las que se encontrarían en este tipo de ambientes.

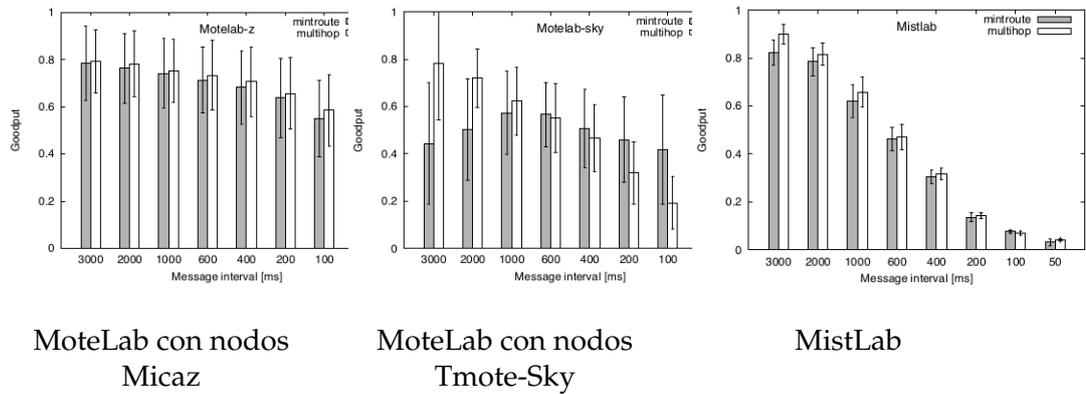
### **3.3. Análisis, problemas no resueltos y posibles mejoras en el campo de los laboratorios remotos de WSN**

En esta sección se resume y analiza la información brindada en las secciones anteriores de este capítulo, y se citan los problemas no resueltos y posibles mejoras a realizar en los laboratorios remotos de WSN actuales.

### 3.3.1. Diferentes resultados para un mismo experimento

A través de experimentos, Koen Langendoen (Langendoen, 2006) muestra que distintos laboratorios remotos de WSN producen resultados diferentes para un mismo experimento.

Los experimentos llevados a cabo por Langendoen consistieron en correr una aplicación en tres laboratorios remotos de WSN: MoteLab-z, MoteLab-sky y MistLab. Esta aplicación hace que los nodos envíen mensajes periódicamente al nodo sumidero, informando lecturas de sus sensores de luz. Fueron empleados dos protocolos de comunicación: MultiHop y MintRoute. La figura 3.11 muestra como varía el goodput (fracción de mensajes correctamente entregados al nodo sumidero) en función del tiempo entre mensajes. Puede verse claramente las diferencias en los resultados producidos por los tres laboratorios remotos.



**Figura 3.11:** Goodput producido por tres laboratorios remotos de WSN para dos protocolos de ruteo. Figuras obtenidas de (Langendoen, 2006)

Son numerosos los factores que contribuyen a que los laboratorios remotos de WSN produzcan distintos resultados para un mismo experimento, entre estos:

- Diferentes tipos de nodos. Esto deriva en el uso de diferentes tipos de antenas, lo cual afecta significativamente la reproducibilidad de los experimentos, ya que las comunicaciones inalámbricas dependen en gran medida del tipo de antenas utilizado.
- Diferente número de nodos.
- Diferentes topologías de red.
- Diferentes entornos en los cuales se encuentran desplegados los laboratorios remotos de WSN. Esto afecta la forma en que se desvanecen las señales de radio, ya

que el desvanecimiento de señales de radio depende fuertemente del entorno en el cual la WSN se encuentra desplegada. También afecta la forma en que se produce el multi-path, fenómeno, por el cual las ondas de radio siguen diferentes caminos, de diferentes longitudes, hasta llegar al destino, debido a las reflexiones producidas en diferentes objetos. La forma en que el multi-path afecta a una comunicación inalámbrica depende del entorno.

- Diferencias en la zona gris. Esta zona es una parte del rango de transmisión de un nodo que posee un comportamiento irregular.

Debido a estos factores, Langendoen declara que cada laboratorio remoto es único. Como resultado, reproducir resultados obtenidos a partir de un laboratorio remoto en otro laboratorio remoto es muy difícil, quizás imposible.

Estos resultados llevan a Langendoen a declarar "comparar los resultados producidos por diferentes laboratorios remotos de WSN es como comparar manzanas y naranjas".

Sin embargo, es muy difícil, sino imposible, que dos WSNs reales sean iguales. El tipo tipo y número de nodos, la ubicación de los nodos, la topología, la tasa de transmisión de datos, etc., dependen de los requisitos de cada aplicación. Además, para diferentes experimentos, no es posible garantizar que las interrupciones se produzcan en el mismo momento, el que los mensajes siempre lleguen cuando el hardware está en el mismo estado para responder al evento, que el ruido electromagnético sea el mismo, etc. Por lo tanto, los resultados serán siempre estadísticos, no determinísticos.

Además, para la mayoría de los experimentos, los laboratorios remotos de WSN y sus diferencias son una buena representación de la realidad, y probablemente el usuario necesite más laboratorios remotos de WSN diferentes a los existentes para llevar adelante un experimento. Por ejemplo, si un usuario diseña un protocolo de ruteo, debe verificar el funcionamiento del mismo en distintas WSN, y el protocolo de ruteo debería funcionar adecuadamente para todas las WSNs para las cuales haya sido diseñado. Por lo que puede concluirse que la desventaja citada por Langendoen, es en realidad una ventaja, ya que es necesario que los laboratorios remotos de WSN sean diferentes entre si, y produzcan resultados diferentes.

Para llevar a cabo un experimento, el usuario debe ejecutar el experimento varias veces, en diferentes WSN, luego analizar la dispersión de los resultados, y decidir si los resultados son útiles para la aplicación, y presentar resultados estadísticos.

### **3.3.2. Variabilidad de resultados**

Langendoen (Langendoen, 2006) observó significativa variabilidad en algunos parámetros, al comparar los resultados producidos por diferentes laboratorios remotos, o por el

mismo laboratorio remoto. Afirma no saber las razones de esta variabilidad. Entre estos parámetros están:

- La conectividad en función del número de nodos en servicio para diferentes experimentos varía de manera diferente.
- La formación de diferentes árboles de ruteo de datos para dos ejecuciones seguidas del mismo experimento. Dos árboles de ruteo diferentes implicará que los mensajes se transmitirán por diferentes caminos para llegar al sumidero, y el comportamiento de la WSN será diferente.
- La variación de las topologías de red. Esta variación es debida a nodos que salen de servicio por fallas, o nodos que vuelven a entrar en servicio luego de que la falla ha sido solucionada.

Sin embargo, esta variabilidad es propia de una aplicación real de WSN. Por lo que los usuarios probablemente necesitarán que los laboratorios remotos de WSN presenten esta variabilidad.

### **3.3.3. Empleo de un número reducido de plataformas de hardware**

Otro problema no resuelto, citado por Ze Zhao *et al.* (Zhao et al., 2010), es que por lo general los laboratorios remotos de WSN actuales utilizan una única plataforma de hardware, lo cual impide poder realizar experimentos sobre redes heterogéneas y de compatibilidad entre distintos fabricantes.

Los autores citan la necesidad de laboratorios remotos de WSN que empleen radios IEEE802.15.4 de distintos fabricantes. Además, algunos autores declaran que las futuras WSNs interactuarán con otras tecnologías, como ser teléfonos celulares (Hellbruck et al., 2011). Por lo que declaran la necesidad de desplegar laboratorios remotos de WSN que combinen el uso de distintos estándares de comunicación, por ejemplo IEEE802.15.4, IEEE802.11, Ethernet, telefonía móvil, etc.

### **3.3.4. Necesidad de un mayor número de laboratorios remotos de WSN**

Como se mencionó en la sección 3.3.1, y de acuerdo con (Hellbruck et al., 2011), son necesarios más laboratorios remotos de WSN heterogéneos disponibles públicamente. Como resultado, varios autores fomentan el despliegue de nuevos laboratorios remotos de WSN. Además, los autores de varios laboratorios remotos de WSN proporcionan a los usuarios los componentes de software de sus propios laboratorios remotos, entre estos,

MoteLab (Werner-Allen et al., 2005a) y WISEBED (Hellbruck et al., 2011). De este modo los usuarios pueden desplegar sus propias versiones de estos laboratorios remotos.

Algunos laboratorios remotos de WSN permiten a los usuarios asociar sus laboratorios remotos de WSN locales a su infraestructura (Hellbruck et al., 2011). Para este propósito, el laboratorio remoto hijo debe cumplir ciertos parámetros impuestos por el laboratorio remoto de WSN padre.

### **3.3.5. Falta de laboratorios remotos para WSN desplegados al aire libre o industriales**

Muchas aplicaciones de WSN requieren despliegues al aire libre. Una WSN desplegada al aire libre se comportará de manera diferente a una WSN desplegada en el interior de un edificio. Por un lado, las señales de radiofrecuencia encontrarán distintos tipos de obstáculos que las atenuarán o reflejarán de manera diferente. Además los nodos estarán expuestos a condiciones de funcionamiento más severas y tendrán mayor probabilidad de fallar. Sin embargo ninguno de los laboratorios remotos para WSN citados en este trabajo posee instalaciones al aire libre.

En cuanto a laboratorios remotos para WSN pensados para ambientes industriales, uno solo ha sido encontrado en la literatura: WINTeR (Slipp et al., 2008).

### **3.3.6. Nodos que no son alimentados por baterías**

Los nodos de casi todos los laboratorios remotos de WSN actuales son alimentados con energía de la red eléctrica, USB u otra fuente de alimentación no finita (no baterías), mientras que en condiciones de campo normales, los nodos son alimentados por baterías. El tiempo de vida de un nodo en una WSN se determina en gran parte por la vida útil de sus baterías, de modo que la autonomía de las mismas es un parámetro importante de diseño. El uso de una fuente de energía no finita es una desviación de las condiciones reales de funcionamiento en campo.

Sólo uno de los laboratorios remotos de WSN citados en este trabajo, SensLab (Metcalf et al., 2010), utiliza baterías. Permite a los usuarios seleccionar entre baterías o corriente directa. Las baterías se recargan cuando los nodos se alimentan mediante corriente directa.

### **3.3.7. No es posible realizar experimentos de larga duración**

En la literatura se han reportado experimentos de corta duración, por ejemplo un par de horas. Pero en la práctica los investigadores pueden requerir experimentos de larga duración, es decir, semanas o meses (Günes et al., 2008).

### **3.3.8. Necesidad de conocimientos avanzados de programación y de WSN**

La mayoría de los laboratorios remotos citados en este trabajo usan nodos de la familia TelosB, que deben ser programados con lenguajes C o NesC. Esto requiere conocimientos avanzados de programación, como resultado, los usuarios sin este conocimiento no podrán utilizar estos laboratorios remotos. Además, los laboratorios remotos de WSN actuales requieren que el usuario cree los programas a verificar y cargue el sistema operativo que necesita. Ésto tiene grandes ventajas cuando se desea realizar un experimento con fines científicos, o bien con fines educativos en cursos donde los estudiantes poseen conocimientos avanzados sobre WSN. Pero usuarios con pocos conocimientos sobre programación o WSN verán complejo el acceso a estos laboratorios remotos de WSN.

Sería útil la construcción de laboratorios remotos de WSN que puedan ser utilizados por usuarios sin conocimientos avanzados. Éstos deberían permitir que los usuarios puedan configurar experimentos sin necesidad de crear programas complejos ni lidiar con detalles del sistema operativo. Los programas deberían estar parcialmente predefinidos, para que el usuario pueda crear un experimento modificando los valores de ciertos parámetros o escribiendo parte del código de esos programas.

### **3.3.9. Experimentos centrados en los nodos de la WSN**

Los laboratorios remotos de WSN actuales han sido construidos de manera que el usuario pueda realizar experimentos sobre los componentes de software de los nodos, sobre el sistema operativo de los mismos o sobre los protocolos de comunicaciones de los nodos. Pero los usuarios no pueden realizar experimentos que permitan conocer como se comportará la WSN como parte de una aplicación específica, con restricciones de funcionamiento determinadas. Para ello, el usuario debería crear un modelo de funcionamiento de la WSN según la aplicación específica, que cumpla los requisitos de la misma y simule los posibles comportamientos de la aplicación, y hacer que el modelo interactúe con el laboratorio remoto.

### 3.4. Aplicabilidad de los laboratorios remotos de WSN actuales a formación y entrenamiento de usuarios

#### 3.4.1. Componentes o características que afectan la aplicabilidad de los laboratorios remotos de WSN a formación y entrenamiento de usuarios

En el capítulo 2, sección 2.3.2, se presentó una clasificación de los laboratorios remotos, distinguiendo laboratorios remotos destinados a formación y entrenamiento de usuarios y destinados a investigación. Teniendo en cuenta esa clasificación, y el estudio realizado en esta sección sobre los diferentes enfoques empleados para construir laboratorios remotos de WSN, puede deducirse:

- Cuáles componentes o características influirán para permitir que un laboratorio remoto de WSN pueda ser utilizado con fines de formación y entrenamiento de usuarios.
- Cuáles son las carencias de los laboratorios remotos de WSN actuales para poder destinarlos a formación y entrenamiento de usuarios.

La tabla 3.2 lista los diferentes componentes y características de los laboratorios remotos de WSN estudiados en este capítulo, y en función de la clasificación propuesta en la sección 2.3.2 del capítulo 2, propone una clasificación de los componentes según si su implementación influye en la aplicabilidad de los laboratorios remotos de WSN a formación y entrenamiento de usuarios.

**Tabla 3.2:** Influencia de los diferentes componentes o características de los laboratorios remotos de WSN sobre su aplicabilidad en formación y entrenamiento de usuarios.

<b>Componentes y características</b>	<b>Influencia de su diseño sobre la aplicabilidad en formación y entrenamiento de usuarios</b>
Nodos	No influye. Son el sistema bajo prueba.
Sistema de Interconexión	No influye. Su diseño no afecta la aplicabilidad a formación y entrenamiento de usuarios.
Operación en modo por lotes o tiempo real	Influye. Es preferible la interacción en tiempo real para permitir mayor interacción con el equipo bajo prueba.
Inyección de datos	No influye. Es preferible la interacción en tiempo real.
Almacenamiento de datos	Influye. Se debe almacenar información sobre la actividad realizada por los usuarios, con el fin de poder seguirlos y evaluarlos, sobre todo si el laboratorio remoto se usa en tareas de docencia.

Plataformas replicables	No influye. Es preferible que el laboratorio posea acceso remoto.
Interface web	Influye. Debe ser tal que el uso del laboratorio remoto sea simple, intuitivo y didáctico.
Sistema de turnos	No influye. Todos los laboratorios remotos poseerán un sistema de turnos.
Nodos móviles y sistemas de localización	No influye. Es parte del sistema bajo prueba.
Movilidad virtual para laboratorios remotos de WSN estáticos	No influye. Es una herramienta para adaptar el laboratorio remoto a un caso particular de WSN.
Laboratorios remotos de WSN de escala real y miniaturizados	No influye. Definen los tipos de experimentos que se pueden realizar, independientemente de si se realizan con fines educativos o científicos.
Herramientas de análisis para laboratorios de WSN	Influye. Pueden resultar útiles para que los usuarios comprendan el funcionamiento de la WSN.
Laboratorios remotos de WSN geográficamente distribuidos	No influye. Si el equipo bajo prueba está distribuido o concentrado en un solo sitio no afecta la aplicabilidad en formación y entrenamiento de usuarios.

Se puede deducir de la tabla 3.2 que la forma de implementar los siguientes componentes o aspectos de un laboratorio remoto de WSN afectará su aplicabilidad a formación y entrenamiento de usuarios:

- Almacenamiento de datos
- Interface web
- Operación en tiempo real
- Herramientas de análisis para laboratorios de WSN

Además, de la sección 2.3.2 se pueden incluir también los siguientes aspectos:

- Experimentos predefinidos.
- Tipos de usuarios.

### 3.4.2. Análisis de la aplicabilidad de los laboratorios remotos de WSN actuales a formación y entrenamiento de usuarios

En la sección 3.4.1 se analizó cuales componentes deberían ser diseñados de manera adecuada para que el laboratorio remoto de WSN pueda ser aplicado a formación y entrenamiento de usuarios.

La tabla 3.3 analiza la forma en que los laboratorios remotos de WSN actuales implementan estos componentes o características, y analiza si la implementación de estos componentes es adecuada para su aplicación en formación y entrenamiento de usuarios.

**Tabla 3.3:** Análisis de la aplicabilidad de los laboratorios remotos de WSN actuales a formación y entrenamiento de usuarios

<b>Componente o característica</b>	<b>Forma en que está implementado en los laboratorios remotos de WSN actuales</b>	<b>¿Es adecuado para su aplicación en formación y entrenamiento de usuarios?</b>
Datos que se almacenan	Se almacenan resultados de experimentos, cuentas de usuarios, estados de los nodos, estado del laboratorio remoto y experimentos pendientes (sección 3.1.5)	No. Deberían almacenar actividades realizadas por los usuarios, en un formato que permita su supervisión por parte de profesores, con la finalidad de seguir sus progresos en el proceso de aprendizaje y evaluarlos.
Interface web	No son didácticas, están pensadas para que los usuarios tengan total control de los nodos. Algunos laboratorios remotos de WSN no implementan interface web (sección 3.1.7).	No. Deberían ser didácticas, de uso fácil e intuitivo.
Operación en tiempo real	Pocos laboratorios implementan operación en tiempo real, la mayoría operan por lotes (sección 3.1.3).	Si (en los laboratorios remotos que implementan interacción en tiempo real)
Herramientas de análisis para laboratorios de WSN	Pocos laboratorios remotos las implementan (sección 3.1.12).	Si (existen dos herramientas que pueden ser de utilidad, ver sección 3.1.12)

Experimentos predefinidos	Ningún laboratorio remoto de WSN implementa experimentos predefinidos	No. Los usuarios deben programar los experimentos desde cero.
Tipos de usuarios	Los laboratorios remotos de WSN actuales no diferencian entre tipos de usuarios.	No. Debería diferenciarse al menos entre usuarios estudiantes y profesores o entrenadores.

En análisis de los laboratorios remotos de WSN realizado en este capítulo junto con la clasificación de laboratorios remotos destinados a investigación o formación y entrenamiento de usuarios realizada en el capítulo 2 nos permite obtener la tabla 3.3. Esta tabla muestra que los laboratorios remotos de WSN actuales no están diseñados para ser utilizados para fines educativos o de formación y entrenamiento, orientados a usuarios como pocos conocimientos de programación o WSN. Sin embargo, el diseño adecuado de algunos de sus componentes puede permitir diseñar laboratorios remotos de WSN con fines educativos o de formación y entrenamiento.

### 3.5. Conclusiones

En general, los nodos utilizados por los laboratorios remotos de WSN actuales son de la familia TelosB o MicaZ. Estos nodos pueden ser programados completamente por los usuarios, incluso pueden elegir el sistema operativo a utilizar. Esto los hace de gran utilidad para realizar experimentos científicos en cuanto a la programación y funcionamiento interno de los nodos o desarrollo de nuevos componentes de software.

Sin embargo, un usuario sin conocimientos avanzados necesitará realizar inicialmente experimentos simples, y luego, a medida que adquiera mayores conocimientos, podrá realizar experimentos más complejos. Por lo tanto, para este usuario será muy difícil realizar sus primeras prácticas empleando los laboratorios remotos de WSN disponibles actualmente, ya que éstos no cuentan con interfaces web didácticas, intuitivas y de fácil uso, ni con experimentos predefinidos. Por otro lado, un laboratorio remoto destinado a tareas de educación requerirá de herramientas especiales que permitan al profesor realizar tareas de seguimiento y evaluación, no presentes en los laboratorios remotos de WSN actuales.

Se concluye que es de gran importancia la construcción de laboratorios remotos orientados a formación y entrenamiento, para usuarios con pocos conocimientos sobre WSN y programación. La tabla 3.3 muestra los elementos que estos laboratorios remotos deberían poseer.

# Laboratorios Remotos de WSN Orientados a Entrenamiento, Experimentación y Desarrollo

## 4.1. Introducción

El capítulo 2 resume como han sido construidos los laboratorios remotos presentados en la bibliografía, mientras que el capítulo 3 describe específicamente los laboratorios remotos de WSN. Una de las conclusiones a las cuales se llega es que todos los laboratorios remotos de WSN presentados en la bibliografía están destinados a investigación científica o experimental, pero centrados en el funcionamiento interno de las WSN. Los usuarios por lo general pueden programar los nodos, cargar el sistema operativo de los mismos, modificar los protocolos de comunicaciones, etc. Para ello requieren conocimientos avanzados de WSN y programación. El capítulo 3 finaliza con un análisis de la aplicabilidad de los laboratorios remotos de WSN a formación y entrenamiento de usuarios. Se llega a la conclusión de que no han sido pensados para ser utilizados en tareas de formación y entrenamiento de usuarios, o en desarrollo de aplicaciones que utilicen WSN.

Como uno de los aportes de esta tesis, se ha propuesto un modelo de capas para implementar laboratorios remotos de WSN orientados a usuarios que deseen formarse, entrenarse o experimentar en programación y funcionamiento de las WSN, o en desarrollo de aplicaciones que utilicen WSN. Se ha realizado también una implementación del modelo propuesto. Este capítulo presenta el modelo propuesto y describe como se ha realizado la implementación del mismo.

Este capítulo comienza presentando la hipótesis de trabajo. Luego presenta el modelo de capas mencionado en el párrafo anterior y se describe en detalle como se realizó la implementación de dicho modelo.

## **4.2. Hipótesis de trabajo**

*"Se pueden desarrollar laboratorios remotos de WSN, que puedan ser utilizados por usuarios que deseen formarse, entrenarse o experimentar en programación y funcionamiento de las WSN, o en desarrollo de aplicaciones que utilicen WSN, mediante el adecuado diseño e implementación de sus módulos de gestión".*

## **4.3. Modelo de capas propuesto**

Con el fin de que el laboratorio remoto cumpla con la hipótesis planteada, se han impuesto una serie de requisitos de diseño, los cuales han sido extraídos de la bibliografía presentada los capítulos 2 y 3. Estos requisitos se presentan en esta sección. Posteriormente se describe el modelo para laboratorios remotos que se utilizó como base, las modificaciones que se realizaron para aplicar dicho modelo a un laboratorio remoto de WSN, y finalmente el modelo para laboratorios remotos propuesto.

### **4.3.1. Requisitos de diseño de los módulos de gestión**

Para cumplir el objetivo planteado en la hipótesis de trabajo, de acuerdo al análisis realizado en la sección 2.3.2, el laboratorio remoto propuesto debe cumplir los siguientes requisitos:

- Interface web didáctica, de fácil uso e intuitiva.
- Almacenamiento de la actividad realizada por los usuarios (esta información es distinta a los datos producidos por los experimentos).
- Experimentos predefinidos.
- Distinción entre distintos tipos de usuarios: clientes y supervisores, con diferentes permisos de acceso cada uno.

Además, se imponen también los siguientes requisitos de diseño:

- Posibilidad de acceso desde cualquier dispositivo con conexión a Internet (PC, notebook, teléfono celular, etc) con cualquier navegador web, sin importar la velocidad de la conexión.
- Fácil adaptación a cualquier tipo de nodos y cualquier tipo de modo de funcionamiento.
- Alta disponibilidad y robustez.

#### **4.3.1.1. Interface web didáctica, de fácil uso e intuitiva.**

El cumplimiento de este requisito es parte central de este trabajo. A través de esta interface web, el usuario puede modificar los parámetros de programación de los nodos, y visualizar los valores de estos parámetros, como también visualizar los valores medidos por los pines de entrada de los nodos. El usuario puede acceder a la información sobre cada parámetro de configuración de la WSN haciendo clic sobre el parámetro, y puede elegir un valor de configuración entre una lista de posibles valores, con una explicación para cada valor.

#### **4.3.1.2. Almacenamiento seguro de la actividad realizada por los usuarios**

Para permitir que el laboratorio remoto pueda ser utilizado como una herramienta en un curso o en tareas de docencia, es necesario almacenar información relacionada con la actividad realizada por los usuarios.

Por este motivo, el laboratorio remoto almacena en el servidor toda la actividad realizada por los usuarios, y permite a los usuarios autorizados (supervisores) visualizar dicha información. Esta información puede ser de utilidad para que profesores o instructores sigan la actividad realizada por los usuarios que están siendo formados en el uso de las WSN.

Pensando en futuras aplicaciones relacionadas con docencia, debe considerarse que esta información puede llegar a ser utilizada para evaluar o hacer seguimiento a estudiantes que tomen un curso de WSN. Por este motivo, esta información debe ser almacenada de manera segura, evitando que personas no autorizadas puedan tener acceso a la misma, y con sistemas de respaldo para asegurar su integridad. Este requisito se logra mediante la utilización de servicios de almacenamiento de datos y autenticación de usuarios de Cloud Computing.

#### **4.3.1.3. Experimentos predefinidos**

Con la finalidad de que el usuario no deba programar un experimento en su totalidad, los experimentos deben estar predefinidos. En el caso de una WSN, esto puede lograrse preseleccionando los parámetros y valores que pueden configurarse. De este modo el usuario puede configurar una WSN rápidamente, sin tener que programar el experimento completamente.

#### **4.3.1.4. Distinción entre distintos tipos de usuarios**

En el caso de aplicaciones destinadas a usuarios que deseen formarse o entrenarse en programación y funcionamiento de las WSN, es necesario una distinción entre distintos usuarios, con la finalidad de que unos puedan hacer el papel de tutores o supervisores de los usuarios que necesiten formación o entrenamiento. Con esta finalidad, se han propuesto dos tipos de usuarios: clientes y supervisores. La diferencia entre estos es que los segundos pueden seguir la actividad realizada por los primeros.

#### **4.3.1.5. Posibilidad de acceso desde cualquier dispositivo con conexión a Internet**

Los usuarios acceden al laboratorio remoto a través de Internet utilizando redes o dispositivos que pueden ser de diferentes velocidades y características. Debido a que se desea que cualquier usuario pueda acceder al laboratorio remoto independientemente de la calidad de su conexión a Internet, el laboratorio remoto debe adaptarse a cualquier demora que produzca la conexión de Internet del usuario.

Para lograr este objetivo, todas las comunicaciones entre el laboratorio remoto y los usuarios son asíncronas. La aplicación servidor está constantemente en estado de espera de comandos o datos, ya sea desde el usuario o desde la red de sensores. Esta aplicación realiza diferentes acciones según el origen y tipo de datos recibido, como se indica a continuación. Cuando la aplicación servidor recibe:

- Un comando desde un usuario, envía a la WSN el comando y almacena en la base de datos información sobre el usuario y el comando que envió.
- Una respuesta a un comando desde el laboratorio de WSN, la aplicación servidor busca en la base de datos el usuario que envió dicho comando, y le envía la respuesta correspondiente.
- Un reporte de datos desde la red de sensores, envía a todos los usuarios que estén visualizando la interface web en ese momento los nuevos valores reportados por la red de sensores.

#### **4.3.1.6. Fácil adaptación a cualquier tipo de nodos**

Para permitir que el laboratorio remoto pueda adaptarse fácilmente a otro tipo de nodos y a cualquier tipo de funcionamiento, se ha diseñado en base a un sistema de comandos y respuestas. A través de la interface web, los usuarios envían comandos, los cuales son procesados por los módulos de gestión del laboratorio remoto, y enviados a los nodos. Las respuestas de los nodos a dichos comandos son enviadas a los módulos de gestión del laboratorio remoto y luego al usuario que envió el comando, para que visualice estas respuestas a través de la interface web. De este modo, agregar otro tipo de nodo implica solo agregar un módulo nuevo a la interface web, que envíe al nuevo nodo los comandos que éste acepta, y permita visualizar las respuestas que el nodo pueda producir, sin necesidad de modificar el resto del sistema.

#### **4.3.1.7. Alta disponibilidad y robustez**

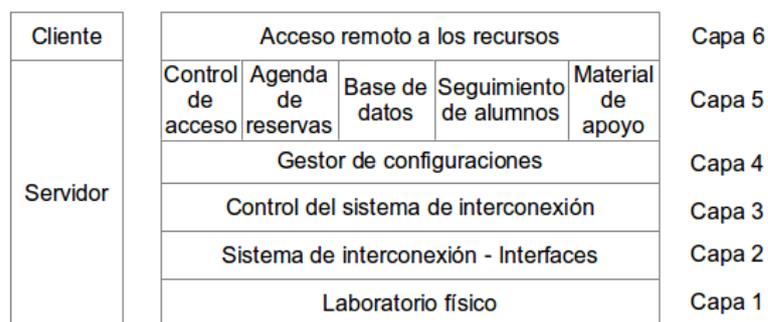
Para dar robustez al sistema, de modo que pueda recuperarse ante fallas externas, como fallas en la alimentación eléctrica o de conexión a Internet, los componentes de software del sistema verifican constantemente el buen funcionamiento de la comunicación con los demás componentes con los cuales se comunican, e intentan restablecer la comunicación en caso de falla.

La implementación de los módulos de gestión en el Cloud asegura que los mismos estarán siempre disponibles y en funcionamiento. De este modo, estos módulos pueden verificar el buen funcionamiento de los componentes del laboratorio físico, y tomar las medidas correspondientes en caso de falla (intentar solucionar la falla, alertar a los administradores, avisar al usuario del desperfecto, etc.).

### **4.3.2. Modelo de capas de Marianetti**

El punto de partida para el laboratorio remoto presentado en esta tesis es el modelo de Marianetti, un modelo de capas para laboratorios remotos destinados a docencia planteado en (Marianetti, 2006) y presentado en la sección 2.3.5 de esta tesis. Este modelo, de acuerdo a la clasificación propuesta en (El-Darymli and Ahmed, 2012), es un laboratorio remoto tipo cluster, ya que consiste de un laboratorio físico en el cual se encuentra el equipo bajo prueba, y un sistema de gestión instalado en una sola computadora, que permite el acceso remoto para realizar los experimentos. El modelo de Marianetti fue utilizado para implementar un laboratorio remoto de PLD.

La figura 4.1 repite el modelo de capas de Marianetti, la descripción de cada una de las capas fue presentada en la sección 2.3.5. El módulo de seguimiento de alumnos muestra



**Figura 4.1:** Modelo de capas para laboratorios remotos destinados a educación propuesto por Marianetti. Figura obtenida de (Marianetti, 2006)

la orientación de este modelo a tareas de docencia. El laboratorio remoto presentado en esta tesis tiene como objetivo poder adaptarse para ser utilizado en tareas de docencia, pero sin limitar su aplicación a las mismas, sino que también pueda aplicarse a formación y entrenamiento de usuarios, realización de experimentos e investigación científica.

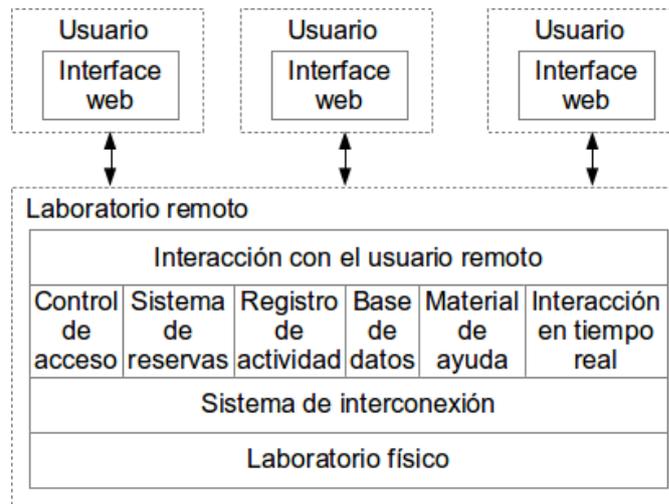
#### 4.3.3. Adaptación del Modelo de Marianetti para implementar un laboratorio remoto de WSN

A partir del modelo de Marianetti, del estudio del estado del arte sobre laboratorios remotos presentado en la sección 2.3, y del estudio del estado del arte sobre la construcción de laboratorios remotos de WSN presentado en la sección 3.1, se elabora y propone el modelo de capas para laboratorios remotos de WSN presentado en la figura 4.2.

Del modelo de Marianetti se toman varios módulos. Uno de ellos es el módulo de seguimiento de alumnos, al que hemos denominado módulo de registro de actividad. Este módulo registrará la actividad realizada por los usuarios. Solo los usuarios autorizados podrán acceder a esta información. Este módulo tiene la función de permitir que el laboratorio remoto pueda ser utilizado en tareas de docencia, para que los estudiantes puedan realizar trabajos prácticos y/o exámenes, y los docentes puedan monitorear y evaluar la actividad realizada por los alumnos.

Del modelo de Marianetti también se toman los módulos de control de acceso, agenda de reservas, base de datos y material de ayuda.

Con respecto al modelo de Marianetti, las tres capas que permiten conectar el sistema bajo prueba con los demás módulos de gestión del laboratorio remoto, que son las capas de gestor de configuraciones, control del sistema de interconexión y sistema de interconexión-interfaces, se han unificado en una sola capa denominada sistema de interconexión. Este módulo está compuesto tanto por componentes de software como



**Figura 4.2:** Modelo de capas de para laboratorios remotos de WSN

componentes de hardware. La forma en que esté construido este sistema depende del equipo bajo prueba. Para el caso de una WSN, que posee una conexión estática, sin elementos móviles ni posibilidad de diferentes conexiones físicas con el laboratorio remoto, el diseño del sistema de interconexión es simple.

El módulo de interacción con el usuario remoto se construye empleando alguna tecnología de generación de contenido web dinámico, por ejemplo PHP o RPC (Remote Procedure Call), de modo que las funciones más complejas de la interacción con el usuario puedan ejecutarse en el servidor. De esta manera, la computadora del usuario remoto solo ejecuta código html y JavaScript en un navegador web estándar.

Conceptualmente, las capas de interacción con el usuario remoto y el sistema de interconexión son las que permiten que el laboratorio pueda ser accedido remotamente. Las demás capas añaden funcionalidades al laboratorio remoto, como controlar el acceso de usuarios, permitir el acceso ordenado a través de un sistema de turnos o almacenar la actividad realizada por los usuarios.

Como puede verse en la figura 4.2, varios usuarios pueden acceder al mismo tiempo al laboratorio remoto, pero realizando diferentes tareas. Un usuario puede estar accediendo a la base de datos para recuperar resultados, otro puede estar interactuando en tiempo real con el laboratorio remoto, mientras que otro puede estar reservando un turno para la realización de un experimento. Por otro lado, si el equipamiento del laboratorio físico fuese suficiente en cantidad, varios usuarios podrían realizar experimentos de manera simultánea, accediendo cada usuario a una parte del equipo bajo prueba. Para el laboratorio remoto de redes de sensores inalámbricos presentado en esta tesis, esto sería posible si el laboratorio físico contara con varias WSN, cada una con un nodo coordinador conectado a la computadora del laboratorio físico.

#### 4.3.4. Modelo de capas para implementar el laboratorio remoto de WSN en el Cloud

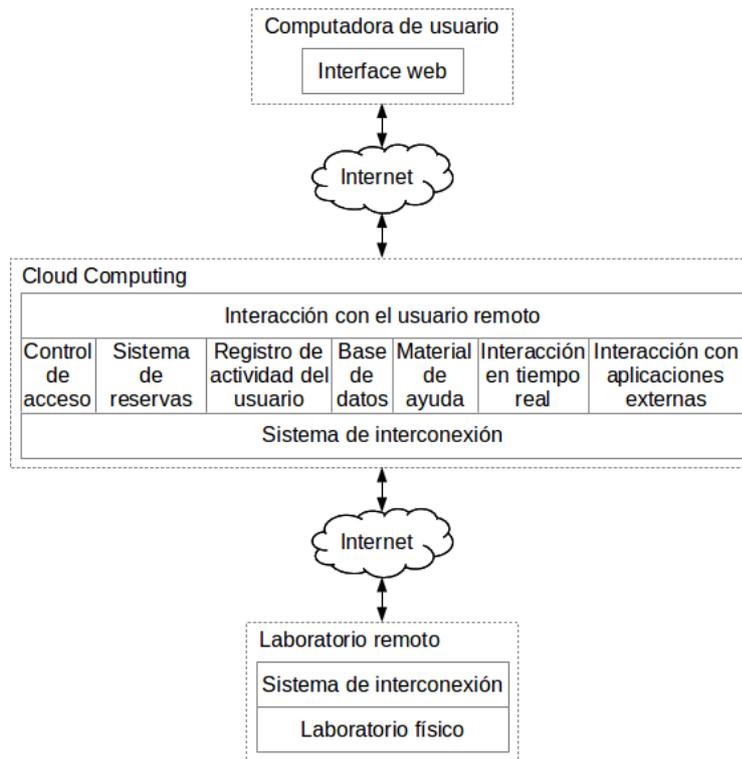
Un tema no tratado por los laboratorios remotos actuales es brindar al usuario características como calidad de servicio, seguridad y almacenamiento de información seguro (Maiti and Tripathy, 2013). En la actualidad, los proveedores de servicios de Cloud Computing ofrecen altos niveles de calidad de servicio, seguridad y almacenamiento seguro información (con copias de respaldo, protección ante intrusos, etc.). Por ello, implementar algunos de los componentes de los laboratorios remotos mediante servicios de Cloud Computing puede aportar estas características a los laboratorios remotos. Además, emplear servicios de Cloud Computing puede aportar otras características ventajosas a los laboratorios remotos, por ejemplo:

- Alta disponibilidad: Los componentes implementados mediante servicios de Cloud Computing estarán siempre disponibles y operativos, de modo que, aún en caso de que falla en los equipos de laboratorio físico, en su conexión a Internet o suministro de energía eléctrica, los usuarios podrán acceder a los datos almacenados en las bases de datos, reservar turnos, etc.
- Mayor robustez: Los componentes desplegados en el Cloud estarán siempre operativos, pudiendo llevar a cabo las acciones necesarias en caso de que los equipos del laboratorio físico fallen (ejecutar algoritmos de detección y corrección de fallas, derivar el experimento a otro laboratorio físico, informar al administrador, etc.).
- Escalabilidad: Pueden agregarse más WSN, en el mismo laboratorio físico o en otro, sin que sea necesario hacer cambios en la aplicación servidor, ni en la forma en que el usuario accede.

La implementación de los módulos de gestión del laboratorio remoto en el Cloud se hace de acuerdo al modelo de capas de la figura 4.3. Este modelo se deriva del modelo presentado en la figura 4.2. Todos los módulos de gestión del laboratorio remoto son implementados completamente en el Cloud, a excepción del sistema de interconexión, el cual es dividido en dos. Una parte del sistema de interconexión se implementa en el Cloud, la otra parte se implementa en el laboratorio físico.

Con respecto a un laboratorio remoto convencional (no implementado en el Cloud), como el mostrado en la figura 4.2, solo la implementación del sistema de interconexión requiere modificaciones importantes. Los demás módulos del laboratorio remoto pueden ser implementados en el Cloud sin modificaciones mayores.

El único antecedente hallado en la biografía al momento de escribir esta tesis de un laboratorio remoto desplegado en el Cloud es (Orduna et al., 2015). Dicho trabajo



**Figura 4.3:** Laboratorio remoto implementado mediante servicios de Cloud Computing

presenta una descripción de una arquitectura de laboratorio remoto desplegada en un Cloud privado, pero declara que la implementación de esta arquitectura no está aun disponible públicamente ya que está en etapa de prueba y desarrollo. Dicho trabajo declara la necesidad de realizar la implementación de la arquitectura en un Cloud público, como Amazon o GAE.

#### 4.3.4.1. Interacción con aplicaciones externas

El modelo de capas presentado en la figura 4.3 añade un módulo denominado "Interacción con aplicaciones externas". Este módulo tiene la finalidad de extender las funcionalidades del laboratorio remoto, siendo el objetivo principal que aplicaciones externas puedan interactuar con el laboratorio remoto de acuerdo al comportamiento de modelos simulados de diferentes tipos. El módulo de interacción con aplicaciones externas puede operar de varias maneras:

- Procesando instrucciones enviadas por el modelo (por ejemplo: desconectar un nodo de la red o cambiar cualquiera de los parámetros que controlan el funcionamiento del nodo, como la potencia o frecuencia de transmisión de datos).

- Reemplazando o modificando los datos generados por los sensores reales del laboratorio remoto por datos generados por el modelo, para simular mediciones de los sensores o fallas en los mismos.
- Alimentando al modelo con datos producidos por los nodos reales, de manera que el modelo pueda evolucionar de acuerdo a estos datos.
- Simulando fallas en la transmisión de datos.

#### **4.4. Descripción e Implementación de los Componentes del Laboratorio Remoto**

A continuación se describen cada uno de los módulos del laboratorio remoto y actores involucrados.

##### **4.4.1. Proveedor de servicios de Cloud Computing**

El sistema implementado utiliza servicios de Cloud Computing. Por lo tanto, es necesario elegir un proveedor de servicios de Cloud Computing. Se necesita que el proveedor seleccionado proporcione como mínimo las siguientes características:

- Permita almacenar datos en el Cloud.
- Permita implementar un mecanismo de comunicación con una aplicación remota. Esta aplicación remota estará a cargo de controlar el equipo bajo prueba. Esta aplicación debe poder acceder a los puertos (USB, seriales, etc) y al sistema operativo de la computadora remota.
- Brinde datos de fecha y hora.

Además, los siguientes requisitos no son necesarios, pero si deseables:

- Que el proveedor de servicios de Cloud Computing ofrezca un mecanismo de autenticación de usuarios.

Los primeros tres requisitos citados son necesarios para poder implementar el modelo propuesto. El proveedor de servicios de Cloud Computing debe permitir almacenar datos y debe permitir consultar la fecha y hora para poder implementar el sistema de seguimiento de usuarios en el Cloud. De esta manera, el almacenamiento de datos será seguro e independiente del laboratorio físico, de modo que pueda funcionar aún si el laboratorio físico está fuera de servicio.

La comunicación con una aplicación remota es necesaria para implementar el sistema de interconexión con el equipo bajo prueba. Esta aplicación remota se encargará de controlar el equipo bajo prueba y de comunicar el mismo con los módulos de gestión desplegados en el Cloud. Debe poder acceder a los puertos de la computadora en la cual se ejecute para poder comunicarse con el equipo bajo prueba. La comunicación puede realizarse por diversos mecanismos que permitan comunicación confiable, por ejemplo: socket TCP, llamada a procedimientos remotos (RPC), sincronización de archivos, etc.

En cuanto al servicio de autenticación de usuarios, si bien el programador podría implementar su propio servicio, es preferible que el proveedor provea un servicio de autenticación de usuarios seguro, que sea actualizado de manera permanente ante nuevos tipos de ataques y vulnerabilidades.

Los requisitos necesarios del proveedor de servicios de Cloud Computing son básicos, ya que no se necesita elevado poder de procesamiento o gran espacio para almacenamiento de datos. Por este motivo, es esperable que la mayoría de los proveedores de servicios de Cloud Computing que ofrezcan servicios de Platform as a Service (PaaS) puedan satisfacer los requisitos mencionados. Para esta primera implementación, el proveedor de servicios de Cloud Computing elegido es Google App Engine (GAE) (Google Inc., 2013), que ofrece servicios de Platform as a Service (PaaS). Estos servicios permiten crear aplicaciones para que se ejecuten en la infraestructura de Google.

#### **4.4.2. Usuarios**

Los usuarios se han categorizado en tres grupos diferentes: clientes, supervisores y administradores. Los usuarios supervisores y administradores pueden acceder al registro de actividades realizadas por los demás usuarios del laboratorio remoto. De esta manera, pueden realizar un seguimiento de la actividad realizada por los usuarios clientes.

Los usuarios administradores están a cargo de la administración de todo el sistema. También se encargan de indicar cuales usuarios son supervisores y cuales clientes. Para ello, almacenan en la base de datos la información de los usuarios habilitados como supervisores.

Cada tipo de usuario verá un entorno diferente del laboratorio remoto.

#### **4.4.3. Interface Web**

Esta interface permite que los usuarios interactúen con el laboratorio remoto. Permite a los usuarios realizar las siguientes tareas:

- Solicitar un turno para acceder al laboratorio remoto.

- Modificar y visualizar los valores de los parámetros de los nodos.
- Permite a los usuarios supervisores visualizar la actividad realizada por los demás usuarios.

Como se dijo anteriormente, la interface web debe ser didáctica, de fácil uso e intuitiva. Para construir la interface web, se han seleccionado un conjunto de parámetros para que puedan ser modificados por los usuarios a través del laboratorio remoto. Para cada uno de estos parámetros, se han seleccionado un conjunto de posibles valores. Estos parámetros son los mismos que el usuario programaría si tuviera los nodos conectados a su computadora. Por ejemplo, para los nodos XBee, que son los nodos elegidos para implementar el laboratorio remoto, si el usuario desea programar la potencia de transmisión, deberá configurar los parámetros PM (power mode) y PL (power level), y si desea programar la frecuencia de reporte de datos, deberá programar los parámetros ST (time before sleep), SP (cyclic sleep period) y SN (number of cycles to power down IO) (Digi International Inc., 2008).

Cada tipo de nodo posee un conjunto de parámetros de programación característicos. Por lo tanto, si se desea agregar al laboratorio remoto otro tipo de nodo, es necesario modificar la interface web para que se adapte a los parámetros del nuevo tipo de nodo. Este problema se solucionaría con una interface web genérica, que genere los comandos necesarios según la variable que se desee modificar, por ejemplo, que calcule y programe los valores de los parámetros ST, SP y SN para la frecuencia de reportes que el usuario seleccione. Sin embargo, una interface web genérica quitaría transparencia y realismo al laboratorio remoto. En este trabajo se ha optado por transparencia y realismo en lugar de interface genérica. Para evitar tener que modificar la interface web para otro tipo de nodos, se estudiará incorporar una biblioteca de componentes para WSN.

También podría optarse por permitir al usuario enviar cualquier comando y cualquier valor a un nodo. Si bien se ha contemplado la opción, esto haría más complejo el uso del laboratorio remoto para un usuario sin conocimientos avanzados en WSN. Por este motivo se ha optado por seleccionar una serie de parámetros y posibles valores, los cuales el usuario puede configurar con un par de clics del mouse de su computadora.

La figura 4.4 muestra una captura de parte de la interface web. Puede verse que para cada nodo se muestra una lista de parámetros de configuración. En el caso de los pines de entrada/salida, el usuario puede configurar su función (deshabilitado, entrada, salida, conversor analógico digital) y visualizar el valor registrado en cada pin.

La interface web ha sido desarrollada empleando lenguaje de programación HTML y JavaScript. Estos lenguajes son aceptados por los navegadores de Internet más populares, tanto para computadoras como para teléfonos celulares.

Turno asignado. Tiempo restante: 51 minutos. Para modificar un parámetro de un nodo, haga clic:

Ir a turnos		Ver Instrucciones		Experimentos		Ver Explicación de parámetros	
Nodo 0		Nodo 1		Nodo 2		Nodo 3	
panID: AAAA		panID: AAAA		panID: AAAA		panID: AAAA	
ST: nodata		ST: 0003E8		ST: 0003E8		ST: 0003E8	
SP: nodata		SP: 000064		SP: 000064		SP: 000064	
SN: nodata		SN: 000003		SN: 000003		SN: 000003	
SO: nodata		SO: 0004		SO: nodata		SO: 0004	
IR: nodata		IR: nodata		IR: nodata		IR: nodata	
PM: nodata		PM: 0001		PM: 0001		PM: 0001	
PL: nodata		PL: 0004		PL: 0004		PL: nodata	
DB: nodata		DB: nodata		DB: nodata		DB: nodata	
Pin valor configuracion		Pin valor configuracion		Pin valor configuracion		Pin valor configuracion	
DIO0: No data x		DIO0: No data E F		DIO0: 0204 ADC		DIO0: No data x	
DIO1: No data x		DIO1: No data x		DIO1: 1 Input		DIO1: No data x	
DIO2: No data x		DIO2: 1 x		DIO2: 0 OutputL		DIO2: 1 x	
DIO3: No data x		DIO3: 1 OutputH		DIO3: 01EC ADC		DIO3: No data x	
DIO12: No data x		DIO12: 1 OutputH		DIO12: No data Disabled		DIO12: No data x	

Figura 4.4: Configuración de parámetros en la interface web

#### 4.4.4. Módulo de interacción con el usuario remoto

La interacción de los usuarios con el laboratorio remoto es a través de la interface web, cuyo código es generado dinámicamente en el servidor por el módulo de interacción con el usuario remoto. Este módulo genera y modifica la información mostrada en la interface web en función de la información brindada por todos los módulos que tiene por debajo en el modelo de capas de la figura 4.3. Al mismo tiempo recibe la información proveniente desde la interface web y la transfiere a los respectivos módulos del modelo de capas.

La implementación de la interface de usuario consiste en el desarrollo de una aplicación web dinámica. La misma se implementó empleando el mecanismo RPC, clases de Java Servlet en el Cloud y las herramientas de desarrollo de GAE.

#### 4.4.5. Control de acceso

El control de acceso se realiza a través de una cuenta de usuario y una contraseña. GAE ofrece el uso de autenticación de usuarios a través de cuentas de Gmail (Google Inc., 2013), por lo que la implementación del control de acceso empleando la herramienta provista por GAE es directa.

Los usuarios, para poder acceder al sistema de reservas de turnos, y para poder realizar experimentos en el laboratorio remoto, deben primero acceder a su cuenta de Gmail con su nombre de usuario y contraseña. Una vez que el usuario ha accedido a su cuenta de Gmail, será redireccionado a la página web del laboratorio remoto y podrá realizar la reserva de un turno y realizar experimentos. Un usuario también puede acceder

al laboratorio remoto sin entrar en su cuenta de Gmail, pero solo podrá visualizar el laboratorio remoto, sin poder realizar experimentos ni realizar la reserva de turnos.

Podría emplearse un mecanismo de autenticación mediante cuenta de usuario y contraseña sin emplear una cuenta de Gmail, para lo cual debería implementarse una base de datos de usuarios y contraseñas, como también mecanismos de seguridad típicos (detección de inicio de sesión en una computadora diferente a la habitual, mecanismo CAPTCHA, etc). Para esta implementación, se ha optado por confiar en la herramienta provista por GAE, es decir, la autenticación a través de cuenta de Gmail. No se descarta en el futuro emplear otro mecanismo de autenticación, como podría ser la cuenta de usuario de la página web de la Universidad en la cual el laboratorio remoto se utilice como herramienta para la enseñanza.

#### **4.4.6. Sistema de reserva de turnos**

Para poder realizar un experimento sobre el laboratorio remoto, el usuario debe reservar un turno a través del formulario web creado para este fin. El usuario solo podrá realizar experimentos dentro del horario de su turno. Fuera de este horario, el usuario solo podrá visualizar el laboratorio remoto, con los resultados producidos por los experimentos de otros usuarios.

Para implementar este sistema, cada turno se almacena como un registro en la base de datos del servidor. Cada registro incluye el usuario, la fecha y la hora de inicio y fin del experimento. Una aplicación, escrita en lenguaje Java, que se ejecuta en el servidor, mantiene ordenados estos registros, verifica la disponibilidad cuando el usuario solicita un turno, e indica al módulo de control de acceso cuál es el usuario propietario del turno actual, cada vez que este módulo lo solicite.

Para obtener la fecha y hora actual, y para almacenar datos, el algoritmo emplea los respectivos servicios de GAE. Todos los usuarios poseedores de una cuenta de Gmail pueden acceder al laboratorio remoto y realizar la reserva de un turno. El usuario administrador puede agregar turnos o modificar los turnos ya reservados.

Cada turno se implementa mediante un objeto de una clase creada a tal fin. Cada objeto posee tres variables, la fecha y hora de inicio del turno, la fecha y hora de fin del turno, y el nombre de usuario propietario del turno. Para representar las fechas y horas de inicio y fin de los turnos se utiliza el tiempo Unix, es decir, el número de segundos desde el momento de referencia (medianoche UTC del 1 de enero de 1970).

#### **4.4.7. Registro de actividad de los usuarios**

El laboratorio remoto registra las acciones realizadas por los usuarios. Cada registro almacena la fecha y hora de la acción, el usuario, y la acción realizada. Los registros se almacenan en la base de datos en el servidor. Las acciones que el laboratorio remoto registra son:

- Envío de un comando.
- Reserva de un turno de usuario para utilizar el laboratorio remoto.

Solo los usuarios supervisores pueden acceder a esta información. Esta función tiene el objetivo de permitir en el futuro adaptar el laboratorio remoto a tareas de enseñanza o docencia. El registro de actividades de los usuarios puede ser utilizado por los docentes para seguimiento y evaluación de los estudiantes.

Desde el punto de vista de la programación, cada registro constituye un objeto de una clase llamada "registro de actividad". Este objeto tiene tres campos, fecha y hora, usuario, y acción realizada. El objeto se crea en el momento de guardar un registro en la base de datos, se almacena y se destruye. Cuando un usuario necesita acceder a la información almacenada en la base de datos, se recupera una lista de objetos de esta clase, se transforman en una cadena de caracteres, y se envían al usuario como valor de retorno de un procedimiento que ha sido invocado remotamente a través del mecanismo de RPC que permite GAE. El motivo de transformar la información en una cadena de caracteres es que la misma pueda mostrarse en cualquier computadora simple o dispositivo móvil que pueda presentar páginas web.

En caso de utilizar otro proveedor de servicios de Cloud Computing, los datos pueden recuperarse de la base de datos, y enviarse al usuario remoto por el mecanismo que el proveedor de servicios Cloud provea, que podría ser socket TCP, transferencia o sincronización de archivos, etc.

#### **4.4.8. Base de datos**

Se utiliza una base de datos para almacenar la siguiente información:

- Actividad de los usuarios.
- Nombre de usuario de usuarios supervisores y administradores.
- Turnos reservados.

Para mantener y acceder a la base de datos, se utilizó la implementación de Java Data Objects (JDO) que provee GAE. JDO es una especificación para el almacenamiento de objetos de Java en bases de datos.

Cada uno de los ítem mencionados se almacena como un tipo de dato diferente. La actividad de los usuarios se almacena como se mencionó en la sección 4.4.7, a través de objetos de una clase creada para registrar la actividad de los usuarios. Los nombres de usuarios supervisores y administradores se almacenan como cadena de caracteres. El sistema de reserva de turnos se implementa como una lista ordenada de objetos de una clase creada para tal fin, como se indicó en la sección 4.4.6. Estos objetos se almacenan directamente en la base de datos.

#### **4.4.9. Material de ayuda**

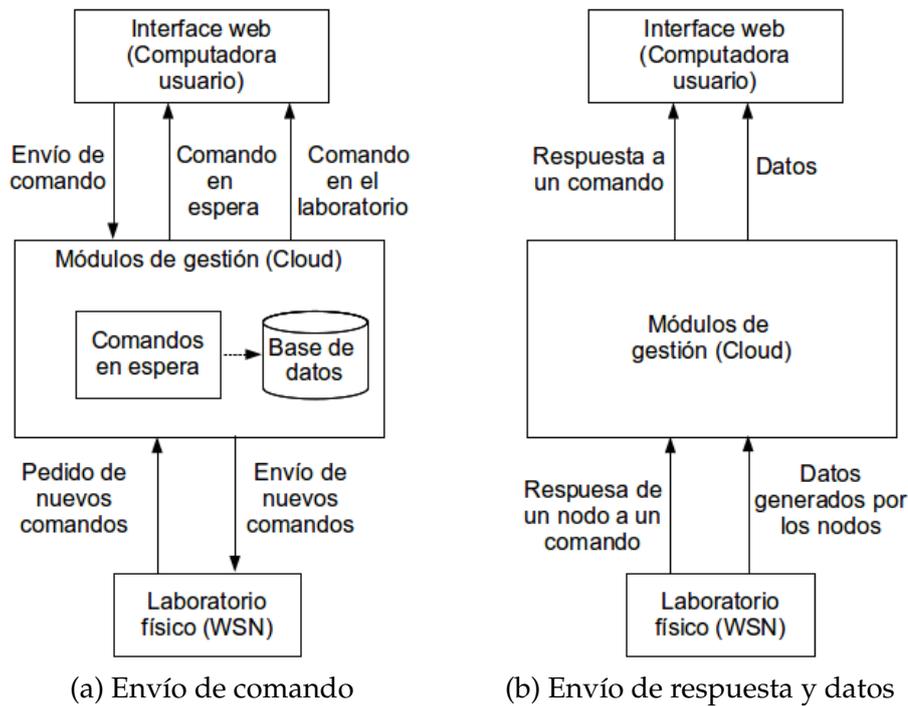
Debido a que el laboratorio remoto está diseñado para ser utilizado por usuarios que desean formarse en el uso de las WSN, es necesario brindar toda la información y ayuda necesaria a los mismos. El módulo de material de ayuda se encarga de esta tarea.

#### **4.4.10. Interacción en tiempo real**

Los laboratorios remotos pueden funcionar, como se describió en la sección 3.1.3, en dos modos: por lotes o en tiempo real. Se ha empleado el modo de operación en tiempo real para el laboratorio remoto propuesto en este trabajo, ya que permite una interacción más cercana a utilizar el laboratorio en modo presencial. En el modo en tiempo real, el usuario puede enviar comandos a los nodos y ver en el mismo momento los cambios.

La figura 4.5.a muestra el proceso de envío de un comando desde la interfaz de usuario al laboratorio remoto. Cuando el usuario selecciona un nuevo valor para un parámetro de un nodo, la interface web envía una petición RPC a los módulos de gestión en el Cloud. Esta petición tiene como parámetros el comando y el nuevo valor. Los módulos de gestión en el Cloud, luego de comprobar que el usuario es el propietario del turno actual, almacenan temporalmente el comando y su valor en un registro, y envían a la interface web una respuesta indicando que el comando está en el servidor esperando ser enviado al laboratorio. Al mismo tiempo, el comando se almacena en la base de datos. Cuando el laboratorio remoto solicita nuevos comandos, los módulos de gestión en el Cloud le envían los comandos en espera, y envían un mensaje a la interface web para indicar al usuario que los comandos ya han sido enviados al equipo bajo prueba.

En el laboratorio físico, un programa escrito en lenguaje Java, denominado aplicación cliente (porque actúa como cliente de una aplicación cliente-servidor en el sistema de interconexión, como se describe en la sección 4.4.12) genera la trama de datos ZigBee



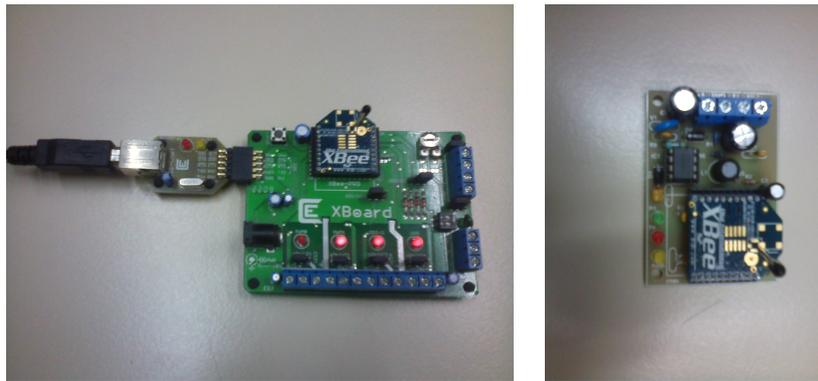
**Figura 4.5:** Comunicación de comandos y datos durante la realización de un experimento

correspondiente a dicho comando, y la envía al equipo bajo prueba (en este caso la WSN). El equipo bajo prueba procesa este comando, realiza las acciones indicadas por el mismo, y envía las respuestas que se puedan generarse a los módulos de gestión en el Cloud (a través de una petición RPC realizada por la aplicación cliente). El equipo bajo prueba también puede enviar reportes de datos propios del experimento que se esté realizando, como se muestra en la figura 4.5.b.

#### 4.4.11. Laboratorio físico

Los equipos del laboratorio físico, o sistema bajo prueba, son los nodos sensores, que forman una red de sensores inalámbricos. Esta red está compuesta por nodos sensores, los cuales se han programado uno como coordinador, uno como ruteador y los demás como end-devices.

Para este laboratorio remoto se han seleccionado nodos XBee, fabricados por la empresa Digi (Digi International Inc., 2008). Estos nodos emplean el protocolo de comunicaciones ZigBee (ZigBee Alliance, 2008b). Los nodos pueden configurarse mediante comandos, donde cada comando indica cuál parámetro modificar y el nuevo valor de dicho parámetro. Estos comandos se envían en tramas de datos preestablecidas por el fabricante. Los nodos pueden configurarse de manera que reporten periódicamente los valores



(a) Nodo coordinador

(b) Nodo end-device

**Figura 4.6:** Nodos Coordinador y End-Device

leídos en sus pines, que pueden ser entradas/salidas digitales o conversores analógicos digitales.

La figura 4.6 muestra el nodo coordinador en su placa de desarrollo y uno de los nodos end devices.

Los nodos han sido desplegados en el Laboratorio de Producción Integrada por Computadora (LAPIC), ubicado en las instalaciones de la carrera de Redes y Telecomunicaciones del Instituto Tecnológico Universitario (ITU), perteneciente a la Universidad Nacional de Cuyo.

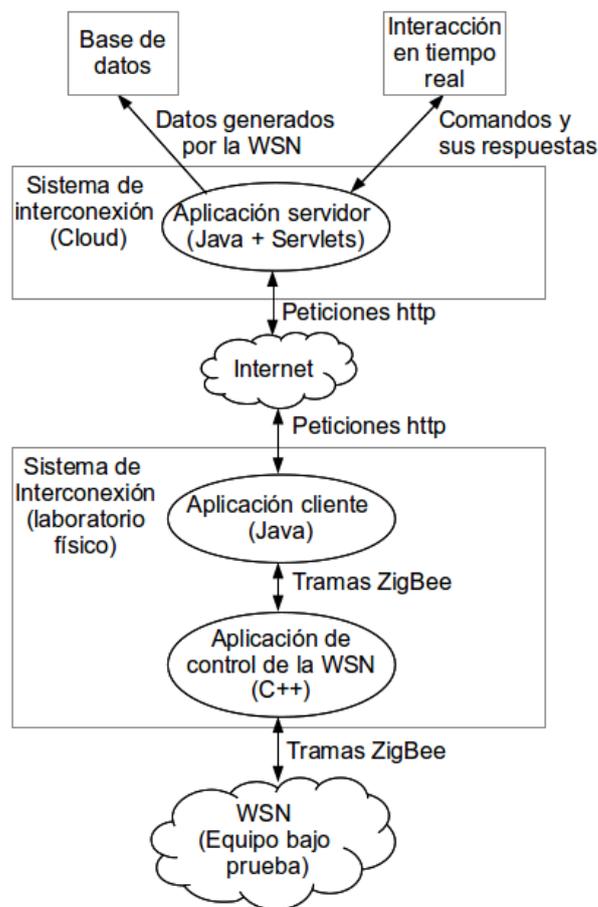
#### 4.4.12. Sistema de interconexión

El sistema de interconexión es el único módulo del laboratorio remoto formado por componentes de hardware y componentes de software.

El sistema de interconexión debe interconectar el laboratorio físico con los módulos de gestión implementados en el Cloud. Para lograr este objetivo, el sistema de interconexión se implementa como una aplicación cliente servidor. La aplicación servidor se ejecuta en el Cloud, mientras que la aplicación cliente se ejecuta en la computadora que controla la WSN, localizada en el laboratorio físico. Ambas aplicaciones se comunican a través de Internet, como se muestra en la figura 4.7.

La aplicación cliente se comunica cada 0.5 segundos con la aplicación servidor para:

- Enviar a la aplicación servidor los datos que los nodos sensores hayan obtenido en ese intervalo. Estos datos pueden ser estados lógicos de sus pines o valores analógicos digitalizados, según la configuración que el usuario haya dado a los pines. El tiempo de muestreo de los nodos es configurado por el usuario.



**Figura 4.7:** Sistema de interconexión

- Solicitar nuevos comandos para programar los nodos (comandos que los usuarios hayan enviado a través de la interface web).

Los nodos pueden generar respuestas a los comandos que reciben. Estas respuestas son enviadas sin demora a la aplicación servidor, para que la misma las procese. La aplicación cliente también se comunica con un programa, escrito en lenguaje C++, el cual se encarga de la comunicación con la WSN a través del puerto USB, como se muestra en la figura 4.7.

Debido a que el sistema de interconexión es una aplicación cliente servidor, los comandos enviados por los usuarios arriban y permanecen en la aplicación servidor hasta que la aplicación cliente (en el laboratorio) los solicita. La aplicación cliente pregunta y solicita nuevos comandos cada 0.5 segundos, por lo tanto, este esquema produce una latencia entre cero y 0.5 segundos.

La aplicación servidor se implementó utilizando lenguaje de programación Java y se ejecuta en el Cloud. Por un lado, la aplicación servidor recibe comandos desde el

usuario (a través del módulo de interacción en tiempo real) y los envía al laboratorio físico a través de Internet. Por otro lado, la aplicación servidor recibe respuestas a estos comandos o reportes de datos desde la WSN. Las respuestas a estos comandos son enviadas al módulo de interacción en tiempo real, y este módulo las envía al usuario correspondiente. Los reportes de datos son enviados a la base de datos para ser almacenados temporalmente. La aplicación cliente fue escrita en lenguaje Java y se ejecuta en la computadora que controla la WSN, localizada en el laboratorio físico.

El sistema de interconexión fue implementado usando la herramienta RPC provista por GWT. GWT-RPC permite a los usuarios crear aplicaciones web tipo cliente servidor empleando Servlet. Los Servlet son clases de Java que permiten crear contenido web dinámico. RPC-GWT crea aplicaciones cliente escritas en JavaScript para ser ejecutadas en navegadores web. Para la implementación de este laboratorio remoto, la aplicación cliente creada por GWT fue reemplazada por un programa escrito en lenguaje Java que ejecuta las mismas funciones, pero además invoca, gestiona y se comunica con el programa de control de la WSN, escrito en lenguaje C++.

#### **4.5. Implementación basada en el Cloud vs implementación tradicional**

La primera implementación de este laboratorio remoto no fue realizada en el Cloud, fue realizada empleando un esquema tradicional para laboratorios remotos: una PC que controla al equipo bajo prueba y permite el acceso remoto a través de una red LAN. La implementación en el Cloud no estaba entre los objetivos iniciales o aportes esperados de esta tesis.

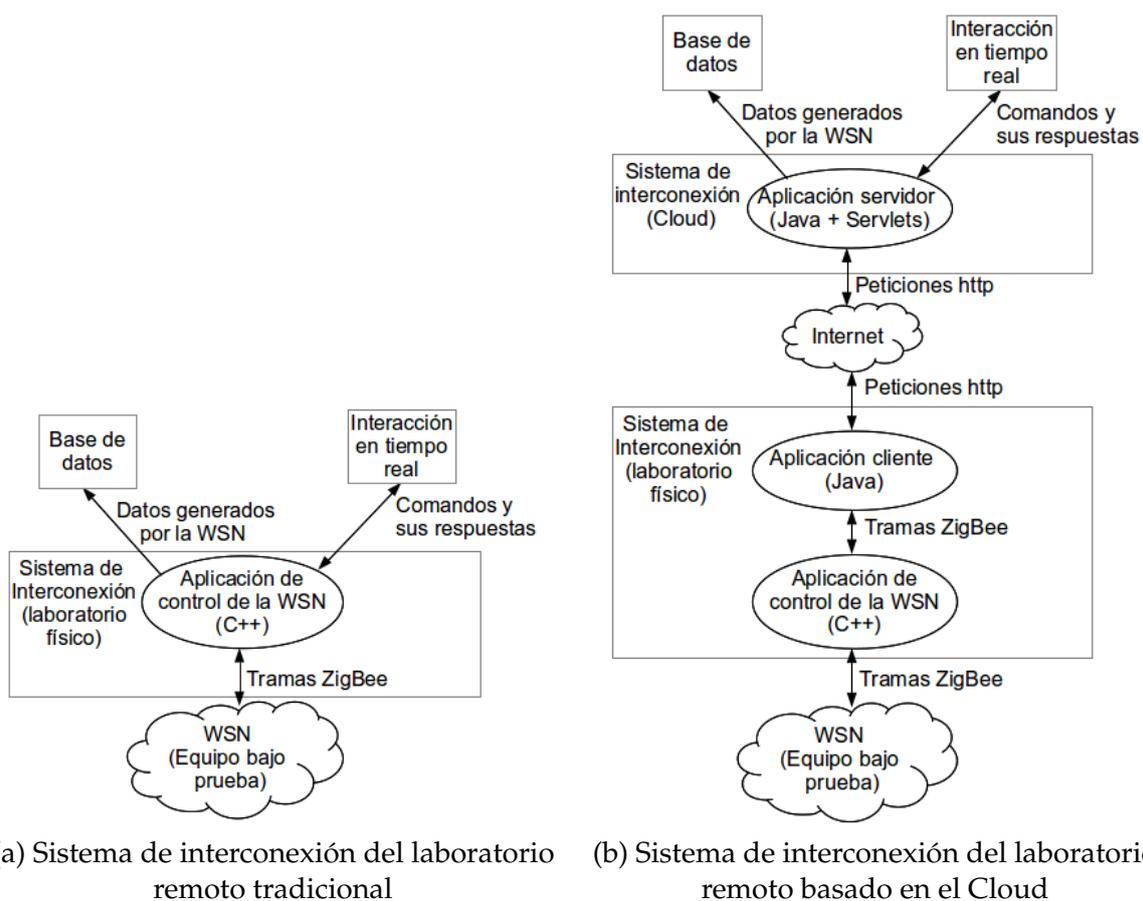
Posteriormente, la lectura de publicaciones recientes (Orduna et al., 2013)(Orduna et al., 2015)(Maiti and Tripathy, 2013) motivo el interés por investigar las ventajas y desventajas de integrar el laboratorio remoto con tecnologías de Cloud Computing. Por los motivos expuestos en la sección 4.3.4 se decidió realizar el despliegue en el Cloud. Para ello se propuso el modelo de capas presentado en la figura 4.3.

Este camino permitió poder comparar las diferencias constructivas en ambos despliegues, implementación en Cloud Computing e implementación tradicional. Pudo verse que el sistema de interconexión es el único componente que requiere modificaciones importantes.

Para el despliegue tradicional, los componentes de hardware del sistema de interconexión consisten en cables USB y placas de desarrollo. Los componentes de software consisten en los drivers del sistema operativo y un programa escrito en lenguaje C++, que

recibe y envía tramas de datos desde y hacia la WSN a través del puerto USB. La arquitectura del sistema de interconexión para el laboratorio remoto tradicional se muestra en la figura 4.8.a.

Cuando se implementan los módulos de gestión del laboratorio remoto en el Cloud, el sistema de interconexión debe interconectar el laboratorio físico con los módulos de gestión implementados en el Cloud. Para lograr este objetivo, a los componentes mencionados anteriormente se agrega la aplicación cliente servidor descrita en la sección 4.4.12 que permite interconectar el equipo bajo prueba con los módulos de gestión desplegados en el Cloud. En otras palabras, el sistema de interconexión fue dividido en dos. Una parte es implementada en el Cloud, y la otra parte es implementada en el laboratorio físico. La figura 4.8 permite comparar el sistema de interconexión para ambos despliegues.



**Figura 4.8:** Sistema de interconexión para laboratorio remoto tradicional y para laboratorio remoto basado en Cloud Computing

## 4.6. Conclusiones

El estudio del estado del arte realizado en el capítulo 3 permitió detectar varios problemas no resueltos en el campo de los laboratorios remotos de WSN. Entre ellos, la falta de laboratorios remotos de WSN destinados a usuarios que deseen formarse, entrenarse o experimentar en programación y funcionamiento de las WSN, o en desarrollo de aplicaciones que utilicen WSN. Se llegó a la conclusión de que hay numerosos laboratorios remotos de WSN, pero en general el uso de éstos requiere de elevados conocimientos de programación y configuración de WSN, lo que dificulta el acceso a usuarios que desean realizar sus primeras prácticas con estas tecnologías.

En base al estudio de los laboratorios remotos destinados a educación en otras áreas, realizado en el capítulo 2, se propuso una arquitectura para laboratorios remotos de WSN orientados a usuarios que deseen entrenarse o experimentar en programación y funcionamiento de las WSN. Se incluyó un módulo de registro de la actividad de los usuarios que permita seguir la actividad realizada por los usuarios, para poder adaptar, en trabajos futuros, el uso del laboratorio remoto a tareas de docencia. Por último se construyó un prototipo del mismo.

Los componentes distintivos de este laboratorio remoto son la interface web didáctica y de fácil uso, junto con el módulo para almacenar información sobre la actividad realizada por los usuarios.

El modelo propuesto y su implementación están basados en el Cloud, como se presenta en la figura 4.3. Se utilizaron los servicios de Cloud Computing de PaaS provistos por GAE. Respecto a un laboratorio remoto tradicional, los módulos de gestión del laboratorio remoto no sufren modificaciones mayores. En cambio, el sistema de interconexión, que permite conectar los equipos bajo prueba con el resto del sistema, requirió modificaciones importantes. El sistema de interconexión permite interconectar el equipo bajo prueba, en este caso una WSN, con los módulos de gestión desplegados en el Cloud. Para implementar el sistema de interconexión se empleó una aplicación cliente servidor basada en el mecanismo RPC que implementa GAE. Se deja para trabajos futuros el estudio de otro tipo de mecanismos para implementar el sistema de interconexión, como sockets TCP o sincronización de archivos.

Se concluyó que las características propias del paradigma de Cloud Computing puede dar solución a algunos problemas como también dotar de características ventajosas a los laboratorios remotos, como el almacenamiento seguro de información importante o alta robustez y disponibilidad. Por este motivo, el modelo propuesto emplea servicios de Cloud Computing para desplegar los módulos de gestión del laboratorio remoto. Del estudio del estado del arte se concluye que, al tiempo de escribir esta tesis, este modelo es uno de los primeros modelos propuestos para integrar laboratorios remotos a Cloud

Computing. También se puede deducir que, con el incremento de las propuestas de educación a distancia y sobretodo de los denominados cursos online abiertos y masivos o MOOCs por sus siglas en inglés (Massive Open Online Course) los laboratorios remotos y virtuales atraerán una mayor atención en los próximos años (Waldrop, 2013a)(Waldrop, 2013b). Se espera que la alternativa propuesta y el trabajo futuro derivado de esta tesis contribuyan con el desarrollo de nuevos laboratorios remotos integrados al Cloud que satisfagan las necesidades de las aplicaciones mencionadas.



# Capítulo 5

## Experimentos

El presente capítulo está dividido en tres secciones. La primera sección presenta experimentos de prueba de concepto, que son experimentos que los usuarios pueden realizar en el laboratorio remoto. Se presentan estos experimentos en el formato en el cual los usuarios los realizarían. La segunda sección presenta experimentos para evaluar el desempeño del laboratorio remoto. En la última sección se presentan las conclusiones.

### 5.1. Experimentos de prueba de concepto

Esta sección tiene la finalidad de mostrar algunos de los posibles experimentos que los usuarios pueden realizar en el laboratorio remoto.

#### 5.1.1. Tiempo de respuesta de la WSN

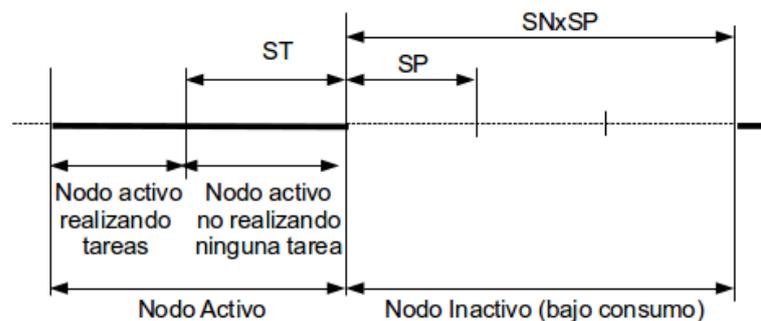
Con este experimento el usuario puede ver como varía el tiempo de respuesta de un nodo según como ha sido programado. El usuario puede programar el tiempo de respuesta de los nodos mediante la programación de los parámetros ST, SP y SN, que tienen las siguientes funciones:

- ST: Tiempo de inactividad que debe transcurrir antes de entrar en modo bajo consumo.
- SP: Intervalo de tiempo que permanece el nodo en bajo consumo.
- SN: Número de ciclos SP durante el cual el nodo permanece en modo bajo consumo.

La figura 5.1 muestra como se configura el ciclo de trabajo de un nodo sensor XBee en función de los valores dados a los parámetros anteriores. El nodo puede estar en dos estados: activo o bajo consumo. La figura muestra tres intervalos de tiempo diferentes, que son:

- **Nodo activo realizando tareas:** Cuando el nodo se despierta del modo bajo consumo y pasa al estado activo, se comunica con el nodo coordinador para recibir mensajes que el mismo tenga para enviarle, sensa sus pines de acuerdo a la configuración de los mismos, procesa los mensajes que el coordinador le haya enviado, y envía al coordinador la información que requiera ser enviada.
- **Nodo activo no realizando ninguna tarea:** Una vez que el nodo ha finalizado de realizar todas sus tareas, espera un tiempo  $ST$  por nuevas comunicaciones con el nodo coordinador. Si durante el tiempo  $ST$  no hay ningún mensaje proveniente del nodo coordinador, el nodo pasa al estado bajo consumo.
- **Nodo en estado bajo consumo,** el nodo esperará un tiempo  $SP \times SN$  antes de volver al estado activo.

Cuando el usuario envía un comando a un nodo de la WSN, este comando puede arribar al nodo coordinador en cualquier momento del ciclo de trabajo mostrado, y el tiempo de respuesta dependerá del momento en el cual arribe la petición al nodo coordinador. El tiempo de respuesta variará entre un valor mínimo que depende de la velocidad de respuesta del protocolo ZigBee (del orden de los 100 milisegundos), y un valor máximo que depende de los valores dados a los parámetros  $ST$ ,  $SN$  y  $SP$ . El usuario debe conocer cómo se configura el ciclo de trabajo de los nodos en una WSN para poder interpretar el valor del tiempo de respuesta que puede obtener al realizar un experimento.



**Figura 5.1:** Configuración del ciclo de trabajo de un nodo XBee

Para conocer el tiempo de respuesta y la latencia total, el usuario envía un comando cualquiera a un nodo cualquiera, y cuando la respuesta del nodo vuelve al usuario, la interface web muestra el tiempo de respuesta de la WSN. Para este experimento, se utilizó

el comando DB, que solicita el valor del RSSI (Indicador de Fuerza de la Señal Recibida) al nodo seleccionado.

La tabla 5.1 muestra mediciones de tiempo para diferentes configuraciones de los parámetros ST, SP y SN. Para realizar el experimento se utilizó el nodo 1 y el nodo coordinador. Se realizó una serie de 30 mediciones para cada combinación de los parámetros, mostrándose en la tabla el valor promedio, el menor valor, el mayor valor y la desviación estándar para cada serie de 30 mediciones correspondiente a cada combinación de parámetros.

Puede verse que el menor valor del tiempo de respuesta no depende de los valores de los parámetros, mientras que el mayor valor, el promedio y la desviación estándar de los tiempos de respuesta dependen de los valores dados a ST, SP y SN. Las filas 1 y 2 muestran que al aumentar el tiempo ST el tiempo de respuesta promedio disminuye. Esto es debido a que al aumentar el intervalo de tiempo durante el cual el nodo permanece encendido antes de entrar a modo bajo consumo, aumenta la probabilidad de que la petición llegue cuando el nodo está encendido, y por lo tanto pueda responder inmediatamente. Por otro lado, al incrementar el tiempo SP o el parámetro SN, no se incrementa el menor tiempo de respuesta registrado en cada serie de 30 mediciones, pero sí se incrementan el tiempo promedio de respuesta, el mayor tiempo de respuesta y la desviación estándar.

**Tabla 5.1:** Tiempos de respuesta del nodo en milisegundos

Valor parámetros			Tiempo promedio de respuesta de la WSN	Menor tiempo de respuesta	Mayor tiempo de respuesta	Desviación estándar
ST	SP	SN				
30	30	1	827 ms	465 ms	1297 ms	244 ms
3E8	30	1	667 ms	349 ms	869 ms	227 ms
3E8	3E8	1	4433 ms	776 ms	10213 ms	3268 ms
3E8	3E8	3	10093 ms	630 ms	20283 ms	8098 ms

Para realizar este experimento, el usuario debería desplegar los nodos de la WSN en su laboratorio, y programarlos con el firmware adecuado según su función. Además, debería utilizar o escribir un programa para comunicarse con el nodo coordinador a través del puerto serie, y enviarle las tramas de datos correspondientes para que los nodos de la WSN realicen las operaciones deseadas. Por último, debería interpretar las tramas recibidas por el coordinador para visualizar los resultados.

El laboratorio remoto, además de permitir al usuario realizar el experimento con hardware real, facilita al usuario la realización del mismo, ya que el laboratorio remoto realiza todas las tareas mencionadas en el párrafo anterior. El usuario solo debe seleccionar los valores deseados para los parámetros e interpretar los resultados. Se cumple de este modo la premisa de permitir al usuario realizar un experimento de manera simple, usando una interface intuitiva, y sin exigir al usuario conocimientos avanzados de programación.

### 5.1.2. Potencia de RF recibida

La potencia de señal de RF recibida por un nodo depende de la potencia de RF con que transmite el nodo emisor y la atenuación en el medio en el cual la señal se propaga. La atenuación depende de muchos factores, entre ellos, la distancia entre los nodos, la presencia de objetos, la altura de los nodos respecto del plano de tierra, la posición de las antenas, etc. La mayoría de los nodos disponibles en el mercado ofrecen algún mecanismo para medir la potencia de la señal de RF recibida. Esto permite a los programadores ajustar la potencia de transmisión al valor necesario.

Los nodos XBee entregan el valor de RSSI (Indicador de fuerza de la señal recibida) que indica la potencia de RF recibida por un nodo en dBm. Este valor de RSSI corresponde al último paquete de datos recibido por el nodo (que puede ser un paquete de datos, una respuesta, etc.). Además, dos parámetros permiten ajustar la potencia transmitida y la sensibilidad del receptor. Por un lado, el comando PL (Power Level) permite ajustar la potencia de transmisión de un nodo permitiendo elegir entre 5 valores, como muestra la tabla 5.2. Por otro lado, el parámetro PM, cuando está en valor 1, incrementa la potencia de transmisión en 2 dB respecto a los valores mostrados en la tabla 5.2 y la sensibilidad del receptor en 1 db.

Con este experimento, el usuario puede observar cómo varía la potencia recibida por un nodo, en función de potencia transmitida por el nodo emisor y la distancia entre los nodos.

**Tabla 5.2:** Potencia transmitida por el nodo según el valor del parámetro PL

Valor de PL	Potencia de transmisión
0	-7 dBm
1	-3 dBm
2	-1 dBm
3	+1 dBm
4	+3 dBm

Para realizar este experimento el usuario debe emplear el comando DB. Este comando solicita a un nodo el valor del RSSI (Indicador de fuerza de la señal recibida) del último paquete de datos recibido por el nodo.

Una metodología sugerida para realizar este experimento, y que se utiliza en este trabajo, consiste en modificar la potencia de transmisión del nodo coordinador, y verificar el valor del RSSI que reciben los nodos finales, considerando que los mismos se encuentran a diferentes distancias del nodo coordinador, siendo el nodo 1 el más cercano, luego el nodo 2, y así sucesivamente.

La tabla 5.3 muestra los resultados de realizar este experimento con algunos nodos de la WSN que forman el laboratorio remoto. Los valores de RSSI están expresados en dBm. Puede verse como aumenta la potencia de RF recibida por el nodo a medida que se incrementa la potencia de transmisión o disminuye la distancia.

**Tabla 5.3:** RSSI recibido por los nodos para distintas potencias de transmisión del nodo coordinador (en dBm).

Parámetros del Nodo Coordinador		Nodo 1	Nodo 2	Nodo 6	Nodo 7
PM	PL				
00	00	-30 dBm	-43 dBm	-67 dBm	-89 dBm
01	00	-28 dBm	-37 dBm	-61 dBm	-85 dBm
00	01	-28 dBm	-34 dBm	-61 dBm	-84 dBm
01	01	-27 dBm	-31 dBm	-57 dBm	-76 dBm

La realización de este experimento requiere que los usuarios conozcan conceptos de configuración de WSN y transmisión de señales de RF, pero no necesitan desplegar una WSN, programar sus nodos, ni armar las tramas de datos, ya que esa tarea la realiza el laboratorio remoto.

### 5.1.3. Tiempo de respuesta en función de la ocupación del canal de comunicaciones

Los nodos XBee utilizados en este laboratorio remoto implementan su capa física y capa MAC de acuerdo al estándar IEEE 802.15.4 (IEEE, Institute of Electrical and Electronic Engineers, 2003), el cual emplea el protocolo CSMA/CA para acceder al canal de comunicación inalámbrico. De acuerdo a lo establecido en el estándar IEEE 802.15.4, cuando un nodo debe transmitir un dato, primero verifica tres veces si el canal de comunicación está desocupado, esperando intervalos de tiempo aleatorios entre cada

verificación. Si en las tres verificaciones se detecta que el canal de comunicaciones está desocupado, el nodo transmite el dato. Si en alguna verificación se detecta que el canal de comunicaciones está ocupado, el nodo espera un tiempo aleatorio y luego reinicia el proceso de verificación. Este mecanismo tiene la finalidad de disminuir al mínimo la probabilidad de colisiones.

Mientras mayor el número de nodos o la frecuencia con la que transmiten datos, mayor probabilidad de que un nodo encuentre el canal de comunicaciones ocupado y deba esperar para enviar un dato, y por consiguiente será mayor la probabilidad de que se incremente el tiempo de respuesta del nodo. Este experimento tiene como objetivo medir el tiempo de respuesta de un nodo a un comando que solicita alguna información, en función de la frecuencia de transmisión de los demás nodos de la red.

Para realizar el experimento, se programó el nodo 1 de manera que su tiempo de respuesta sea aproximadamente entre 1.5 y 2 segundos. Por otro lado, se configuraron otros 7 nodos, del 2 al 8, para que transmitan datos periódicamente. Se midió el tiempo de respuesta del nodo 1 a un comando para diferentes velocidades de transmisión periódica de los otros 7 nodos, para verificar si la mayor o menor ocupación del canal de comunicaciones afecta el tiempo de respuesta de un nodo.

Los nodos 2 al 8 se configuraron para que transmitan datos periódicamente cada 0.1 segundos, 0.5 segundos, 5 segundos, 50 segundos y nunca. Para cada una de estos periodos de transmisión de datos, se tomaron 30 mediciones del tiempo de respuesta del nodo 1 al comando DB, y se calcularon el valor promedio y la desviación estándar. La tabla 5.4 muestra los valores obtenidos de tiempo de respuesta promedio y desviación estándar para el nodo 1, para los diferentes periodos de transmisión de datos de los nodos 2 al 8. En la última fila, los nodos 2 al 8 se desconectaron de la red, por lo cual solo el nodo 1 ocupa el canal de comunicaciones.

**Tabla 5.4:** Tiempo de respuesta (en segundos) en función de la ocupación del canal de comunicaciones

Periodo de reporte de datos de los nodos 2 al 8	Tiempo de respuesta promedio del nodo 1	Desviación estándar
0.1 s	9.60 s	9.69 s
0.5 s	8.19 s	7.10 s
5 s	2.54 s	1.93 s
50 s	2.07 s	0.75 s
nunca	1.99 s	0.46 s

Puede verse como al disminuir el periodo de reportes de los demás nodos de la red, y por tanto aumentar la ocupación del canal de comunicaciones, aumenta el tiempo de respuesta promedio de un nodo a un comando. Puede verse que cuando los nodos 2 al 7 transmiten datos cada 100 milisegundos o cada 500 milisegundos, el tiempo de respuesta promedio y su desviación estándar aumentan considerablemente. Sin embargo, para tiempos de reporte de datos de los nodos 2 al 7 de 5 segundos para arriba, el tiempo de respuesta promedio del nodo 1 no varía considerablemente. Este resultado puede atribuirse a que mientras mayor sea la ocupación del canal de comunicaciones, mayor será la probabilidad de que un nodo deba esperar a que el canal de comunicaciones esté desocupado para poder transmitir un dato.

Debe mencionarse que para realizar este experimento no se tuvo en cuenta la interferencia que pueden ocasionar otras señales que transmiten en la banda de 2.4 GHz presentes en la zona donde se realizó el experimento, como ser señales de Wi-Fi.

#### **5.1.4. Formación de dos redes**

El protocolo ZigBee permite que en un mismo ambiente puedan coexistir varias WSN si se asigna a cada una un identificador de red (PAN-ID) distinto. Los nodos y coordinador de una misma red deben poseer el mismo PAN-ID para poder comunicarse.

El laboratorio remoto permite programar dos valores diferentes para el PAN-ID. El usuario puede modificar el PAN-ID de cualquiera de los nodos, incluido el coordinador, y verificar la conectividad. De este modo, el usuario podría verificar que solo los nodos que posean el mismo PAN-ID que el coordinador pueden comunicarse.

## **5.2. Experimentos de evaluación de desempeño**

### **5.2.1. Influencia de la latencia sobre los experimentos**

Introducir el laboratorio remoto entre el equipo o sistema bajo prueba y el usuario permite que el mismo pueda realizar los experimentos desde cualquier ubicación del mundo, sin tener que trasladarse al lugar donde está físicamente instalado el equipo bajo prueba. Pero también introduce una latencia que puede afectar los experimentos o la experiencia del usuario.

Disminuir esta latencia al mínimo puede requerir, entre otras medidas, mejorar la infraestructura de acceso a Internet donde se encuentre el laboratorio físico, como también contratar mejores servicios por parte del proveedor de servicios de Cloud Computing. Ambas medidas significan mayores costos. Esta sección analiza y mide esta latencia.

### **5.2.1.1. Influencia de la latencia sobre los resultados experimentales**

Los resultados producidos por un experimento pueden o no ser afectados por la latencia introducida por los módulos de gestión del laboratorio remoto, según la forma en que se implemente el laboratorio remoto y la forma en que se realicen los experimentos. Un ejemplo de un experimento en el cual la latencia puede afectar los resultados experimentales se presenta cuando el usuario necesita que un algoritmo interactúe con el sistema bajo prueba. Si el usuario ejecuta dicho algoritmo en su computadora, la latencia de los módulos de gestión del laboratorio remoto puede afectar los resultados del experimento.

Puede eliminarse la influencia de la latencia introducida por el laboratorio remoto sobre los resultados experimentales si se toman las medidas para excluirla. Entre estas medidas se encuentran (Maiti and Tripathy, 2013):

- Realizar todas las mediciones que dependan del tiempo en la computadora del laboratorio físico (no en el Cloud ni en la computadora del usuario remoto).
- Si el usuario necesita que un algoritmo interactúe con el sistema bajo prueba, permitir que sus algoritmos se ejecuten en las computadoras del laboratorio físico.

Si esto no fuera posible (por ejemplo, si el usuario no desea compartir su algoritmo), la latencia de los módulos de gestión pueden influir en la realización de los experimentos.

Si se toman estas medidas, puede lograrse que la latencia introducida por el laboratorio remoto no influya en los resultados experimentales. Para el laboratorio remoto propuesto en esta tesis, la latencia no afecta ninguno de los resultados de los posibles experimentos que pueden realizarse, ya que todas las mediciones se hacen del lado del laboratorio físico.

### **5.2.1.2. Influencia de la latencia en la experiencia del usuario**

Dado que el laboratorio remoto opera en tiempo real, de modo de lograr una alta interactividad del usuario con los equipos bajo prueba, la latencia puede afectar la experiencia del usuario, aunque no afecte los resultados experimentales. Si la latencia es grande, el usuario podrá percibirla, pudiendo resultar molesta, o pudiendo disminuir la cantidad de experimentos que el usuario puede realizar en el periodo de tiempo que se le asigna.

## 5.2.2. Medición de las latencias introducidas por los módulos de gestión del laboratorio remoto

### 5.2.2.1. Intervalos de tiempos y latencias involucradas

Debido a que un comando y su respuesta pasan a través de varios subsistemas, la latencia total puede dividirse en tres:

- Latencia introducida por la red del usuario (a través de la cual el usuario accede a Internet).
- Latencia introducida por los módulos de gestión del laboratorio remoto.
- Tiempo de respuesta de la WSN.

La figura 5.2 muestra estas latencias. Las mismas se midieron de manera indirecta, a partir de tres intervalos de tiempo:  $T_{WSN}$ ,  $T_{CLOUD}$ ,  $T_{TOTAL}$ , que también se muestran en la figura 5.2.

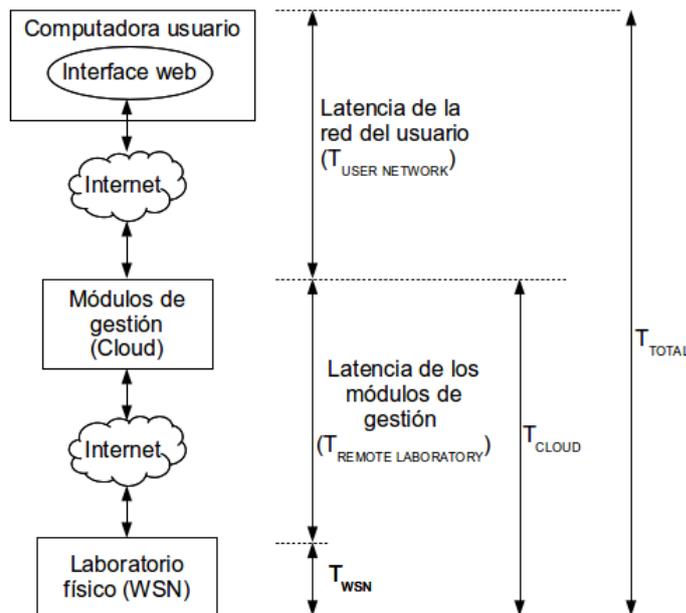


Figura 5.2: Latencias y tiempos de respuesta

**Latencia introducida por la computadora y la red del usuario ( $T_{USER NETWORK}$ )** Esta latencia depende de la computadora del usuario y la velocidad de la red a través de la cual accede a Internet. No puede considerarse como un parámetro característico del laboratorio remoto, ya que puede variar significativamente según la velocidad de la conexión a Internet de cada usuario. Por otro lado, esta latencia no puede controlarse desde el punto de vista del diseño del laboratorio remoto.

Se calculó esta latencia para poder calcular las demás latencias y tiempos de respuesta. Se empleó para ello la red de la Universidad en la cual se encuentra instalado el laboratorio remoto (sección 4.4.11) para acceder a Internet.

**Tiempo de respuesta de la red de sensores ( $T_{WSN}$ )** Este tiempo depende de la red de sensores (que es el sistema bajo prueba para este laboratorio remoto) y la computadora a través de la cual se accede a la red de sensores. Es el tiempo de respuesta o demora que el usuario observaría si hiciera el experimento presencialmente.

El tiempo  $T_{WSN}$  es una característica propia del sistema bajo prueba, no del laboratorio remoto. Se medirá su valor para poder calcular otros tiempos y para propósitos ilustrativos. Debido a que el usuario puede configurar el tiempo de respuesta de los nodos como parte de los experimentos, su valor dependerá de la configuración del experimento que el usuario realice. Por otro lado, debido al comportamiento propio de las WSN, este tiempo puede tener una alta variabilidad.

#### **Latencia introducida por los módulos de gestión del laboratorio remoto**

( $T_{REMOTE LABORATORY}$ ) Esta latencia depende del tiempo de procesamiento de los módulos de gestión del laboratorio remoto, de la velocidad de la red que conecta estos módulos con la computadora del laboratorio físico y de la velocidad de procesamiento de la computadora del laboratorio.

Esta latencia es un parámetro característico del laboratorio remoto y puede ser disminuida mediante un diseño adecuado. La latencia introducida por los módulos de gestión y la computadora del laboratorio depende de la implementación de estos componentes. La latencia introducida por la red a través de la cual el laboratorio físico se conecta a Internet depende de la infraestructura de la cual se disponga.

#### **5.2.2.2. Metodología experimental**

Para medir los intervalos de tiempo  $T_{WSN}$ ,  $T_{CLOUD}$ ,  $T_{TOTAL}$  se utilizaron rutinas de programación y los relojes internos de las computadoras o servicios de Cloud Computing que forman parte del sistema, como se menciona a continuación:

- Tiempo de respuesta total ( $T_{TOTAL}$ ). Intervalo de tiempo desde que el usuario envía un comando al laboratorio remoto desde la interface web, hasta que recibe su respuesta. Se midió utilizando el reloj interno de la computadora del usuario y una rutina, programada en lenguaje JavaScript, que permite capturar el valor de este reloj.

- Tiempo de respuesta desde el servidor ( $T_{CLOUD}$ ). Intervalo de tiempo desde que un comando llega al sistema de gestión desplegado en el Cloud, procedente de la interface web de un usuario remoto, hasta que la respuesta a este comando ha sido procesada por el equipo bajo prueba, devuelta a los módulos de gestión en el Cloud, y está lista para ser devuelta a la interface web del usuario. Se midió empleando un servicio de Cloud Computing, en este caso provisto por Google App Engine y programado en lenguaje Java, que permite determinar la hora de los servidores de Google.
- Tiempo de respuesta de la red de sensores ( $T_{WSN}$ ). Intervalo de tiempo desde que un comando se envía a la WSN desde la computadora del laboratorio, hasta que la respuesta de la WSN se recibe en dicha computadora. Se midió utilizando el reloj interno de la computadora del laboratorio, y una rutina, programada en lenguaje Java, que permite determinar la hora de la computadora en la cual se ejecuta la rutina.

Como estos tres intervalos de tiempos se miden de manera independiente, utilizando distintos relojes provistos por los distintos componentes que forman el laboratorio remoto, no es necesario que estos relojes estén sincronizados.

A partir de estos valores, se obtiene:

Tiempo de respuesta de la red de sensores:

$$T_{WSN} = T_{WSN} \text{ (medido directamente)}$$

Latencia introducida por los módulos de gestión del laboratorio remoto:

$$T_{REMOTE LABORATORY} = T_{CLOUD} - T_{WSN}$$

Latencia introducida por la computadora y la red del usuario:

$$T_{USER NETWORK} = T_{TOTAL} - T_{CLOUD}$$

### 5.2.2.3. Resultados experimentales

Se realizaron 90 mediciones en diferentes momentos de tiempo en tres días diferentes de la semana, obteniéndose los valores promedios mostrados en la Tabla 5.5.

**Tabla 5.5:** Tiempos de respuesta y latencias en milisegundos

Parámetro	Latencia promedio	Desviación estándar
Tiempo de respuesta de la WSN	2091 ms	1356 ms
Latencia introducida por los módulos de gestión del laboratorio remoto	1081 ms	343 ms
Latencia introducida por la computadora y la red del usuario	500 ms	49 ms

La latencia introducida por la red del usuario no puede considerarse como un parámetro característico del laboratorio remoto, ya que puede variar significativamente según la velocidad de la conexión a Internet de cada usuario.

El tiempo de respuesta de la WSN es propio del sistema bajo prueba. Puede verse en la tabla 5.5 que posee una alta variabilidad, lo cual es propio del funcionamiento de las WSN. Su valor dependerá de la configuración que el usuario de a los nodos.

El parámetro característico del laboratorio remoto es la latencia introducida por sus módulos de gestión. Su valor promedio es de 1081 milisegundos. Sin embargo, este valor difiere significativamente según el día y la hora en el cual se toma la medición. En la tabla 5.6 se presenta el valor obtenido para la latencia introducida por los módulos de gestión del laboratorio remoto para tres días diferentes.

**Tabla 5.6:** Latencia introducida por los módulos de gestión del laboratorio remoto para diferentes días en milisegundos

Día y hora	Latencia	Desviación estándar
Martes, 9:00 horas	1497 ms	241 ms
Viernes, 19:00 horas	890 ms	167 ms
Domingo, 10:00 horas	856 ms	69 ms

La tabla 5.6 muestra que la latencia introducida por los módulos de gestión del laboratorio remoto depende significativamente del día y la hora a la cual se mida. Esta diferencia puede atribuirse a las diferentes velocidades de la conexión entre el laboratorio físico y los servidores del proveedor de servicios de Cloud Computing (Google App Engine), según el día y la hora. Dicha conexión es a través de Internet, empleando la red de la Universidad. Puede verse que los momentos de mayor latencia se corresponden a los momentos de mayor concurrencia de personal en la Universidad.

### **5.2.3. Latencias introducidas por implementar los módulos de gestión en el Cloud respecto a un laboratorio remoto tradicional**

Implementar los componentes del sistema de gestión en el Cloud, como muestra la figura 4.3, puede producir un incremento en la latencia introducida por el laboratorio remoto, respecto a un laboratorio remoto tradicional, como el mostrado en la figura 4.2. En esta sección se mide este incremento en la latencia total

#### **5.2.3.1. Metodología experimental**

Para conocer el incremento en la latencia al implementar los módulos de gestión del laboratorio remoto en el Cloud, se midió la latencia producida por los módulos de gestión del laboratorio remoto para dos despliegues, para luego realizar su comparación:

- Despliegue tradicional: consiste en desplegar los módulos de gestión del laboratorio remoto en la computadora del laboratorio físico, es decir, sin emplear servicios de Cloud Computing. Esta es la misma computadora a la cual se conecta la WSN. El modelo de capas de este laboratorio remoto se presentó en la figura 4.2.
- Despliegue en el Cloud: los módulos de gestión desplegados en el Cloud, comunicados con el laboratorio físico a través de Internet. Este es el despliegue final del laboratorio remoto presentado en esta tesis, y operativo en la actualidad. El modelo de capas de este laboratorio remoto se presentó en la figura 4.3.

Para ambos despliegues se utilizan los mismos módulos de gestión, pero desplegados en diferentes sistemas: una PC local del laboratorio para el despliegue tradicional y el Cloud. Para el laboratorio remoto tradicional, se empleó un entorno de desarrollo y testeo provisto por GAE, que permite testear los componentes de software localmente, antes de ser desplegados en el Cloud. Este entorno de desarrollo se provee como una extensión de la plataforma de desarrollo de aplicaciones Eclipse. Para el despliegue en el Cloud, se emplearon los servicios de Cloud Computing provistos por GAE.

La latencia de la red del usuario se excluyó de las mediciones y del cálculo. Por lo que, si bien es necesario ejecutar la interface del usuario para realizar el experimento, la latencia introducida por la red del usuario no influye en el experimento.

La latencia introducida por los módulos de gestión del laboratorio remoto fue medida indirectamente como se presentó en la sección 5.2.2. Esta latencia se midió para ambos despliegues, sin Cloud y con Cloud.

### 5.2.3.2. Resultados experimentales

Para determinar estas latencias, se realizaron 60 mediciones para ambos despliegues en diferentes días, y se calcularon los valores promedios y desviaciones estándar. El experimento se realizó enviando diferentes comandos al nodo 1, y midiendo las latencias mencionadas anteriormente.

La tabla 5.7 muestra las latencias obtenidas para ambos despliegues, y la diferencia entre éstas.

**Tabla 5.7:** Latencia de los módulos de gestión con y sin Cloud en milisegundos

	Valor promedio	Desviación estándar
Latencia de los módulos de gestión para el laboratorio remoto tradicional	505 ms	94 ms
Latencia de los módulos de gestión para el laboratorio remoto desplegado en el Cloud	1081 ms	343 ms
Incremento en la latencia	576 ms	

Puede verse que desplegar los módulos de gestión del laboratorio remoto en el Cloud incrementa la latencia, en promedio, en aproximadamente 576 milisegundos, respecto a un laboratorio remoto tradicional. Este incremento en la latencia puede atribuirse a la conexión a Internet que hay de por medio cuando los módulos de gestión se despliegan en el Cloud, y al mecanismo cliente-servidor empleado.

## 5.3. Conclusiones

Se realizaron experimentos de prueba de concepto del laboratorio remoto. La realización de cada uno de estos experimentos requiere un tiempo de entre media hora o una hora. Estos experimentos permitieron demostrar que el usuario puede, muy rápidamente, poner a prueba conceptos relacionados con las redes de sensores, sin tener que desplegar una WSN real, ni desarrollar programas que procesen tramas de datos o puertos seriales. Sin embargo, el usuario realiza los experimentos con hardware real, lo que le permite comprobar comportamientos difíciles de modelar en un simulador.

Se realizaron experimentos de desempeño del laboratorio remoto implementado. Primero se midieron las latencias producidas por los diferentes módulos del laboratorio remoto ya desplegado empleando los servicios de GAE. Luego se midió el incremento de la latencia debido al despliegue en el Cloud respecto a la latencia cuando se despliegan los módulos de gestión del laboratorio remoto en la computadora del laboratorio físico.

Los tiempos medidos poseen una baja precisión, ya que las mediciones se realizaron empleando los relojes de diferentes computadoras, y estos son de baja exactitud. Sin embargo, las demoras propias de una WSN son del orden de los segundos y poseen alta variabilidad, por lo que los errores introducidos por estos relojes no son significativos. Por otro lado, las latencias introducidas no afectan los resultados de los experimentos que los usuarios pueden realizar, sino la experiencia del usuario, por lo que no se requiere conocer los valores de las latencias con elevada exactitud.

Puede observarse en la tabla 5.5 que la latencia introducida por los módulos de gestión del laboratorio remoto es del mismo orden que las latencias introducidas por la WSN y la red del usuario. El tiempo de respuesta de la WSN no puede disminuirse porque este es el sistema bajo prueba. La latencia de la red del usuario tampoco puede disminuirse a través del diseño del laboratorio remoto, ya que depende de la red a través de la cual accede el usuario, y no se tiene control sobre la misma. Por lo tanto, disminuir la latencia introducida por los módulos de gestión del laboratorio remoto no disminuirá significativamente la latencia total del laboratorio remoto. En la tabla 5.7 puede verse que desplegar los módulos de gestión del laboratorio remoto en el Cloud incrementa la latencia alrededor de 576 milisegundos respecto a un laboratorio remoto tradicional. Este tiempo puede afectar la experiencia del usuario, pero no imposibilita una interacción en tiempo real.



## Conclusiones y Trabajos Futuros

Este capítulo cierra esta tesis, listando las contribuciones de la misma, líneas de trabajo futuro generadas a partir del trabajo realizado, y por último presenta las observaciones finales.

### **6.1. Contribuciones**

Esta sección menciona y resume las contribuciones de esta tesis.

#### **6.1.1. Una arquitectura para laboratorios remotos de WSN orientados a experimentación y generación de casos de estudio**

Dicha arquitectura se presenta en el modelo de capas presentado en la figura 4.2. Como componentes novedosos para un laboratorio remoto de WSN se incluyen módulos para seguimiento de la actividad realizada por los usuarios y una interface web didáctica, intuitiva y de fácil uso.

#### **6.1.2. Una propuesta para el despliegue de laboratorios remotos empleando servicios de Cloud Computing**

Se propone un nuevo modelo de implementación de laboratorios remotos de WSN no documentado en la bibliografía, que consiste en el despliegue de varios componentes propios de un laboratorio remoto empleando servicios de Cloud Computing, e integrando el laboratorio físico o sistema bajo prueba al Cloud mediante un sistema de interconexión basado en una aplicación cliente servidor. Este modelo se presenta en la figura 4.3, y el sistema de interconexión en la figura 4.7.

Este modelo permite dotar a los laboratorios remotos de las características ventajosas de los sistemas de Cloud Computing, como almacenamiento seguro y confiable de datos, alta disponibilidad y robustez. Además, los requisitos computacionales necesarios para implementar los módulos de gestión del laboratorio remoto se trasladan al Cloud, permitiendo disminuir la complejidad de la computadora del laboratorio físico.

La integración del laboratorio remoto a Cloud Computing resulta ventajosa para uno de los trabajos futuros a realizar: implementar un sistema híbrido de laboratorio remoto con modelos simulados, como se presenta en la sección 6.3.5. Los modelos simulados pueden necesitar grandes capacidades de cómputo, las cuales pueden obtenerse mediante el empleo de servicios de Cloud Computing (ver capítulo 4).

### **6.1.3. Una propuesta para integrar dispositivos electrónicos a Cloud Computing**

Para la interconexión de los equipos del laboratorio físico con el Cloud, se propone un mecanismo basado en una aplicación cliente servidor, ejecutando la aplicación cliente localmente, en el laboratorio físico en el cual se encuentra el equipo bajo prueba, y la aplicación servidor en el Cloud.

Este mecanismo puede ser útil para la interconexión de cualquier dispositivo electrónico con aplicaciones que se ejecutan en el Cloud. Para ello es necesario que el dispositivo electrónico, o una computadora a la cual esté conectado, pueda ejecutar la aplicación cliente del sistema de interconexión.

### **6.1.4. Construcción de una interface web didáctica**

Esta interface permite a los usuarios programar la WSN de manera fácil y didáctica, de modo que los usuarios no necesiten conocimientos avanzados de programación ni WSN.

La interface web lista una serie de parámetros y posibles valores para cada uno de los mismos. El usuario programa los nodos seleccionando valores para estos parámetros.

### **6.1.5. Una nueva clasificación de los laboratorios remotos**

Se ha esbozado en la literatura la existencia de laboratorios remotos destinados a tareas de investigación científica, desarrollo o educación, pero ningún autor ha propuesto una clasificación formal ni ha realizado una comparación exhaustiva. Por este motivo, se propone en esta tesis una clasificación de los laboratorios remotos, distinguiendo entre laboratorios remotos destinados a investigación científica y destinados a educación y desarrollo. Esta clasificación se presenta en la sección 2.3.2.

En términos constructivos, estos dos tipos de laboratorios remotos requieren diferentes formas de implementar las interfaces web, almacenar diferentes tipos de datos y diferentes maneras para realizar los experimentos, como se analizó en la sección 2.3.2. Desde el punto de vista del usuario, para un usuario sin conocimientos avanzados de programación o WSN, será difícil acceder a laboratorios remotos que no posean experimentos predefinidos o interfaces web didácticas e intuitivas, que son características de los laboratorios remotos de WSN destinados a educación. Por otro lado, el usuario que necesite realizar experimentos más complejos, como experimentos científicos, tendrá otras necesidades, como mayor posibilidad de control sobre el equipo bajo prueba, la cual es una característica de los laboratorios remotos destinados a investigación científica.

#### **6.1.6. Construcción de un prototipo que emplea nodos disponibles en el mercado local**

Existe una gran variedad de nodos sensores comerciales disponibles en el mercado. Los laboratorios remotos de WSN presentados en la bibliografía típicamente emplean nodos de la familia TelosB o MicaZ (Crossbow Technology, Inc., 2010), que pueden ser programados completamente. Sin embargo, estos nodos no están disponibles en el mercado local, ni son los más adecuados para usuarios que aún no poseen los conocimientos de programación necesarios para programar este tipo de nodos. Por ello, el prototipo presentado en esta tesis emplea nodos de fácil programación y disponibles en el mercado local.

## **6.2. Limitaciones**

### **6.2.1. Reducido número de nodos**

El número de nodos sensores es reducido en comparación con los laboratorios remotos montados en universidades de Estados Unidos o Europa, ya que algunos de estos laboratorios remotos cuentan con cientos de nodos. Por ejemplo, el laboratorio remoto de WSN *FIT IoT-Lab* (FIT consortium, 2014), el cual fue montado con una subvención inicial de 5.8 millones de euros, cuenta con 2700 nodos. El laboratorio remoto presentado en esta tesis no pretende destacarse por su número de nodos ni recursos tecnológicos, sino por su arquitectura y modelo de capas propuesto, que incluye el despliegue en el Cloud, y su diseño pensado para usuarios que deseen formarse, entrenarse o experimentar en programación y funcionamiento de las WSN.

### **6.2.2. Latencia**

De acuerdo a los experimentos realizados en la sección 5.2.2, el laboratorio remoto desplegado introduce una latencia que está entre 1 y 1.5 segundos aproximadamente. Esta latencia depende de la conexión a Internet que posee el laboratorio físico, y varía según el día y la hora a la cual se accede.

Sin embargo, esta latencia no afecta los resultados producidos por los experimentos, solo tiene efecto sobre la experiencia del usuario.

### **6.2.3. Necesidad de probar la adaptabilidad a otro tipo de equipo bajo prueba**

El modelo planteado para laboratorios remotos ha sido pensado y puesto a prueba para laboratorios remotos de WSN, ya que éste es el objetivo principal de la tesis. No ha sido puesto a prueba para otro tipo de equipo, por lo que es necesario un análisis sobre su adaptabilidad a otro tipo de equipo bajo prueba.

### **6.2.4. Necesidad de estudiar su aplicación a docencia desde el punto de vista pedagógico**

El laboratorio remoto ha sido dotado de componentes pensados para permitir su aplicación a tareas de docencia, como un módulo de seguimiento de usuarios, que permita a los docentes seguir la actividad realizada por los estudiantes, y una interface web didáctica e intuitiva, que permita a los estudiantes realizar experimentos de manera rápida, sin tener que montar completamente un laboratorio de WSN. Sin embargo, no se ha realizado un estudio de los aspectos pedagógicos que permitan la aplicación del laboratorio remoto a tareas de docencia. Por este motivo no se declara en esta tesis que el laboratorio remoto propuesto e implementado sea apto para tareas de docencia.

## **6.3. Trabajos futuros**

Los trabajos futuros se dividen en dos: posibles mejoras al laboratorio remoto desplegado y profundizar la investigación en cuanto a la integración de laboratorios remotos al Cloud.

### **6.3.1. Posibles mejoras**

Entre los trabajos futuros se incluyen algunas ideas para mejorar o agregar funcionalidades al laboratorio remoto, que han surgido durante el proceso de implementación del

modelo propuesto, durante la realización de experimentos, y la lectura de recientes publicaciones (Orduna et al., 2015). Estas mejoras posibles se describen a continuación.

#### **6.3.1.1. Estudiar formas para disminuir la latencia introducida por el laboratorio remoto**

Como se describió en la sección 5.2.2, la latencia introducida por el laboratorio remoto depende fundamentalmente de la velocidad del enlace entre el equipo bajo prueba y los componentes de gestión en el Cloud. Disminuir esta latencia es difícil porque depende de la infraestructura disponible en el laboratorio donde está montado el laboratorio remoto. Sin embargo, se propone estudiar si otras opciones para interconectar el laboratorio físico con el Cloud pueden ayudar a disminuir esta latencia.

#### **6.3.1.2. Mejorar el sistema de seguimiento de usuarios**

El sistema de seguimiento lista todos los usuarios que han accedido al laboratorio remoto y las acciones que han realizado (reserva de turnos y envío de comandos). Sin embargo, se propone estudiar posibles mejoras e implementarlas, como por ejemplo: mejorar el diseño de las diferentes páginas web que conforman la interface web de acuerdo a las opiniones y sugerencias de los usuarios, y posibilidad de que los supervisores carguen guías de trabajos prácticos o exámenes para realizar on-line.

#### **6.3.1.3. Incrementar el número de nodos y el tamaño del despliegue**

Incrementar el número de nodos y el tamaño del despliegue, agregando otros nodos coordinadores conectados a la computadora del laboratorio, de modo que el usuario pueda configurar varias WSN. Esta mejora está sujeta a la disponibilidad de dichos nodos.

#### **6.3.1.4. Estudiar la integración a plataformas de aprendizaje**

Debido a que el laboratorio remoto ha sido pensado en parte como una herramienta para usuarios que deseen formarse, entrenarse o experimentar en programación y funcionamiento de las WSN, puede llegar a ser una ventaja su integración con plataformas de aprendizaje.

Implementar esta integración implicará realizar un estudio del estado del arte sobre plataformas de aprendizaje y sobre la conveniencia de realizar esta integración, y en ese caso, proponer mecanismos para realizar la integración.

#### **6.3.1.5. Más opciones para la realización de los experimentos**

El laboratorio remoto implementado permite a los usuarios configurar y realizar experimentos de manera rápida a través de una interface web que permite configurar los nodos seleccionando los parámetros a configurar y los valores de los mismos, seleccionando desde los valores predefinidos, o enviando los comandos y valores que el usuario desee.

Sin embargo se estudiarán otras opciones para permitir a los usuarios realizar experimentos más fácil y rápidamente, o bien para que puedan tener acceso a características de más bajo nivel de los nodos utilizados.

#### **6.3.2. Profundizar la investigación en cuanto a la integración de laboratorios remotos al Cloud**

Del estudio del estado del arte y los trabajos relacionados, pudo observarse que la integración de laboratorios remotos con tecnologías de Cloud Computing es un campo con pocas y muy recientes publicaciones. Sin embargo, como pudo verificarse en esta tesis, el paradigma de Cloud Computing puede brindar características beneficiosas a los laboratorios remotos. Por otro lado, también se llegó a la conclusión de que en este campo hay varios problemas por resolver, como proponer y analizar distintas arquitecturas para la interconexión de los equipos bajo prueba con los componentes del laboratorio remoto implementados en el Cloud, disminuir las latencias introducidas, etc. En este trabajo se propuso un mecanismo para lograr la interconexión, pero pueden proponerse y estudiarse otros mecanismos y otros esquemas de interconexión.

#### **6.3.3. Estudio de los aspectos pedagógicos para su aplicación a tareas de docencia**

Como se mencionó anteriormente en las limitaciones, el laboratorio remoto ha sido diseñado de manera que una de sus aplicaciones futuras sea en tareas de docencia. Pero previo a emplear y presentar el laboratorio remoto como apto para esta aplicación, es necesario realizar un estudio de su aplicabilidad a tareas de docencia desde el punto de vista pedagógico.

Se propone como trabajo futuro realizar este análisis, partiendo de un estudio del estado del arte sobre condiciones a cumplir y métodos para evaluar la aptitud de un laboratorio remoto, o herramienta tecnológica en general, para ser aplicado a docencia. Dicho estudio del estado del arte permitirá conocer las mejoras a implementar y los experimentos a realizar para poder aplicar el laboratorio remoto a tareas de docencia.

#### **6.3.4. Aplicación del laboratorio remoto a tareas de docencia**

Una vez realizado el trabajo futuro mencionado en el punto 6.3.3, se espera que el laboratorio remoto implementado pueda emplearse como una herramienta de enseñanza en la carrera de licenciatura en ciencias de la computación de la Universidad Nacional de Cuyo, la cual cuenta con una materia relacionada con las WSN. Además, luego de estudiar la aplicación del laboratorio remoto a otro tipo de equipos bajo prueba, se espera que el laboratorio remoto pueda ser empleado para su aplicación en enseñanza y docencia de otro tipo de equipo bajo prueba.

#### **6.3.5. Implementación de un modelo simulado para interactuar con el laboratorio remoto**

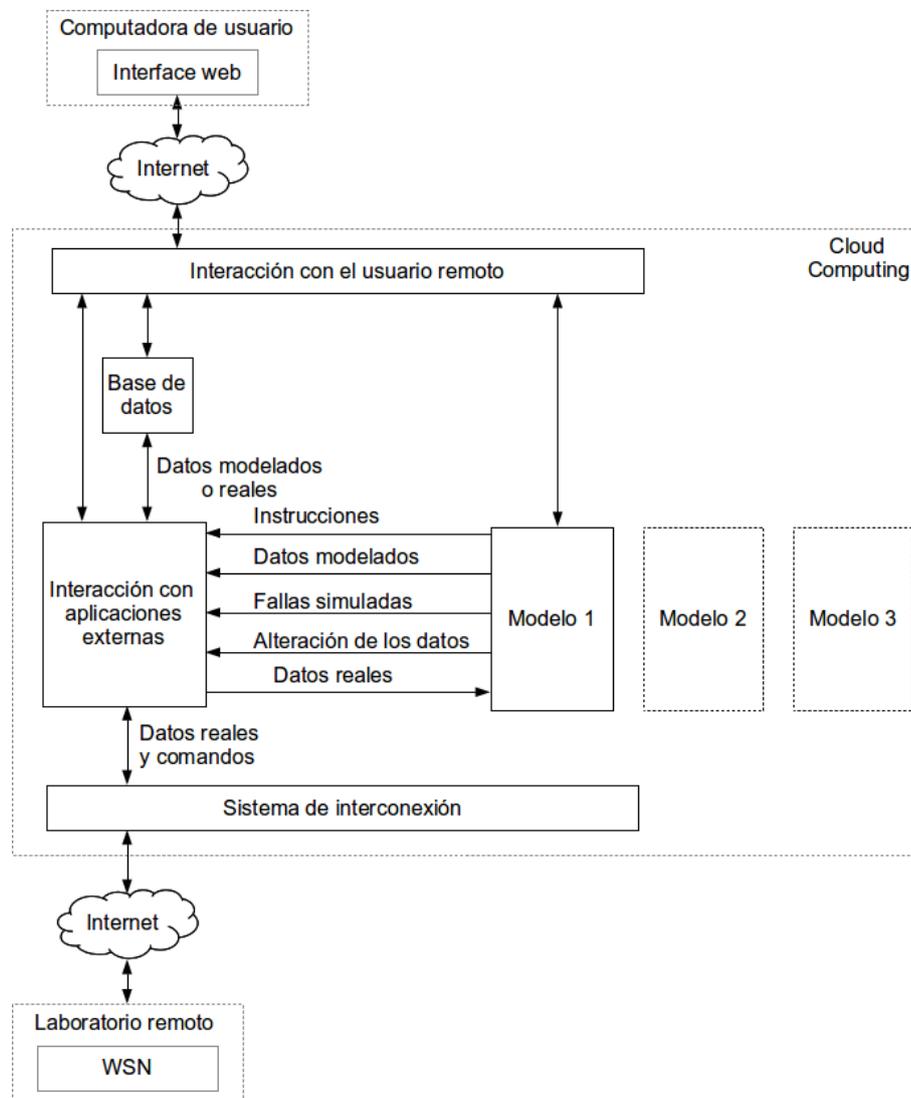
Las WSN tienen un gran número de aplicaciones, algunas de las cuales fueron descritas en la sección 2.1.2. Cada aplicación tiene características de funcionamiento diferentes en cuanto a aspectos como: tipos de datos, frecuencias de muestreo de datos, requisitos respecto a la forma de sensar datos, tipos de posibles fallas, tiempos de procesamiento de datos, acciones a tomar según los datos sensados, etc. Esto hace imposible que un laboratorio remoto de WSN pueda adaptarse a todas las aplicaciones posibles. Por otro lado, de acuerdo al estudio del estado del arte realizado en los capítulos 2 y 3, puede afirmarse que casi todos los laboratorios remotos de WSN presentados en la bibliografía están diseñados para realizar investigación sobre la programación y desarrollo de software para las WSN, y no sobre el comportamiento de las WSN en distintas aplicaciones.

Se propone implementar diferentes tipos de modelos que interactúen con el laboratorio remoto a través de aplicaciones creadas para tal fin. Con este propósito fue incorporado en el modelo de capas del laboratorio remoto el módulo de interacción con aplicaciones externas (sección 4.3.4.1). Este módulo permite la interacción con aplicaciones que pueden realizar diferentes acciones en función del comportamiento de modelos simulados. Estos modelos estarán implementados en software y serán ejecutados en el Cloud, para aprovechar una de las características constructivas más importantes del laboratorio remoto, el uso de servicios de Cloud Computing, que provee grandes capacidades de cómputo y almacenamiento de datos. Ejemplos de posibles modelos pueden ser:

- Aplicaciones de sensado en cultivos: donde los datos son recolectados con baja frecuencia (cada 30 minutos por ejemplo), corresponden a variables ambientales, y los sensores están expuestos a condiciones climáticas adversas.

- Domótica: los nodos transmiten datos en base a detección eventos o periódicamente, y la velocidad de respuesta puede ser importante (detección de eventos peligrosos, etc).
- Aplicaciones industriales: Alto nivel de ruido e interferencias que pueden ocasionar pérdida de datos.

La figura 6.1 muestra en detalle la interacción de este módulo con los demás módulos del laboratorio remoto y diferentes modelos.



**Figura 6.1:** Modelo de integración de laboratorio remoto con modelos simulados

## 6.4. Observaciones finales

En esta tesis se han estudiado los laboratorios remotos de WSN y su aplicación en formación y entrenamiento de usuarios en el uso de las WSN. El trabajo comenzó estudiando los laboratorios remotos de WSN. Pudo verse que los mismos han sido contruidos con fines de investigación científica y desarrollo de software para las WSN, brindando una gran cantidad de herramientas para este propósito. Sin embargo, se llegó a la conclusión de que su uso por parte de usuarios que desean formarse o entrenarse en el uso de las WSN es difícil, sobre todo pensando en usuarios con pocos conocimientos de programación y WSN.

A fin de elaborar propuestas para la implementación de laboratorios remotos de WSN orientados a usuarios que deseen formarse en programación y funcionamiento de WSN, se realizó un estudio de antecedentes y trabajos relacionados, que incluyó el estudio de otro tipo de laboratorios remotos destinados a educación, los módulos y componentes que se usan para construir los laboratorios remotos de WSN actuales, y cuáles módulos deben ser modificados para poder aplicar los laboratorios remotos de WSN a formación y entrenamiento de usuarios. Se incluyó el estudio de laboratorios remotos de redes de computadoras y dispositivos programables, por su similitud con un laboratorio remoto de WSN. Se estudiaron los requisitos y modificaciones que deben realizarse a los módulos de los laboratorios remotos de WSN actuales. Se llegó a la conclusión de que varios de estos requisitos pueden cumplirse mediante el empleo de tecnologías de Cloud Computing.

Los análisis realizados en los capítulos 2 y 3 llevaron a proponer una hipótesis para la implementación de laboratorios remotos de WSN destinados a usuarios que deseen formarse, entrenarse o experimentar en programación y funcionamiento de las WSN, o en desarrollo de aplicaciones que utilicen WSN, mediante el adecuado diseño e implementación de sus módulos de gestión. Se propuso un modelo para el despliegue de laboratorios remotos que cumplan con esta hipótesis empleando servicios de Cloud Computing, y se implementó un prototipo del mismo para la realización de experimentos.

Se dejan planteadas varias posibles mejoras a la arquitectura propuesta y al prototipo implementado, y se deriva de esta tesis una línea de investigación, la integración de laboratorios remotos a Cloud Computing, especialmente el diseño de sistemas de interconexión.



## Publicaciones Realizadas

### A.1. Artículos publicados

Los siguientes artículos, publicados en diferentes congresos y revistas, han sido realizados como parte del trabajo de investigación llevado adelante durante la realización de esta tesis.

1. Iacono, L., Godoy, P., Marianetti, O., and García Garino, C. Estudio de plataformas de hardware empleadas en redes de sensores inalámbricas. Congreso Argentino de Ciencias de la Computación (CACIC) 2010, V Workshop Arquitectura, Redes y Sistemas Operativos (WARSO). Universidad de Morón. ISBN: 978-950-9474-49-9.
2. Godoy, P., Iacono, L., Marianetti, O., and García Garino, C. Aplicaciones de redes de sensores inalámbricos al sensado de variables ambientales y agrícolas - estudio de los efectos de la potencia de transmisión en la vida útil y en la confiabilidad del enlace. Segundo Congreso Virtual de Microcontroladores y sus aplicaciones. Universidad Tecnológica Nacional - Facultad Regional Paraná. ISBN: 978- 987-25855-5-6.
3. Iacono, L., Godoy, P., Cayssials, R., Marianetti, O., and García Garino, C. Programmable logic devices in sensor networks: A survey. 40 Jornadas Argentinas de Informática e Investigación Operativa (JAIIO) 2011, 12 Argentine Symposium on Technology (AST). Sociedad Argentina de Informática e Investigación Operativa (SA-DIO). ISSN: 1850-2806.
4. Iacono, L., Godoy, P., Marianetti, O., García Garino, C., and Párraga, C. Estudio de la integración entre WSN y redes TCP/IP. Memoria de Trabajos de Difusión Científica y Técnica 10. ISSN 1510-7450.

5. Godoy, P., Iacono, L., Cayssials, R., and García Garino, C. A survey of WSN testbeds deployment. Congreso Argentino de Sistemas Embebidos (CASE) 2012. Universidad de Buenos Aires, Facultad de Ingeniería. ISBN: 978-987-9374-82-5.
6. Godoy, P., Iacono, L., Cayssials, R., and García Garino, C. Simulación de redes de sensores inalámbricos mediante autómatas celulares. Congreso Argentino de Ciencias de la Computación (CACIC) 2012, VII Workshop Arquitectura, Redes y Sistemas Operativos (WARSO). Universidad Nacional del Sur. ISBN: 978-987-1648-34-4.
7. Godoy, P., Iacono, L., Cayssials, R., Párraga, C., and García Garino, C. Effect of working conditions over the performance in ZigBee WSN. IEEE ArgenCon 2012, Capítulo Sistemas de Control. IEEE Argentina y Facultad de Ciencias Exáctas, Física y Naturales de la Universidad Nacional de Córdoba. ISBN: 987-572-076-3.
8. Godoy, P., Cayssials, R., García Garino, C. Zigbee WSN Round Trip Latency in Function of Channel Occupation and Nodes Configuration. Aceptado para ser publicado en IEEE ArgenCon 2016.
9. Godoy, P., Cayssials, R., García Garino, C. A Cloud Based WSN Remote Laboratory for User Training. Aceptado para ser publicado en Tecnología en Educación y Educación en Tecnología (TE&ET) 2016.

## **A.2. Presentaciones en Conferencias**

Los siguientes trabajos fueron presentados en los congresos mencionados. El primero como un tutorial, y el segundo como un poster.

1. Godoy, P. Hardware para redes de sensores inalámbricos (WSN). Situación actual, alternativas y futuro. Sesión Tutoriales, Simposio Argentino de Sistemas Embebidos (SASE) 2011. Buenos Aires, Argentina.
2. Pra Baldi, A., Iacono, L., Godoy, P., Marianetti, O., and García Garino, C. Utilización de herramientas Freescale de desarrollo de aplicaciones Zigbee sobre entornos GNU/Linux en WSN. Congreso EnIDI 2011. Universidad Tecnológica Nacional - Facultad Regional Mendoza. ISBN: 978-950-42-0134-2.

### **A.3. Artículos en estado de evaluación**

Los siguientes artículos han sido escritos y enviados a evaluar a revistas científicas. Los tres artículos se encuentran, al momento de terminar la redacción de esta tesis, en estado de evaluación.

- P. Godoy, R. Cayssials, C. García Garino. A WSN Testbed for Teaching Purposes. Enviado para ser evaluado en la revista IEEE Latin America Transaction.
- P. Godoy, R. Cayssials, C. García Garino. Implementation of the Management Modules of a Remote Laboratory in the Cloud. Enviado para ser evaluado en la revista Computing and Informatics.

### **A.4. Artículos en escritura**

Los siguientes artículos, al momento de entregar esta tesis para su evaluación, se encuentran en etapa de escritura, y serán presentados para su evaluación en congresos o revistas adecuadas durante Junio o Julio de 2016. Los resultados publicados en estos artículos han sido obtenidos mediante el empleo del laboratorio remoto implementado y durante el trabajo de estudio del estado del arte.

- P. Godoy, R. Cayssials, C. García Garino. Actual transmission channel occupation in function of calculated transmission channel occupation.
- A Survey of Different Approaches Used for Building WSN Testbeds.



# Bibliografía

- Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422.
- Alemdar, H. and Ersoy, C. (2010). Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688 – 2710.
- Amazon Web Services, I. (2012). Amazon elastic compute cloud (amazon ec2) website (<http://aws.amazon.com/es/ec2>).
- Baronti, P., Pillai, P., Chook, V. W., Chessa, S., Gotta, A., and Hu, Y. F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, 30(7):1655 – 1695.
- Bekmezci, I. and Alagöz, F. (2009). Energy efficient, delay sensitive, fault tolerant wireless sensor network for military monitoring. *International Journal of Distributed Sensor Networks*, 5(6):729–747.
- Bouckaert, S., Vandenberghe, W., Jooris, B., Moerman, I., and Demeester, P. (2010). The w-ilab. t testbed. In *TridentCom 2010: proceedings CD of the 6th International conference on Testbeds and Research Infrastructures for the Development of Networks and Communities*.
- Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599 – 616.
- Crossbow Technology, Inc. (2010). *MicaZ hoja de datos*.
- Crossbow Technology Inc. (2015). *Imote2, High-performance Wireless Sensor Network Node hoja de datos*. [http://wsn.cse.wustl.edu/images/e/e3/Imote2\\_Datasheet.pdf](http://wsn.cse.wustl.edu/images/e/e3/Imote2_Datasheet.pdf).
- Digi International Inc. (2008). XBee ZB low power ZigBee module with integrated wire antenna hoja de datos. Accedido en 2015.

- Dimitriou, T., Kolokouris, J., and Zarokostas, N. (2007). Sensenet: a wireless sensor network testbed. In *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, MSWiM '07, pages 143–150, New York, NY, USA. ACM.
- Drago, I., Mellia, M., M. Munafo, M., Sperotto, A., Sadre, R., and Pras, A. (2012). Inside Dropbox: Understanding personal Cloud storage services. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 481–494, New York, NY, USA. ACM.
- Durisic, M., Tafa, Z., Dimic, G., and Milutinovic, V. (2012). A survey of military applications of wireless sensor networks. In *Embedded Computing (MECO), 2012 Mediterranean Conference on*, pages 196–199.
- El-Darymli, K. and Ahmed, M. (2012). *Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management*, chapter Wireless Sensor Network Testbeds, pages 148–205. IGI Global.
- Ertin, E., Arora, A., Ramnath, R., Nesterenko, M., Naik, V., Bapat, S., Kulathumani, V., Sridharan, M., Zhang, H., and Cao, H. (2006). Kansei: a testbed for sensing at scale. In *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, pages 399–406.
- FIT consortium (Accedido en 2014). Iot-lab website, accedida en agosto de 2014 (<https://www.iot-lab.info/>).
- Gholamzadeh, B. and Nabovati, H. (2008). Concepts for designing low power wireless sensor network. *World Academy of Science, Engineering and Technology*, 45:559–565.
- Gilbert, E. (2012). Research issues in wireless sensor network applications: a survey. *International Journal of Information and Electronics Engineering*, 2(5):702–706.
- Girod, L., Elson, J., Cerpa, A., Stathopoulos, T., Ramanathan, N., and Estrin, D. (2004). EmStar: a software environment for developing and deploying wireless sensor networks. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, ATEC '04, pages 24–24, Berkeley, CA, USA. USENIX Association.
- Girod, L., Ramanathan, N., Elson, J., Stathopoulos, T., Lukac, M., and Estrin, D. (2007). EmStar: A software environment for developing and deploying heterogeneous sensor-actuator networks. *ACM Transactions on Sensor Networks (TOSN)*, 3.
- GNU Radio project (2006). GNU Radio website, versión 3.7.7.1 (<http://gnuradio.org/redmine/projects/gnuradio/wiki>). Accedido en 2015.

- Google Inc. (2013). Google App Engine website (<https://appengine.google.com/>). Accedido en 2015.
- Google Inc. (2013). Google Docs website. (<https://drive.google.com/>). Accedido en 2015.
- Günes, M., Blywis, B., and Schiller, J. (2008). A hybrid testbed for long-term wireless sensor network studies. In *Int. Workshop on Sensor Network Engineering (IWSNE)*.
- Handziski, V., Köpke, A., Willig, A., and Wolisz, A. (2006). TWIST: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality, REALMAN '06*, pages 63–70, New York, NY, USA. ACM.
- Hellbruck, H., Pagel, M., Kroller, A., Bimschas, D., Pfisterer, D., and Fischer, S. (2011). Using and operating wireless sensor network testbeds with WISEBED. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, pages 171–178.
- Hempstead, M., Lyons, M. J., Brooks, D., and Wei, G.-Y. (2008). Survey of hardware systems for wireless sensor networks. *Journal of Low Power Electronics*, 4(1):11–20.
- Huangfu, W., Sun, L., and Liu, J. (2010). A high-accuracy nonintrusive networking testbed for wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.*, 2010:2:1–2:15.
- Iacono, L. (2015). *Gestión Remota de Redes de Sensores Inalámbricos Mediante Tecnologías de Cloud Computing*. PhD thesis, Universidad de Mendoza.
- Iacono, L., Godoy, P., Marianetti, O., García Garino, C., and Párraga, C. (2012). Estudio de la integración entre WSN y redes TCP/IP. *Memoria de Trabajos de Difusión Científica y Técnica 10*. ISSN 1510-7450.
- IEEE, Institute of Electrical and Electronic Engineers (2003). Ieee std 802.15.4-2003.
- Imran, M., Said, A., and Hasbullah, H. (2010). A survey of simulators, emulators and testbeds for wireless sensor networks. In *Information Technology (ITSim), 2010 International Symposium in*, volume 2, pages 897–902.
- Information, N. and Communications Technology of Australia, NICTA, R. C. (2010). Castalia wireless sensor network simulator. Accedido en diciembre de 2014.
- Jona, K. and Uttal, D. (2013). Don't forget the teacher: New tools to support broader adoption of remote labs. In *Remote Engineering and Virtual Instrumentation (REV), 2013 10th International Conference on*, pages 1–2.

- Ju, X., Zhang, H., and Sakamuri, D. (2012). Neteye: a user-centered wireless sensor network testbed for high-fidelity, robust experimentation. *International Journal of Communication Systems*, 25(9):1213–1229.
- KanseiGenie (2006). Kanseigenie website (<http://kansei.cse.ohio-state.edu/kanseigenie/>). Accedido en 2015.
- Kanzaki, A., Hara, T., Ishi, Y., Wakamiya, N., and Shimojo, S. (2009). X-Sensor: A sensor network testbed integrating multiple networks. In *Complex, Intelligent and Software Intensive Systems, 2009. CISIS '09. International Conference on*, pages 1082–1087.
- Kelly, S., Suryadevara, N., and Mukhopadhyay, S. (2013). Towards the implementation of IoT for environmental condition monitoring in homes. *Sensors Journal, IEEE*, 13(10):3846–3853.
- Kumar, P. and Lee, H.-J. (2011). Security issues in healthcare applications using wireless medical sensor networks: A survey. *Sensors*, 12(1):55–91.
- Lambrou, T. P. and Panayiotou, C. G. (2011). A testbed for coverage control using mixed wireless sensor networks. *Journal of Network and Computer Applications*, 35(2):527–537.
- Langendoen, K. (2006). Apples, oranges, and testbeds. In *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 387–396.
- Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., et al. (2005). Tinyos: An operating system for sensor networks. *Ambient intelligence*, 35.
- Liu, B. and Towsley, D. (2004). A study of the coverage of large-scale sensor networks. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 475–483.
- M. Doddavenkatappa, M. C. C. and Ananda, A. L. (2011). Indriya: A low cost, 3D wireless sensor network testbed. In *TRIDENTCOM 2011*.
- Ma, X. and Luo, W. (2008). The analysis of 6LowPAN technology. In *Computational Intelligence and Industrial Application, 2008. PACIA '08. Pacific-Asia Workshop on*, volume 1, pages 963–966.
- Maiti, A. and Tripathy, B. (2013). Remote laboratories: Design of experiments and their web implementation. *Journal of Educational Technology & Society*, 16(3).
- Marianetti, O. (2006). *Laboratorios remotos, un aporte para su diseño y gestión*. Universidad de Mendoza, Facultad de Ingeniería, Maestría en Teleinformática.
- Mell, P. and Grance, T. (2011). The NIST definition of Cloud Computing (Draft). *NIST Special Publication*, 800:145.

- Metcalf, C., Camp, T., Colagrosso, M., and Chase, O. (2010). TestbedProfiler: A validation tool for wireless sensor network testbed deployment. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, pages 188–191.
- Mikhaylov, K., Tervonen, J., Heikkila, J., and Kansakoski, J. (2012). Wireless sensor networks in industrial environment: Real-life evaluation results. In *Future Internet Communications (BCFIC), 2012 2nd Baltic Congress on*, pages 1–7.
- NS-2 (2009). Network Simulator NS-2 website, versión 2.34 (<http://www.isi.edu/nsnam/ns>). Accedido en 2015.
- Oliveira, L. and Rodrigues, J. (2011). Wireless sensor networks: a survey on environmental monitoring. *Journal of Communications*, 6(2).
- Oppermann, F., Boano, C., and Römer, K. (2014). A decade of wireless sensing applications: Survey and taxonomy. In Ammari, H. M., editor, *The Art of Wireless Sensor Networks, Signals and Communication Technology*, pages 11–50. Springer Berlin Heidelberg.
- Orduna, P., Gomez-Goiri, A., Rodriguez-Gil, L., Diego, J., Lopez-de Ipina, D., and Garcia-Zubia, J. (2015). wCloud: Automatic generation of WebLab-Deusto deployments in the Cloud. In *Remote Engineering and Virtual Instrumentation (REV), 2015 12th International Conference on*, pages 223–229.
- Orduna, P., Larrakoetxea, X., Bujan, D., Angulo, I., Dziabenko, O., Rodriguez-Gil, L., Lopezde-IPina, D., and Garcia-Zubia, J. (2013). Weblab-deployer: Exporting remote laboratories as saas through federation protocols. In *Remote Engineering and Virtual Instrumentation (REV), 2013 10th International Conference on*, pages 1–5.
- PlanetLab Consortium (2003). Planetlab research network (<http://www.planetlab.org/>). Accedido en 2015.
- Polastre, J., Szewczyk, R., and Culler, D. (2005). Telos: enabling ultra-low power wireless research. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, pages 48–es. IEEE Press.
- Potdar, V., Sharif, A., and Chang, E. (2009). Wireless sensor networks: A survey. In *Advanced Information Networking and Applications Workshops, 2009. WAINA '09. International Conference on*, pages 636–641.
- Puccinelli, D. and Giordano, S. (2009). ViMobiO: Virtual mobility overlay for static sensor network testbeds. In *World of Wireless, Mobile and Multimedia Networks Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a*, pages 1–6.

- Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., Siracusa, R., Liu, H., and Singh, M. (2005). Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1664–1669. IEEE.
- Rensfelt, O., Hermans, F., Ferm, C., Larzon, L.-A., and Gunningberg, P. (2009). Sensei - a flexible testbed for heterogeneous wireless sensor networks. In *Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on*, pages 1 –2.
- Rensfelt, O., Hermans, F., Gunningberg, P., and Larzon, L. (2010). Repeatable experiments with mobile nodes in a relocatable WSN testbed. In *Distributed Computing in Sensor Systems Workshops (DCOSSW), 2010 6th IEEE International Conference on*, pages 1 –6.
- Ruiz-Garcia, L., Lunadei, L., Barreiro, P., and Robla, I. (2009). A review of wireless sensor technologies and applications in agriculture and food industry: state of the art and current trends. *Sensors*, 9(6):4728–4750.
- Rutgers University (2005). ORBIT network testbed website (<http://www.orbit-lab.org/>). Accedido en 2015.
- SensLAB Consortium (2010). IoT-LAB: a very large scale open testbed website (<https://www.iot-lab.info/>). Accedido en 2015.
- Shakshuki, E. M., Malik, H., and Sheltami, T. R. (2011). A comparative study on simulation vs. real time deployment in wireless sensor networks. *Journal of Systems and Software*, 84(1):45 – 54.
- Sheu, J.-P., Chang, C.-J., Sun, C.-Y., and Hu, W.-K. (2008). WSNTB: A testbed for heterogeneous wireless sensor networks. In *Ubi-Media Computing, 2008 First IEEE International Conference on*, pages 338 –343.
- Singh, C., Vyas, O., and Tiwari, M. (2008). A survey of simulation in sensor networks. In *Computational Intelligence for Modelling Control Automation, 2008 International Conference on*, pages 867 –872.
- Singh, S., Singh, M., and Singh, D. (2010). Routing protocols in wireless sensor networks—a survey. *International Journal of Computer science and engineering Survey (IJCES)*, 1(2):63–83.
- Slipp, J., Ma, C., Polu, N., Nicholson, J., Murillo, M., and Hussain, S. (2008). WINTeR: Architecture and applications of a wireless industrial sensor network testbed for radio-harsh environments. In *Communication Networks and Services Research Conference, 2008. CNSR 2008. 6th Annual*, pages 422 –431.

- Steyn, L. P. and Hancke, G. P. (2011). A survey of wireless sensor network testbeds. In *IEEE Africon'11*.
- Sundani, H., Li, H., Devabhaktuni, V., Alam, M., and Bhattacharya, P. (2011). Wireless sensor network simulators a survey and comparisons. *International Journal of Computer Networks*, 2(5):249–265.
- Tyagi, M. and Jain, M. (2012). Comparative study of routing protocols in wireless sensor network. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(9).
- University of Utah (2006). Emulab website (<http://www.emulab.net/tutorial/mobilewireless.php3>). Accedido en 2013.
- University of Utah (2008). Mobile Emulab website (<http://www.emulab.net/tutorial/mobilewireless.php3>). Accedido en 2015.
- Waldrop, M. (2013a). Campus 2.0. *Nature*, 495(7440):160–163.
- Waldrop, M. M. (2013b). Education online: the virtual lab. *Nature*, 499(7458):268–270.
- Wang, J.-C., Lin, C.-H., Siahaan, E., Chen, B.-W., and Chuang, H.-L. (2014). Mixed sound event verification on wireless sensor network for home automation. *Industrial Informatics, IEEE Transactions on*, 10(1):803–812.
- Wayne state University (2012). NetEye, wireless sensor network experimental facility website (<http://neteye.cs.wayne.edu/neteye/home.php>). Accedido en 2014.
- Werner-Allen, G., Swieskowski, P., and Welsh, M. (2005a). MoteLab: a wireless sensor network testbed. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 483 – 488.
- Werner-Allen, G., Swieskowski, P., and Welsh, M. (2005b). Motelab: Harvard sensor network testbed website (<http://motelab.eecs.harvard.edu/index.php>). Accedido en 2014.
- Yibo, C., mean Hou, K., Zhou, H., Shi, H.-L., Liu, X., Diao, X., Ding, H., Li, J.-J., and de Vaulx, C. (2011). 6lowpan stacks: A survey. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, pages 1–4.
- Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330.
- Zhang, Q., Cheng, L., and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18.

Zhao, Z., Yang, G.-H., Liu, Q., Li, V. O., and Cui, L. (2010). EasiTest: A multi-radio testbed for heterogeneous wireless sensor networks. In *Wireless Sensor Network, 2010. IET-WSN. IET International Conference on*, pages 104–108.

ZigBee Alliance (2008a). Zigbee-2007 specification.

ZigBee Alliance (2008b). Zigbee alliance website (<http://www.zigbee.org/>). Accedido en 2014.

Zvkovic, M., Nikolic, B., Protic, J., and Popovic, R. (2014). A survey and classification of wireless sensor networks simulators based on the domain of use. *Ad-hoc & sensor wireless networks*, 20(3-4):245–287.