

Universidad de Mendoza
Facultad de Ingeniería
Doctorado en Ingeniería

Gestión Remota de Redes de Sensores Inalámbricas Mediante Tecnologías de Cloud Computing

por
Lucas Emanuel Iacono

Director: Dr. Carlos García Garino

Co-Director: Dra. Cristina Párraga

Tesis presentada como requisito parcial
para optar por el título de
Doctor en Ingeniería

Mendoza, Julio de 2015

Resumen

Las redes de sensores inalámbricas (Wireless Sensor Networks ó simplemente WSNs) permiten brindar soluciones al estudio de fenómenos climáticos que causan pérdidas en cultivos como heladas y tormentas. Estas redes están compuestas por dispositivos denominados nodos sensores, los cuales permiten recolectar datos en campo y enviarlos al usuario vía Internet mediante una máquina denominada estación base.

Las WSNs utilizan protocolos de comunicaciones específicos para minimizar el consumo de energía en los nodos, por lo que no son compatibles con TCP/IP. Para incorporar las WSNs a TCP/IP, se deben implementar gateways que actúen a modo de traductor entre ambas tecnologías o redes overlay, las cuales solapan el protocolo TCP/IP sobre el nativo de la WSN. Una vez incorporadas a TCP/IP, las WSNs pueden ser gestionadas en forma remota mediante tecnologías tradicionales de Internet ó de computación distribuida como Grid y Cloud Computing. Cloud Computing es un paradigma basado en la posibilidad de ofrecer al usuario recursos informáticos (procesamiento, almacenamiento, software) en forma de servicios. Gracias a esta posibilidad, los usuarios pueden acceder a los servicios disponibles en el Cloud sin ser expertos en la gestión de los recursos que usan.

A diferencia de las tecnologías tradicionales de Internet, la gestión remota de WSNs mediante tecnologías cloud permite contar con recursos computacionales escalables, acordes a los requerimientos del usuario. Estos recursos tienen la ventaja de estar disponibles las 24 horas al día, los 365 días del año. La incorporación de las WSNs a Cloud ha dado lugar a infraestructuras a gran escala denominadas Clouds de Sensores, las cuales cuentan con recursos de medición en campo y recursos computacionales de alto rendimiento para almacenar y procesar la información recopilada en campo.

Actualmente, existen distintas plataformas que permiten gestionar Clouds de Sensores. Si bien la mayoría de dichas plataformas realizan con éxito la gestión remota de WSNs mediante tecnologías cloud, se ha detectado que la mayoría presentan los siguientes inconvenientes, no son apropiadas para gestionar las WSNs disponibles en el mercado local, sólo algunas plataformas realizan la gestión remota conservando las condiciones propias del buen funcionamiento de las WSNs, generalmente no están diseñadas en base a parámetros de eficiencia, y pueden ser optimizadas en aspectos como simplicidad de diseño e implementación.

Con el objetivo de solucionar los aspectos mencionados, en esta tesis se propone una nueva plataforma de gestión remota denominada Sensor Cirrus. Esta plataforma tiene una arquitectura basada en gateways y servicios cloud estándar no programados en la WSN. Además Sensor Cirrus brinda soluciones a problemas reales como lo son los daños en cultivos debido a fenómenos agro-meteorológicos como heladas en la Región de Cuyo,

Argentina.

El fenómeno de las heladas causa severas pérdidas en la economía de la región. Como ejemplo puede citarse que en la Provincia de Mendoza, República Argentina, las pérdidas económicas debido a las heladas alcanzan al 9 % del total del área cultivada bajo riego. En particular, en Octubre del año 2013 se decretó emergencia agropecuaria en la provincia debido a los daños causados por heladas generales. Dicho fenómeno ocurrió durante el mes de septiembre y afectó en algunos distritos hasta el 80 % de los cultivos.

Los experimentos realizados en esta tesis han permitido validar que Sensor Cirrus puede gestionar en forma remota, sencilla y eficiente WSNs conformadas por recursos acotados de hardware y fuente de energía, mediante tecnologías cloud, adecuando las tecnologías disponibles en el mercado local y con un bajo costo económico. Además, los experimentos permitieron realizar la prueba de concepto del prototipo de servicio de predicción de heladas de Sensor Cirrus, el cual permite determinar la probabilidad de ocurrencia de heladas.

En resumen, la presente tesis ha dado lugar a una nueva plataforma para la gestión de Clouds de Sensores, la cual estará al alcance de todos los productores agrícolas de la Región de Cuyo, por lo que permitirá reducir los daños causados en los cultivos por eventos como las heladas y contribuirá así al desarrollo local de la Región.

Palabras Clave: WSN, Cloud Computing, Predicción de Heladas.

Agradecimientos

Después de mucho esfuerzo y dedicación, quisiera agradecer a aquellas personas que hicieron que esto sea hoy posible.

A mi Director de tesis Carlos García Garino por su dirección y paciencia. Además, por compartir su experiencia y sabiduría, valores que me guiaron a través de la fase más difícil de mi vida.

A mi Co-Directora y amiga Cristina Párraga por su acompañamiento a lo largo de mi carrera de doctorado, sus consejos permanentes y su calidad de persona.

Agradezco a la Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT), al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) y al programa Bec.AR por financiar mis estudios de Doctorado.

Quisiera también agradecer al Ing. Osvaldo Marianetti por ser quien me inició en el camino del doctorado y brindó su amistad en todo momento desde el grado al doctorado.

Estoy muy agradecido al Dr. José Luis Vázquez Poletti, por compartir sus conocimientos e ideas sobre ciencia y Cloud Computing conmigo.

Un párrafo aparte merecen mis compañeros y amigos del ITIC. Juntos compartimos momentos graciosos, duros y finalmente muy valiosos. En especial a Elina Pacini, David Monge y Carlos Catania por sus consejos y aportes sobre mi tesis.

Muy especialmente quisiera agradecer a mi esposa Roxana por su carisma, amor, y acompañamiento constante a lo largo de estos años de estudio y superación. También a mis hijos quienes dieron parte del tiempo de su “papa” para que hoy pueda terminar mi doctorado.

Finalmente le agradezco a mis padres Ana y Manuel por no dejarme nunca bajar los brazos y a mi hermana Belén por toda las discusiones y consejos “agronómicos” relacionados con mi tesis.

Lucas E. Iacono

Índice general

1. Introducción	1
1.1. Motivación	3
1.2. La Tesis	5
1.3. Contribuciones	7
1.4. Organización	8
2. Antecedentes	11
2.1. Redes de Sensores Inalámbricas	11
2.1.1. Características de las WSN	12
2.1.1.1. Ventajas	12
2.1.1.2. Requerimientos de Diseño	13
2.1.2. Protocolos de WSNs	14
2.1.2.1. IEEE 802.15.4	15
2.1.2.2. Protocolos de Red	16
2.2. Grid y Cloud Computing	18
2.2.1. Grid Computing	19
2.2.1.1. Modelo Conceptual de Grid Computing	20
2.2.1.2. Funcionamiento de Grid Computing	21
2.2.1.3. Ejemplos de Infraestructuras Grid	22
2.2.1.4. Aplicaciones de Grid Computing	24
2.2.2. Cloud Computing	26

2.2.2.1.	Modelo de Servicios Cloud	26
2.2.2.2.	Servicios de Infraestructura y Plataforma	28
2.2.2.3.	Servicios de Aplicación	29
2.3.	Integración de WSNs a Grid y Cloud Computing	32
2.4.	Conclusiones	34
3.	Trabajos Relacionados	37
3.1.	Integración de WSNs a TCP/IP	37
3.1.1.	Gateway	38
3.1.2.	Redes Overlay	40
3.2.	Gestión de WSNs desde Internet	42
3.2.1.	Gestión con Tecnologías Tradicionales	43
3.2.1.1.	Ad-hoc	43
3.2.1.2.	Basadas en Estándares	44
3.2.2.	Gestión mediante Tecnologías Grid y Cloud	47
3.2.2.1.	Middleware ó Servicio Cloud no embebido en la WSN	48
3.2.2.2.	Middleware ó Servicio Cloud Embebido en la WSN	52
3.3.	Resumen de los Trabajos Relacionados	56
3.4.	Discusión	57
3.4.1.	Integración de WSNs a TCP/IP	57
3.4.2.	Gestión Remota	60
3.4.3.	Factibilidad Tecnológica y Costo Económico	63
3.4.4.	Eficiencia	64
3.5.	Conclusiones	65
4.	Sensor Cirrus	69
4.1.	Hipótesis de Trabajo	71
4.2.	Arquitectura	71
4.3.	Sensor Cirrus: Descripción de Módulos	74
4.3.1.	WSN Experimental	74

ÍNDICE GENERAL

4.3.2.	Módulo WSN - TCP/IP	75
4.3.3.	Módulo TCP/IP - Cloud	78
4.3.4.	Portal Web	80
4.3.5.	Servicio de Predicción de Heladas	84
4.3.5.1.	Método de Predicción de Heladas	84
4.3.5.2.	Implementación del Servicio de Predicción	85
4.4.	Conclusiones	86
5.	Experimentos en la WSN	89
5.1.	Estudio de Plataformas de Hardware	89
5.1.1.	Nodos Sensores Disponibles en el Mercado Local	90
5.1.2.	Ensayos de Consumo	92
5.1.2.1.	Metodología Experimental	92
5.1.2.2.	Resultados Experimentales	92
5.2.	Efecto de la Potencia de Transmisión en la Vida Útil y Distancia de Comunicación Confiable.	96
5.2.1.	Ensayo de Consumo y Vida Útil	97
5.2.1.1.	Metodología experimental	97
5.2.1.2.	Resultados Experimentales	99
5.2.2.	Ensayo de Distancia de Comunicación Confiable	102
5.2.2.1.	Metodología Experimental	102
5.2.2.2.	Resultados Experimentales	103
5.3.	Efecto de las Condiciones de Trabajo sobre la Comunicación de Nodos Zig-Bee	104
5.3.1.	Metodología Experimental	104
5.3.2.	Resultados Experimentales	106
5.4.	Conclusiones	109

6. Experimentos en Sensor Cirrus	111
6.1. Pérdida de Paquetes en el Proceso de Comunicación	111
6.1.1. Metodología Experimental	112
6.1.2. Resultados Experimentales	115
6.2. Servicio de Predicción de Heladas	117
6.2.1. Heladas en la Región de Cuyo, Argentina	118
6.2.1.1. Heladas. Definición y Tipos	118
6.2.1.2. Daños en Cultivos	120
6.2.1.3. Métodos de Defensa	120
6.2.1.4. Sistemas de Alarmas y Procedimiento de Guardias Contra Heladas	124
6.2.1.5. Impacto en la Región de Cuyo	124
6.2.2. Metodología Experimental	126
6.2.3. Resultados Experimentales en Instancias Individuales	126
6.2.3.1. Modelos Teóricos de Predicción de Rendimiento	128
6.2.3.2. Comparación de Instancias de Procesamiento en EC2	129
6.2.4. Precisión de los Modelos Propuestos para el Caso Típico de Uso	132
6.2.4.1. Ejecución del Servicio de Predicción de Heladas para las Mejores Instancias del Caso Típico de Uso	133
6.2.4.2. Comparación de Resultados	134
6.2.4.3. Ejecución del Servicio de Predicción de Heladas para Resultados Corregidos	134
6.2.5. Resultados Experimentales en Clústeres Virtuales	136
6.3. Conclusiones	137
7. Conclusiones y Trabajos Futuros	141
7.1. Contribuciones	142
7.2. Limitaciones	143
7.3. Trabajos Futuros	145
7.3.1. Implementación de Estaciones Bases basadas en Android OS	145
7.3.2. Paralelización del Servicio de Predicción de Heladas de Sensor Cirrus	145

<i>ÍNDICE GENERAL</i>	XI
7.3.3. Validación de la Calidad de la Predicción de Heladas Realizada por Sensor Cirrus	145
7.3.4. Utilización de Clouds de otros Proveedores y Selección del más Adecuado para Procesar el Servicio de Predicción de Heladas . . .	146
7.4. Observaciones Finales	146
A. Publicaciones Realizadas	149
Bibliografía	153

Índice de figuras

1.1. Arquitectura de Sensor Cirrus.	6
2.1. Nodo sensor.	12
2.2. Elementos de una WSN.	12
2.3. Pila de protocolos WSNs.	14
2.4. Pila del protocolo IEEE 802.15.4.	16
2.5. Pila del protocolo ZigBee.	17
2.6. Pila del protocolo Wireless Hart.	17
2.7. Pila del protocolo 6lowPAN.	18
2.8. Pila del protocolo ISA 100.11.a	19
2.9. Modelo Conceptual de Grid Computing.	20
2.10. Funcionamiento de la Grid.	22
2.11. Servicios Cloud.	27
2.12. Transferencia de archivos en Globus Online.	31
2.13. Arquitectura General Plataforma de Integración WSN - Grid/Cloud. . .	32
2.14. Etapa WSN - TCP/IP.	33
2.15. Integración por gateway.	33
2.16. Integración mediante redes overlay.	34
2.17. Etapa TCP/IP - Grid/Cloud.	34
2.18. Plataforma WSN - Grid/Cloud.	35

3.1. Arquitectura en capas de SYLPH.	39
3.2. Arquitectura desarrollada por Annamraju et al.	39
3.3. Arquitectura de comunicación de Interoperable Test-Bed.	40
3.4. Arquitectura de SPMIPv6.	42
3.5. Componentes de SYLPH.	44
3.6. Enfoques de integración WSN - Grid/Cloud	47
3.7. Arquitectura de National Weather Sensor Grid.	49
3.8. Arquitectura de MPAS.	49
3.9. Arquitectura de Tangible Cloud.	51
3.10. Arquitectura propuesta por Aloisio et. al.	53
3.11. Nodo desarrollado por Zhang et al.	54
4.1. Arquitectura de Sensor Cirrus.	72
4.2. Módulo WSN - TCP/IP.	72
4.3. Módulo TCP/IP - Cloud.	73
4.4. Nodo Sensor WSN Experimental.	75
4.5. Gestión de una WSN mediante Sensor Cirrus.	76
4.6. Adquisición de Datos en el Nodo Sensor.	76
4.7. Módulo WSN - TCP/IP.	78
4.8. Funcionamiento Módulo TCP/IP - Cloud	80
4.9. Portal Web Sensor Cirrus.	81
4.10. Gráficos de Datos Recolectados por la WSN.	81
4.11. Proceso de Visualización de Datos.	82
4.12. Cambio de Configuración desde Portal Web	83
4.13. Servicio de Predicción de Heladas.	86
5.1. Ensayo de Consumo Plataforma XBee.	94
5.2. Ensayo de la Plataforma Freescale.	96
5.3. Cálculo del Consumo.	97
5.4. Ensayo de Vida Útil XBee.	100

5.5. Ensayo de Vida Útil Freescale.	101
5.6. Escenarios de Prueba.	107
6.1. Elementos y Enlaces del Proceso de Comunicación.	113
6.2. Nodo Sensor Utilizado.	114
6.3. Daños por Heladas.	120
6.4. Métodos de Defensa contra Heladas.	122
6.5. Daños por Heladas en Mendoza.	125
6.6. Tiempos de Ejecución y Costos Económicos para cada Tipo de Instancia.	127
6.7. Tiempos y Costos de Ejecución Teóricos Calculados con los Modelos de Rendimiento.	129
6.8. Tiempos de Ejecución Empíricos versus Teóricos Calculados con los Modelos de Rendimiento.	130
6.9. Costos Económicos Empíricos versus Teóricos Calculados con los Modelos de Rendimiento.	131

Índice de tablas

2.1. Ejemplos de infraestructuras Grid.	25
2.2. Comparativa de servicios IaaS y PaaS en Cloud.	30
3.1. Ventajas y Desventajas de utilizar XML en WSN.	46
3.2. Componentes de SEGEDMA.	54
3.3. Resumen de Trabajos Previos	58
5.1. Características de Plataformas de Nodos Sensores Disponibles en el Mercado Local.	93
5.2. Estados del Ciclo de Funcionamiento del Nodo.	99
5.3. Características de los Sensores Simulados.	103
5.4. Resultados Ensayo de Distancia Confiable Nodo XBee.	103
5.5. Resultados Ensayo de Distancia Confiable Nodo Freescale.	104
5.6. Escenarios de Prueba.	105
5.7. Resultados Obtenidos en cada Escenario de Prueba.	109
6.1. Enlaces del Proceso y PLR Correspondiente.	113
6.2. Escenarios Experimentales.	115
6.3. Paquetes Recibidos y Emitidos por cada Módulo de Sensor Cirrus.	115
6.4. Resultados WSN de Media Escala.	116
6.5. Paquetes Recibidos y Emitidos por cada Módulo de Sensor Cirrus.	117
6.6. Resultados WSN de Gran Escala.	117

6.7. Métodos de Defensa.	121
6.8. Características de las Instancias Utilizadas.	127
6.9. Coeficientes de los Modelos de Predicción de Rendimiento Propuestos para cada Tipo de Instancia.	128
6.10. Nodos Sensores Procesados en una Hora en Cada Tipo de Instancia. . .	132
6.11. Tiempos de ejecución empíricos obtenidos para el caso típico de uso. . .	133
6.12. Comparación entre los tiempos de ejecución empíricos y teóricos obtenidos para el caso típico de uso.	134
6.13. Corrección del número de nodos sensores procesados para el caso típico de uso.	135
6.14. Resultados obtenidos para la cantidad de nodos sensores corregidos en el caso típico de uso.	135
6.15. Tiempos de ejecución en clústeres virtuales para el caso típico de uso. .	137

Introducción

Los avances en diseño y miniaturización de circuitos electrónicos han dado lugar al surgimiento de nuevos dispositivos llamados nodos sensores. Un nodo sensor es un sistema embebido compuesto por sensores de distinto tipo, un microcontrolador, memoria, interfaz inalámbrica de conexión y baterías. Debido a que los nodos sensores cuentan con recursos acotados de memoria, procesamiento y fuente de energía, utilizan protocolos de comunicación inalámbricos diseñados específicamente para minimizar el uso de dichos recursos. Algunos de los protocolos más populares son ZigBee [ZigBee Alliance, 2006], 6lowPAN [Bormann et al., 2006], y IEEE 802.15.4 [IEEE, 2006].

Los protocolos de comunicación utilizados por los nodos sensores permiten agrupar los nodos y formar redes llamadas Redes de Sensores Inalámbricas (Wireless Sensor Networks o simplemente WSNs). Estas redes brindan soluciones a aplicaciones como agricultura de precisión [Morais et al., 2008], meteorología y monitoreo del clima [Ghanem et al., 2004, O'Reilly et al.], salud [Corchado et al., 2010, Preve, 2011], prevención de desastres naturales [Hughes et al., 2008], entre otras.

En aplicaciones como agricultura de precisión y meteorología, el uso de WSNs ofrece distintas ventajas frente a otras tecnologías de medición como instrumentos y estaciones meteorológicas. Entre otras ventajas, las WSNs permiten cubrir áreas extensas con nodos de bajo costo, alimentados por baterías y re-programables. La eficiencia energética de los nodos y su vida útil (mayor de 2 años) permiten acceso en tiempo real a los datos recolectados y mínimo mantenimiento de la WSN.

Debido a que las WSNs se encuentran desplegadas en campo (generalmente a kilómetros del usuario), se requieren tecnologías que permitan gestionar los recursos de las WSNs desde Internet, es decir tecnologías que permitan al usuario visualizar, almacenar y procesar los datos adquiridos por los nodos sensores en campo, y además realizar cambios en la configuración de dichos dispositivos.

Tradicionalmente, la gestión remota de WSNs ha sido realizada mediante tecnologías y protocolos de redes y/o de sensores. Dichos protocolos generalmente están desplegados sobre servidores que funcionan de manera aislada. Los principales problemas de realizar la gestión de WSNs mediante tecnologías tradicionales son, en primer lugar, que dichas tecnologías no cuentan con mecanismos de alta disponibilidad y tolerancia a fallos, en segundo lugar, no están preparadas para asegurar la escalabilidad de la provisión de recursos computacionales en base al aumento de nodos sensores, y finalmente, que implican altos costos de mantenimiento para asegurar la disponibilidad de los recursos computacionales, las 24 horas del día y los 365 días del año.

Con el objetivo de solucionar los problemas mencionados, en la última década han surgido una serie de propuestas que plantean el uso de tecnologías de computación distribuida para gestionar las WSNs desde Internet [Humble et al., 2005, Hori and Kawashima, 2010, Lee et al., 2010, Ahmed and Gregory, 2011, Alamri et al., 2013]. A diferencia de las tecnologías tradicionales, las tecnologías de computación distribuida como Grid y Cloud Computing están preparadas brindar soporte a la escalabilidad de las WSNs.

En primer lugar, el concepto de Grid Computing [Foster et al., 2001] propone la interconexión y utilización de recursos computacionales heterogéneos y descentralizados, que permiten plantear la noción de una gran computadora virtual con recursos escalables y administración descentralizada. La tecnología Grid incluye además soluciones de seguridad, gestión de protocolos y servicios, entre otros. En particular, el desarrollo de Grid Computing ha permitido que infraestructuras compuestas por grandes cantidades de recursos de hardware heterogéneo sean utilizadas para ejecutar aplicaciones de distinto tipo cuyo común denominador es la gran cantidad de cómputo y tareas simultáneas. Además, en la última década, Grid se ha extendido brindando soluciones a aplicaciones científicas que implican la operación remota y el análisis de datos de instrumentos, satélites y sensores [Foster and Kesselman, 2011].

Por otra parte, Cloud Computing es un paradigma basado en la posibilidad de ofrecer al usuario recursos informáticos (procesamiento, almacenamiento, software) en forma de servicios. Gracias a esta posibilidad, los usuarios pueden acceder a los servicios disponibles en el Cloud sin ser expertos en la gestión de los recursos que usan.

El concepto de Cloud Computing [Buyya et al., 2009] se define como *“un sistema de computación distribuida orientado al mercado que consiste en un conjunto de computadoras interconectadas y virtualizadas. Dicho sistema se provee y presenta en forma dinámica como uno o más recursos de cómputo unificado (s) en base a acuerdos de servicio establecidos mediante negociaciones entre el proveedor y los consumidores de servicios”*.

En Cloud, el usuario accede a los servicios desde cualquier dispositivo, móvil o fijo, mediante una conexión a Internet independientemente de su ubicación geográfica y además,

permite que el usuario aumente las peticiones de recursos informáticos en forma escalable de acuerdo a su necesidad y pague sólo por el consumo efectuado.

Finalmente, gracias a la utilización de tecnologías Grid y Cloud, surgen nuevas infraestructuras a gran escala conformadas por: nodos sensores heterogéneos y recursos computacionales de altas prestaciones denominadas Grids y Clouds de Sensores. Estas infraestructuras permiten que distintas aplicaciones puedan realizar la gestión remota de las WSN mediante los recursos computacionales disponibles en la Grid ó Cloud.

En resumen, las Grids y Clouds de Sensores permiten brindar solución a distintos problemas a través de la provisión de recursos computacionales y de sensado de gran escala, escalables y disponibles las 24 horas del día y los 365 días del año.

1.1. Motivación

Las heladas son un fenómeno agro-meteorológico que afecta a cultivos de distinto tipo. Como ejemplo puede citarse que en la Provincia de Mendoza, República Argentina, las pérdidas económicas debido a las heladas alcanzan al 9 % del total del área cultivada bajo riego. En particular, en Octubre del año 2013 se decretó emergencia agropecuaria en la provincia debido a los daños causados por heladas generales. Dicho fenómeno ocurrió durante el mes de septiembre y afectó en algunos distritos hasta el 80 % de los cultivos.

Una estimación del impacto económico causado por las heladas puede plantearse a partir del caso de la industria vitivinícola Argentina. Las exportaciones de vinos (que han alcanzando un total de 733.9 millones de dólares en el año 2010) han hecho que en los últimos 10 años Argentina se haya incorporado a los principales países exportadores a nivel mundial. Argentina, actualmente posee una superficie cultivada con vid de 228.575 hectáreas, base congelada al cierre de la cosecha 2010, de las cuales, la provincia de Mendoza posee una superficie cultivada con vid de 160.704 hectáreas lo cual representa el 70,31 % del total de la superficie cultivada con vid del país.

Dado que las heladas es un fenómeno que sucede todos los años, existen distintos métodos de defensa para minimizar los daños, entre los cuales se puede mencionar el riego por aspersores, uso de quemadores diesel y ventiladores. Sin embargo, los métodos de defensa deben ser activados en base a alarmas provistas por sistemas de monitoreo agrícola denominados *sistemas de alarma contra heladas*. Los sistemas de alarma contra heladas, realizan la adquisición de datos en campo y permiten el acceso a los datos recolectados para su posterior análisis.

Las Grids y Clouds de Sensores presentan distintas ventajas para el desarrollo de sistemas de alarmas contra heladas. Por un lado, el uso de sensores inalámbricos permite contar con varios puntos de medición de temperatura, humedad y otros parámetros dentro de

un mismo establecimiento agrícola. El acceder a gran cantidad de puntos de medición es de especial interés para la predicción de heladas, ya que el fenómeno depende de las características particulares de cada sector de la finca (presencia de árboles, laderas de montañas, etc.). Como ejemplo, se puede mencionar que en algunos casos la helada se ha presentado en sectores de la finca expuestos a corrientes de aire frío (como aquellos colindantes con laderas de montañas) y en otros sectores cercanos (como los rodeados por cortinas de árboles) no se ha presentado el fenómeno. Además, cabe destacar que el costo económico de los sensores es menor comparado con otras tecnologías como estaciones meteorológicas, por lo que se puede reducir significativamente el costo de instrumentar las fincas.

Por otro lado, el acceso a infraestructuras computacionales de gran escala como Grid y Cloud permite contar con recursos computacionales de altas prestaciones. Gracias a esta posibilidad, los servicios de predicción de heladas pueden mejorar la confiabilidad de la predicción mediante el procesamiento de grandes volúmenes de datos, generados por cientos de nodos sensores. Además, si se utilizan servicios Cloud se puede incorporar los beneficios de dicha tecnología (replicación de datos, tolerancia a fallos, escalabilidad de recursos, etc.) a los sistemas de alarma contra heladas.

Actualmente, existen distintas plataformas que permiten gestionar Grids y Clouds de Sensores [Humble et al., 2005, Yujie et al., 2005, Hirafuji et al., 2011, Yuriyama et al., 2011]. Estas plataformas (de ahora en adelante plataformas de gestión remota de WSNs) utilizan los protocolos y servicios de Grid ó Cloud para gestionar la configuración y los datos obtenidos de las WSNs desde Internet. Además, las plataformas de gestión remota permiten la incorporación de las WSNs como otro recurso ó servicio más de Grid ó Cloud. Si bien la mayoría de dichas plataformas realizan con éxito la gestión remota de WSNs mediante tecnologías de computación distribuida, se ha detectado que aún existen algunos problemas a resolver, entre los cuales se encuentran:

- *Incompatibilidad de protocolos entre ambas tecnologías.* Grid y Cloud utilizan para su funcionamiento middlewares ó servicios basados en protocolos pertenecientes a la pila TCP/IP, mientras que las WSNs funcionan bajo protocolos específicamente diseñados para minimizar el consumo de energía, los cuales son incompatibles con TCP/IP [Zuñiga and Krishnamachari, 2003]. Para resolver la incompatibilidad de protocolos entre las WSNs y las tecnologías de computación distribuida se deben desarrollar plataformas que, en primer lugar, integren las WSNs a TCP/IP, y luego, utilicen un middleware Grid ó servicio Cloud para gestionar los datos y configuración de las WSNs desde Internet.
- *Factibilidad Tecnológica.* Otro de los problemas a resolver es la gestión remota de plataformas de hardware de WSNs disponibles en el mercado local mediante tec-

nologías Grid y Cloud. Las plataformas de hardware de WSNs disponibles en Argentina, utilizan microcontroladores con recursos acotados de memoria y procesamiento. Además, emplean baterías como fuente de energía y generalmente utilizan protocolos de comunicación basados en IEEE 802.15.4 [IEEE, 2006] como por ejemplo ZigBee [ZigBee Alliance, 2006].

- *Conservar las condiciones propias del buen funcionamiento de las WSNs.* Se debe realizar la gestión remota de WSNs en forma eficiente, por lo que se requieren plataformas de gestión remota de WSNs que permitan cumplir con parámetros de eficiencia tales como, bajo consumo de energía en los nodos sensores y tasas de errores aceptables en la entrega de datos.
- *Disponibilidad y Escalabilidad de Recursos.* Si bien tanto Grid como Cloud permiten gestionar grandes cantidades de nodos sensores, sólo los servicios Cloud garantizan la alta disponibilidad de los recursos (las 24 horas del día y los 365 días del año) y la provisión escalable de recursos en base a los requerimientos del usuario. Es por este motivo, que se deben desarrollar plataformas de gestión remota de WSNs preferentemente mediante servicios Cloud.

Tomando en cuenta los supuestos discutidos anteriormente, existe la necesidad de un nuevo enfoque que aborde eficazmente éstas cuestiones. La siguiente sección describe la presente tesis, la cual cubre los principales retos y provee una nueva alternativa para lidiar con los problemas de los enfoques existentes.

1.2. La Tesis

En esta tesis se propone un nuevo enfoque cuyo objetivo es el de desarrollar un prototipo de un Cloud de Sensores adecuado a los recursos disponibles en el contexto local. La hipótesis principal detrás de esta tesis doctoral afirma que *“se puede gestionar en forma remota, sencilla y eficiente WSNs conformadas por recursos acotados de hardware y fuente de energía, mediante tecnologías de computación distribuida como Cloud Computing, adecuando las tecnologías disponibles en el mercado local y con un bajo costo”*.

Para este propósito, en esta tesis se desarrolla una nueva plataforma de gestión remota de WSNs denominada *Sensor Cirrus*. *Sensor Cirrus* permite incorporar las WSNs a Internet y utiliza tecnologías Cloud para permitir que los datos de las WSNs sean accedidos, almacenados y procesados en forma sencilla y confiable, que se puedan realizar cambios en la configuración de la WSN, que no se generen consumos extras de energía y recursos de hardware y finalmente que se puedan gestionar en forma remota y eficiente WSNs tecnológicamente factibles.

Además, Sensor Cirrus ofrece nuevas soluciones a los productores de la Región de Cuyo, en Argentina, ya que permite evitar pérdidas económicas debidas a daños en cultivos causados por heladas. En la Figura 1.1 se presenta la arquitectura de Sensor Cirrus. Como se puede observar, Sensor Cirrus está conformada por dos módulos principales (Módulo WSN - TCP/IP y Módulo TCP/IP - Cloud), un portal web y un servicio de predicción de heladas.

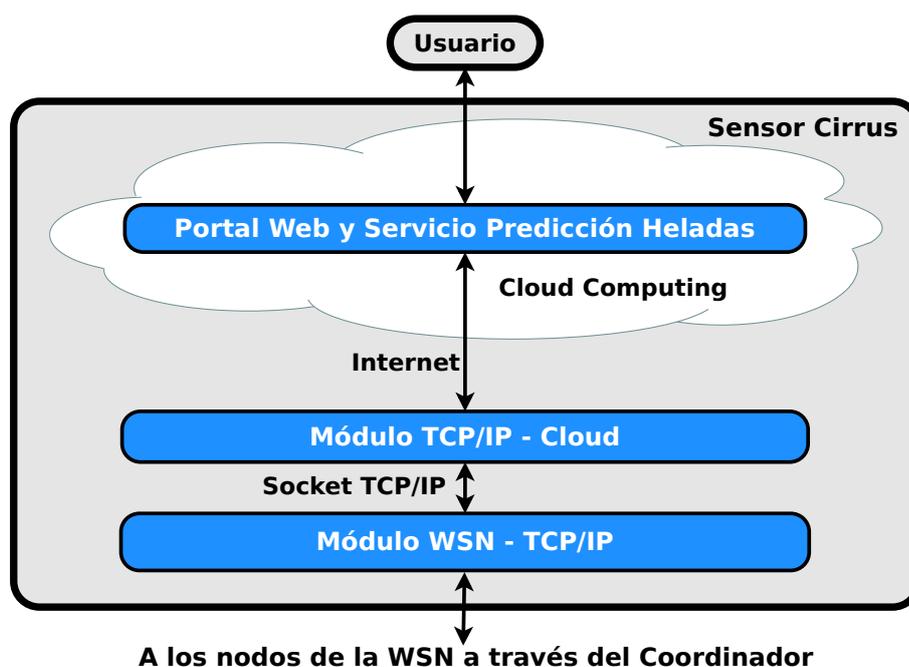


Figura 1.1: Arquitectura de Sensor Cirrus.

A continuación, se describen cada uno de los módulos de Sensor Cirrus:

- *Módulo WSN - TCP/IP.* Permite incorporar las WSNs a TCP/IP. Este módulo se desarrolla mediante el enfoque de gateway, el cual permite reducir los consumos de recursos de hardware y fuente de energía en los nodos de la WSN. Dicho módulo está programado en la estación base de la WSN y actúa a modo de traductor entre el protocolo nativo de la WSN (ZigBee) y los protocolos pertenecientes a la pila TCP/IP.
- *Módulo TCP/IP - Cloud.* Este módulo utiliza servicios estándares de transferencia de archivos de Cloud como Google Drive para gestionar en forma remota los recursos de las WSNs. El Módulo TCP/IP - Cloud está desplegado sobre una máquina externa a la WSN. Gracias a esta posibilidad, la integración de las WSNs a Cloud Computing se realiza conservando las condiciones propias del buen funcionamiento de las WSNs. Además, el módulo recibe los datos provenientes de las WSN y permite que los scripts alojados en la *portal web* de Sensor Cirrus generen gráficos

para que el usuario pueda visualizar los datos recolectados por los sensores. Finalmente, el Módulo TCP/IP - Cloud permite recibir los pedidos de cambio de configuración (realizados por el usuario a través del portal web) y enviar estos pedidos al Módulo WSN - TCP/IP para cambiar la configuración de los nodos sensores.

- *Portal web.* El sitio web de Sensor Cirrus está compuesto por una serie de scripts que permiten al usuario visualizar los datos provenientes del campo, procesarlos y realizar cambios en la configuración de los sensores. Dichos scripts han sido desarrollados con servicios de plataforma Cloud. Además, tanto los scripts como el sitio web están alojados en un Cloud Público, lo que garantiza la confiabilidad y alta disponibilidad del portal y sus componentes las 24 horas del día y los 365 días del año.
- *Servicio de Predicción de Heladas.* El Servicio de Predicción de heladas de Sensor Cirrus utiliza servicios de infraestructura Cloud para procesar los datos provenientes de las WSNs y detectar la probabilidad de ocurrencia de heladas en la finca. Este servicio, ha sido desarrollado a partir del método de predicción de heladas propuesto por Snyder y Melo-Abreu, el cual ha sido publicado y recomendado por la Food and Agriculture Organization of the United Nations (FAO) en [Snyder and Melo-Abreu, 2005].

1.3. Contribuciones

Esta tesis presenta importantes contribuciones al área de Clouds de Sensores, a saber:

- Una *nueva plataforma para la gestión remota de Clouds de Sensores*, la cual es adecuada a los recursos y tecnologías disponibles en Argentina. La plataforma utiliza gateways para incorporar WSNS a TCP/IP, y servicios Cloud estándar no programados en la WSN. Esto permite que las WSNs puedan ser gestionadas en forma remota mediante tecnologías Cloud, sin que se generen consumos extras de recursos de hardware y de fuente de energía en los elementos de las WSNs.
- Un *Cloud de Sensores tecnológicamente factible*, compuesto por plataformas de hardware disponibles en el mercado local y de bajo costo económico. Sensor Cirrus permite el desarrollo de Clouds de Sensores de bajo costo económico que permitan ejecutar servicios para activar sistemas de defensa contra heladas y mitigar los daños producidos en los cultivos.
- Una *nueva interfaz* que permite realizar la *gestión remota de WSNs en forma sencilla*, sin que se requiera por parte del usuario conocimientos en la programación

de WSNs. Esto se ha logrado mediante el uso de servicios Cloud estándar como Google Drive y tecnologías de redes simples como sockets TCP.

- Una *prueba de concepto de un servicio Cloud de predicción de heladas*, basado en información proveniente de WSNs. El servicio de predicción de heladas de Sensor Cirrus permite realizar pronósticos de heladas de radiación específicos para cada sector de la finca donde está desplegada la WSN.
- La *materialización de los aportes mencionados* en un nuevo prototipo de plataforma de gestión de Clouds de Sensores denominado Sensor Cirrus. Sensor Cirrus ha sido diseñada para gestionar WSNs en forma sencilla y eficiente. La plataforma integra WSNs de bajo costo económico y servicios Cloud escalables para visualizar, almacenar y procesar grandes volúmenes de datos recolectados en campo. Además, Sensor Cirrus permite que el usuario cambie la configuración de las WSNs de acuerdo a sus necesidades. Por último, la plataforma estará al alcance de todos los productores agrícolas de la Región de Cuyo, por lo que se contribuirá así al desarrollo local de la Región.

1.4. Organización

La presente tesis está estructurada de la siguiente manera. El Capítulo 2 introduce los conceptos esenciales sobre las WSNs y describe sus características distintivas. Además, este capítulo brinda un resumen de Grid y Cloud Computing a través de los modelos conceptuales de ambas tecnologías, ejemplos de aplicaciones, infraestructuras, middlewares y servicios. Finalmente, el capítulo ofrece una introducción al fenómeno de las heladas y su impacto en la economía de la Provincia de Mendoza, Argentina.

En el Capítulo 3, se relevan los trabajos que discuten la integración de WSNs a infraestructuras Grid y Cloud. Además, el capítulo, está estructurado en dos partes bien diferenciadas. En la primer parte, se estudian trabajos que resuelven la integración de WSNs a TCP/IP. El estudio de integración de WSNs a TCP/IP, ha sido presentado en [Iacono et al., 2012]. Luego, en la segunda parte del Capítulo 3, se analizan propuestas de distintos autores que abordan la gestión remota de WSNs mediante tecnologías tradicionales y de computación distribuida. Del análisis de los trabajos relevados surgen algunos temas abiertos de potencial interés, los cuales han permitido plantear la hipótesis de trabajo de esta tesis.

En el Capítulo 4, se presenta a Sensor Cirrus, una plataforma de gestión remota de WSNs mediante tecnologías Cloud. En este capítulo, se describen algunos de los principales aportes de esta tesis como la arquitectura basada en gateways y tecnologías Cloud estándar no programadas en la WSN, y el servicio de predicción de heladas. La mayoría de

los contenidos discutidos en este capítulo han dado lugar al trabajo publicado en [Iacono et al., 2013].

Para simplificar la lectura de la presente tesis, los resultados experimentales se discuten en dos capítulos diferentes. Por un lado, el Capítulo 5 presenta distintos experimentos con plataformas de WSN. El principal objetivo de los experimentos es analizar las plataformas de hardware de WSNs disponibles en el mercado argentino, estudiar su comportamiento en campo y seleccionar las más adecuadas para implementar una WSN experimental. Una vez seleccionada la instancia más adecuada, se evalúa la eficiencia de Sensor Cirrus para ofrecer soluciones a problemas de la región de Cuyo, como lo son los daños en cultivos causados por las heladas. Los resultados de este capítulo han dado lugar a las siguientes publicaciones [Iacono et al., 2010, Godoy et al., 2010, 2012].

Por otro lado, el Capítulo 6 muestra los resultados obtenidos de los experimentos realizados con Sensor Cirrus. En la primer parte de este capítulo, se estudia la tasa de pérdida de paquetes en los distintos enlaces del proceso de comunicación de Sensor Cirrus. Luego, en la segunda parte del Capítulo 6, se realizan experimentos con el servicio de predicción de heladas de Sensor Cirrus. Los experimentos realizados, demuestran como se pueden predecir heladas mediante WSNs y servicios Cloud. Además, se presentan modelos teóricos que permiten optimizar el uso de Cloud mediante la selección de la instancia de Amazon EC2 más adecuada para procesar el servicio de predicción de heladas de Sensor Cirrus. Los resultados de las pruebas realizadas con el servicio de predicción de heladas de Sensor Cirrus fueron publicados en [Iacono et al., 2014].

Finalmente, el Capítulo 7 resume los aportes de esta tesis al área de Clouds de Sensores, presenta sus limitaciones y algunas perspectivas para investigaciones futuras.

Capítulo 2

Antecedentes

En este capítulo se introducen las técnicas y conceptos requeridos para la comprensión de la Gestión Remota de WSNs mediante tecnologías de Computación Distribuida. En la Sección 2.1 se detallan las características y protocolos de comunicación de las WSNs. En la Sección 2.2, se presentan los paradigmas de Grid y Cloud Computing. Posteriormente, en la Sección 2.3, se discute el problema de la integración de WSNs a Grid y Cloud Computing. Finalmente, la Sección 2.4 concluye el presente capítulo.

2.1. Redes de Sensores Inalámbricas

En los últimos años el surgimiento de dispositivos compuestos por sensores de distinto tipo, microcontrolador, memoria e interfaz inalámbrica han reemplazado en muchos casos a los sensores tradicionales. Dichos dispositivos, denominados nodos sensores pueden interconectarse, formar redes inalámbricas e interactuar entre ellos. Este tipo de redes denominadas WSNs, permiten estudiar el entorno gracias a la medición y observación de distintos parámetros (temperatura, humedad, presión, etc.) [Akyildiz et al., 2007]. La Figura 2.1 ilustra los principales elementos que componen un nodo sensor.

En una WSN (Ver Figura 2.2), los datos son adquiridos por nodos llamados fuentes y enviados en forma inalámbrica a un nodo especial llamado estación base. La estación base coordina el funcionamiento de la WSN y puede ser una computadora personal (PC) ó un sistema embebido. Además, la estación base almacena o transmite vía Internet los datos registrados por la red. Los nodos deben cumplir con requerimientos tales como autonomía, bajo consumo de energía, bajo costo, robustez y confiabilidad.

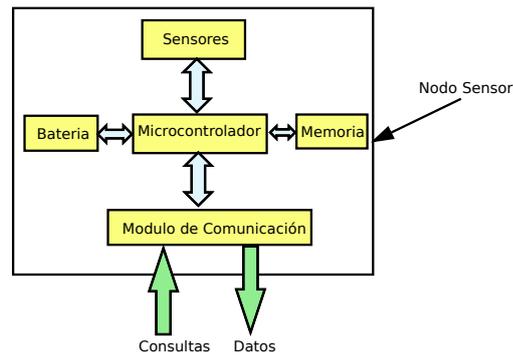


Figura 2.1: Nodo sensor.

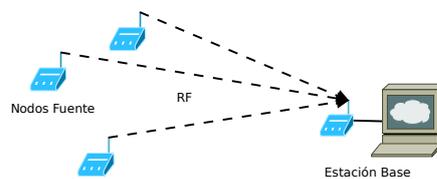


Figura 2.2: Elementos de una WSN.

2.1.1. Características de las WSN

En comparación con los métodos de medición tradicionales (instrumentos, estaciones meteorológicas, etc.), las WSNs ofrecen, además de bajo costo económico, distintos beneficios para aplicaciones que requieran cubrir áreas extensas. A continuación se mencionan las ventajas que ofrecen las WSNs y los requerimientos de diseño que deben considerarse al momento de diseñar este tipo de redes.

2.1.1.1. Ventajas

Debido a la capacidad de formar una red los nodos pueden:

- Detectar eventos: los nodos reportan a la estación base la ocurrencia de un evento específico (por ejemplo: ocurrencia de heladas).
- Mediciones periódicas: los sensores pueden ser programados para reportar periódicamente los valores medidos.
- Aproximar funciones: la forma en como un valor físico varía de un lugar a otro del campo, puede ser considerada como función de la ubicación. Con el objetivo de aproximar la función desconocida (por ejemplo: variación de la temperatura según la ubicación) se pueden utilizar las muestras tomadas por cada nodo de una WSN.

- Localización: en aplicaciones donde la fuente de un evento es móvil (por ejemplo detección y seguimiento de intrusos), la WSN puede ser utilizada para reportar cambios en la posición de la fuente del evento a la estación base.

2.1.1.2. Requerimientos de Diseño

A continuación, se detallan algunos requerimientos a considerar al momento de diseñar WSNs:

- Tipo de servicio: en las WSNs, a diferencia de otras redes de comunicaciones, el servicio no implica solamente trasladar bits [Karl and Willig, 2007]. Además, debe considerarse que en las redes de sensores la información e interpretación de la misma son parte del servicio a brindar, por lo tanto debe considerarse que “espera” la aplicación de la WSN. “La gente busca respuestas no números” (Steven Glaser, UC Berkeley citado en [Huang, 2003]).
- QoS (Quality of Service): los requerimientos de calidad varían de acuerdo a la aplicación. Como ejemplo puede citarse que en aplicaciones de tiempo real el retardo es un factor importante a considerar mientras que en aplicaciones industriales la tasa de pérdida de paquetes debe ser minimizada.
- Tolerancia a fallas: los nodos, pueden presentar diversas fallas: pérdida de alimentación, daños físicos, pérdida del enlace entre dos nodos, etc. La WSN debe ser capaz de soportar las fallas, utilizando mayor cantidad de nodos (redundancia) y redes con topología mesh.
- Tiempo de vida: debido a que los nodos trabajan con fuentes limitadas de energía (baterías) y suelen estar ubicados en el campo (algunas veces a kilómetros del usuario), la WSN debe funcionar minimizando el mantenimiento externo y el consumo de energía. Se deben entonces utilizar técnicas de reducción de consumo tanto en el hardware como en el software de los nodos. Además, en caso de ser necesario, se pueden utilizar fuentes alternativas de energía. En [Morais et al., 2008], se puede observar como los autores utilizan celdas solares y micro generadores hidráulicos en nodos de WSNs aplicadas a vitivinicultura de precisión.
- Escalabilidad: una WSN puede estar conformada por decenas de nodos, por lo que los protocolos y la topología de red a utilizar en la WSN deben ser escalables a esas cantidades.
- Reconfiguración: no sólo es necesario que los nodos procesen información. Además, deben ser capaces de permitir cambios en sus tareas (por ejemplo: variar la fre-



Figura 2.3: Pila de Protocolos WSNs. Figura basada en [Zheng and Jamalipour, 2009].

cuencia de muestreo). Por ello, los nodos deben admitir la posibilidad de realizar cambios en su software durante el funcionamiento de la WSN.

- **Mantenibilidad:** debido a que pueden presentarse cambios en el entorno y la WSN (baterías agotadas, interferencia por obstáculos, fallas en nodos, etc.), el sistema debe garantizar su correcto funcionamiento. Ante la presencia de fallas, la WSN debe cambiar su configuración y negociar entre diversos aspectos. Como ejemplo puede citarse que se puede disminuir la calidad del servicio proporcionalmente a la disminución de la carga de las baterías.

2.1.2. Protocolos de WSNs

A diferencia de las redes inalámbricas tradicionales, las WSNs están integradas por nodos con recursos acotados de hardware y fuente de energía. Debido a este motivo, los nodos utilizan para comunicarse protocolos diseñados específicamente para utilizar poca memoria, ahorrar energía y procesamiento.

Con el fin de minimizar el consumo de energía, los protocolos de WSNs deben evitar algunas de las siguientes fuentes de consumo de energía implicadas en la comunicación entre nodos: colisiones, escucha ociosa, overhearing (escuchar paquetes destinados a otros nodos) y sobrecarga [Gholamzadeh and Nabovati, 2008].

Otra diferencia que presentan las WSNs respecto a las redes inalámbricas tradicionales es que los protocolos de comunicación utilizados en WSNs no son compatibles con los protocolos de la pila TCP/IP.

La mayoría de las WSNs, tienen protocolos que pueden encontrarse en alguna de las capas del siguiente modelo análogo al modelo TCP/IP (Ver Figura 2.3).

- **Capa física.** En esta capa se convierten los datos digitales, provenientes de la capa de enlace de datos, a señales analógicas que pueden ser transmitidas en forma

inalámbrica.

- Capa de enlace de datos (DLL) y subcapa MAC. La capa de enlace de datos permite establecer enlaces de comunicación directos (sin repetidores) y confiables entre nodos. Esta capa es la encargada de gestionar el acceso de varios nodos a un medio de transmisión compartido mediante la subcapa de control de acceso al medio (MAC).
- Capa de red. La principal tarea de la capa de red es la de establecer rutas dentro de la WSN, que permitan que los datos recolectados por los nodos sean entregados a la estación base.
- Capa de transporte. La capa de transporte es la responsable de garantizar la entrega confiable de datos entre los nodos sensores y la estación base.
- Capa de aplicación. En la capa de aplicación se incluyen aplicaciones que permiten al usuario disponer de la información de la WSN y realizar distintas acciones de control ó cambios de configuración en la WSN. Como ejemplo de aplicaciones pueden citarse agro-meteorología, prevención de incendios y domótica.

2.1.2.1. IEEE 802.15.4

Existen distintos protocolos de capa física y subcapa MAC para redes inalámbricas. Sin embargo, el estándar IEEE 802.15.4 [IEEE, 2006] se ha posicionado como uno de los más populares en el área de las WSNs. La popularidad de IEEE 802.15.4 se debe, entre otros motivos, a que es sencillo de implementar y cumple ampliamente con los requerimientos de funcionamiento y diseño de las WSNs. Además, en los últimos años, ha aumentado la cantidad de fabricantes que proveen hardware compatible con IEEE 802.15.4. Por los motivos expuestos, en el presente trabajo sólo se analizará el protocolo IEEE 802.15.4.

El Institute of Electrical and Electronics Engineers (IEEE) finalizó el estándar IEEE 802.15.4 en Octubre del 2003. IEEE 802.15.4 abarca la capa física y la subcapa MAC del modelo TCP/IP, para Wireless Personal Area Networks (WPANs) de baja tasa de transferencia de datos (Ver Figura 2.4). El estándar brinda soluciones a aplicaciones de WSNs, domótica, monitoreo de salud, entre otras.

IEEE 802.15.4 admite topologías de red estrella y punto a punto IEEE [2006]. Además, el protocolo define dos tipos de dispositivos o nodos: de función completa (Full Function Device) y de función reducida (Reduced Function Device).

- Full Function Device (FFD). Puede operar en tres roles diferentes: coordinador de PAN (Personal Area Network), coordinador simple (sólo coordina un cluster y no toda la PAN) o nodo sensor.

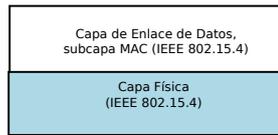


Figura 2.4: Pila del Protocolo IEEE 802.15.4.

- **Reduced Function Device (RFD).** En IEEE 802.15.4 los RFDs sólo pueden operar como nodos sensores.

2.1.2.2. Protocolos de Red

Diversos protocolos han sido desarrollados para utilizar las plataformas de hardware basadas en IEEE 802.15.4. Los protocolos de red proporcionan por sobre las capas definidas por IEEE 802.15.4: topologías del tipo malla (de ahora en adelante mesh), servicios de seguridad y soporte a la aplicación. Sin embargo debe considerarse que el aumento de funcionalidades es proporcional al consumo de energía y al uso de recursos de hardware (memoria, procesamiento, etc.). Por lo tanto, para seleccionar un protocolo de red para WSN, debe encontrarse el balance óptimo entre funcionalidades, uso de recursos de hardware y consumo de fuente de energía.

ZigBee Desarrollado por la ZigBee Alliance [ZigBee Alliance, 2006], ZigBee es un protocolo de comunicaciones diseñado para reducir costos y consumo de energía en dispositivos que necesiten comunicarse inalámbricamente. ZigBee, es estudiado y analizado por diversos autores [Eady, 2007, Kinney, 2003, Baronti et al., 2007, Ergen, 2004], por lo que sólo se mencionarán algunas características principales de este estándar.

ZigBee utiliza los servicios de 802.15.4 y añade por sobre este, la capa de red (Ver Figura 2.5). Gracias a la capa de red, se pueden utilizar topologías Mesh, Estrella (Star) y Árbol (Tree) [ZigBee Alliance, 2006]. Además, ZigBee proporciona la subcapa de soporte de la aplicación (APS) y servicios de seguridad. Algunas de sus aplicaciones son domótica, redes de sensores, monitoreo de salud, etc.

ZigBee utiliza tres tipos de dispositivos: Coordinador, Router y End-Device. El Coordinador y el Router son FFD de IEEE 802.15.4 y el End Device es un RFD de IEEE 802.15.4. En una red ZigBee pueden existir varios End Devices y Routers, pero solo un Coordinador.

Wireless Hart Wireless Hart [Hart Foundation, 2011], surge como una opción inalámbrica al protocolo industrial Highway Addressable Remote Transducer (HART). Wire-



Figura 2.5: Pila del Protocolo ZigBee.

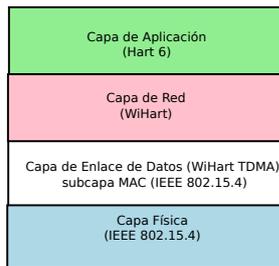


Figura 2.6: Pila del protocolo Wireless Hart.

lessHART utiliza los servicios de la capa física y subcapa MAC de IEEE 802.15.4 para la banda de frecuencia Industrial Científica y Médica (ISM) de 2.4 GHz.

A diferencia de ZigBee, en WirelessHart no existen los RFD. La topología de la red es Mesh y cualquier dispositivo puede actuar como nodo fuente o como router. Por ello, WirelessHart garantiza la entrega de datos inclusive en caso de que la ruta se vea interrumpida por cualquier eventualidad (por ejemplo ruido eléctrico). Además la capa de enlace de datos (ver Figura 2.6) soporta gestión dinámica del ancho de banda utilizando TDMA, por lo que cada dispositivo cuenta con un intervalo asignado de ancho de banda.

6lowPAN IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) [Bormann et al., 2006] ha sido desarrollado por el Internet Engineering Task Force (IETF) y tiene como principal objetivo lograr la compatibilidad entre dispositivos basados en IEEE 802.15.4 y redes TCP/IP. 6lowPAN define la implementación del stack IPv6 sobre las capas física y subcapa MAC de IEEE 802.15.4.

El principal inconveniente de integrar IPv6 y IEEE 802.15.4 es que IPv6 define una unidad máxima de transmisión (MTU) de 1280 bytes, mientras que IEEE 802.15.4 permite solo 127 bytes para todo el paquete (incluido el encabezado y los datos). Para solucionar la diferencia de tamaños entre las tramas, 6lowPAN define una capa de adaptación que utiliza técnicas de fragmentación y reconstrucción para transmitir IPv6 sobre IEEE802.15.4. En la Figura 2.7 puede observarse la pila del protocolo 6lowPAN.

Cabe destacar que si bien 6lowPAN provee compatibilidad con IPv6, los nodos no pueden



Figura 2.7: Pila del protocolo 6lowPAN.

ser conectados directamente a Internet, debe existir en la red un dispositivo llamado router de borde o “edge router” [Shelby and Bormann, 2010] que implemente la última etapa de la adaptación IEEE 802.15.4 - IPv6. El protocolo 6LowPAN ha sido ampliamente analizado y puede consultarse en las siguientes referencias [Shelby and Bormann, 2010, Kushalnagar et al., 2005, Ma and Luo].

ISA-100.11a El estándar ISA-100.11a [International Society of Automation (ISA), 2009], ha sido desarrollado por la International Society of Automation (ISA). ISA-100.11a tiene como principal objetivo permitir la implementación de aplicaciones no críticas de monitoreo y control industrial en forma inalámbrica, confiable y segura. Sus características principales son: robustez hacia la interferencia, baja complejidad, bajo consumo e interoperabilidad con redes cableadas ya existentes en planta.

ISA-100.11.a trabaja con radios IEEE 802.15.4 en la banda ISM de 2.4 GHz, utilizando TSCH (Time-Synchronized Channel-Hopping). El uso de TSCH permite evitar interferencias y minimizar el consumo de energía. La principal ventaja del protocolo desarrollado por ISA es la posibilidad de que los Full Function Devices (FFD) puedan funcionar en modo bajo consumo. Respecto a la capa de enlace de datos, ISA-100.11.a, utiliza TDMA con salto de canal (Channel Hopping) y permite topología de red Mesh.

2.2. Grid y Cloud Computing

En esta sección se presentan las tecnologías de computación distribuida que pueden ser utilizadas para la gestión remota de WSNs. En la Subsección 2.2.1, se discute el modelo conceptual y el funcionamiento de Grid Computing. Además, se detallan algunas infraestructuras Grid y las características de las aplicaciones que pueden ejecutarse sobre estas infraestructuras. Luego, en la Subsección 2.2.2 se presenta el paradigma de Cloud

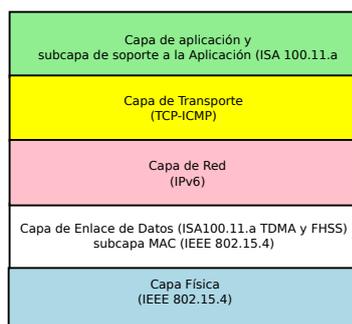


Figura 2.8: Pila del protocolo ISA 100.11.a.

Computing mediante una descripción del modelo de servicios Cloud y los servicios que este paradigma ofrece a los usuarios.

2.2.1. Grid Computing

El paradigma de Grid Computing brinda soluciones a aplicaciones científicas y comerciales que requieren grandes cantidades de recursos computacionales de altas prestaciones.

Como definición de Grid Computing puede citarse la planteada por Foster y Kesselman en [Foster et al., 2001]. Según los autores, el paradigma de Grid Computing propone la interconexión y utilización de recursos computacionales heterogéneos y descentralizados que permiten plantear la noción de una gran computadora virtual con recursos escalables y administración descentralizada.

El término Grid ha sido aplicado a diversos problemas y sistemas: Grid Computacional, Grid de Escritorio, Grid de Datos, Grid de Clusters, etc. Todos estos sistemas buscan, a través de la integración de recursos computacionales, formar recursos mas poderosos. Sin embargo, varían significativamente en distintos aspectos. En [Foster, 2002] Foster propone una lista de control, integrada por tres puntos, que permite en forma sencilla identificar si una infraestructura IT es una Grid o no. Según Foster, una Grid cumple con las siguientes características:

1. Coordina recursos no sujetos a control centralizado.
2. Usa protocolos e interfaces abiertos y de proposito general, basados en estándares.
3. Ofrece calidad de servicio.

Como ejemplo de utilización de la lista de Foster puede mencionarse a Internet. Si bien Internet permite la comunicación entre máquinas y el intercambio de información, no es una Grid debido a que no implica el uso coordinado de recursos en múltiples sitios.

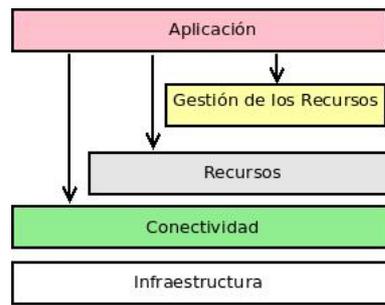


Figura 2.9: Modelo Conceptual de Grid Computing. Figura basada en [Foster et al., 2001].

2.2.1.1. Modelo Conceptual de Grid Computing

El modelo conceptual de Grid (Ver Figura 2.9) fue originalmente propuesto por Foster, Kesselman y Tuecke en [Foster et al., 2001]. El modelo describe los requerimientos generales para los componentes de la Grid. Los componentes de cada capa tienen características en común y utilizan los servicios proporcionados por las capas inferiores [Foster et al., 2001].

A continuación se detallan las capas del modelo conceptual de Grid Computing y sus respectivos componentes:

- Capa de infraestructura. Los recursos pueden ser: computadoras, clusters, servidores de archivo, instrumentos, sensores, y otros. Además, esta capa incluye la infraestructura y las políticas de administración de la red.
- Capa de conectividad. Brinda los protocolos de comunicación y seguridad necesarios para lograr el intercambio de información entre dispositivos (por ejemplo: DNS, SSL, etc.).
- Capa de recursos. Incluye los protocolos necesarios para controlar y administrar un recurso en particular. La información del recurso comprende la utilización, características técnicas, costo de acceso, etc. Además, se puede controlar el inicio de actividad de un recurso, el progreso de la misma, etc. Algunos de los protocolos y servicios utilizados son GridFTP, GRAM (Grid Resource Allocation Management), GRIS (Grid Resource Information Service).
- Capa de gestión de los recursos. Agrupa protocolos, servicios, Application Programming Interfaces (APIs) y Software Development Kits (SDK). Estos elementos no están asociados con un recurso específico, sino que son de naturaleza global y capturan interacciones entre conjuntos de recursos. Entre estos protocolos y servicios se encuentran: servicios de directorio, planificadores de tareas distribuidos, servicios de seguimiento y diagnóstico de ejecución, etc. La capa de gestión de los

recursos debe garantizar la seguridad de los procesos y arbitrar las demandas de los clientes para garantizar que si se produce la caída de un recurso, *no* peligran el desarrollo del proceso.

- Capa de aplicación. Finalmente la capa de aplicación contiene los protocolos que permiten que la aplicación de usuario acceda a uno o varios niveles de la Grid.

2.2.1.2. Funcionamiento de Grid Computing

Como base para explicar el funcionamiento de Grid Computing, se tomará el conjunto de servicios Open Grid Services Architecture propuesto por Foster et. al. [Foster et al., 2003]. Open Grid Services Architecture (OGSA), define una arquitectura común, estándar, abierta y basada en servicios para aplicaciones de Grid Computing. El objetivo de OGSA es estandarizar prácticamente todos los servicios que se pueden encontrar en aplicaciones que trabajan sobre la Grid (servicios de descubrimiento de recursos, de seguridad, etc.).

En la arquitectura definida en OGSA, se introduce el concepto de servicio Grid, el cual es simplemente una extensión de un servicio web. La diferencia entre ambos servicios, es que el servicio Grid cuenta con un estado que permite seguir su ejecución. La principal ventaja de representar los recursos de la Grid utilizando servicios radica que se facilita la interoperabilidad entre recursos pertenecientes a infraestructuras Grid.

El Open Grid Service Infraestructure (OGSI) brinda una especificación formal y técnica de servicios Grid, los cuales se han implementado en el middleware Globus Toolkit. Globus Toolkit [Foster and Kesselman, 1999, Foster et al., 2001] es una arquitectura compuesta por servicios y librerías de software de código libre diseñada para brindar soporte a aplicaciones e infraestructuras Grid [Foster et al., 2003]. Los componentes principales de Globus son: Grid Resource Allocation and Management (GRAM), Monitoring and Discovery System (MDS) y Grid Security Infraestructure (GSI).

A continuación se explican los componentes de Globus mencionados y su interacción, la cuál permitirá finalmente comprender el funcionamiento de la Grid.

- GRAM: brinda al usuario la posibilidad de enviar, monitorear y cancelar trabajos remotos en recursos que integran la Grid.
- MDS: permite el descubrimiento y acceso tanto a la configuración como a información del recurso (capacidad de procesamiento, ancho de banda de la red, tipo de almacenamiento, etc.).
- GSI: permite que las comunicaciones entre los recursos de la Grid se den en un entorno seguro garantizando la autenticación de usuarios y la confidencialidad e integridad de datos.

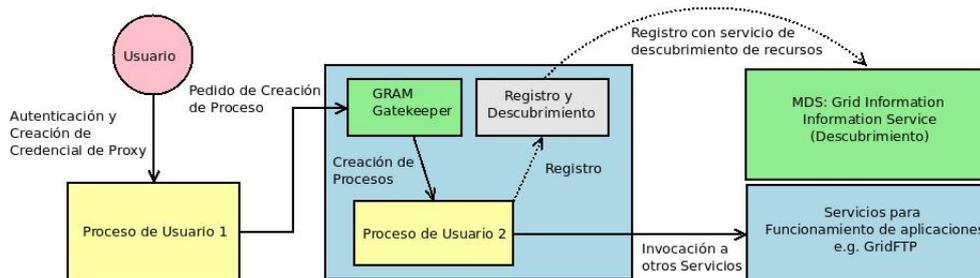


Figura 2.10: Funcionamiento de la Grid. Figura basada en Foster et al. [2003].

Con el fin de ejemplificar el funcionamiento de Grid Computing, en la Figura 2.10 se muestra la interacción de un usuario con los servicios de Globus Toolkit.

El proceso de interacción del usuario con Globus es el siguiente:

1. El "Proceso de Usuario 1" comienza cuando el usuario se autentica. En la autenticación, GSI crea y proporciona al usuario una credencial de proxy [Foster et al., 2003]. Luego, la credencial será utilizada por cualquier programa que necesite invocar los servicios remotos de la Grid en nombre del usuario.
2. Una vez finalizada la autenticación, se solicita la creación de un proceso a través del componente "Gatekeeper" de GRAM. El resultado obtenido, es la creación del "Proceso de Usuario 2" y su correspondiente credencial de proxy.
3. El "Proceso de Usuario 2" se registra y le solicita a MDS servicios de descubrimiento de recursos.
4. Finalizado el "Proceso de Usuario 2" los programas, que actúan en nombre del usuario, invocan los servicios que permiten ejecutar la aplicación del usuario (por ejemplo GridFTP para manejo de archivos).

2.2.1.3. Ejemplos de Infraestructuras Grid

Actualmente hay varias infraestructuras Grid en funcionamiento, distribuidas geográficamente en distintos países del mundo y en muchos casos interconectadas. En la presente sección se detallarán los objetivos, plataformas de hardware y middlewares de algunas de las infraestructuras Grid existentes.

DEGISCO Desktop Grids for Scientific Collaboration (DEGISCO) es un proyecto destinado a solucionar las necesidades de capacidad de cómputo de distintas comunidades científicas, mediante el desarrollo e implementación de una Grid de computadoras tipo

PC. Las máquinas que integran DEGISCO, pertenecen a distintos voluntarios conectados a infraestructuras Grid ya existentes (EGEE, SEE-GRID, etc.).

El proyecto DEGISCO está integrado por instituciones pertenecientes a Brasil y a distintos países de Europa y Asia. Actualmente la infraestructura Grid cuenta con alrededor de 180000 PCs registradas. Mediante la tecnología Generic Grid-Grid (3G Bridge), se logra interconectar las distintas infraestructuras Grid a las que pertenecen las PCs, permitiendo que el usuario los acceda utilizando gLite.

EELA y EELA-2 E-Infraestructura compartida entre Europa y Latinoamérica (EELA), tiene como principal objetivo desarrollar una infraestructura Grid que permita “construir un puente digital entre las iniciativas de e-Infraestructuras existentes, las que están en proceso de consolidación en Europa, y aquellas que están emergiendo en América Latina”.

EELA brinda soporte para el desarrollo y puesta en marcha de aplicaciones de Medicina, Física de Alta Energía, e-Learning y Clima. Luego, en base a los resultados obtenidos con EELA, surge EELA-2. El principal objetivo de EELA-2 es consolidar la e-Infraestructura EELA. El proyecto esta constituido por 78 instituciones de 16 países, 5 europeos (Francia, Irlanda, Italia, Portugal y España) y 11 de Latinoamérica (Argentina, Brazil, Chile, Colombia, Cuba, Ecuador, Mexico, Panama, Peru, Uruguay y Venezuela).

EELA-2 esta integrada por 29 centros de recursos, con un total de 6100 CPUs y 560 TB de almacenamiento. Respecto al middleware Grid, EELA-2 utiliza los servicios de gLite y OurGrid. Debido a que la Grid esta integrada por nodos que cuentan con ambos middlewares, el proyecto EELA-2 incluye entre sus prioridades la de lograr la interoperabilidad entre OurGrid y gLite.

GISELA Grid Initiatives for e-Science Virtual Communities in Europe and Latin America (GISELA), es un proyecto de Grid que surge a partir de EELA y EELA-2. GISELA permite que distintas comunidades científicas puedan ejecutar aplicaciones que requieran gran cantidad de recursos de cómputo y almacenamiento gracias a la integración de infraestructuras Grid de países de Latinoamérica y Europa..

En abril del 2012 GISELA contaba con aproximadamente 1233 CPU y 63 TB de almacenamiento distribuido en 21 centros de cómputo gestionados utilizando gLite. Actualmente, el proyecto tiene como objetivo el llegar a integrar 2660 CPU, 105 TB y 56 centros de cómputos.

TERA GRID-XSEDE TeraGrid es una infraestructura Grid fundada en Estados Unidos en 2001. La infraestructura de TeraGrid, está integrada por computadoras de alto rendimiento.

to y recursos de almacenamiento. El proyecto finalizaba originalmente en 2004, pero fue extendido hasta 2011. En mayo de 2011 se aprobó el proyecto XSEDE como continuación de TeraGrid. Ambos proyectos están destinados a científicos e ingenieros que necesiten recursos Grid para dar soporte a la ejecución de sus aplicaciones. XSEDE está formada por 17 instituciones que proveen 16 máquinas multiprocesador tipo Symmetric Multiprocessing (SMP), Massively Parallel Computer (MPP) y Clusters.

TeraGrid a fines del 2009 contaba con alrededor de 2 petaFLOPS de capacidad de cómputo y más de 60 PB de espacio de almacenamiento interconectados con un enlace de fibra óptica de 10 Gb/s.

Entre los recursos con que cuenta la infraestructura, se puede citar como ejemplo a Blacklight, la cual se encuentra en el Pittsburgh Supercomputing Center y es del tipo SMP. Blacklight cuenta con dos nodos con 256 procesadores Intel Xeon X7560 de 8 núcleos cada uno y 150 TB de almacenamiento.

TeraGrid tiene arquitectura orientada a servicios. Los recursos son gestionados utilizando los distintos programas incluidos en un framework llamado Coordinated TeraGrid Software and Services (CTSS). CTSS incluye entre otros paquetes Globus Toolkit, Condor y Condor-G.

NorduGrid y NDGF El proyecto NorduGrid tiene como objetivo principal crear una infraestructura Grid en los países Nórdicos (Dinamarca, Noruega, Suecia y Finlandia), destinada mayormente a brindar soporte a aplicaciones de física de alta energía. En 2002, el proyecto sirvió como base para el proyecto Nordic Data Grid Facility (NDGF).

Desde 2006, NorduGrid y NDGF contribuyen a Nordic Grid Infrastructure. NorduGrid contribuye con el middleware ARC y NDGF con los recursos. Al mes de marzo del 2012, Nordic Grid Infrastructure contaba con aproximadamente 25000 cpus y 5,7 PB de almacenamiento.

A comienzos del proyecto NorduGrid se utilizaba Globus, posteriormente y tomando como base algunos componentes de Globus se desarrolló el middleware ARC (Advanced Resource Connector). Actualmente, ARC es el middleware Grid utilizado en Nordic Grid.

En la Tabla 2.1, se realiza un resumen de las infraestructuras Grid analizadas en esta sección.

2.2.1.4. Aplicaciones de Grid Computing

A continuación se detallan las características que tienen en común las aplicaciones que maximizan su rendimiento al ser ejecutadas en infraestructuras Grid:

Tabla 2.1: Ejemplos de infraestructuras Grid.

Infraestructuras Grid	Ambito Geográfico	Recursos	Middleware
DEGISCO	Europa, Asia y Brazil.	Alrededor de 180000 PCs registradas.	3G Bridge y gLite.
EELA y EELA-2	Latinoamérica y Europa.	6100 CPUs y 560 TB de almacenamiento.	OurGrid y gLite.
GISELA	Latinoamérica y Europa.	1233 CPU y 63 TB de almacenamiento. (a abril del 2011).	gLite.
TeraGrid-XSEDE	USA.	Alrededor de 2 petaFLOPS de capacidad de cómputo y mas de 60 PB de espacio de almacenamiento (a fines de 2009).	Condor-G.
NorduGrid y NGDF	Dinamarca, Noruega, Suecia y Finlandia.	25000 cpus y 5,7 PB de almacenamiento (a marzo de 2012).	ARC (basado en Globus).

1. Procesos débilmente acoplados. Una aplicación que cumple con esta característica está compuesta por procesos que al ejecutarse no requieren gran intercambio de información entre ellos. Por lo tanto, las aplicaciones compuestas por procesos débilmente acoplados “se adaptan mejor a la forma cambiante de los recursos Grid subyacentes” [Foster and Kessellman, 2011].
2. Implican el uso de cómputación de alta disponibilidad y están compuestas por gran cantidad de computos y tareas. La infraestructura Grid les proporciona a dichas aplicaciones recursos de cómputo y almacenamiento para ejecutar sus tareas.
3. Operación remota y acceso a los datos de instrumentos, sensores, etc. Gran cantidad de aplicaciones científicas implican la operación remota y el análisis de datos de instrumentos, satélites, sensores, etc. [Foster and Kessellman, 2011]. Estas aplicaciones pueden ser ejecutadas sobre la Grid y acceder tanto a los datos como a la configuración de distintos dispositivos de medición.

Algunas de las aplicaciones que cumplen con las características antes mencionadas son: Física de Partículas de Alta Energía [Lamanna, 2004, Jin, 2005], Astronomía [Anderson et al., 2002] y Estudios Paramétricos [Catania et al., 2008].

2.2.2. Cloud Computing

Cloud Computing, proporciona a los usuarios un nuevo paradigma para el desarrollo de aplicaciones y la utilización de recursos de cómputo y almacenamiento. Gracias al uso de virtualización y servicios web, los recursos de hardware y las aplicaciones pueden ser provistas dinámicamente al usuario. Como definiciones de Cloud Computing pueden citarse las siguientes:

- Buyya et al. [Buyya et al., 2009] definen a Cloud Computing en términos de utilidad por parte de los usuarios: *“Un Cloud es un sistema de computación distribuida orientado al mercado que consiste en un conjunto de computadoras interconectadas y virtualizadas. Dicho sistema se provee y presenta en forma dinámica como uno o más recursos de cómputo unificado (s) en base a acuerdos de servicio establecidos mediante negociaciones entre el proveedor y los consumidores de servicios”*.
- Según Armbrust et al. [Armbrust et al., 2010]: *“Cloud Computing se refiere tanto a las aplicaciones ofrecidas como servicios a través de Internet como al hardware y software de los datacenters que proveen estos servicios”*.
- Por otro lado Foster et al. [Foster et al., 2008] presentan la siguiente definición de Cloud: *“Es un paradigma de computación distribuida a gran escala, impulsado por grandes economías, en el que un grupo abstracto, virtualizado y dinámicamente escalable de servicios de almacenamiento, plataforma y cómputo es entregado bajo demanda a clientes externos a través de Internet”*.

Una de las principales ventajas de Cloud Computing es la escalabilidad de recursos, la cual permite solucionar los requerimientos de potencia de cómputo y espacio de almacenamiento de las aplicaciones. Además, los usuarios pueden acceder en forma sencilla a plataformas de desarrollo de aplicaciones que utilizan distintos servicios web para escalar y acceder a los recursos necesarios para la ejecución de la aplicación sobre infraestructuras Cloud.

En la presente Sección, se describirá el paradigma de Cloud. Además, se presentarán y analizarán los distintos servicios que ofrece Cloud.

2.2.2.1. Modelo de Servicios Cloud

Cloud Computing ofrece a los usuarios (clientes) tres tipos de servicios: Infraestructura, Plataforma y Aplicación (Ver Figura 2.11), los cuales son generados por distintos proveedores y consumidos por clientes.

A continuación se detallan cada uno de los tipos de servicios ofrecidos por Cloud Computing:

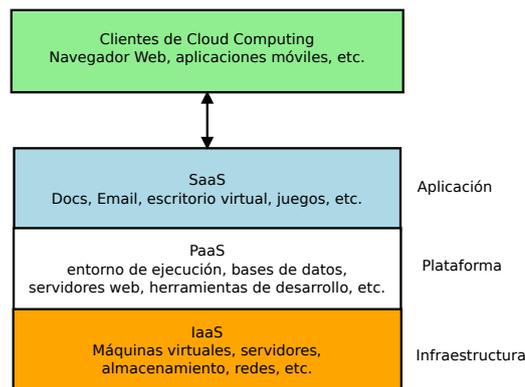


Figura 2.11: Servicios Cloud.

1. Servicios de Infraestructura o “Infrastructure as a Service” (IaaS): son servicios que ofrecen al usuario acceso a recursos computacionales de altas prestaciones virtualizados (CPU’s, dispositivos de almacenamiento, etc.). En los servicios de infraestructura el proveedor entrega al cliente recursos acordes a los requerimientos específicos solicitados: tipo y potencia de CPU, cantidad de memoria, espacio de almacenamiento y tipo de sistema operativo, entre otros.

Los recursos computacionales virtualizados se agrupan en infraestructuras denominadas Clouds. Un Cloud puede ser público, privado ó híbrido. Un Cloud es público si el usuario accede a sus servicios mediante “pago por consumo” (pay-as-you-go) [Armbrust et al., 2010]. Por otro lado, se denomina Cloud Privado al que pertenece a una organización ó empresa que lo utiliza internamente y no permite el acceso público al Cloud. Finalmente un Cloud híbrido, como lo sugiere su nombre, es una infraestructura que permite acceso a recursos tanto privados como públicos.

2. Servicios de Plataforma o “Platform as a Service” (PaaS): proveen interfaces de programación de aplicaciones (APIs) y entornos de desarrollo para que los usuarios desarrollen e implementen sus propias aplicaciones para Cloud. Además, en algunos casos los proveedores de servicios de plataforma también son de servicios de infraestructura y suministran, además del entorno de desarrollo, recursos de hardware virtualizados para que el cliente ejecute sus aplicaciones desarrolladas con servicios de PaaS.
3. Servicios de Aplicación o “Software as a Service” (SaaS): incluyen aplicaciones que pueden ser accedidas por un usuario final. Dichas aplicaciones pueden ser desarrolladas con PaaS y ejecutadas mediante IaaS.

Según Armbrust y Fox [Armbrust et al., 2010], el concepto de SaaS presenta las siguientes ventajas tanto para el usuario como para el proveedor del servicio:

- No es necesario que los proveedores de servicios sean propietarios de un Cloud. Las empresas de desarrollos de aplicaciones pueden pagar por el uso de infraestructuras existentes y disponibles.
- Los proveedores de servicios centralizan la instalación, mantenimiento y actualización de versiones de software.
- El servicio puede ser accedido por los usuarios “cuando y donde” quieran. Además, el usuario no debe preocuparse por el almacenamiento de sus datos.
- Los proveedores de Cloud son los responsables del mantenimiento y la calidad de servicio de la infraestructura IT, por lo que los usuarios y proveedores de SaaS no deben preocuparse por el mantenimiento del Cloud.

El paradigma de Cloud Computing ha sido tratado en forma detallada por distintos autores y puede consultarse en las siguientes referencias [Armbrust et al., 2010, Buyya et al., 2009, Ograph and Morgens, 2008, Foster et al., 2008, Pacini et al., 2012]

2.2.2.2. Servicios de Infraestructura y Plataforma

Amazon EC2. En 2006 Amazon lanzó la plataforma Amazon Elastic Compute Cloud (EC2) [Amazon EC2, 2012]. EC2 permite que el cliente desarrolle mediante APIs, servicios web para procesar aplicaciones con recursos de hardware de centros de datos de Amazon mediante máquinas virtuales (denominadas instancias).

Los servicios de EC2 brindan escalabilidad a las aplicaciones ya que el usuario puede crear, lanzar y finalizar instancias según sus necesidades. Respecto al costo de uso, el usuario solo paga por las horas en las que el servicio estuvo activo.

Además, la plataforma EC2 brinda acceso a máquinas físicas que proveen altos niveles de redundancia y permiten optimizar latencias debido a la ubicación geográfica. Actualmente, Amazon ofrece distintos tipos de instancias, acordes a las necesidades de CPU, memoria y almacenamiento del usuario.

Google App Engine. Google AppEngine [Google, 2013a] es una plataforma que permite el desarrollo de aplicaciones web que pueden luego ser ejecutadas sobre el Cloud de Google. App Engine provee un conjunto de APIs y modelos de aplicaciones que permiten que el usuario utilice otros servicios provistos por Google como correo electrónico, almacenamiento de datos, etc. Las APIs y modelos de AppEngine permiten la creación de aplicaciones en Java, Python y JRuby.

Las aplicaciones desarrolladas por los usuarios funcionan sobre una capa denominada SandBox. La principal función de SandBox es la de abstraer al usuario y a su aplicación

de los recursos de hardware subyacentes. Además, AppEngine provee servicios de escalabilidad que permiten aumentar la potencia de cómputo de acuerdo a las necesidades de la aplicación.

Windows Azure. Window Azure [Microsoft, 2013] tiene como principal objetivo el brindar a los usuarios un entorno de desarrollo y ejecución de aplicaciones Cloud. Gracias a Windows Azure, los usuarios pueden desarrollar aplicaciones que son luego ejecutadas en los centros de datos de Microsoft. Además, el entorno de desarrollo es multiplataforma ya que permite trabajar con máquinas virtuales Windows y Linux.

Entre otras, pueden desarrollarse aplicaciones web, móviles, de bases de datos y multimedia. Otra de las características de Azure es la posibilidad de escalar la potencia de cómputo y capacidad de almacenamiento en base a las necesidades del usuario. Además la plataforma permite la integración de aplicaciones ya desarrolladas para otros Clouds públicos.

En la Tabla 2.2, se puede observar una comparación de las características principales de Amazon EC2, Google App Engine y Windows Azure.

2.2.2.3. Servicios de Aplicación

Google Docs. El servicio de aplicación Cloud Google Docs [Google, 2013b], permite la creación y edición en línea de: documentos de texto, presentaciones, hojas de cálculo, y formularios de encuesta. Los documentos son almacenados en el Cloud de Google por intermedio de Google Drive y pueden ser compartidos con otros usuarios que tengan cuentas en Google.

Si bien originalmente el servicio tenía como principal objetivo la creación de documentos, actualmente ha sido totalmente integrado a Google Drive y permite almacenar todo tipo de documentos en el Cloud. Además, los documentos pueden ser editados y accedidos por el usuario desde dispositivos móviles con sistema operativo Android.

El servicio también admite la capacidad de sincronizar archivos con la PC y almacenamiento gratuito de hasta 15 GB.

Globus Online. Desarrollado por el Computation Institute de la Universidad de Chicago y el Argonne National Laboratory, Globus Online (GO) [Foster, 2011] tiene como principal objetivo el proveer servicios SaaS que faciliten la transferencia entre usuarios de grandes volúmenes de datos [Allen et al., 2012]. Gracias a esta posibilidad, los usuarios de Grid, pueden compartir archivos sin lidiar con la complejidad de instalación del middleware Grid y la configuración de la infraestructura IT. Los servicios que provee GO son Globus Transfer y Globus Nexus.

Tabla 2.2: Comparación de servicios IaaS y PaaS en Cloud Computing.

	Amazon EC2	Google App Engine	Windows Azure
Servicios Cloud	IaaS y PaaS	PaaS	PaaS
Plataformas Soportadas	Todas las distribuciones de Linux, Windows Server, Oracle Enterprise, IBM DB2 y Informix Dynamic Server, Microsoft SQL Server, Enterprise Oracle Database, Hadoop, Condor, OpenMPI, JBoss Enterprise, Application Platform, Ruby on Rails, IBM WebSphere Application Server, Java Application Server	Entornos de ejecución tales como Máquina Virtual Java e Intérprete de Python. Permite la integración con cuentas de Google, gestión de tareas, etc. No se pueden ejecutar aplicaciones desarrolladas con software externo	Linux, Windows Server, Windows 7, Windows Vista
Lenguajes de Programación Soportados	Cualquiera	Java, Python, JRuby	VB.NET, C# y PHP
Servicios Cloud y herramientas	Amazon CloudWatch API Tools, Auto Scaling API Tools , Elastic Load Balancing API Tools, AWS Toolkit for Eclipse, AWS Management Console, Amazon EC2 API Tools, Amazon EC2 AMI Tools, Elasticfox Firefox Extension for Amazon EC2, Javascript Scratchpad for Amazon EC2, Amazon S3 Authentication Tool for Curl CloudBerry, Explorer for Amazon S3 and CloudFront, Manager for Amazon CloudFront Firefox, Organizer for Amazon S3 and Amazon CloudFront (S3Fox), AWSzone.com, Javascript Scratchpad for Amazon SQS, Amazon Mechanical Turk Developer Sandbox, Amazon Mechanical Turk Command Line Tools, y otros.	Google Secure Data Connector, Private gadgets, Google Visualization API, Google Apps APIs, Google web toolkit, IDE support	Windows Azure Platform Training Kit, Windows Azure Software Development Kit, Microsoft Visual Studio 2008 Service Pack 1, Windows Azure platform AppFabric SDK V1.0, Windows 7 Training Kit For Developers
Panel de Control	Interfaz Web	Interfaz Web	Interfaz Web

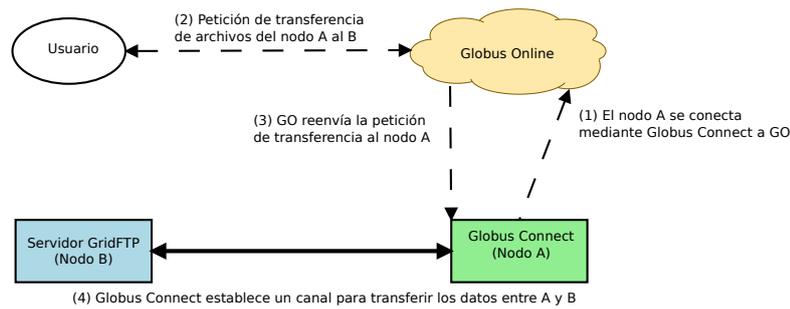


Figura 2.12: Transferencia de archivos en Globus Online. Figura basada en [Foster et al., 2003, 2012].

Globus Transfer es un servicio que simplifica el proceso de intercambio seguro de datos. El servicio utiliza GridFTP y técnicas de SaaS para permitir en forma sencilla, confiable y segura transferencias de datos. El usuario puede acceder al servicio de transferencia de Globus Transfer en tres formas distintas: interfaz web estándar, programas del tipo script ó interfaz de línea de comando. Respecto al proceso de autenticación, Globus Transfer utiliza los servicios de GSI.

El servicio de transferencia de Globus Transfer utiliza dos tipos de nodos: fuente y sumidero. El nodo fuente es la máquina desde donde el usuario desea enviar los datos y el sumidero es la que los recibe. En Globus Online ambos nodos son denominados “endpoints”. Gracias a la utilización de SaaS, el usuario sólo debe instalar en el endpoint una aplicación sencilla denominada Globus Connect y no todo el middleware Globus. En la Figura 2.12, se detalla el proceso de transferencia de archivos utilizado por Globus Online.

Actualmente, el servicio de transferencia de archivos de GO no permite en su versión gratuita la transferencia directa entre endpoints, dicha opción solo está disponible en la versión paga de GO denominada Globus Plus [Globus, 2014]. Motivo por el cual, el proceso requiere que los datos se envíen a un tercer nodo desde el cual serán luego descargados. Este nodo es un servidor GridFTP que puede ser propio del usuario ó de la infraestructura IT de Globus Online. Si el usuario realiza la transferencia con un servidor público de GO, la cantidad de archivos a transferir estará restringida a 10MB.

Por otro lado, Globus Nexus provee servicios de identificación y gestión de grupos. Además, gestiona el uso de los certificados X.509 del usuario, los cuales son requeridos cuando se realiza el acceso remoto a los recursos.

Respecto al uso de Globus Online en infraestructuras Grid existentes, TeraGrid y XSEDE han comenzado a utilizar los servicios SaaS de GO. Además, distintos centros de investigación como el Australian eResearch Organization (ARCS) y el National Center for Supercomputing Applications (NCSA) utilizan los servicios de Globus Online para transferencias de archivos.



Figura 2.13: Arquitectura General Plataforma de Integración WSN - Grid/Cloud.

2.3. Integración de WSNs a Grid y Cloud Computing

Generalmente, las tecnologías Grid y Cloud han sido utilizadas para ejecutar aplicaciones que requieren acceso a recursos de hardware de altas prestaciones y escalables. Sin embargo, en la última década han surgido una serie de propuestas que plantean la posibilidad de gestionar desde Internet la configuración e información de instrumentos y sensores mediante tecnologías Grid y Cloud. Dichas propuestas han permitido incorporar a Grid y Cloud dispositivos que permiten observar y medir el medio físico. Gracias a esta posibilidad, los usuarios de Grid y Cloud pueden validar experimentos y observaciones con datos provenientes del entorno donde están desplegados los instrumentos y sensores.

La integración de WSNs a Grid y Cloud implica dos aspectos a resolver: la incompatibilidad de protocolos entre ambas tecnologías y la utilización de un middleware Grid ó un servicio Cloud para administrar los recursos de las WSNs. Ambos problemas pueden ser solucionados implementando una plataforma de integración WSN - Grid/Cloud adecuada. En la Figura 2.13, se puede observar una arquitectura genérica de una plataforma para la integración de WSNs a Grid y Cloud.

El primer aspecto a resolver por parte de la plataforma es la incompatibilidad de protocolos entre las WSNs y Grid ó Cloud Computing. El inconveniente surge debido a que las WSNs utilizan protocolos de comunicación diseñados para minimizar el consumo de energía (por ejemplo ZigBee) mientras que Grid y Cloud utilizan middlewares y servicios basados en protocolos TCP/IP (por ejemplo GridFTP y Globus Online). Para solucionar esta incompatibilidad se debe implementar en la plataforma un módulo denominado "Módulo WSN - TCP/IP" que actúe a modo de traductor entre los protocolos de las WSNs y TCP/IP.

En la Figura 2.14 se observa el Módulo WSN - TCP/IP de la plataforma de integración WSN - Grid/Cloud. Dicha figura es a modo ilustrativo y no se especifican detalles sobre compatibilización de señales y datos en capa física y superiores.

Actualmente, la incorporación de WSNs a TCP/IP es un tema abierto en el campo de las WSN [Rajesh et al., 2010]. La integración WSN - TCP/IP implica resolver una arquitectura de comunicación que permita compatibilizar los protocolos de ambas tecnologías. Dicha

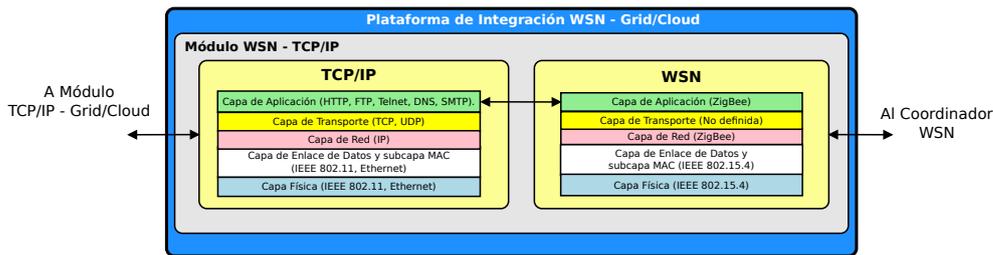


Figura 2.14: Módulo WSN - TCP/IP de la plataforma de integración WSN - Grid/Cloud.

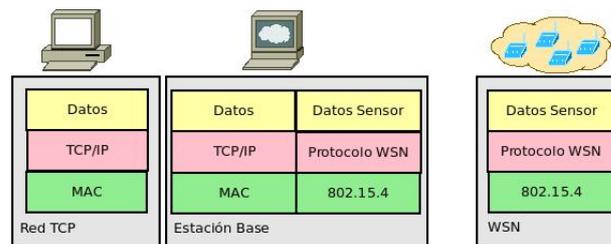


Figura 2.15: Integración por Gateway. Figura basada en [Roman and Lopez, 2009].

arquitectura define cual es el equipo de la WSN que implementará el protocolo TCP/IP y se resuelve mediante alguno de los siguientes enfoques: *gateway* y *redes overlay* [Roman and Lopez, 2009, Zuñiga and Krishnamachari, 2003].

Por un lado, la arquitectura de *gateway* (ver Figura 2.15) define que la estación base tiene dirección IP y actúa como gateway, traduciendo los protocolos de ambas redes (TCP/IP y por ejemplo ZigBee) [Akyildiz et al., 2007, Roman and Lopez, 2009, Zuñiga and Krishnamachari, 2003, Liutkevicius et al., 2010]. El gateway tiene las interfaces físicas y los protocolos de ambas redes, además es el único punto de acceso a la red. Dicho dispositivo permite tomar los datos y comandos de las WSNs en el protocolo nativo y compatibilizarlos con TCP/IP. Luego, por intermedio de alguna red mayor (como Internet) los datos son entregados a la aplicación de gestión remota para ser presentados al usuario. De igual forma, el usuario solicita cambios de configuración a la aplicación para que a través de Internet y TCP/IP lleguen al gateway y finalmente en protocolo nativo a la WSN.

Debido a que no requiere cargas extras de protocolo en los nodos sensores, el enfoque de *gateway* permite trabajar con nodos sensores que cuenten con recursos acotados de cómputo, memoria y fuente de energía.

Por otro lado, el enfoque de integración con redes del tipo *overlay* permite compatibilizar redes con distintos protocolos mediante el solapamiento de uno de los protocolos sobre el otro [Akyildiz et al., 2007, Roman and Lopez, 2009, Zuñiga and Krishnamachari, 2003, Liutkevicius et al., 2010]. En el caso de la integración WSN - TCP/IP, se denomina TCP/IP “*overlay*” Sensor Networks [Roman and Lopez, 2009], debido a que se programa parte o

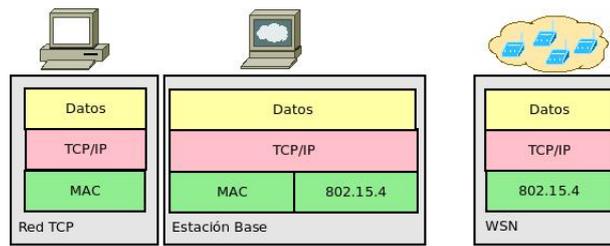


Figura 2.16: Integración mediante redes overlay. Figura basada en Roman and Lopez [2009].

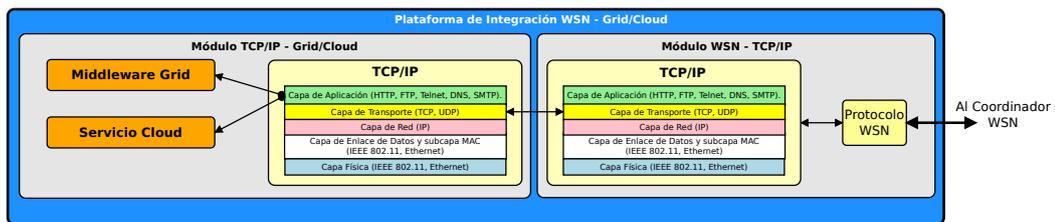


Figura 2.17: Módulos WSN - TCP/IP y TCP/IP - Grid/Cloud de la Plataforma de Integración.

toda la pila TCP/IP en los nodos sensores (Ver Figura 2.16).

Luego del Módulo WSN - TCP/IP, se debe realizar en la plataforma de integración WSN - Grid/Cloud un segundo módulo denominado “Módulo TCP/IP - Grid/Cloud”. El módulo TCP/IP - Grid/Cloud permitirá finalmente que el middleware Grid ó el servicio Cloud administre a la WSN como recurso Grid ó servicio Cloud. El estudio del estado del arte ha permitido detectar que existen dos enfoques para solucionar la integración de WSNs a Grid y Cloud Computing: middleware Grid (ó servicio Cloud) embebido y middleware Grid (ó servicio Cloud) no embebido en la WSN. Ambos enfoques serán estudiados en detalle en el Capítulo 3.

La Figura 2.17, ilustra los Módulos WSN - TCP/IP y TCP/IP - Grid/Cloud de la plataforma de integración WSN - Grid/Cloud.

Finalmente, se deben implementar en la capa de aplicación un módulo de software, en el presente trabajo denominado “Módulo de Interconexión”, que permita a la aplicación del usuario acceder al middleware Grid ó servicio Cloud para gestionar desde Internet los datos y la configuración de las WSNs. En la Figura 2.18, se observa el módulo de interconexión y los dos módulos de la plataforma WSN - Grid/Cloud.

2.4. Conclusiones

En este capítulo se discutieron las técnicas y conceptos necesarios para el desarrollo de la presente tesis doctoral. En primer lugar, se presentaron las redes de sensores inalámbricas. Luego, se analizaron las tecnologías de computación distribuida que pueden ser

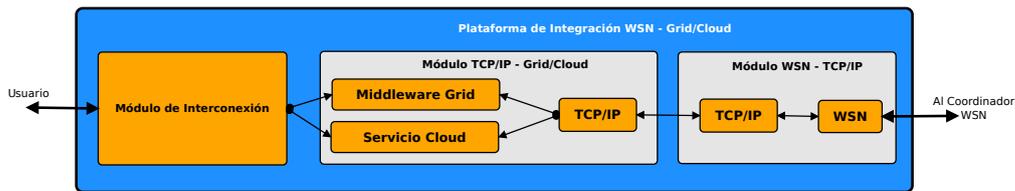


Figura 2.18: Plataforma WSN - Grid/Cloud con Módulo de Interconexión.

utilizadas para procesar y almacenar los datos adquiridos por las WSNs y gestionar la configuración de los nodos sensores. Además, se detalló la problemática de la integración de WSNs a Grid y Cloud. Finalmente, se presentó el fenómeno de las heladas, como afecta a la economía de la Región de Cuyo, Argentina y que medidas se pueden implementar para combatirlas.

Las WSNs permiten acceder a información real del medio físico mediante dispositivos denominados nodos sensores. Los nodos sensores tienen bajo costo económico, mínimo mantenimiento y pueden ser reconfigurados en forma remota. Debido a que las WSNs están desplegadas en campo, generalmente a kilómetros del usuario, es necesario contar con tecnologías que posibiliten la gestión de la información adquirida en campo y la configuración de los nodos sensores desde Internet. Las tecnologías Grid y Cloud permiten gestionar los datos y configuración de WSNs (conformadas por cientos de nodos sensores) mediante infraestructuras compuestas por recursos computacionales de altas prestaciones denominadas Grids y Clouds. Gracias a las tecnologías de computación distribuida, se puede acceder a los recursos de las WSNs y de la Grid ó el Cloud para estudiar distintos fenómenos meteorológicos como tormentas, inundaciones, granizo y heladas (las que se discuten en profundidad en el Capítulo 6).

Por un lado, Grid Computing permite la integración de recursos computacionales heterogéneos y de altas prestaciones para formar recursos más poderosos. Estos recursos permiten cubrir las necesidades de procesamiento y almacenamiento de los usuarios y son administrados con protocolos específicos (por ejemplo GridFTP y GRS). Además, los protocolos de Grid Computing permiten incorporar a las Grids instrumentos de distinto tipo y redes de sensores inalámbricas.

Por otro lado, el paradigma Cloud ofrece nuevas oportunidades para realizar la gestión remota de WSNs. El modelo de servicios Cloud, permite al usuario abstraerse de la administración de la infraestructura necesaria para almacenar y procesar los datos de las WSNs, contar con recursos bajo demanda adecuados a sus necesidades puntuales, y que los recursos estén a su disposición las 24 horas del día los 365 días del año.

Para gestionar las WSNs mediante plataformas Grid ó Cloud, se requieren middlewares que compatibilicen los protocolos de ambas tecnologías y permitan la interacción con

middlewares Grid ó servicios Cloud. Además, la gestión remota debe ser implementada conservando el funcionamiento eficiente de las WSNs.

En el próximo capítulo se analiza el estado del arte de la gestión de WSNs desde Internet. Se estudian trabajos de distintos autores en dos áreas principales: (i) integración de WSNs a TCP/IP y (ii) gestión remota de WSNs mediante tecnologías tradicionales ó de computación distribuida. Además, el capítulo brinda un análisis de las principales ventajas y limitaciones de los enfoques utilizados en cada área, las cuales son las bases para el desarrollo de esta tesis doctoral.

Trabajos Relacionados

En el presente capítulo se presentan trabajos de distintos autores que realizan la gestión remota de WSNs desde Internet. En la Sección 3.1 se analizan trabajos sobre la integración de WSNs a TCP/IP. Luego, en la Sección 3.2 se estudian propuestas que utilizan tecnologías tradicionales ó de computación distribuida para gestionar las WSNs desde Internet. La literatura ha sido clasificada de acuerdo al enfoque que utilizan los autores para resolver la integración WSN - TCP/IP y la gestión remota de WSNs desde Internet. En la Sección 3.3 se resúmen los trabajos previos. Luego, en la Sección 3.4 se discuten las ventajas y desventajas de cada enfoque y otros aspectos a considerar al momento de diseñar plataformas de gestión remota de WSNs. Finalmente en la sección 3.5 se presentan las conclusiones de este capítulo.

3.1. Integración de WSNs a TCP/IP

El principal inconveniente que surge al momento de incorporar las WSNs a TCP/IP es la incompatibilidad de protocolos entre ambas tecnologías. Con el objetivo de solucionar este inconveniente, se debe implementar una plataforma de integración WSN - TCP/IP que permita resolver el problema.

Para desarrollar una plataforma WSN - TCP/IP se debe primero definir cuales de los equipos integrantes de la WSN emplean el protocolo TCP/IP. Luego, surgen dos tipos de arquitecturas de comunicación: *gateway* y *overlay*. En la presente sección, se detallan las características de ambas soluciones y se analizan trabajos previos que integran las WSNs a TCP/IP.

3.1.1. Gateway

Como se mencionó en el Capítulo 2, la arquitectura de gateway define que la estación base implementará el protocolo TCP/IP. Este equipo realiza la traducción de los protocolos del Grid ó el Cloud y de la WSN [Akyildiz et al., 2007, Roman and Lopez, 2009, Zuñiga and Krishnamachari, 2003, Liutkevicius et al., 2010]. A continuación, se analizan diferentes propuestas disponibles en la literatura.

Sensor Base - cSDSP. En Reddy et al. [Reddy et al., 2007], se desarrolló una arquitectura basada en gateway para compartir los datos recolectados y variar la configuración de redes de sensores. Los sensores se comunican con uno o más gateways utilizando el protocolo específico de la red de sensores. Luego, los gateways envían los datos y parámetros de la red de sensores a bases de datos y los indexan en buscadores de sensores. El gateway se implementa sobre una máquina tipo PC y permite capturar datos que son utilizados para estudiar hábitos de aves. Finalmente los datos son enviados a una aplicación desarrollada en forma ad-hoc que permite el acceso desde Internet a los servicios de la WSN.

SYLPH. En [Corchado et al., 2010], los autores desarrollaron un sistema destinado a prestación de asistencia sanitaria y seguimiento de pacientes en sus hogares. La integración a TCP/IP es realizada mediante una arquitectura de gateway. La plataforma desarrollada permite incorporar WSNs ZigBee y Bluetooth a TCP/IP.

En el trabajo los autores conectan, vía USB, el coordinador ZigBee y el nodo maestro Bluetooth al gateway programado en una PC. El gateway recibe los datos provenientes de ambos nodos y mediante una aplicación los compatibiliza con TCP/IP. Finalmente los datos son almacenados en bases de datos y enviados al usuario vía servicios web. La Figura 3.1, muestra las capas de SYLPH.

Annamraju et al. Annamraju, Gumudavelli et. al. [Annamraju et al., 2010], propusieron un middleware que permite la integración de WSNs ZigBee a TCP/IP. El middleware utiliza una plataforma con arquitectura de gateway y es programado en nodos sensores Mobeet y gateways MobeetNet [Mobitrum Corporation, 2010].

El gateway posee un sistema embebido con interfaces físicas ZigBee, WiFi y Ethernet. Además, el gateway permite representar los datos recolectados por la WSN utilizando el estándar de sensores IEEE 1451 [Lee, 2000]. La aplicación desarrollada se basa en los estándares SNMP e IEEE 1451 y se explicará en forma detallada en la Subsección 3.2.1. En la Figura 3.2 se ilustra la arquitectura propuesta en [Annamraju et al., 2010].

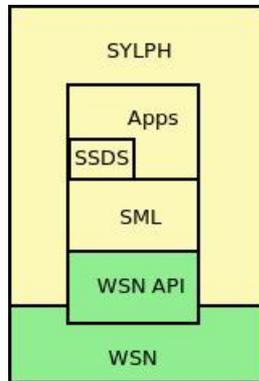


Figura 3.1: Arquitectura en capas de SYLPH. Figura basada en Corchado et al. [2010].

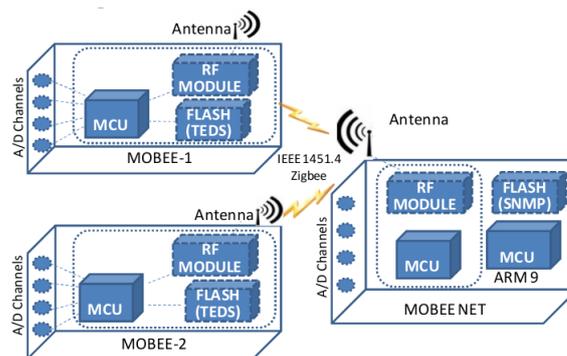


Figura 3.2: Arquitectura desarrollada por Annamraju et al. Figura extraída de [Annamraju et al., 2010].

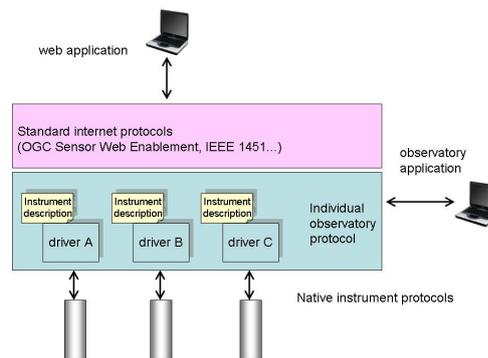


Figura 3.3: Arquitectura de comunicación de Interoperable Test-Bed. Figura extraída de [O’Reilly et al., 2009].

Interoperable Test-Bed. O’Reilly et al. [O’Reilly et al., 2009], demostraron como integrar los datos de una red de sensores compuesta por instrumentos de tres observatorios oceanográficos, para luego accederlos desde Internet. Interoperable Test-Bed está compuesto por instrumentos y sensores oceanográficos heterogéneos tanto en hardware como en protocolos de comunicación. Cada instrumento y sensor está conectado mediante USB, RS232 ó RS485 a una computadora que actúa a modo de gateway.

El proceso de integración de Interoperable Test-Bed, comienza con el almacenamiento de los datos recolectados por los instrumentos en la plataforma. Luego, la plataforma de integración tiene una serie de módulos de software, que permiten convertir los datos de los protocolos propios de cada observatorio a IEEE 1451. Finalmente, los datos se acceden mediante una aplicación que permite al usuario la gestión y uso de la red de instrumentos y sensores. En la Figura 3.3 se puede observar el modelo conceptual de la propuesta.

3.1.2. Redes Overlay

Las arquitecturas de compatibilización basadas en redes overlay realizan la compatibilización de las WSNs y TCP/IP mediante el solapamiento de uno de los protocolos sobre el otro [Akyildiz et al., 2007, Roman and Lopez, 2009, Zuñiga and Krishnamachari, 2003, Liutkevicius et al., 2010]. Como se discutió en el Capítulo 2, para el caso particular de las WSNs (TCP/IP “overlay” Sensor Networks), se debe programar toda la pila TCP/IP ó parte de dicha pila de protocolos, en los nodos sensores de la WSN.

Actualmente existen varias propuestas que utilizan el enfoque overlay para integrar WSNs a TCP/IP. Sin embargo, la mayoría se basan en la implementación desarrollada por Adam Dunkels denominada μ TCP/IP [Dunkels, 2003]. Por lo tanto, se presentará en la presente

subsección el trabajo desarrollado por Dunkels y otras propuestas basadas en μ TCP/IP.

μ TCP/IP. Adam Dunkels [Dunkels, 2003], presentó una optimización de mecanismos utilizados en TCP/IP para permitir que la pila pueda ser programada en nodos sensores compuestos por microcontroladores de 8 bits. La optimización propuesta por Dunkels posibilita que los nodos sensores utilicen una versión reducida de TCP/IP (denominada μ TCP/IP) sin generar grandes consumos de memoria, procesamiento y energía. Las soluciones propuestas por Dunkels en μ TCP/IP son las siguientes:

1. Arquitectura de direccionamiento IP. Debido a que en la mayoría de las WSN los datos generados por los nodos necesitan estar asociados con la ubicación geográfica, μ TCP/IP utiliza una asignación de IP basada en la situación espacial de los nodos. Este método se basa en proveer una dirección IP “*semi*” exclusiva a cada nodo sensor.
2. Sobrecarga en el encabezado. Los encabezados pueden comprimirse utilizando diversas técnicas de compresión, por ejemplo: si todos los nodos son parte de la misma subred IP no hay necesidad de transmitir toda la dirección IP completa en los encabezados de los paquetes que provienen o que están destinados a los nodos de cada WSN.
3. Recursos limitados de hardware. Gracias al desarrollo e implementación de μ TCP/IP, Dunkels demostró que se pueden implementar versiones de la pila TCP/IP en microcontroladores de 8 bits con poca memoria RAM.
4. Eficiencia energética. En μ TCP/IP, se minimiza el consumo de energía en los nodos gracias a un mecanismo que combina almacenamiento en caché de segmentos de datos y retransmisiones locales en cooperación con la capa de enlace.

El trabajo de Dunkels sentó las bases para el desarrollo de tecnologías específicas para WSNs como el sistema operativo Contiki OS [Dunkels et al., 2004] y el protocolo 6lowPAN [Bormann et al., 2006].

SPMIPv6. M. M. Islam et al. [Islam et al., 2010], implementaron un protocolo para WSNs llamado Sensor Proxy Mobile IPv6 (SPMIPv6). Dicho protocolo está basado en Proxy Mobile IPv6 (PMIPv6) [Lee et al., 2008] y en 6lowPAN [Bormann et al., 2006].

PMIPv6 es un protocolo basado en Mobile IPv6 (MIPv6). La principal diferencia entre PMIPv6 y MIPv6 es que PMIPv6 que permite trabajar con nodos móviles IPv6 pero sin la gran carga de señales de control que utiliza MIPv6. PMIPv6 reduce las funcionalidades

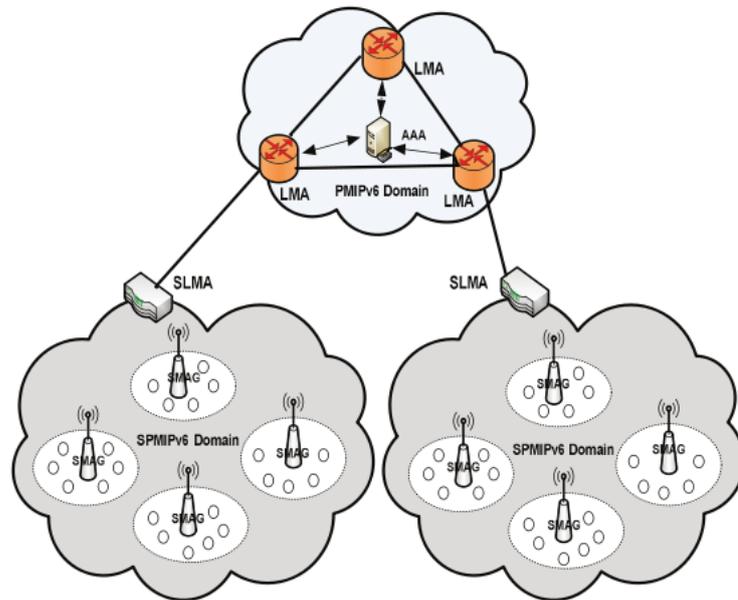


Figura 3.4: Arquitectura de SPMIPv6. Figura extraída de [Islam et al., 2010].

en el nodo móvil, aumentándolas en un nuevo dispositivo de la red llamado MAG (Mobile Access Gateway) [Islam et al., 2010]. SPMIPv6 toma las ventajas de PMIPv6 ya que minimiza las funcionalidades del nodo móvil pero, a diferencia de PMIPv6, el MAG de SPMIPv6 trabaja con 6LoWPAN.

Sensor Mobility Proxy. Otra propuesta de integración de WSN vía redes overlay es la de Silva et al. [Silva et al., 2010]. Los autores desarrollaron un protocolo llamado Sensor Mobility Proxy (SMP) basado en MIPv6. SMP, se implementó en una plataforma que permitió dar solución a los requerimientos específicos de una aplicación de monitoreo de salud: movilidad, baja latencia [Akyol et al., 2010] y confiabilidad.

La propuesta implementa en los nodos móviles el protocolo μ IPv6 y establece comunicación con una red TCP/IP por intermedio de un gateway programado en la estación base. Si bien la plataforma utiliza un gateway, la arquitectura es del tipo overlay debido a que la estación base y cada uno de los nodos sensores tienen dirección IP.

3.2. Gestión de WSNs desde Internet

En la presente sección se analizan el estado del arte de la gestión de WSNs desde Internet. En la Subsección 3.2.1, se detallan enfoques y propuestas basadas en tecnologías

tradicionales de redes, sensores e Internet y en la Subsección 3.2.2, se estudian los enfoques y propuestas basadas en tecnologías de computación distribuida.

3.2.1. Gestión con Tecnologías Tradicionales

La gestión de la información y configuración de las WSN desde Internet, ha sido implementada tradicionalmente mediante módulos de software de capa de aplicación. Dichos módulos serán denominados como aplicaciones de gestión remota y pueden ser desarrolladas en forma ad-hoc ó mediante estándares. Cabe destacar que en el presente trabajo, las aplicaciones que combinan el uso de estándares con componentes desarrollados en forma ad-hoc, serán consideradas como aplicaciones de gestión ad-hoc.

3.2.1.1. Ad-hoc

El enfoque ad-hoc propone el desarrollo de aplicaciones de gestión remota que consideren las características específicas de la WSN y el uso que se le dará a la red de sensores. El enfoque tiene como principal ventaja el adecuarse al hardware y a las tareas que realizará la WSN. Además, el desarrollo de aplicaciones ad-hoc permite minimizar el uso de recursos de hardware y el consumo de energía.

La principal desventaja del enfoque ad-hoc es que presenta problemas para lograr la interoperabilidad entre distintas plataformas de WSNs. El inconveniente surge debido a que la aplicación de gestión remota no está basada en estándares y se deben desarrollar aplicaciones específicas para cada nueva WSN a integrar a Internet.

A continuación se detallan algunas aplicaciones de gestión remota tipo ad-hoc que funcionan sobre las propuestas estudiadas en la Subsección 3.1.

Sensor Base - SDSP. Reddy et. al. [Reddy et al., 2007], desarrollaron una aplicación ad-hoc para que el usuario acceda a los datos y configuración de la WSN. La aplicación de gestión remota incluye un protocolo desarrollado por los autores llamado Sensor Data Stream Protocol (SDSP). SDSP transporta datos en formato XML en forma similar a Really Simple Syndication (RSS) RSS Advisory Board [2005].

La ventaja de SDSP es que funciona en forma bidireccional, ya que permite modificar desde Internet la configuración de la WSN. Además, SDSP permite que la información de la WSN sea almacenada en bases de datos, e indexada en motores de búsqueda de datos y sensores. Finalmente, el usuario accede a los servicios de SDSP gracias a una plataforma web desarrollada para gestión de WSNs denominada SensorBase [Chang et al., 2006].

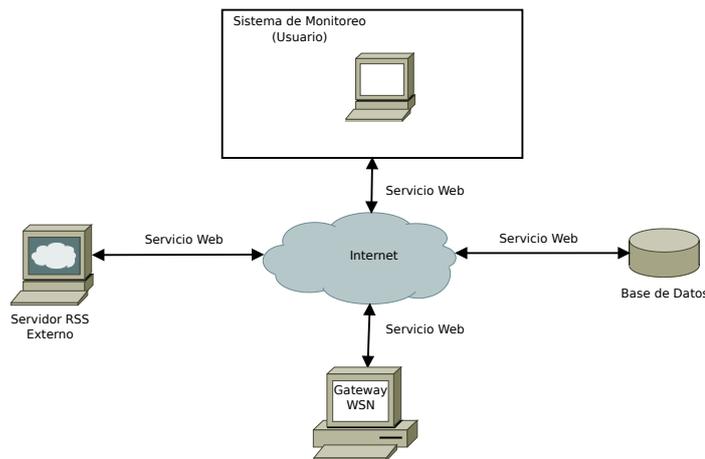


Figura 3.5: Aplicación de gestión remota de SYLPH. Figura basada en [Corchado et al., 2010].

SYLPH. Services layers over light physical devices (SYLPH) utiliza una aplicación de gestión remota basada en el modelo de arquitectura orientada a servicios (SOA) la cual permite utilizar servicios web para realizar la gestión remota de WSNs. Para acceder a los datos y configuración de los sensores SYLPH interactúa con estaciones bases de WSNs. Luego, los exporta a Internet como servicios web.

Los datos y configuración de las WSNs están codificados en un lenguaje de descripción (basado en XML) denominado SYLPH Services Definition Language (SSDL). Dicho lenguaje está adaptado a los recursos acotados de hardware y fuente de energía de los nodos sensores y permite estructurar los datos desde el momento en el que son adquiridos en campo. Además, SSDL permite la comunicación entre componentes de software sin importar su hardware ni protocolo de comunicación inalámbrico. La Figura 3.5, muestra los distintos componentes de SYLPH.

3.2.1.2. Basadas en Estándares

Las aplicaciones de gestión remota basadas en estándares realizan la gestión de datos y configuración de la WSN mediante servicios web, lenguajes de descripción como XML y WSDL, estándares de sensores (IEEE 1451, SensorML) y protocolos web (SNMP, HTML, etc). Si bien el uso de estándares implica el consumo de mayores recursos de memoria y procesamiento en los nodos sensores, tiene como principal ventaja la posibilidad de estandarizar los datos prácticamente desde el momento en que son obtenidos por el sensor ubicado en el campo.

A continuación se describen algunos de los estándares y lenguajes de descripción utilizados en aplicaciones de gestión remota basadas en estándares.

IEEE 1451. El estándar IEEE 1451 [Lee, 2000], describe un conjunto abierto de interfaces de comunicación para conectar sensores y actuadores a microprocesadores, instrumentos y redes de comunicación. Además, proporciona un conjunto de protocolos para redes cableadas e inalámbricas (CAN, ZigBee, Bluetooth, Ethernet) y permite que los sensores se conviertan en dispositivos “*plug-and-play*”. IEEE 1451 define tres elementos principales:

- Transducer Electronic Data Sheets (TEDS). Consisten en un método estandar de almacenamiento de información sobre el sensor como ID, calibración, rango de medición, información de fábrica y otros.
- Transducer Interface Module (TIM). Incluye sensores, actuadores, circuito de acondicionamiento de señal, conversores analógico - digitales y microcontrolador. Además, cuenta con una memoria que almacena el conjunto de TEDS correspondientes a los sensores y una interfaz de comunicación cableada o inalámbrica que permite comunicarse con el Network Capable Application Processor (NCAP). Cabe destacar que si el TIM es inalámbrico se lo denomina WTIM.
- NCAP. Es una computadora o un sistema embebido que tiene como principal objetivo proporcionar una pasarela entre los diferentes TIM y una red de comunicación mayor (LAN, Internet, etc).

Como ejemplo de aplicación de interconexión basada en IEEE 1451, puede citarse la desarrollada por Annamraju et al. [Annamraju et al., 2010]. Los autores desarrollaron un middleware que permite al usuario acceder desde Internet a WSNs ZigBee mediante IEEE 1451.

Cada nodo de la WSN es un WTIM de IEEE 1451. Los nodos tienen un radio ZigBee, microcontrolador, sensores y una memoria que permiten almacenar las TEDS. En la estación base de la WSN se encuentra el NCAP IEEE 1451. En la capa de aplicación del NCAP, se programan los módulos de software que permiten al usuario enviar mensajes en formato IEEE 1451 para acceder a los servicios de cada WSN desde Internet.

XML. Extensible Markup Language (XML) [Bray et al., 1997], es un lenguaje que gracias a una sintaxis estándar permite el intercambio de datos entre diferentes sistemas y aplicaciones. Los documentos XML son representados en forma textual para permitir la legibilidad humana del documento. A través de la representación textual de XML se almacenan datos independientemente del software y hardware utilizado, por lo que XML es una herramienta que permite intercambiar datos entre distintas aplicaciones y sistemas no compatibles [Höller, 2010].

Diversas propuestas se han hecho para acceder con XML a los servicios de las WSNs [Aberer et al., 2006, Gluhak and Schott, 2007, Bernat Vercher et al., 2008]. Höeller [Höller, 2010],

Tabla 3.1: Ventajas y Desventajas de utilizar XML en WSN.

Ventajas
Permite la interoperabilidad de distintos nodos sensores en la capa de aplicación.
Facilita la integración WSNs a redes TCP/IP
Permite implementar Arquitectura Orientada a Servicios (SOA).
Desventajas
La representación textual de XML puede ser mayor al 400 % del tamaño de su representación binaria [Hoeller et al., 2008].
Debido a que el proceso de transmisión de datos es el que más energía consume en una WSN, transmitir un bit puede ser 1000 veces más costoso que computar una sola operación [Madden et al., 2002].
Complejidad de programación de nodos sensores: usar XML requiere sólidos conocimientos de cómo representar XML con los lenguajes comúnmente utilizados para la programación de sistemas embebidos en WSN, por ejemplo Embedded C.

realiza un análisis de las ventajas y desventajas al utilizar XML en redes de sensores. La Tabla 3.1, sintetiza las principales ventajas y desventajas detectadas por Höeller.

SWE-SensorML. El grupo OGC-SWE (Open Geospatial Consortium - Sensor Web Enablement) se dedica al desarrollo de estándares abiertos, diseñados para generar sensores y sistemas de sensores “*Web-connected*”. Dichos sensores y sistemas pueden ser dispositivos de polución ambiental, medidores de temperatura, celdas de carga en estructuras sismo - resistentes, cámaras ubicadas en satélites, instrumentos oceanográficos, etc. [Botts et al., 2008]. OGC-SWE se focaliza en el desarrollo de estándares para permitir el descubrimiento, intercambio y procesamiento tanto de las observaciones de los sensores como de las tareas que realiza cada sensor. Una de las normas desarrolladas por este grupo de tareas es SensorML.

SensorML [Botts and Robin, 2007], es un modelo conceptual y un esquema XML que permite describir sensores, sistemas, y procesos. El estándar, provee la información necesaria para realizar el descubrimiento y conocer la localización de sensores. Además, permite configurar las WSNs y realizar el pre-procesamiento de las observaciones de los sensores.

Servicios Web (SOAP-XML). Un servicio WEB [Booth et al., 2004], es un software diseñado para brindar interoperabilidad en interacciones máquina a máquina dentro de una red. Dichos servicios tienen interfaces descritas en XML y permiten que distintos sistemas puedan interactuar entre si. La interacción puede ser realizada utilizando mensajes

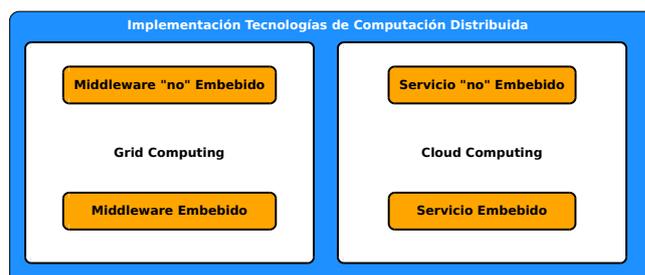


Figura 3.6: Enfoques de implementación de tecnologías Grid y Cloud.

SOAP, generalmente estandarizados con XML y transmitidos mediante HTTP.

El propósito de implementar servicios web en WSNs, es lograr que los nodos sensores actúen a modo de proveedores de servicios y las aplicaciones sean los clientes que consuman los servicios provistos por los sensores.

Dentro de las aplicaciones de interconexión que integran distintos estándares, se puede mencionar a Interoperable Test-Bed [O'Reilly et al., 2009]. Gracias al uso de protocolos de interconexión estándar, el usuario de Interoperable Test-Bed puede acceder a Internet, conectarse a un servidor IEEE 1451 propio de cada observatorio y gestionar la configuración e información de los instrumentos y sensores.

Para desarrollar la aplicación de interconexión, los autores utilizan servicios web especiales denominados Smart Transducer Web Services (STWS). STWS es un conjunto de servicios web diseñados para gestionar Smart Transducers de IEEE 1451 [Song and Lee, 2007, 2008]. STWS está descrito en WSDL, basado en arquitectura SOA y en IEEE 1451. Otra ventaja de STWS es que permite integrar instrumentos que utilicen el estándar SWE-SensorML [Song and Lee, 2009] a IEEE 1451.

3.2.2. Gestión mediante Tecnologías Grid y Cloud

Las tecnologías Grid y Cloud pueden implementarse para gestionar las WSNs desde Internet mediante alguno de los dos siguientes enfoques: middleware grid (ó servicio cloud) “*embebido*” y middleware grid (ó servicio cloud) “*no embebido*” en la WSN. Estos enfoques dependen de si se programan o no componentes de software del middleware grid ó del servicio cloud en los nodos y/o estaciones base de la WSN. La Figura 3.6, ilustra los enfoques mencionados.

3.2.2.1. Middleware ó Servicio Cloud no embebido en la WSN

En este caso, el middleware grid o el servicio cloud están programados en una máquina externa a la WSN. Una vez que la información y configuración de la WSN están disponibles en TCP/IP, el middleware grid (Globus, gLite, GridGain, etc.) ó el servicio cloud (Globus Online, Google Drive, etc.) se encarga de gestionarlos. En este tipo de enfoques, la WSN funciona sin cargas extras de software en los nodos, por lo que se minimiza el uso de recursos de cómputo y el consumo de batería en los nodos.

Además, se debe tener en cuenta que en este enfoque los nodos no pueden ser administrados directamente como un recurso de Grid y Cloud, ya que esto se resuelve en una máquina externa a la WSN.

National Weather Sensor Grid. En [Lim et al., 2007], los autores presentaron una plataforma de gestión de WSNs denominada National Weather Sensor Grid (NWSG). Esta plataforma brinda soporte en Singapur a estudios medioambientales a gran escala. NWSG (Ver Figura 3.7) está compuesta por diversas estaciones meteorológicas, las que registran datos de: temperatura, humedad, precipitaciones, velocidad y dirección del viento, etc. Las estaciones meteorológicas de NWSG, están interconectadas y se encuentran ubicadas en distintas escuelas de Singapur.

Cada estación meteorológica de NWSG se conecta a una PC que contiene un servidor proxy, el que actúa como gateway permitiendo que los datos sean enviados a una base de datos. A partir de dicha base de datos, la información es procesada, analizada y almacenada utilizando una infraestructura Grid.

Respecto al middleware Grid utilizado, la base de datos y los nodos de la Grid son administrados con Globus Toolkit. Además, los usuarios pueden acceder a los datos, desde Internet, a través de Google Earth y Microsoft Sensor Map.

MPAS. Yujie et al. [YuJie et al., 2005] propusieron una plataforma basada en protocolos Grid estándar llamada MPAS (Ver Figura 3.8). MPAS utiliza permite integrar WSNs a Grids que funcionan con Globus. Los nodos y la estación base tienen programados el middleware para WSNs TinyOS. Dicho middleware permite administrar la configuración de los nodos y realizar el proceso de adquisición de datos. Una vez adquiridos, los datos son enviados desde la estación base de la WSN en formato hexadecimal y vía TCP/IP hacia una máquina donde está programado MPAS. Luego, MPAS los convierte a XML y estandariza según WSRF [Czajkowski et al., 2004] para finalmente integrarlos a la Grid.

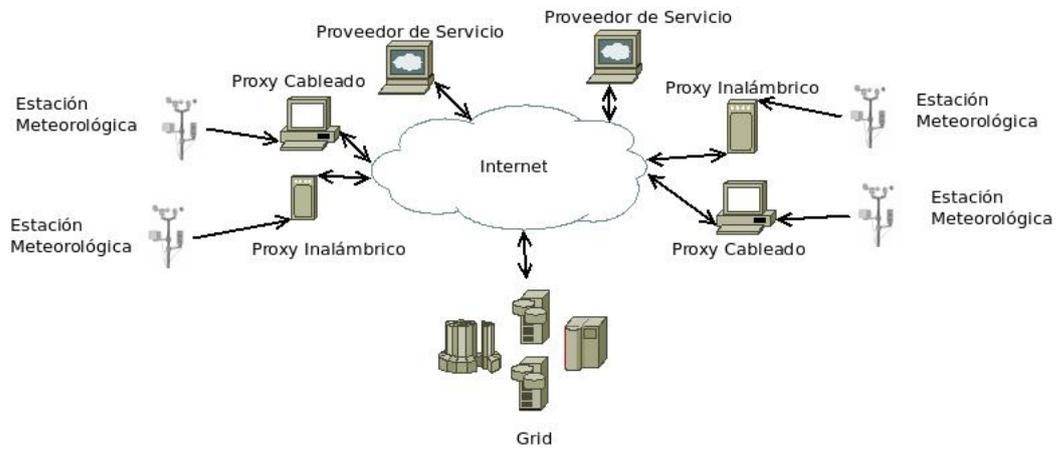


Figura 3.7: Arquitectura de National Weather Sensor Grid. Figura basada en [Lim et al., 2007].

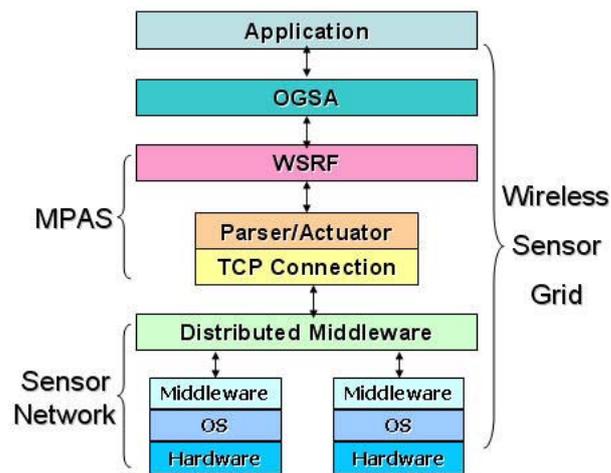


Figura 3.8: Arquitectura de MPAS. Figura extraída de YuJie et al. [2005].

Sensor Grids for Air Pollution Monitoring. En [Ghanem et al., 2004], los autores diseñaron una grid de sensores para medir contaminación ambiental. En este caso, la incorporación de WSNs a Grid es necesaria debido a la gran cantidad de información (8 GB por día) producida por cada sensor. La integración de sensores vía servicios grid se realiza a través de Discovery Net [Alsairafi et al., 2003].

Discovery Net, está basada en protocolos y estándares como Globus, pero además define sus propios protocolos como el Discovery Process Mark-up Language para gestión de flujo de datos. Las capacidades de cada sensor pueden ser accedidas a través de SensorML [Botts and Robin, 2007], lo que implica que metadatos (datos que describen las propiedades del sensor) y los datos del sensor puedan ser accedidos a través de protocolos estándar. En la grid de sensores propuesta por los autores, cada nodo sensor envía los datos a un servicio grid remoto. Dicho servicio grid gestiona el almacenamiento de la información en bases de datos, que finalmente son accedidas a través de Globus Toolkit.

Humble et al. En [Humble et al., 2005] los autores presentaron una arquitectura en la que cada uno de los sensores que componen cada nodo sensor de la WSN es un proveedor de servicios grid OGSA. A diferencia de otras propuestas analizadas [Ghanem et al., 2004], los autores plantearon que cada sensor físico (Temperatura, Humedad, etc.) fuera el proveedor de servicios grid.

El cliente accede a servicios grid compuestos por los datos proporcionados por cada sensor asociado a los nodos de la WSN. De igual forma, el cliente accede a un servicio para realizar cambios en la configuración del nodo o de alguno de los sensores (por ejemplo variar la frecuencia de muestreo del sensor de temperatura).

Internet and Integration Controller interaction Architecture. En [Rajesh et al., 2010], los autores desarrollaron una plataforma que mediante SOAP y XML, permite que los sensores sean proveedores de datos. La arquitectura propuesta por Rajesh et al. utiliza un dispositivo llamado *Integration Controller* (IC) a través del cual pasan las comunicaciones con el cliente. El cliente, utiliza mensajes SOAP para invocar servicios de descubrimiento de sensores. Los servicios pasan por el IC, el cual permite la autenticación del usuario y funciona como gateway a la WSN. El trabajo de Rajesh et al. demostró que los usuarios pueden realizar de forma sencilla la recolección, procesamiento, visualización, almacenamiento y búsqueda de gran cantidad de datos recolectados por sensores en Clouds.

Tangible Cloud Computing. Lee et al. [Lee and Hughes, 2010], los autores sostienen que para integrar y compartir recursos de WSNs en el Cloud, se debe contar con tec-

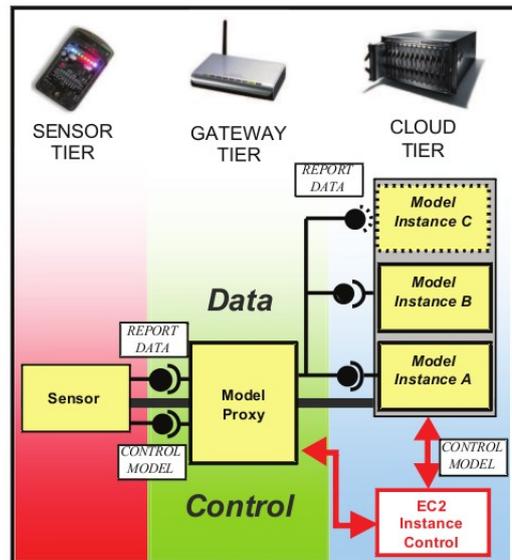


Figura 3.9: Arquitectura de Tangible Cloud Computing. Figura extraída de [Lee et al., 2010].

nologías de virtualización adecuadas para sistemas embebidos. Dichas tecnologías de virtualización permiten que los dispositivos de sensado sean gestionados en forma remota mediante servicios web.

Existen ciertas plataformas de hardware para WSNs como Sun SPOT [Oracle Microsystems, Diciembre 2012], que cuentan con el entorno de ejecución Java y permiten la virtualización necesaria para compartir recursos. Sin embargo, cabe destacar que estas plataformas no siempre están disponibles en el mercado local. Además, requieren de mayores recursos de hardware y consumo de energía.

Los autores propusieron una arquitectura llamada Tangible Cloud Computing, la cual toma conceptos de Cloud tales como:

- Recursos abstractos o virtualizados.
- SaaS.
- Modelo de fijación de precios Pay-Per-Use.

“Tangible Cloud Computing” tiene como principal objetivo extender el dominio de Cloud a sensores, cámaras, etc. En [Lee et al., 2010] los autores implementaron la plataforma con Java y Amazon EC2 [Amazon EC2, 2012]. En Tangible Cloud Computing, EC2 permite solicitar al Cloud recursos de procesamiento a medida que crece la cantidad de información recolectada por los sensores. En la Figura 3.9, se puede observar la arquitectura desarrollada por Lee et. al.

Ahmed et al. Ahmed and Gregory [Ahmed and Gregory, 2011], presentaron una plataforma para integrar WSNs a Cloud. La plataforma desarrollada por los autores tiene como principal objetivo “*facilitar el intercambio de datos entre WSNs e infraestructuras de Cloud Computing*”. Además, los autores concluyen que la integración de WSNs a Cloud permite que los datos de las WSNs sean almacenados en infraestructuras de dominio público. Gracias a la posibilidad de almacenar los datos en infraestructuras públicas, se puede optimizar el uso de la información recolectada por la WSN debido a que distintos usuarios y aplicaciones comparten dicha información.

3.2.2.2. Middleware ó Servicio Cloud Embebido en la WSN

El enfoque de middleware grid ó servicio cloud embebido en la WSN se basa en programar el middleware grid ó servicio cloud en los nodos o en la estación base de la WSN. En dicho enfoque el hardware de la WSN debe contar con los recursos necesarios para soportar los componentes del middleware grid ó del servicio cloud correspondiente, por lo que esta solución dependerá de los recursos de hardware disponibles en la WSN. Otras consecuencias relacionadas a programar componentes del middleware grid (ó servicio cloud) en la WSN son retardos y aumento de consumos de energía en la WSN.

Globus Monitoring and Discovery Service. Aloisio et. al. presentaron en [Aloisio et al., 2006] una arquitectura que utiliza el servicio Monitoring and Discovery Service (MDS) de Globus para integrar las WSNs a infraestructuras grid. La arquitectura propuesta (Ver Figura 3.10) utiliza SensorML para estructurar la información proveniente de las WSNs. Luego, el servicio de monitoreo de MDS analiza la información proveniente del recurso WSN y permite la integración básica a la Grid. Finalmente, Globus provee los servicios necesarios para que el recurso sea totalmente incorporado a la Grid.

Los autores plantean que las WSNs presentan las siguientes características que las hacen aptas para ser integradas a la Grid:

- Los nodos sensores son heterogeneos, tanto en la información como en el formato de los datos que proveen.
- Cada WSN puede pertenecer a distintas organizaciones por lo que se necesita regular la forma en que éstas son compartidas.
- Al trabajar de forma ininterrumpida, los sensores pueden llegar a producir una gran cantidad de información.

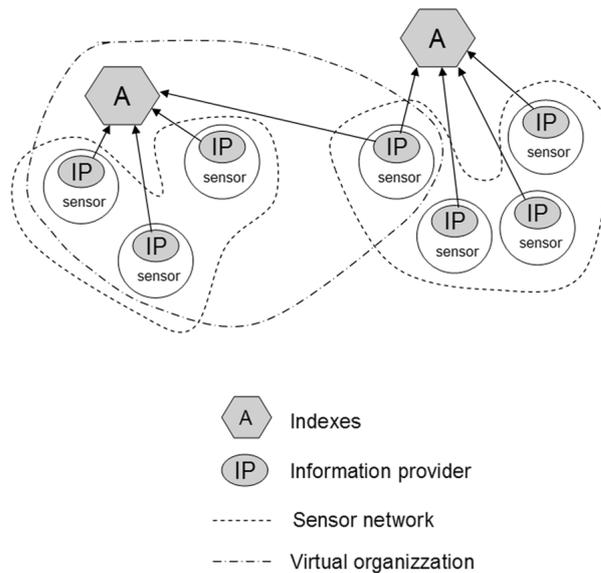


Figura 3.10: Arquitectura propuesta por Aloisio et. al. Figura extraída de [Aloisio et al., 2006].

En el trabajo de Aloisio et. al., se observa que si bien se pudo realizar la implementación de la arquitectura sin mayores inconvenientes, la misma no ha sido aplicada en sensores con recursos limitados de hardware y fuente de energía.

Zhang. et. al. Los autores desarrollaron en [Zhang et al., 2010], una implementación de WSN que permite la incorporación de WSNs a Grid mediante estándares de servicios web. Zhang. et. al. desarrollaron la interfaz necesaria para conectar una WSN con la Grid utilizando la plataforma para desarrollo de servicios web gSOAP [Van Engelen and Galivan, 2002].

El nodo que utilizaron los autores (ver Figura 3.11) es una plataforma basada en un microprocesador ARM 9 con Sistema Operativo Linux embebido, por lo que la plataforma tiene los recursos de hardware y software lo suficientemente potentes como para recibir los datos de los sensores y guardarlos en un documento. En el trabajo de Zhang et. al., el usuario accede a los datos por intermedio de un servidor web embebido BOA [Doolittle and Nelson, 2000] y un programa del tipo *Common Gateway Interface* (CGI) [Robinson and Coar, 2004].

SEGEDMA. Sensor Grid Enhancement Data Management System for Health Care Computing (SEGEDMA) [Preve, 2011], es una plataforma que permite que las WSNs sean descubiertas y accedidas por cualquier aplicación grid que necesite de sus servicios. SEGED-

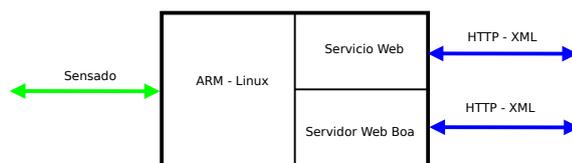


Figura 3.11: Nodo desarrollado por Zhang et. al. Figura basada en [Zhang et al., 2010].

Tabla 3.2: Componentes de SEGEDMA.

Componente del Gateway	Función
Data Management	Realiza la conversión de datos de la WSN a formato Grid compatible (por ejemplo XML). Soporta diversos métodos para transferir datos a la aplicación del usuario como GridFTP.
Information Services	Gestiona el descubrimiento y monitoreo de los recursos WSN. Utiliza mecanismos como el IS y el MDS de OGSA.
WSN Connectivity	Provee servicios para realizar la interface entre los protocolos de las WSN y los protocolos de red de la Grid.
Power Management	Se encarga de seguir el consumo de los sensores y realizar acciones para mantenerlo al mínimo.
Security	Implementa técnicas de seguridad compatibles con las tecnologías Grid de OGSA, para, por ejemplo, realizar mediante GSS autenticación entre el Proxy y la WSN.
Availability	Garantiza la disponibilidad de la WSN, monitoreando fallas en sensores, nivel de señal débil, y otras causas que lleven a migrar las tareas a otros nodos.
QoS	Gestiona junto al WSN Scheduler la ubicación y reserva de recursos WSN de acuerdo a los requerimientos de calidad de servicio de las tareas.

MA toma los datos provenientes de los sensores y los convierte a formatos compatibles con protocolos estándar OGSA.

La plataforma utiliza un gateway para vincular la Grid con dos componentes llamados WSN Scheduler y WSN Manager. Estos componentes son los que administran los recursos y el acceso a las WSNs. El gateway desarrollado por Preve et al. tiene siete componentes de software que permiten incorporar las WSNs a Grid Computing (Ver Tabla 3.2).

Hughes et. al. Los autores desarrollaron en [Hughes et al., 2008] una plataforma llamada GridStix, la cual funciona en nodos basados en computadoras Intel XScale PXA255 CPU de 400 Mhz, con 64 MB de memoria RAM y 16 MB de memoria flash. Los nodos permiten que cada nodo tenga conexión IP, sin necesidad de utilizar una estación base como gateway.

La principal ventaja de esta propuesta es la descentralización del punto de falla de la red. Además, la plataforma soporta Linux embebido y entorno de ejecución Java lo que permite que cada nodo sensor tenga embebido un middleware GridKit [Grace et al., 2004] y sea un recurso de la Grid sin necesidad de utilizar una estación base.

Se observa que la propuesta de Hughes et. al. presenta el inconveniente de requerir conexión permanente a una fuente de energía solar. Además, el consumo de energía en los nodos es 14 veces más en modo bluetooth o 50 veces más en modo Wi-Fi, comparado con un Mote Berkeley [Polastre et al., 2005].

Aneka. Otra plataforma para integrar WSNs a Cloud es Aneka [Gubbi et al., 2013]. Aneka utiliza recursos de Clouds privados y públicos para brindar soporte a aplicaciones de “*entornos inteligentes*” [Cook and Das, 2004] como cuidado de salud, transporte, monitoreo urbano y otras. La plataforma ofrece PaaS que soportan distintos modelos de programación paralela (por ejemplo MapReduce) para que los desarrolladores puedan construir SaaS adaptados a sus necesidades. Respecto a los servicios de infraestructura, Aneka permite acceder a los IaaS suministrados por Microsoft Azure, Amazon EC2 y GoGrid.

Hirafuji et al. Los autores desarrollaron una plataforma aplicada a estudios de rendimiento en cultivos de distinto tipo [Hirafuji et al., 2011]. La plataforma desarrollada utiliza Twitter para almacenar y mostrar los datos recolectados por las WSNs. Los nodos sensores utilizados por los autores están compuestos por un sistema embebido Arduino con interfaz Ethernet, celdas solares y WiFi. La principal innovación del sistema desarrollado por Hirafuji et al. es la simplicidad de la plataforma y el bajo costo económico requerido para gestionar grandes volúmenes de datos recolectados por WSNs.

Hori et al. En [Hori and Kawashima, 2010], se presentó una solución comercial que permite almacenar y procesar datos de distintos tipos de sensores agrícolas en el Cloud. Además, la plataforma permite la integración de los datos recolectados a sistemas de gestión agrícola desplegados como SaaS, lo que facilita el seguimiento de la trazabilidad, la aplicación de normas de calidad y buenas prácticas agrícolas a los cultivos. Finalmente cabe destacar que los autores estiman que en treinta años la cantidad de datos a procesar por la plataforma alcanzaría los 100 PBytes.

3.3. Resumen de los Trabajos Relacionados

En esta sección se resúmen los trabajos previos que resuelven la gestión remota de WSNs desde Internet. La Tabla 3.3 resume las principales características de dichos trabajos. Cada fila de la Tabla 3.3, corresponde a uno de los trabajos previos analizados en este capítulo.

En la segunda columna de la Tabla 3.3, se especifica el tipo de arquitectura utilizado en cada propuesta para compatibilizar las WSNs con TCP/IP. Luego, en la tercer columna se detalla el tipo de tecnología de gestión remota utilizada (tradicional ó de computación distribuida) y en la cuarta columna el enfoque adoptado (ad-hoc, estándar, middleware grid ó servicio cloud embebido, etc.). En la quinta columna, se especifica la tecnología utilizada por los autores para implementar la gestión remota y en la sexta columna la plataforma de hardware utilizada en la WSN. Finalmente en la séptima columna, se detalla el uso que se le dará a cada una de las plataformas desarrolladas.

3.4. Discusión

En esta sección, se discutirán los trabajos resumidos en la sección anterior desde los enfoques mediante los cuales fueron categorizados en la Secciones 3.1 y 3.2.

3.4.1. Integración de WSNs a TCP/IP

En primer lugar, se debe determinar cual es la arquitectura más adecuada para incorporar plataformas de WSNs con recursos acotados de hardware y energía (como las disponibles en el mercado local) a TCP/IP. Por un lado, autores como [Reddy et al., 2007, O'Reilly et al., 2009, Corchado et al., 2010, Annamraju et al., 2010] proponen incorporar las WSNs a TCP/IP mediante arquitectura de gateway. La principal ventaja de este enfoque es que no requiere agregar componentes extras de software en los nodos de la WSN, por lo tanto, los nodos funcionan solo con protocolos diseñados para WSNs y se optimiza tanto el uso de recursos de hardware como así también consumo de energía. Además, el gateway permite reutilizar tecnología existente, debido a que la integración se implementa solamente en una máquina y no se deben realizar modificaciones en el software de WSNs ya instaladas y en funcionamiento.

Uno de los inconvenientes que presenta el enfoque de gateway es el desbalance de tráfico en la WSN. El problema surge debido a que una gran cantidad de datos tienen que ser enviados desde y hacia el gateway. Ante esta situación, los nodos cercanos al gateway serán los que presentarán un mayor consumo de energía. Otro problema de la arquitectura gateway es la centralización del punto de falla en la estación base, lo que genera que la WSN pierda la conexión a Internet en caso de presentarse una falla en el gateway.

El uso de gateways presenta problemas de escalabilidad debido a que es necesario agregar un componente nuevo de software en el gateway por cada nueva WSN a integrar. Además, debe considerarse que no es posible la comunicación extremo a extremo entre la WSN y la red TCP/IP debido a que todas las comunicaciones están centralizadas por el gateway.

Como puede observarse en las propuestas de [Ghanem et al., 2004, YuJie et al., 2005, Humble et al., 2005, Lim et al., 2007, Rajesh et al., 2010, Ahmed and Gregory, 2011], el enfoque de gateway está naturalmente asociado al uso de middlewares grid y servicios cloud no embebidos en la WSN. Sin embargo, en algunos casos pueden embeberse componentes del middleware ó servicio sólo en el gateway de la WSN, por lo que también en estos casos el uso de gateways podría combinarse con el de middlewares ó servicios embebidos.

Por otro lado, autores como [Dunkels et al., 2003, Islam et al., 2010, Silva et al., 2010] presentan propuestas que siguen el enfoque overlay. Las principales ventajas de las redes

Tabla 3.3: Resumen de Trabajos Previos

Propuesta	Integración a TCP/IP	Gestión remota	Enfoque	Tecnología Implementada	Plataforma de hardware WSN	Aplicación
Sensor Base-cSDSP [Reddy et al., 2007]	Gateway	Tradicional	Ad-hoc	SDSP protocolo basado en RSS/XML.	PC, cámara web y sensor de temperatura con display LCD.	Monitoreo de hábitos de animales.
SYLPH [Corchado et al., 2010]	Gateway	Tradicional	Ad-hoc	SSP y SSDL.	Nodos ZigBee con microcontrolador C8051F121, radio CC2420 y nodos Bluetooth BlueCore4-Ext	Asistencia sanitaria y monitoreo de salud.
Annamraju et al. [2010]	Gateway	Tradicional	Estándar	IEEE 1451.4.	Nodos sensores Mobeey y gateways MobeeyNet.	Industria Aeroespacial.
Interoperable Test-Bed [O'Reilly et al., 2009]	Gateway	Tradicional	Estándar	OCC-SWE/SensorML y IEEE 1451	Instrumentos Oceanográficos (SBE37-SM CTD, RBR XR420 CTD, etc.).	Oceanografía.
uTCP/IP [Dunkels et al., 2003]	Overlay	Tradicional	Estándar	uTCP/IP	Nodos sensores Scatterweb y Motes Berkeley.	No se especifica, los nodos pueden ser adaptados a diversas aplicaciones.
SMPIPv6 [Islam et al., 2010]	Overlay	Tradicional	Ad-hoc	SMPIPv6	Diseñado para nodos sensores 6lowPAN. Validación realizada mediante simulación con NS2.	Monitoreo de Pacientes. Nodos sensores móviles.
SMP [Silva et al., 2010]	Overlay	Tradicional	Ad-hoc	MIPv6	Diseñado para nodos sensores 6lowPAN y Contiki OS. Validación realizada mediante simulación con Cooja.	Monitoreo de Salud en Industrias. Nodos sensores móviles.
NWSG [Lim et al., 2007]	Gateway	Middleware grid	Middleware "no" em-bebido.	Globus.	Estaciones meteorológicas y computadoras tipo PC.	Estudio del clima.
MPAS [Yujie et al., 2005]	Gateway	Middleware grid	Middleware "no" em-bebido.	Globus y TinyOS.	Motes Berkeley.	Asistencia sanitaria.
Ghanem et al. [2004]	Gateway	Middleware grid	Middleware "no" em-bebido.	Globus y Discovery Net.	Computadoras tipo PC y sensores CUSTO.	Estudios de polución ambiental.

Propuesta	Integración a TCP/IP	Gestión remota	Enfoque	Tecnología Implementada	Plataforma de hardware	Aplicación
Humble et al. [2005]	Gateway	Middleware grid	Middleware "no" embebido.	Globus.	Dispositivos ad-hoc y estaciones meteorológicas.	Monitoreo de salud y estudios medioambientales.
IICIA [Rajesh et al., 2010]	Gateway	Servicio cloud	Servicio "no" embebido.	SOAP-XML.	Dispositivos ad-hoc e instrumental de planta nuclear.	NPMS (Nuclear Plant Maintenance System).
Ahmed et al. [2011]	Gateway	Servicio cloud	Servicio "no" embebido.	XML.	Dispositivos con capacidades de sensado en general.	Aplicaciones científicas y comerciales en general.
Tangible Cloud Computing [Lee and Hughes, 2010]	Overlay	Servicio cloud	Servicio embebido.	Amazon EC2, SOAP-XML.	SunSpot.	Control y Monitoreo de tráfico, logística, etc.
Zhang et al. [2010]	Overlay	Middleware grid	Middleware embebido.	gSOAP.	Microcontroladores ARM9 con sistema operativo Linux.	No se especifica.
SEGEDMA [Preve, 2011]	Overlay	Middleware grid	Middleware embebido.	Globus.	Motes Berkeley.	Monitoreo de salud.
Hughes et al. [2008]	Overlay	Middleware grid	Middleware embebido.	GridKit.	GridStix.	Monitoreo de inundaciones.
Aneka [Gubbi et al., 2013]	Overlay	Servicio cloud	Servicio embebido.	SOAP-SensorML, Windows Azure y otros.	Dispositivos con capacidades de sensado en general.	Entornos Inteligentes.
Hirafuji et al. [2011]	Overlay	Servicio cloud	Servicio embebido.	Twitter.	Nodos Arduino con Ethernet y WIFI.	Estudios de rendimiento en cultivos.
Hori et al. [2010]	Overlay	Servicio cloud	Servicio embebido.	Fujitsu Cloud Platform.	Dispositivos con capacidades de sensado en general.	Agricultura.

overlay son que permiten la comunicación extremo a extremo entre ambas redes y brindan la posibilidad de trabajar con la pila de protocolos TCP/IP, lo que facilita el trabajo a usuarios familiarizados con protocolos de Internet.

Algunas propuestas como μ TCP/IP [Dunkels et al., 2003], han permitido el desarrollo de versiones especiales de la pila TCP/IP para ser utilizadas en WSNs con recursos acotados de energía. Sin embargo, es necesario implementar algunas capas de la pila TCP/IP en los nodos y otras en la estación base (la cual actúa a modo de gateway) para que sea posible la implementación de TCP/IP en WSNs [Bormann et al., 2006, Islam et al., 2010], por lo que no siempre esta solución logra evitar el punto de falla centralizado de la red.

Otra desventaja que presenta este enfoque es que las propuestas de versiones de TCP/IP analizadas solo son compatibles con algunas plataformas de hardware de WSN. Estas plataformas, presentan mayores costos económicos que las plataformas ZigBee disponibles en el mercado local (por ejemplo Nodos XBee) [Digi International Inc., 2012].

Cabe destacar que el enfoque overlay está asociado al uso de middlewares grid y servicios cloud embebidos en la WSN, como puede observarse en los trabajos de Aloisio et al. [2006], Zhang et al. [2010], Preve [2011], Hughes et al. [2008], Gubbi et al. [2013], Hirafuji et al. [2011], Hori and Kawashima [2010], Lee et al. [2010], Lee and Hughes [2010].

A modo de resumen, se puede afirmar que *el enfoque más apto para incorporar las WSNs a TCP/IP es el de gateways debido principalmente a que no genera consumos extras de recursos de hardware y fuente de energía en los nodos sensores*. Además este enfoque permite incorporar a TCP/IP WSNs ya instaladas y en funcionamiento.

3.4.2. Gestión Remota

Una vez determinada la arquitectura más adecuada para incorporar las WSNs a TCP/IP, se debe determinar que tecnología de gestión remota presenta mayores ventajas para administrar las WSNs desde Internet. En esta sección se analizan las tecnologías de gestión remota que implementan los distintos autores en la literatura revisada. En la tercer columna de la Tabla 3.3, se puede observar que tipo de tecnología se desarrolla en cada propuesta para realizar la gestión remota de las WSNs desde Internet (tradicional, servicio cloud, middleware grid, etc.). Luego, en la cuarta columna se detalla el enfoque seguido en cada propuesta (ad-hoc, estándar, embebido ó no embebido) para implementar la tecnología de gestión remota. Finalmente, en la quinta columna se especifica que estándar, tecnología ad-hoc, middleware ó servicio implementa cada propuesta para solucionar la gestión remota.

El uso de tecnologías tradicionales ad-hoc tiene como principal ventaja facilitar el desarrollo de programas adaptados a las necesidades del usuario y de las aplicaciones. En la

mayoría de los casos, esta técnica no necesita cargar funciones extras a los componentes de software de las WSN. Además, se puede optimizar el consumo de energía y minimizar el uso de los recursos de hardware debido a que los nodos solo necesitan procesar el protocolo de la red de sensores.

De la literatura revisada surge que este tipo de enfoque se utiliza mayormente en arquitecturas basadas en gateway y en menor medida en redes overlay. Además, debido a que las redes overlay han sido desarrolladas para ser utilizadas con protocolos de Internet, el tipo de aplicaciones ad-hoc utilizadas en redes overlay suelen ser modificaciones de estándares de Internet.

En el enfoque ad-hoc no se utiliza estandarización en el software, por lo que el programador de las aplicaciones debe estar familiarizado tanto con la tecnología WSN como con la aplicación de gestión remota desarrollada. Entre otros aspectos relacionados a la WSN, el programador deberá conocer la forma de la trama utilizada por el protocolo WSN nativo para entregar los datos a la estación base.

Otros problemas que presentan las aplicaciones ad-hoc son la escalabilidad e interoperabilidad en etapas post-desarrollo de la aplicación de interconexión. Estas dificultades surgen cuando se deben incorporar nuevas plataformas de WSNs a las ya existentes, lo que implica que se deban agregar nuevos módulos de software en la estación base.

Las tecnologías tradicionales basadas en estándares, permiten estandarizar los datos de los sensores y sus metadatos prácticamente desde el momento en que son adquiridos en los nodos. El uso de estándares como IEEE 1451 y SensorML permite que se acceda a los nodos sensores mediante un formato estándar y facilita la interoperabilidad de sensores que utilizan distintos protocolos de comunicación.

Algunos autores han propuesto utilizar servicios web para acceder a los datos y configuración de sensores. El uso de servicios web en WSNs facilita la interacción con tecnologías como Cloud e independiza al programador de aplicaciones del hardware y del protocolo nativo de las WSNs.

El principal inconveniente respecto al uso de estándares en WSNs es la necesidad de programar en los nodos componentes de software adicionales al protocolo nativo utilizado (por ejemplo ZigBee). Dichos módulos de software incluyen la descripción de cada sensor y sus metadatos (de acuerdo al formato del estándar utilizado). Generalmente esta información no se proporciona estandarizada por el fabricante e implica un mayor trabajo de programación. Además los nodos deben contar con recursos de hardware suficientes para soportar los componentes de software adicionales necesarios para entregar los datos a la estación base estandarizados según el formato pre-establecido por el protocolo implementado.

Otra desventaja que presentan las aplicaciones de gestión remota basadas en estándares

es el consumo de energía en los nodos. Dicho problema surge porque al enviar datos en formato XML, IEEE 1451 o SensorML la cantidad de bits a transmitir es mucho mayor que si sólo se enviaran los datos en el formato nativo WSN, por lo que la radio del nodo debe permanecer encendida más tiempo y consume más energía. Además, en este enfoque el tiempo para acceder a los datos es mayor, ya que se deben expresar los datos de acuerdo al lenguaje de descripción utilizado.

Respecto al uso de tecnologías de *computación distribuida*, el enfoque de middlewares grid ó servicios cloud “no” embebidos permite re-utilizar tecnología ya existente y en funcionamiento [Lim et al., 2007, YuJie et al., 2005, Ghanem et al., 2004, Humble et al., 2005, Rajesh et al., 2010, Lee and Hughes, 2010, Ahmed and Gregory, 2011]. Además, no requiere programar en los nodos de la WSN componentes extras de software relacionados con el middleware grid ó el servicio cloud.

En algunos casos [YuJie et al., 2005], los autores demostraron que la plataforma de gestión de WSNs (desarrollada con tecnologías de Grid Computing) podría presentar retardos o fallas al proveer los servicios al usuario si son varias las WSN conectadas a la plataforma. Los autores dedujeron que dichos inconvenientes podrían deberse al uso compartido de CPU con otros procesos ó al tiempo requerido para tomar los datos y cambiarlos de formato WSN nativo a formato middleware Grid.

Cuando se utiliza el enfoque de middleware ó servicio cloud embebido en la WSN, los nodos sensores pueden ser administrados individualmente como recursos grid [Aloisio et al., 2006, Zhang et al., 2010, Preve, 2011, Hughes et al., 2008] ó servicios cloud [Gubbi et al., 2013, Hirafuji et al., 2011, Hori and Kawashima, 2010, Lee et al., 2010, Lee and Hughes, 2010]. Gracias a esta posibilidad, el enfoque de middleware grid ó servicio cloud embebido permite incorporar las WSNs a TCP/IP mediante redes overlay y evita generar un punto de falla centralizado en la conexión de la WSN a Internet. Otra ventaja de este enfoque es que no se comparten recursos de CPU con otros procesos debido a que la integración WSN - Grid/Cloud se realiza en la estación base y/o en los sensores de la WSN.

La mayoría de las propuestas analizadas que siguen el enfoque de middleware ó servicio programado en la WSN, utilizan nodos con recursos de cómputo, memoria y fuente de energía lo suficientemente potentes como para soportar Linux y el middleware ó servicio necesario. Sin embargo, algunos autores han demostrado que es posible aplicar el enfoque analizado a WSNs con recursos acotados de hardware y fuente de energía. En [Preve, 2011] se accede a la WSN como recurso Grid pero se programa el middleware en la estación base, si bien aparece el punto centralizado de falla, se ha logrado optimizar el consumo de energía y los recursos de hardware dentro de la WSN.

Además, se detectó que sólo algunas plataformas basadas en Cloud Computing utilizan servicios cloud estándar [Lee and Hughes, 2010, Hori and Kawashima, 2010, Hirafuji

et al., 2011, Gubbi et al., 2013]. La principal ventaja del uso de servicios cloud estándar es que se simplifica el diseño e implementación de la plataforma.

Se puede concluir que *las tecnologías más adecuadas para gestionar las WSNs desde Internet son las de Cloud Computing, preferentemente las estándar como Google Drive, Amazon EC2 y Dropbox*. Esto es así debido a que brindan acceso a recursos computacionales acordes a las demandas del usuario, son de fácil implementación y garantizan el acceso a los recursos computacionales y de las WSNs 24 horas al día, 365 días al año. Respecto al enfoque de implementación, *se recomienda el uso de servicios cloud "no" embebidos en la WSN*, lo cual permite evitar consumos extras de recursos de hardware y energía en los nodos de la WSN.

3.4.3. Factibilidad Tecnológica y Costo Económico

En el contexto del presente trabajo, se debe evaluar si las plataformas de gestión remota estudiadas en el estado del arte son factibles de ser implementadas en Argentina y si su costo económico es adecuado para implementar Cloud de Sensores en el contexto local. El análisis de estas plataformas debe ser realizado en base a las WSNs que gestionan y a las tecnologías que emplean para realizar la gestión.

Las plataformas de gestión de WSNs serán tecnológicamente factibles si permiten gestionar plataformas de hardware que estén disponibles en el mercado local. En las propuestas analizadas en el estado del arte, se ha detectado que la mayoría de los autores utilizan en los nodos de la WSN plataformas de hardware que no cumplen con dichos requerimientos. Como ejemplo puede mencionarse las propuestas que utilizan plataformas de hardware basadas en Motes Berkeley [Corchado et al., 2010, Preve, 2011, YuJie et al., 2005, Dunkels, 2003, Dunkels et al., 2003], las cuales son de difícil adquisición en el mercado local. Respecto al costo económico de estas plataformas de hardware, son mayores si se las compara con otras disponibles en el mercado local como por ejemplo los nodos XBee ZB [Digi International Inc., 2012].

Propuestas como [Hughes et al., 2008, Lee and Hughes, 2010, Lee et al., 2010, Zhang et al., 2010] utilizan nodos con recursos de hardware lo suficientemente potentes como para incorporar un sistema operativo (por ejemplo: Linux) o plataformas como Java. Sin embargo, los nodos sensores que pueden utilizar sistema operativo tienen alto costo económico por lo que sólo podrían ser empleadas como estaciones base en el contexto de la presente tesis. A modo de ejemplo se puede mencionar que un nodo que soporte linux embebido compuesto por una plataforma Raspberry Pi 2 y un módulo de comunicaciones GPRS es un 57 % más caro (sin incluir los costos del uso de la red de telefonía celular) que un nodo conformado por una placa Arduino Uno con módulo XBee ZigBee. Además presenta un consumo de baterías mucho mayor.

Respecto a la factibilidad de las tecnologías de gestión remota de WSNs, se ha observado que las tecnologías tradicionales son tecnológicamente factibles, pero son superadas por las tecnologías de computación distribuida (especialmente las de Cloud Computing) en aspectos como escalabilidad, confiabilidad y tolerancia a fallos.

El análisis de trabajos de autores que utilizan Grid y Cloud para gestionar WSNs, ha permitido concluir que ambas tecnologías son aptas para ser implementadas en el contexto local. Debido a que Cloud presenta ventajas frente a Grid en simplicidad de implementación, confiabilidad y tolerancia a fallos, se recomienda utilizar Cloud para gestionar WSNs en forma remota. Además, debe contemplarse el uso de tecnologías cloud sencillas, de bajo costo económico y de rápida implementación, las que no requieran el uso de estándares como XML en los nodos sensores.

En conclusión, *para diseñar plataformas de gestión remota de WSNs tecnológicamente factibles y económicas, se deben utilizar servicios cloud estándar y de bajo costo.* Entre estas tecnologías se encuentran los servicios de transferencia de archivos (provistos por Google Drive y Dropbox) y los de procesamiento (como los ofrecidos por Amazon EC2 y Google Cloud).

3.4.4. Eficiencia

El último aspecto a analizar en el estado del arte es la eficiencia de las plataformas de gestión remota de WSNs. La eficiencia implica que la plataforma de gestión remota debe cumplir con distintos requerimientos denominados parámetros de eficiencia. Los parámetros de eficiencia pueden ser clasificados en dos grupos: específicos y comunes.

Por un lado, los parámetros específicos dependen de la aplicación a la que serán destinadas las WSNs y la plataforma de gestión remota (por ejemplo: tiempo requerido para detectar la posibilidad de ocurrencia de heladas). Por otro lado, los parámetros comunes son transversales a todas las aplicaciones e implican entre otros el tiempo de entrega de datos, tiempo de cambio de la configuración de la WSN, tasa de pérdidas en la entrega de datos y aumentos del consumo de energía en los nodos sensores debido a la integración a Grid ó Cloud.

Algunas de las propuestas analizadas en el estado del arte [Lim et al., 2007, Humble et al., 2005, Aloisio et al., 2006, Rajesh et al., 2010, Reddy et al., 2007, Annamraju et al., 2010], se focalizan en demostrar que la integración es viable, pero no realizan análisis de eficiencia de la plataforma. Otras propuestas [Ghanem et al., 2004, Corchado et al., 2010, Preve, 2011, Hughes et al., 2008, YuJie et al., 2005, Dunkels, 2003, Islam et al., 2010, Silva et al., 2010, Lee and Hughes, 2010, Zhang et al., 2010, Dunkels et al., 2003], si profundizan en la evaluación de la eficiencia. Al momento de diseñar una plataforma de gestión remota de WSNs, se debe tener en cuenta los parámetros de eficiencia específicos y comunes.

Luego y en base a dichos parámetros, se debe evaluar el desempeño de la plataforma a través de experimentos que cuantifiquen los parámetros de eficiencia logrados (por ejemplo: se logra una tasa de pérdida de datos en el proceso de comunicación menor al 10 %). Esto permitirá validar la eficiencia de la plataforma de gestión remota de WSNs y sus correspondientes módulos de software en base a los resultados obtenidos por experimentación.

Por último, se ha observado que *el diseño de una plataforma de gestión remota de WSNs no sólo implica realizar la gestión remota de los nodos sensores, sino que también debe focalizarse en alcanzar la eficiencia de la plataforma* en base a parámetros como la tasa de pérdida de datos y el consumo de energía en los nodos sensores.

3.5. Conclusiones

En el presente capítulo se ha estudiado el estado del arte sobre la gestión de WSNs desde Internet. El análisis realizado ha permitido detectar que la gestión remota de WSNs desde Internet se resuelve a través del diseño e implementación de plataformas de gestión remota. El proceso de diseño e implementación de plataformas de gestión de WSNs implica llevar a cabo una arquitectura de interconexión para integrar las WSNs a TCP/IP y aplicar tecnologías que permitan la gestión remota de los recursos WSNs desde Internet. Este proceso, debe ser resuelto en forma eficiente, considerando tanto la factibilidad tecnológica como el costo económico de las WSNs a integrar y de las tecnologías a utilizar.

Con el objetivo de diseñar e implementar una plataforma de gestión remota de WSNs, en primer lugar se debe seleccionar cual es el enfoque más adecuado para resolver la arquitectura de integración de WSNs a TCP/IP. Actualmente, existen dos enfoques para incorporar las WSNs a TCP/IP, gateway y redes overlay. De la literatura relevada se puede concluir que el enfoque de gateway es el más adecuado para incorporar las WSNs a TCP/IP debido a que evita retrasos en la entrega de datos y permite trabajar con plataformas de WSN de bajo costo económico. Además brinda la posibilidad de integrar a TCP/IP redes de sensores y estaciones meteorológicas ya instaladas y en funcionamiento.

Una vez resuelta la arquitectura de integración de WSNs a TCP/IP, se requiere determinar que tecnologías serán utilizadas para gestionar remotamente las WSNs. En base a las propuestas de otros autores analizadas en el estado del arte, se ha podido distinguir que existen dos tipos principales de tecnologías de gestión remota de WSNs, las tradicionales ó las de computación distribuida. El análisis realizado ha permitido determinar que si bien las tradicionales son más sencillas de aplicar para desarrolladores familiarizados en su utilización, estas tecnologías no cuentan con mecanismos de alta disponibilidad y

tolerancia a fallos, por lo que no están preparadas para asegurar la escalabilidad de recursos de almacenamiento y procesamiento en base al aumento de nodos sensores. Además, implican altos costos de mantenimiento para asegurar la disponibilidad de los recursos computacionales, las 24 horas del día, los 365 días del año.

A diferencia de las tecnologías tradicionales, las de computación distribuida si están preparadas brindar soporte a la escalabilidad de las WSNs. Si bien los primeros trabajos sobre el uso de tecnologías de computación distribuida para gestionar WSNs utilizaban Grid Computing [Lim et al., 2007, YuJie et al., 2005, Ghanem et al., 2004, Humble et al., 2005, Aloisio et al., 2006, Zhang et al., 2010, Preve, 2011, Hughes et al., 2008], el surgimiento del paradigma Cloud dió lugar a nuevos trabajos [Rajesh et al., 2010, Lee et al., 2010, Lee and Hughes, 2010, Ahmed and Gregory, 2011, Gubbi et al., 2013, Hirafuji et al., 2011, Hori and Kawashima, 2010] que evitan lidiar con la complejidad de la instalación y configuración de los middlewares de Grid Computing. Se puede afirmar entonces que Cloud ha permitido, simplificar el proceso de gestión de WSNs en forma remota, disminuir costos económicos, evitar al usuario el mantenimiento de recursos y abstraerlo de la programación de las WSNs.

Respecto al enfoque utilizado para implementar tecnologías de computación distribuida, algunos autores [Aloisio et al., 2006, Zhang et al., 2010, Preve, 2011, Hughes et al., 2008, Gubbi et al., 2013, Hirafuji et al., 2011, Hori and Kawashima, 2010] programan el middleware Grid ó servicio Cloud en la WSN. Otros autores [Lim et al., 2007, YuJie et al., 2005, Ghanem et al., 2004, Humble et al., 2005, Rajesh et al., 2010, Lee et al., 2010, Lee and Hughes, 2010, Ahmed and Gregory, 2011] no embeben componentes del middleware ó servicio en la WSN.

El enfoque de middlewares ó servicios embebidos en la WSN puede utilizarse junto al enfoque de gateway si solo se embeben componentes del middleware ó servicio en el gateway (no en los nodos) de la WSN. Sin embargo, el enfoque de middleware grid ó servicio cloud *no embebido, es el más adecuado* para gestionar WSNs con gateways debido a que no se programan componentes extras de software en ningún elemento de la WSN, por lo que no se generan consumos extras de recursos de hardware y energía en los nodos.

El análisis realizado del estado del arte ha permitido detectar que existen varias plataformas de gestión remota de WSNs desarrolladas con tecnologías de computación distribuida. Sin embargo la mayoría de estas plataformas no son apropiadas para gestionar WSNs de bajo costo económico (como las disponibles en el mercado local) con recursos acotados de hardware y fuente de energía. Además, la mayoría de estas plataformas utilizan el enfoque de gateways en conjunto con componentes del middleware ó servicio embebidos en en el gateway de la WSN, por lo que se generan cargas extras de consumo y procesamiento en el gateway.

Otro aspecto observado es que las plataformas analizadas generalmente no están diseñadas en base a parámetros de eficiencia comunes y específicos. Finalmente, se ha observado que sólo algunas plataformas basadas en Cloud Computing utilizan servicios cloud estándar, lo cual es necesario para simplificar el diseño e implementación de la plataforma de gestión remota.

En base a los aspectos arriba mencionados, en esta tesis se desarrollará e implementará una plataforma de gestión remota de WSNs basada en gateways y servicios cloud no embebidos en los nodos y gateways de la WSN. Esta plataforma será desarrollada con el objetivo de alcanzar parámetros de eficiencia como tasa de pérdida de datos mínimas en el proceso de comunicación y tiempos acordes a la aplicación para procesar los datos recolectados por las WSNs. Por último, esta plataforma permitirá implementar un Cloud de Sensores de bajo costo económico, tecnológicamente factible y que permita disponer de los recursos las 24 horas al día, los 365 días del año.

Sensor Cirrus

La gestión remota de WSNs permite que el usuario acceda a la información recolectada por los nodos sensores y realice cambios en la configuración de estos dispositivos desde Internet. Esta característica permite al usuario monitorear fenómenos y eventos de distinto tipo (agro-meteorológicos, medioambientales, de seguridad, etc.) sin necesidad de desplazarse al campo donde se encuentran desplegados los sensores.

Actualmente, el modelo de servicios de Cloud Computing Foster et al. [2008], Buyya et al. [2009], Armbrust et al. [2010] ofrece nuevas posibilidades para gestionar los recursos de las WSNs desde Internet. Los usuarios pueden acceder a los recursos de las WSNs en forma sencilla gracias a servicios de software desplegados en infraestructuras altamente confiables, compuestas por recursos computacionales de altas prestaciones y escalables.

A partir del análisis del estado del arte presentado en el Capítulo 3, se detectaron los siguientes aspectos a resolver en las plataformas actuales de gestión remota de WSNs.

- *No son apropiadas para para gestionar las WSNs disponibles en el mercado local.* Las plataformas actuales, están diseñadas para WSNs compuestas por dispositivos costosos como nodos sensores que utilizan Java y Linux, o aquellas WSNs conformadas por nodos no disponibles en el mercado local (como los Motes Berkeley).
- *Sólo algunas plataformas combinan el enfoque de gateway (para integrar las WSNs a TCP/IP) y el de servicios clouds no embebidos en la WSN (para realizar la gestión remota) [Rajesh et al., 2010, Ahmed and Gregory, 2011].* Sin embargo la propuesta de Rajesh et al. [2010] utiliza XML en los nodos sensores por lo que genera consumos extras de procesamiento y memoria en los nodos de la WSN.

Se puede entonces concluir que se deben integrar ambos enfoques para (i) minimizar el consumo de energía en los nodos sensores, (ii) acceder a infraestructuras altamente confiables compuestas por recursos computacionales de altas prestaciones,

(iii) contar con mecanismos de tolerancia a fallos y (iv) brindar escalabilidad. Esto debe ser realizado sin utilizar estándares basados en XML en los nodos sensores debido a que pueden generar cargas extras de procesamiento y memoria en los nodos.

- *Generalmente no están diseñadas en base a parámetros de eficiencia.* Se ha observado que la mayoría de las plataformas analizadas, se focalizan en lograr la gestión remota de WSNs, pero no se orientan a cumplir con parámetros de eficiencia como tiempos mínimos y tasa de error aceptable en la entrega de datos.
- *Simplicidad de diseño e implementación.* Del análisis de la literatura relevada, se detectó que sólo algunas de las plataformas basadas en Cloud Computing utilizan servicios cloud estándar [Lee and Hughes, 2010, Gubbi et al., 2013, Hirafuji et al., 2011, Hori and Kawashima, 2010]. Dichas plataformas utilizan generalmente servicios web para acceder a los datos de la WSN y IaaS para almacenarlos y procesarlos. Respecto al almacenamiento y transferencia de datos de las WSNs, en algunos casos se utilizan servicios de transferencia de archivos sencillos y estándar (como los ofrecidos por Dropbox ¹ y Google Drive). Los servicios cloud de Google Drive y Dropbox permiten compartir, sincronizar y transferir archivos en forma sencilla, confiable y con bajo costo económico. Es por este motivo que los servicios cloud resultan ser una potencial alternativa frente a servicios que requieren un mayor trabajo de programación y desarrollo. Además, dichos servicios pueden ser utilizados para modificar la configuración de la WSN, lo cual hasta donde se ha podido estudiar aún no ha sido implementado en ninguna plataforma de gestión remota de WSNs.

En resumen, se requiere contar con una plataforma de gestión remota de WSNs que permita que (i) los datos de las WSNs sean accedidos, almacenados y procesados en forma sencilla y confiable, (ii) se puedan realizar cambios en la configuración de la WSN, (iii) no se generen consumos extras de energía y recursos de hardware y (iv) se puedan gestionar en forma remota y eficiente WSNs tecnológicamente factibles.

Con el objetivo de solucionar los aspectos mencionados, en esta tesis se propone una nueva plataforma de gestión remota denominada Sensor Cirrus. Esta plataforma tiene una arquitectura basada en gateways y servicios cloud estándar no programados en la WSN. Además Sensor Cirrus brinda soluciones a problemas reales como lo son los daños en cultivos debido a fenómenos agro-meteorológicos como heladas en la Región de Cuyo, Argentina.

El presente capítulo está estructurado de la siguiente manera. En la Sección 4.1 se describe la Hipótesis de Trabajo de la presente Tesis Doctoral. Luego, en la Sección 4.2 se presenta

¹<https://www.dropbox.com/>

la arquitectura de Sensor Cirrus. En la Sección 4.3, se realiza la descripción detallada de cada uno de los módulos que componen Sensor Cirrus. Finalmente en la Sección 4.4 se presentan las conclusiones de este capítulo.

4.1. Hipótesis de Trabajo

A partir del análisis de trabajos previos desarrollado en el Capítulo 3 en esta tesis, se plantea la siguiente Hipótesis de Trabajo:

“Se puede gestionar en forma remota, sencilla y eficiente WSNs conformadas por recursos acotados de hardware y fuente de energía, mediante tecnologías de computación distribuida como Cloud Computing, adecuando los servicios cloud a las WSNs disponibles en el mercado local y con un bajo costo”

4.2. Arquitectura

En la Figura 4.1 se presenta la arquitectura general de Sensor Cirrus. La plataforma tiene una arquitectura basada en dos módulos principales (Módulo WSN - TCP/IP y Módulo TCP/IP - Cloud), un portal web y un servicio de predicción de heladas. El *Módulo WSN - TCP/IP* permite incorporar las WSNs a TCP/IP mediante el enfoque de gateway. Dicho módulo actúa a modo de traductor entre el protocolo nativo de la WSN (ZigBee) y TCP/IP. Respecto al *Módulo TCP/IP - Cloud*, posibilita la gestión remota de los recursos de las WSNs mediante servicios de Cloud Computing *no* programados en la WSN. El *portal web* de Sensor Cirrus permite acceder a los servicios cloud del *Módulo TCP/IP - Cloud* para visualizar los datos de las WSNs, procesarlos y cambiar la configuración de los nodos sensores. Finalmente, el *Servicio de Predicción de Heladas* procesa los datos provenientes de la WSN y emite alarmas de ocurrencia de heladas.

Con el propósito de detallar la arquitectura, se explicará con un mayor nivel de abstracción cada uno de sus módulos y la iteración con los otros elementos de la plataforma tanto para un proceso de adquisición y procesamiento de datos como para un proceso de cambio de configuración de la WSN. Cabe destacar que los módulos, portal web y servicio de predicción de heladas serán explicados con mayor detalle en la Sección 4.3.

El primer módulo de Sensor Cirrus es el *Módulo WSN - TCP/IP* y se ilustra en la Figura 4.2. Este módulo se encuentra programado en la estación base de la WSN la cual puede ser una máquina tipo PC ó un sistema embebido. El proceso de *adquisición de datos* de la WSN comienza cuando el Coordinador de la WSN recibe los datos de los nodos sensores (1). Luego, el Coordinador envía los datos a la estación base vía USB ó RS232 (2).

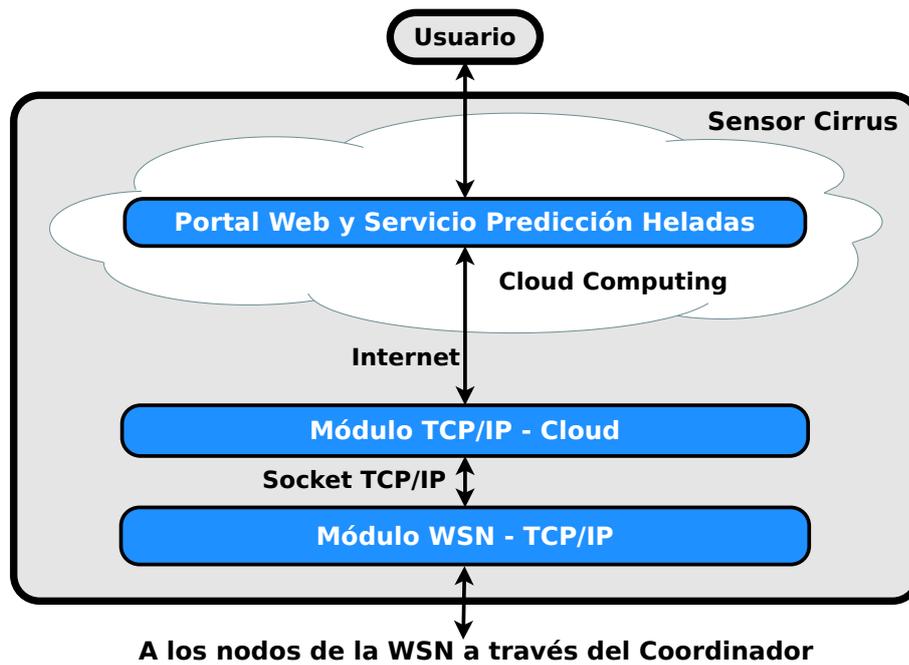


Figura 4.1: Arquitectura de Sensor Cirrus.

El *Módulo WSN - TCP/IP* está compuesto por distintos procesos que permiten en primer lugar la comunicación con el Coordinador ZigBee (3) y en segundo lugar decodificar y extraer los datos encapsulados en tramas ZigBee (4). Finalmente, los datos extraídos de las tramas ZigBee son encapsulados en tramas TCP/IP (5) y enviados a una máquina remota la cual contiene el *Módulo TCP/IP - Cloud*. La comunicación entre ambos módulos es realizada a través de un socket TCP/IP sobre redes 802.11 ó Ethernet (6).

El segundo módulo de Sensor Cirrus, denominado *Módulo TCP/IP - Cloud* se ilustra en la Figura 4.3. Este módulo es el responsable de recibir las tramas provenientes de

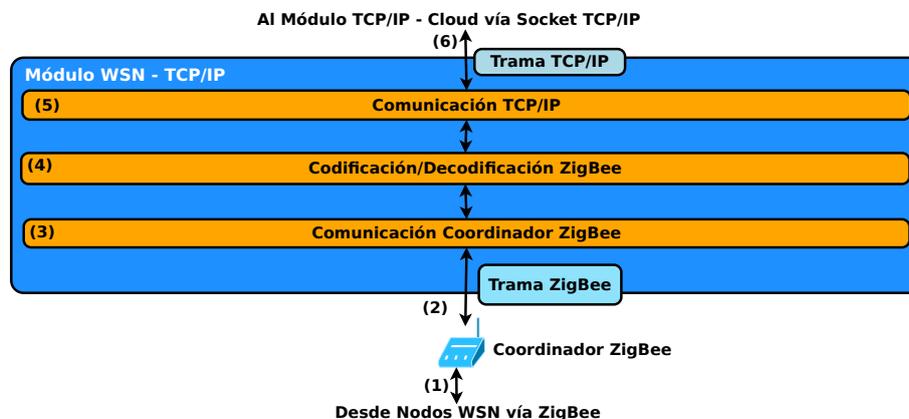


Figure 4.2: Módulo WSN - TCP/IP.

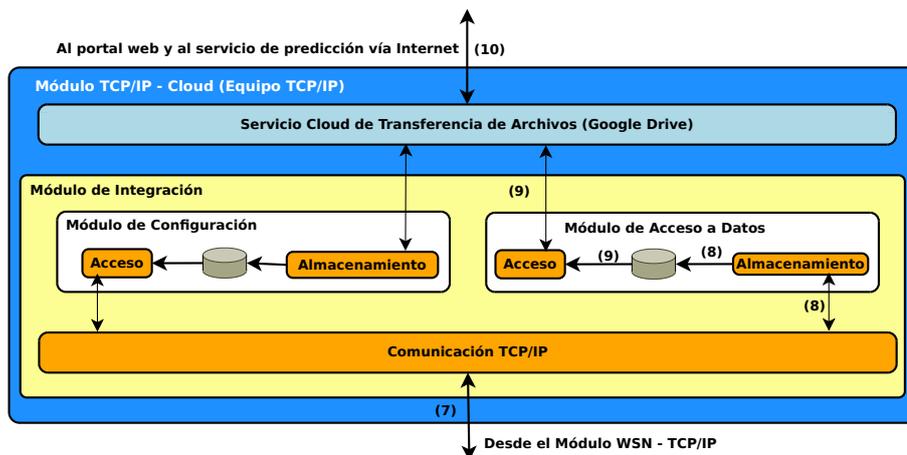


Figure 4.3: Módulo TCP/IP - Cloud.

la estación base vía Socket TCP/IP (7) y está compuesto por el *Módulo de Integración* y un *Servicio Cloud de Transferencia de Archivos Estándar*. El *Módulo de Integración* tiene dos sub-módulos, el de *Acceso a Datos* y el de *Configuración*. En el *proceso de adquisición de datos de la WSN*, sólo interviene el sub-módulo de *Acceso a Datos*. Este sub-módulo realiza dos procesos principales. El primer proceso es denominado *Almacenamiento* (8) y tiene como objetivo generar un archivo con los datos provenientes de la WSN. A continuación, un segundo proceso denominado *Acceso* (9) permite que un servicio cloud estándar de transferencia de archivos (como Google Drive) acceda al archivo generado por el proceso de *Almacenamiento* y lo exporte al Cloud (10) vía Internet.

El usuario final puede acceder a los datos de la WSN a través de un portal web alojado en el Cloud ó también mediante un dispositivo móvil Android que tenga instalado Google Drive. Además en dicho portal se puede observar si hay alarmas de ocurrencia de heladas. Dichas alarmas son emitidas por el *Servicio de Predicción de Heladas* de Sensor Cirrus, el cual accede a los datos de la WSN vía Google Drive y los procesa con instancias de Amazon EC2.

En forma similar a como se lleva a cabo el *proceso de adquisición de datos de la WSN*, se realiza también el proceso de *cambio en la configuración de la WSN*. Con el fin de explicar el proceso de cambio de configuración, se detallará como se realiza el cambio en la frecuencia de muestreo de los nodos.

En primer lugar, el usuario accede al *portal web* y envía la petición de cambio de frecuencia al *Módulo TCP/IP - Cloud* a través de un formulario web y un script alojado en un Cloud público. En segundo lugar, el *Módulo TCP/IP - Cloud* recibe la petición del usuario por intermedio de Google Drive. Luego, el proceso de *Almacenamiento* del sub-módulo de *Configuración* genera un archivo con el pedido de cambio de configuración. Finalmente, el proceso de *Acceso* pone el archivo a disposición del proceso de Comuni-

cación TCP/IP para que se envíe al *Módulo WSN - TCP/IP* a través de un socket TCP/IP

El *Módulo WSN - TCP/IP* extrae del archivo el nuevo valor de frecuencia de muestreo y el ID del nodo que será reprogramado. Luego, encapsula dichos valores en una trama ZigBee y los envía al Coordinador de la WSN. Finalmente el nodo recibe vía Coordinador ZigBee el nuevo valor de frecuencia de muestreo y se lleva a cambio la reprogramación.

4.3. Sensor Cirrus: Descripción de Módulos

En la presente sección se detallan los módulos y elementos que componen a Sensor Cirrus. En primer lugar, en el Apartado 4.3.1 se presenta la WSN Experimental que será gestionada mediante Sensor Cirrus desde Internet. Luego, en las Subsecciones 4.3.2, 4.3.3, 4.3.4 y 4.3.5 se detalla el *Módulo WSN - TCP/IP*, *Módulo TCP/IP - Cloud*, *Portal Web* y *Servicio de Predicción de Heladas*, respectivamente.

4.3.1. WSN Experimental

La red de sensores que se utiliza está destinada a la medición de variables agro-meteorológicas (mas específicamente temperatura y humedad). Cada nodo de la WSN está compuesto por un microcontrolador (Atmega) en una plataforma Arduino Pro [Spa, 2014] y un radio XBee ZB [Digi International Inc., 2012] que utiliza ZigBee como protocolo de comunicación. Además, cada nodo cuenta con un gabinete estanco IP 65, con sensores de temperatura, humedad y su correspondiente batería. La Figura 4.4 muestra el nodo utilizado para implementar la WSN. Cabe destacar que el nodo ha sido desarrollado con tecnologías disponibles en el mercado local, por lo que se ha cumplido con lo establecido en la Hipótesis de la presente tesis doctoral.

La WSN posee routers ZigBee que permiten aumentar la distancia máxima de comunicación entre los nodos y la estación base. La estación base de la WSN está compuesta por el Coordinador ZigBee y una computadora tipo PC. Además, la estación base está ubicada en un punto de la finca que tiene acceso a fuentes permanentes de energía eléctrica e Internet.

La Figura 4.5, muestra la arquitectura de la WSN a gestionar mediante tecnologías Cloud. Debido a que la finca se encuentra ubicada en una zona rural de la Provincia de Mendoza cercana a polos urbanos, la conexión a Internet se realiza con un enlace punto a punto.

La finca tiene frutales y está ubicada en el departamento de General Alvear, al sur de la Provincia de Mendoza. El principal objetivo de las mediciones es obtener información para determinar la probabilidad de ocurrencia de fenómenos agro-meteorológicos que

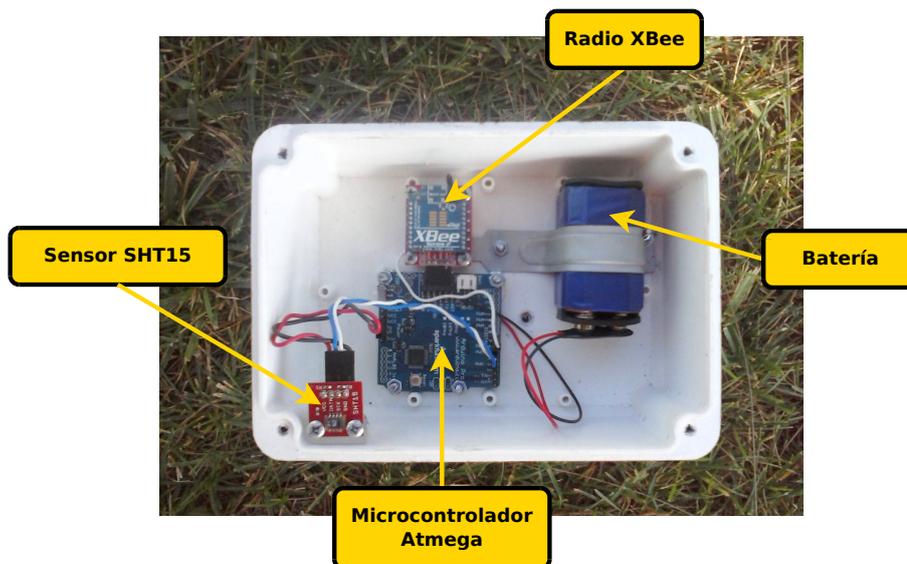


Figura 4.4: Nodo Sensor WSN Experimental.

puedan dañar a los cultivos. Para el caso particular de la presente tesis, se ha focalizado el uso de la WSN y Sensor Cirrus para el estudio de heladas y la prevención de los daños que dicho fenómeno causa en los cultivos.

Respecto al funcionamiento de la WSN, el *proceso de adquisición de datos* se presenta en la Figura 4.6. El proceso comienza cuando el microcontrolador del nodo realiza la adquisición de los datos del sensor. El firmware de los nodos sensores se denomina *Módulo WSN*. El *Módulo WSN* lleva a cabo una serie de procesos que realizan la adquisición de datos a intervalos programados. La adquisición puede ser en forma digital, por ejemplo vía I2C (1),(2) ó analógica (1*), (2*). En caso de que la adquisición se realice de manera analógica, el *Módulo WSN* debe digitalizar la señal para luego transmitir los datos al radio ZigBee (2*).

Una vez realizada la adquisición, el *Módulo WSN* envía una señal de activación al radio (3). Una vez activado, el microcontrolador le envía los datos al radio (4). La comunicación microcontrolador-radio se realiza con una conexión del tipo Universal Asynchronous Receiver/Transmitter (UART). Luego de recibir los datos, el radio los encapsula en tramas ZigBee (5) y transmite las tramas en forma inalámbrica al Coordinador de la WSN (6). La transmisión es en la banda de 2,4 Ghz y puede ser realizada en forma directa o a través de routers ZigBee.

4.3.2. Módulo WSN - TCP/IP

El primer componente de Sensor Cirrus es el *Módulo WSN - TCP/IP* que se ilustra en la Figura 4.7c. Dicho módulo de software está programado en la estación base de la WSN



Figura 4.5: Gestión de una WSN mediante Sensor Cirrus.

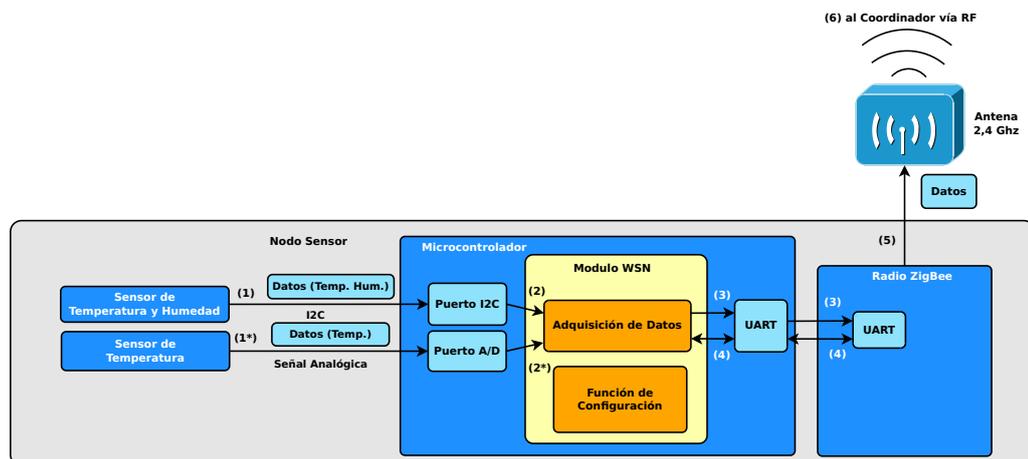


Figura 4.6: Adquisición de Datos en el Nodo Sensor.

y permite compatibilizar el protocolo de la WSN experimental (ZigBee) con TCP/IP. El *Módulo WSN - TCP/IP* está desarrollado en base al enfoque de gateway y ha sido programado en Python sobre Debian 7. Además, puede ser fácilmente migrado a otros sistemas operativos (Windows) y distribuciones Linux (Android, Linux Embebido, etc.). El funcionamiento del *Módulo WSN - TCP/IP* es el siguiente:

La Figura 4.7a, ilustra los dos procesos iniciales del módulo. En (1) se establece la comunicación con el Coordinador ZigBee a través del puerto USB. Luego, en (2) se crean dos sockets TCP/IP con el equipo remoto donde está programado el *Módulo TCP/IP - Cloud*, un socket para recibir los pedidos de cambio de configuración (por ejemplo variar la frecuencia de muestreo) y otro para enviar los datos de la WSN al Cloud.

Una vez establecida la comunicación con el *Módulo TCP/IP - Cloud* y con el Coordinador ZigBee, se recibe la primer trama de dato de un nodo de la WSN (ver Figura 4.7b). La trama proviene de uno de los nodos sensores de la WSN vía ZigBee. El proceso *Comunicación con el Coordinador ZigBee* recibe la trama por el puerto USB de la estación base (3) y la envía al proceso *Codificación/Decodificación ZigBee* (3 b). Dicho proceso, extrae el ID del nodo sensor (del cual proviene la trama) y consulta en un archivo el valor (fijado por el usuario) de período de frecuencia de muestreo del nodo (4).

A continuación, se codifica el valor de frecuencia de muestreo (5) a ZigBee (6) (Ver Figura 4.7c) y luego en (7) se envía al nodo vía Coordinador ZigBee. Dicho proceso debe realizarse haya cambiado o no el valor de frecuencia de muestreo. Esto es así debido a que el nodo espera algunos milisegundos la respuesta del Coordinador, y en caso de no recibir respuesta, el firmware del nodo fija por defecto el valor de frecuencia de muestreo más rápido (una adquisición cada 8 segundos) para evitar así pérdida de datos. Luego, el nodo termina el enlace ZigBee con el Coordinador y entra en modo bajo consumo para minimizar el uso de la batería.

Por último, la Figura 4.7d muestra como el proceso *Codificación/Decodificación ZigBee* envía el dato y el ID del nodo sensor al proceso *Comunicación TCP/IP* (8) y envía el dato al *Módulo TCP/IP - Cloud* vía socket TCP/IP (9).

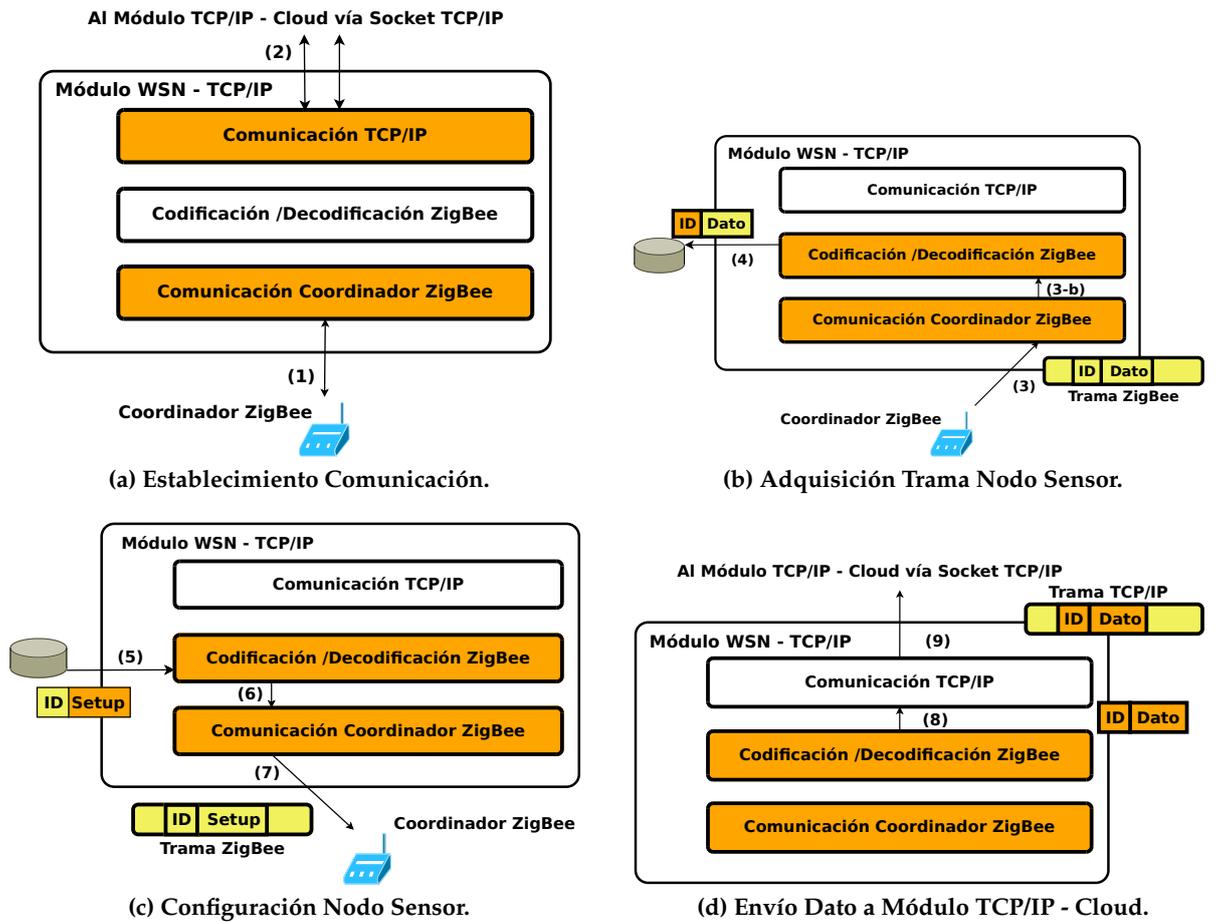


Figure 4.7: Módulo WSN - TCP/IP.

4.3.3. Módulo TCP/IP - Cloud

La Figura 4.8 ilustra el segundo componente de Sensor Cirrus, el *Módulo TCP/IP - Cloud*. Este módulo tiene dos componentes bien diferenciados el *Módulo de Integración* y un servicio Cloud estándar de transferencia y sincronización de archivos (Google Drive).

Por un lado, el *Módulo de Integración* está compuesto por dos sub-módulos (*Configuración* y *Acceso a Datos*) y un proceso denominado *Comunicación TCP/IP*. Estos componentes han sido desarrollados en Python y programados en una PC con sistema operativo Windows 7 externa a la WSN. Además, pueden ser migrados a otras plataformas y sistemas operativos.

Por otro lado, el servicio Cloud ha sido implementado en la misma PC mediante un cliente de Google Drive estándar. Respecto a la migración a otros sistemas operativos, actualmente el servicio cliente de Google Drive sólo puede ser implementado en Windows,

MAC OS y Android. Sin embargo existen alternativas de código abierto para Linux como Grive².

La Figura 4.8 ilustra el funcionamiento del *Módulo TCP/IP - Cloud*. En primer lugar (Ver Figura 4.8a), el *Módulo Comunicación TCP/IP* abre dos sockets TCP/IP con la estación base de la WSN. Los sockets permiten (1) recibir los datos adquiridos por los sensores y (2) enviar pedidos de cambio de configuración a la WSN.

Posteriormente, la Figura 4.8b muestra como el *Módulo Acceso a Datos* recibe los datos provenientes de la estación base (3) y los almacena en archivos (4). Al recibir nuevos datos desde la WSN, se registran cambios en dichos archivos (5) y se sincronizan con el Cloud a través de Google Drive (6).

La Figura 4.8c ilustra el proceso de gestión de los pedidos de cambio de configuración en los nodos sensores. El proceso comienza cuando se almacenan en un archivo de configuración de la WSN alojado en el Cloud (7) los cambios realizados por el usuario en el portal web. Luego se sincroniza el archivo del Cloud con los archivos de configuración guardados en la máquina que contiene el *Módulo TCP/IP - Cloud* vía Google Drive (8).

Finalmente, el *Módulo Cambio de Configuración* (Ver Figura 4.8d) verifica periódicamente el último instante en el que se ha modificado el archivo de configuración de los nodos (9). En caso de que se registren cambios (10), se detecta que ha habido una solicitud de cambio de configuración y se transmite a la estación base el nuevo valor de configuración de los nodos por intermedio del *Módulo Comunicación TCP/IP* (11).

²<http://www.lbreda.com/grive/start>

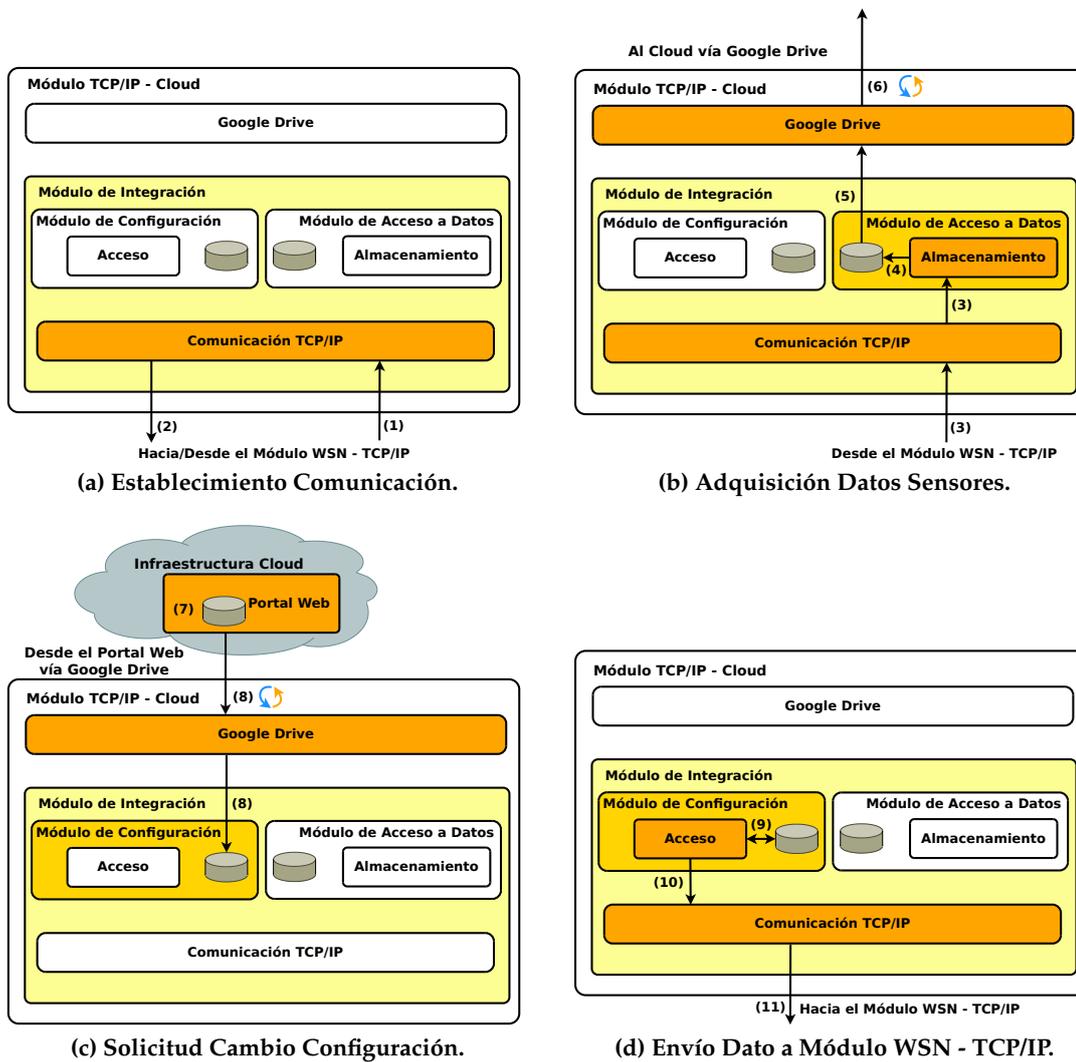


Figure 4.8: Funcionamiento Módulo TCP/IP - Cloud

4.3.4. Portal Web

El *portal web* de Sensor Cirrus permite acceder a la plataforma para visualizar los datos provenientes de las WSNs y realizar cambios de configuración en los nodos sensores. El portal ha sido desarrollado con los Toolkits de Google Sites y Google Drive. Además, el *portal web* se encuentra alojado en el Cloud público de Google, el cual permite alojamiento gratuito, acceso a recursos computacionales de altas prestaciones, visualización en dispositivos Android y disponibilidad del sitio las 24 horas del día, los 365 días del año. La Figura 4.9 muestra el mapa del portal web de Sensor Cirrus.



Figura 4.9: Portal Web Sensor Cirrus.

El portal está estructurado en cinco secciones: Hardware, Arquitectura, Acceso en Tiempo Real a WSNs, Alertas de Heladas y Configuración WSNs.

La Sección *Hardware* incluye una descripción de los nodos sensores gestionados por la plataforma. Actualmente, Sensor Cirrus permite la gestión remota de nodos compuestos por un sistema embebido Arduino Pro 328 [Spa, 2014], un radio XBee - ZigBee [Digi International Inc., 2012] y un sensor de temperatura y humedad digital Sensirion SHT15 [Sen, 2011].

En la Sección *Arquitectura*, se detalla la arquitectura de la plataforma. Además se explica el funcionamiento de los distintos módulos que componen a Sensor Cirrus a través de los procesos de *adquisición de datos* y *cambio de frecuencia de muestreo*.

La Sección *Acceso en Tiempo Real a WSNs* permite visualizar los datos recolectados en campo por las WSNs. En dicha sección se incluyen gráficos que muestran la temperatura y humedad medidas por los nodos sensores y la hora a la que fueron registrados (Ver Figura 4.10a y Figura 4.10b). Para simplificar la visualización, los valores de temperatura, humedad y hora de adquisición para un punto específico del gráfico pueden verse al pasar el puntero del mouse sobre el valor de interés.

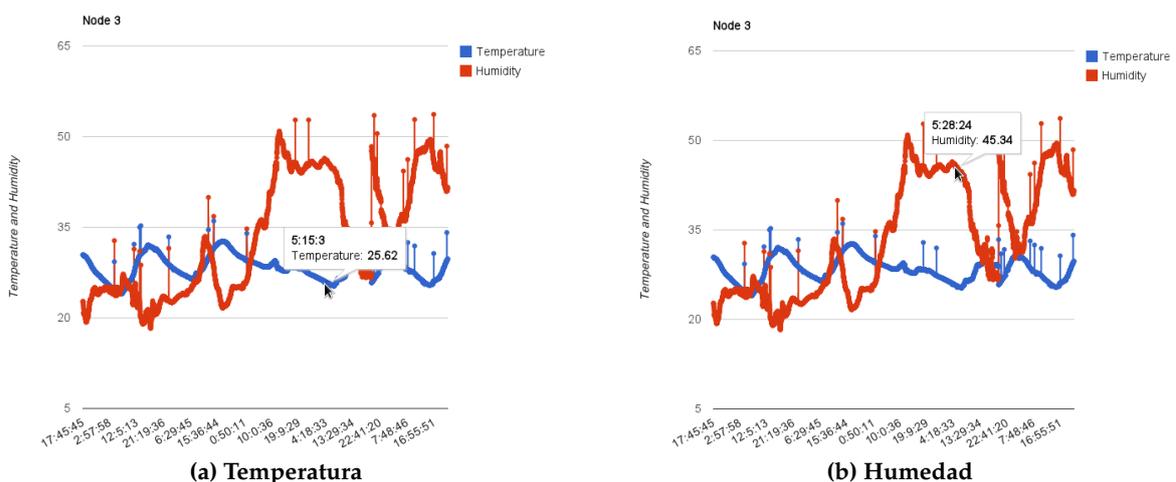


Figura 4.10: Gráficos de Datos Recolectados por la WSN.

Para realizar la visualización de datos, se utilizaron scripts de Google. Gracias a la ayuda del SDK de Google Scripts, se pueden desarrollar programas en Javascript que luego

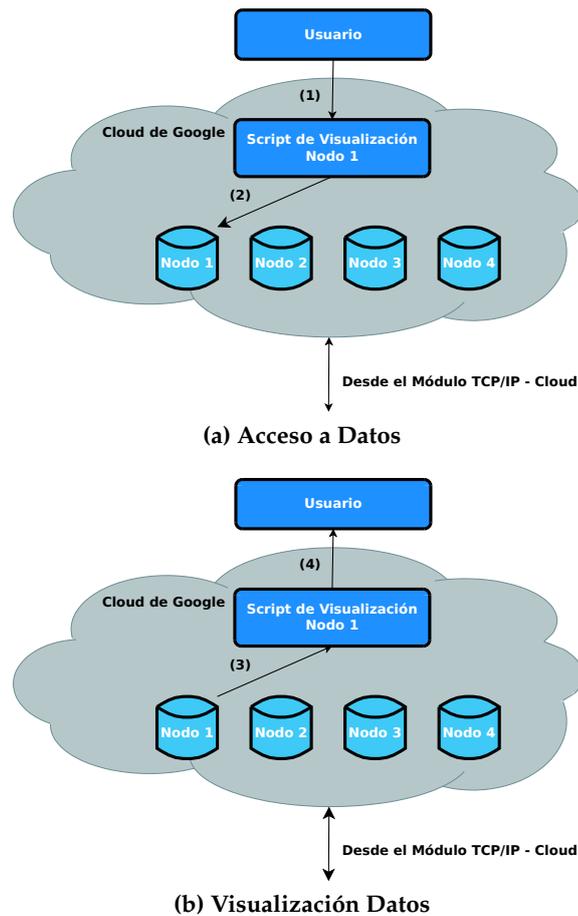


Figura 4.11: Proceso de Visualización de Datos.

se embeben en sitios alojados en el Cloud de Google. Además, los scripts de Google permiten interactuar con Google Drive para gestionar los archivos de los nodos sensores. La Figura 4.11 detalla el funcionamiento de los scripts desarrollados para visualizar los datos recolectados por los nodos sensores de la WSN.

El proceso de *visualización de datos* se ilustra en la Figura 4.11a. Este proceso comienza cuando el usuario accede a la Sección *Acceso en Tiempo Real a WSNs* del portal web (1). Al acceder a la sección, se activa un script que accede a la lista de archivos de Sensor Cirrus alojados en el Cloud de Google (2). Luego, en la Figura 4.11b, el script encuentra el archivo de datos (3) (por ejemplo el del nodo 1), accede al contenido y (4) genera los gráficos en el portal web. De igual forma, el proceso de acceso y visualización se repite para los otros nodos que componen la WSN (nodo 2, nodo 3, etc.).

La Sección *Alertas de Heladas* muestra al usuario los resultados del *Servicio de Predicción de Heladas* de Sensor Cirrus. Este servicio permite predecir heladas en base a información extraída de los datos recolectados por la WSN. Una vez activada la alarma por

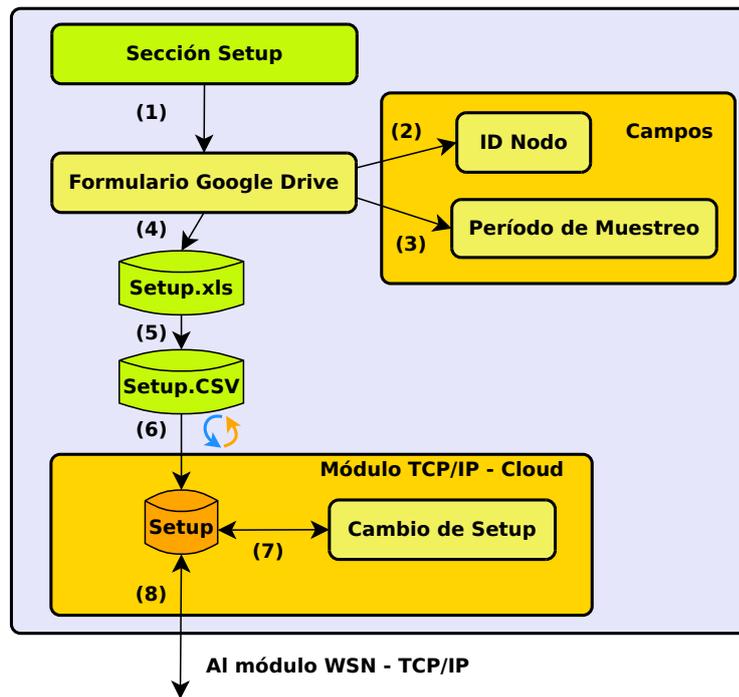


Figura 4.12: Cambio de Configuración desde Portal Web

el *Servicio de Predicción*, el Ingeniero Agrónomo debe acceder a los datos recolectados en tiempo real (*Acceso en Tiempo Real a WSNs*) para activar el sistema de defensa de heladas (calefactores, riego, etc.). El *Servicio de Predicción* es una herramienta que complementa a la adquisición de datos en tiempo real ya que permite anticipar la logística del sistema de guardia y defensa contra heladas. Cabe destacar que el servicio de predicción será explicado en profundidad en el apartado 4.3.5.

La *Sección Configuración WSNs* permite realizar cambios en la configuración de los nodos sensores. La Figura 4.12 ilustra el *proceso de cambio de configuración*. En primer lugar (1) el usuario accede a la *Sección Setup* del portal web de Sensor Cirrus. Luego, el usuario llena los siguientes campos en un formulario de Google Drive: (2) ID Nodo (nodo sensor al cual se le realizará el cambio de configuración) e (3) intervalo de muestreo (el que puede ser de 1, 5, 10 ó 15 minutos). Finalmente envía el formulario y los datos son almacenados en el Cloud de Google Drive en una planilla de cálculo (4). La planilla contiene un script que al detectar un nuevo valor de configuración genera un archivo CSV en Google Drive (5). A continuación, el archivo CSV es sincronizado en el *Módulo TCP/IP - Cloud* de Sensor Cirrus (6). Por último, al detectarse un cambio en el archivo de configuración (7), el nuevo valor es enviado (vía socket TCP/IP) al *Módulo WSN - TCP/IP* (8) y a la WSN.

4.3.5. Servicio de Predicción de Heladas

En la presente sección, se detalla el *Servicio de Predicción de Heladas (SPH)* de Sensor Cirrus. El principal objetivo del SPH es predecir la *temperatura mínima* que sucederá en la noche para determinar si hay probabilidad de que ocurran heladas en la finca donde está desplegada la WSN. Gracias a la predicción realizada, los productores pueden activar sistemas de defensa contra heladas y minimizar los daños en los cultivos. La presente sección está organizada de la siguiente manera: en la Subsección 4.3.5.1, se presenta el método de predicción de heladas utilizado para desarrollar el SPH y en la Subsección 4.3.5.2, se detalla la implementación de dicho servicio.

4.3.5.1. Método de Predicción de Heladas

El servicio de predicción de heladas de Sensor Cirrus ha sido desarrollado a partir del método de predicción de heladas presentado por [Snyder and Melo-Abreu, 2005]. El método se basa en la ecuación de predicción de temperatura mínima de [Allen, 1957] y permite calcular la temperatura mínima que sucederá en la noche.

Si bien el método permite predecir heladas, sólo es aplicable en noches con cielo despejado y sin la presencia de frentes fríos. Por lo tanto, sólo es aplicable a la predicción de heladas por radiación. Cabe destacar que otro tipo de heladas (como las debidas a frentes fríos) son predichas por la Dirección de Agricultura y Contingencias Climáticas de la Provincia de Mendoza y pueden ser consultadas por los productores a través del portal web de contingencias de la Provincia de Mendoza³.

Para calcular la temperatura mínima, el método de Snyder y de Melo-Abreu utiliza datos extraídos de un archivo histórico que contiene diez años de registros de temperatura, humedad, punto de rocío y estado del cielo. El método toma del archivo una muestra con datos correspondientes a cincuenta días (de cualquier año, pero del mes al cual pertenece el día en el que se realizará la predicción) en los que hayan sucedido heladas por radiación.

Formalmente, la temperatura mínima se calcula con la siguiente ecuación de regresión lineal (RL):

$$T_p = s_T * T_o + s_D * D_0 + i, \quad (4.1)$$

donde T_p es la temperatura mínima a predecir, T_o y D_0 son la temperatura mínima y el punto de rocío adquiridos el día de la predicción, dos horas después de la puesta del sol

³<http://contingencias.mendoza.gov.ar/>

(hora denominada como “hora cero”), respectivamente. Finalmente, i es la intercepción de la RL, s_T la pendiente de temperatura y s_D la pendiente de punto de rocío.

Por una parte, los valores de s_T e i , se calculan a partir de las ecuaciones (2) y (3), respectivamente.

$$s_T = \frac{\sum(T_{h0} - \bar{T}_{h0})(T_m - \bar{T}_m)}{\sum(T_{h0} - \bar{T}_{h0})^2}, \quad (4.2)$$

$$i = \frac{\sum T_m - s_T \sum T_{h0}}{n}, \quad (4.3)$$

donde T_{h0} son las temperaturas históricas de los cincuenta días registradas a la *hora cero*, T_m la temperatura mínima registrada en la noche, y n la cantidad de datos de la muestra. Finalmente, \bar{T}_{h0} y \bar{T}_m son los promedios de todos los valores de T_{h0} y T_m de la muestra.

Por otra parte, el valor de la pendiente s_D se calcula mediante la ecuación (4).

$$s_D = \frac{\sum(D_{h0} - \bar{D}_{h0})(R - \bar{R})}{\sum(D_{h0} - \bar{D}_{h0})^2}, \quad (4.4)$$

donde D_{h0} son los puntos de rocío históricos registrados a la *hora cero* y R los residuos. Los parámetros \bar{D}_{h0} y \bar{R} son los promedios de los valores de D_{h0} y R , de la muestra. Finalmente el residuo R se calcula a través de la ecuación (5):

$$R = T_m - s_T * T_o + i \quad (4.5)$$

4.3.5.2. Implementación del Servicio de Predicción

Con el objetivo de implementar el *Servicio de Predicción de Heladas* de Sensor Cirrus, se implementó el método de Snyder y de Melo-Abreu utilizando Java y MySQL. En particular, se utilizó MySQL para, (i) generar bases de datos con los registros históricos de la WSN, (ii) acceder a las bases de datos y (iii) almacenar los resultados de la predicción realizada por el *Servicio de Predicción de Heladas*. Finalmente, se utilizaron instancias de Amazon EC2 para ejecutar el SPH en el Cloud.

La Figura 4.13 resume el funcionamiento global del *Servicio de Predicción de Heladas*. En el proceso (1) la información recolectada por los nodos sensores en campo (disponibles en archivos CSV en Google Drive) se almacena en bases de datos MySQL alojadas en instancias de Amazon EC2. Luego, en el proceso (2), el SPH realiza una consulta a la base de datos para generar la muestra de cincuenta días detallada en la Subsección 4.3.5.1. Dicha muestra contiene todos los datos recolectados por la WSN (temperatura, humedad,

radiación solar, etc.). Luego, en el proceso (3), la aplicación extrae de la muestra sólo los datos requeridos por el método de Snyder y de Melo-Abreu para predecir la temperatura mínima (T_o , D_o , etc.). Finalmente, en (4) se ejecuta el método de predicción de heladas y se obtienen las predicciones de temperatura mínima (5) por cada nodo sensor procesado.

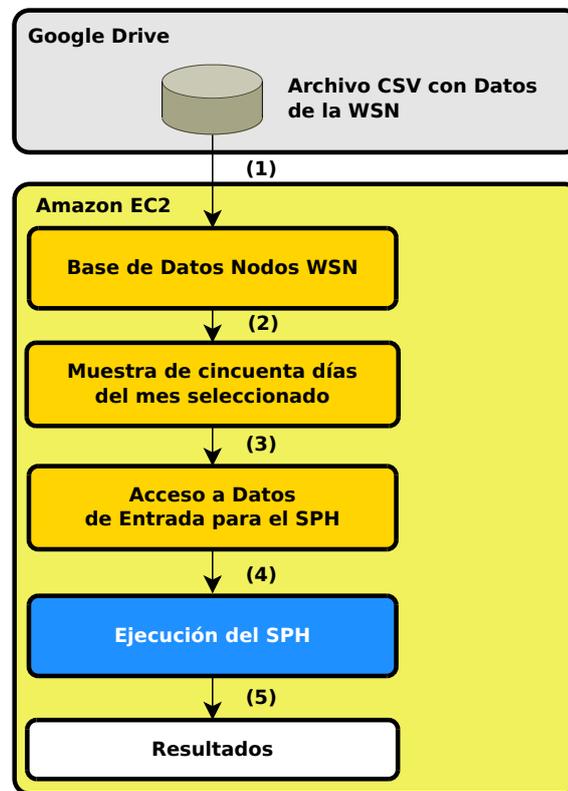


Figure 4.13: Servicio de Predicción de Heladas.

4.4. Conclusiones

El análisis del estado del arte realizado en el Capítulo 3, permitió detectar una serie de aspectos a resolver en las plataformas actuales de gestión remota de WSNs. En primer lugar, se observó que la mayoría de las plataformas desarrolladas por otros autores no son adecuadas para gestionar las WSNs disponibles en el mercado local. Además, sólo algunas plataformas que combinan el enfoque de gateway (para integrar las WSNs a TCP/IP) y el de servicios Cloud no embebidos en la WSN (para realizar la gestión remota). Las ventajas de la combinación de ambos enfoques han sido detalladas al principio del presente capítulo.

Otro aspecto a resolver es que generalmente las plataformas de gestión remota de WSNs no están diseñadas en base a parámetros de eficiencia como tasas de errores aceptable

en la entrega de datos de la WSN. Si bien la mayoría de las plataformas analizadas se focalizan en lograr la gestión remota de WSNs, no se orientan a cumplir con los requerimientos de eficiencia mencionados.

Además, el análisis realizado permitió detectar que la mayoría de las propuestas presentadas por otros autores utilizan servicios Cloud ad-hoc para gestionar los datos de la WSN y no servicios Cloud estándar de transferencia de archivos como Dropbox ó Google Drive.

Con el objetivo de proveer una solución a los aspectos mencionados, en este capítulo se ha presentado una nueva plataforma de gestión remota de WSNs denominada Sensor Cirrus. Uno de los principales aportes de Sensor Cirrus es que combina el enfoque de gateway y de servicios Cloud no embebidos en la WSN. Gracias a la combinación de ambos enfoques se puede minimizar el consumo de energía en los nodos sensores, acceder a infraestructuras confiables de bajo costo económico compuestas por recursos computacionales de altas prestaciones, disponer de mecanismos de tolerancia a fallos y brindar recursos computacionales y de almacenamiento acordes a las necesidades del usuario.

Otro de los aportes de la plataforma es que ha sido diseñada e implementada en base a parámetros de eficiencia como tiempos mínimos y tasas de error aceptables (hasta 10 %) en la entrega de datos. Respecto a la factibilidad tecnológica, cabe destacar que Sensor Cirrus permite integrar plataformas de WSNs de bajo costo y disponibles en el mercado local. Además, se le pueden añadir nuevos módulos de software a la plataforma que permitan acceder a datos de WSNs, estaciones meteorológicas y otros instrumentos ya instalados y en funcionamiento.

A diferencia de las plataformas analizadas en el estado del arte, en Sensor Cirrus se utilizaron sólo servicios Cloud estándar. Para almacenar los datos de las WSNs y transferirlos se utilizó Google Drive. Luego, se emplearon Scripts de Google para visualizar y realizar cambios de configuración de la WSN desde un portal web alojado en Google Sites. Por último, se procesaron los datos mediante instancias de Amazon EC2 para emitir alarmas de ocurrencia de heladas. Las principales ventajas de utilizar servicios Cloud estándar son entre otras la simplicidad de uso, de programación y la gran cantidad de herramientas de desarrollo disponible.

Sensor Cirrus ha sido provista de un Servicio de Predicción de Heladas. Este servicio calcula la temperatura mínima que sucederá en la noche y determina si pueden ocurrir heladas que causen daños en cultivos. Finalmente, Sensor Cirrus permitirá a los productores de la región acceder a tecnologías que eviten pérdidas económicas en los cultivos y contribuyan al desarrollo local.

Posteriormente, en el Capítulo 6, se detallarán los experimentos realizados para evaluar y validar los distintos componentes de Sensor Cirrus.

Experimentos en la WSN

En el presente capítulo, se realizan distintos experimentos con plataformas de WSN. El principal objetivo es analizar las plataformas de hardware de WSNs disponibles en el mercado argentino, estudiar su comportamiento en campo y seleccionar las más adecuadas para implementar una WSN experimental. Dicha WSN se aplicará al estudio de fenómenos agro-meteorológicos en la Provincia de Mendoza, más específicamente al problema de heladas. Además, la WSN permitirá realizar los experimentos necesarios para validar que es posible *gestionar en forma remota, sencilla y eficiente WSNs conformadas por recursos acotados de hardware y fuente de energía, mediante tecnologías de computación distribuída como Cloud Computing*.

Este capítulo está estructurado de la siguiente forma, en la Sección 5.1 se desarrolla un resumen de distintas tecnologías disponibles en el mercado local para ser utilizadas en redes de sensores y sus características fundamentales. Además, se implementa un método de envío de datos para evaluar y comparar el consumo de energía en dos plataformas ZigBee disponibles en el mercado local. Luego, en la Sección 5.2, se estudia la influencia de la potencia de transmisión en el consumo de energía y la distancia de comunicación confiable en las potenciales plataformas de hardware a ser utilizadas en la WSN experimental. Finalmente, en la Sección 5.3 se realizan experimentos para determinar como las condiciones de trabajo en campo (posición de la antena, altura desde el suelo y presencia de obstáculos) afectan la comunicación entre nodos sensores.

5.1. Estudio de Plataformas de Hardware

Actualmente, existen en el mercado distintas plataformas de hardware que pueden ser utilizadas para implementar WSNs. Para seleccionar cuál es la plataforma más adecuada para implementar una WSN, se deben estudiar y comparar las características técnicas

de las plataformas. Como se discutió en el Capítulo 2 Sección 2.1 de la presente tesis, las plataformas de WSN deben cumplir con una serie de requisitos como bajo consumo, reconfiguración, bajo mantenimiento, etc [Akyildiz and Vuran, 2010]. Para el caso particular de WSNs aplicadas a monitoreo agrícola y ambiental, las plataformas de hardware deben contar con suficientes recursos de memoria y procesamiento para implementar aplicaciones de adquisición de datos y cantidades de interfaces analógicas y digitales (I2C, UART, SPI) para acceder a sensores de distinto tipo (temperatura, humedad, radiación solar, etc.). Además deben contar con el tamaño adecuado para su despliegue y manipulación en campo [Abbasi et al., 2014].

En la presente sección, se estudian las plataformas de WSNs disponibles en el mercado local. En primer lugar, se presentan las plataformas y sus principales características técnicas. Luego, se realizan ensayos de consumo sobre dos plataformas de hardware y se determina cual es la más adecuada para implementar la WSN. Los resultados de las pruebas realizadas en la presente sección fueron reportados en Iacono et al. [2010].

5.1.1. Nodos Sensores Disponibles en el Mercado Local

Las plataformas de hardware descritas en este apartado implementan la capa física del estándar IEEE 802.15.4 y poseen suficientes recursos de hardware para implementar ZigBee y aplicaciones de sensado remoto. Además, están preparadas para trabajar en modo de bajo consumo, sus potencias en modo transmisión son de aproximadamente 2 mW, tienen alcances de hasta 200 metros en espacios abiertos (con línea de visión) y son de bajo costo.

Freescale

Freescale Semiconductor Inc. posee dos circuitos integrados para ser usados en redes de sensores inalámbricos, el MC13192 [Freescale Semiconductor, 2008], y el MC13213 [Freescale Semiconductor, 2009]. Por un lado, el MC13213 integra en un mismo chip un microcontrolador de la familia HCS08 y un transceptor de la familia MC1320x. Por otro lado, el MC13192 sólo incluye el transceptor y el fabricante aconseja utilizarlo en conjunto con microcontroladores de la familia HCS08.

Los transceptores mencionados implementan la capa física del estándar IEEE 802.15.4. Debido a que ambos transceptores trabajan con microcontroladores Freescale de la familia HCS08 (en un mismo chip ó vía interfaz UART), cuentan con los recursos suficientes de memoria y procesamiento para soportar ZigBee.

Digi

Digi International Inc. produce varios dispositivos aptos para el diseño de WSN. En el presente trabajo, se analizarán los Xbee ZB y Xbee-PRO ZB ya que ambos implementan el protocolo ZigBee y están disponibles en el mercado local. La principal diferencia entre ambos es la potencia de transmisión y el alcance (200 metros para el Xbee ZB y 2 Km para el Xbee PRO ZB). Respecto al conjunto microcontrolador - transceptor, las plataformas de Digi poseen un chip Ember EM250 compuesto por un microcontrolador y un transceptor 802.15.4. Cabe destacar que debido a su firmware, solo pueden configurarse algunos de sus parámetros, por lo que para aplicaciones más complejas se requiere un microcontrolador de uso general a través de una interfaz serie.

Texas Instruments

Texas Instruments Inc. fabrica dos circuitos integrados para WSNs: CC2430 y CC2480. El CC2430 [Tex, 2007] es un dispositivo que incluye en un mismo chip un transceptor CC2420 (que implementa el estándar IEEE 802.15.4) y un microcontrolador 8051. Este dispositivo está preparado para soportar la pila de protocolos ZigBee. Respecto al CC2480 [Tex, 2008], el mismo es un transceptor ZigBee, el cual se comunica con cualquier microcontrolador utilizando una interfaz UART.

Crossbow

Crossbow Technology Inc. provee las siguientes plataformas: TPR2400 (más conocido como TELOS-B), y MICAz. Ambas plataformas están basadas en Motes Berkeley y poseen el transceptor CC2420 de Texas Instruments. Por un lado, la plataforma TELOS-B utiliza el microprocesador MSP430 y está pensado para ensayos de laboratorio y enseñanza. Por otro lado, el MICAz utiliza un microprocesador ATMEGA 128L. La principal ventaja de la plataforma TELOS-B es la tensión de alimentación mínima. Dicha tensión es de 1.8 V, la cual aumenta la vida útil de las baterías ya que éstas entregarán energía casi hasta su tensión de corte.

La Tabla 5.1 muestra los parámetros significativos para evaluar la aptitud de una plataforma para ser utilizada como nodo en una WSN aplicada a sensado remoto. Cada fila corresponde a una plataforma de hardware de WSN. La primera columna de la Tabla 5.1 indica el fabricante y la segunda el modelo de cada plataforma analizada. Luego, las columnas tres a ocho muestran los principales parámetros que pueden ser utilizados para comparar las plataformas, es decir potencia máxima, tensión de alimentación, protocolo de

comunicación, consumo en modo hibernación ó *sleep* y sensibilidad del radio. Los criterios de selección de los parámetros significativos utilizados en la comparativa, han sido discutidos en detalle por los siguientes autores [Barrenetxea et al., 2008, Zhang, 2009].

5.1.2. Ensayos de Consumo

Uno de los parámetros más significativos utilizados para evaluar el funcionamiento de las plataformas de WSN es el consumo de energía. En este apartado se realiza un ensayo de consumo promedio para evaluar plataformas de WSNs disponibles en el mercado local y seleccionar las más adecuadas para implementar la WSN.

5.1.2.1. Metodología Experimental

El consumo promedio se define como el valor medio de la corriente eficaz consumida durante períodos de transmisión ó recepción de datos, intercalados entre períodos de bajo consumo [Iacono et al., 2010]. El experimento a realizar consiste en medir el consumo de energía en los nodos sensores en tres casos distintos. El primer escenario es cuando el nodo sensor transmite ó recibe una trama de 2 bytes de datos. Este caso simula un envío ó recepción de comandos que permiten modificar la configuración de los nodos sensores en forma remota. El segundo caso es la transmisión de una trama de datos de 20 bytes, lo cual simula el envío de datos como temperatura y humedad del nodo sensor al Coordinador. Por último, el tercer escenario donde se medirá el consumo de energía es en el proceso de asociación del nodo sensor con el Coordinador.

El ensayo se realiza en dos nodos sensores disponibles en el mercado, los módulos XBee ZB y el conjunto Freescale MC13192 - MC9S08GT32. Se han utilizado estas plataformas debido a que son las que tienen menor costo económico. Para medir los consumos de las plataformas citadas se emplea un osciloscopio Fluke 196C y un multímetro Fluke 189. Además, se debe utilizar una resistencia serie de 10 ohm para obtener (por intermedio de su caída de tensión) las formas de onda de corriente en el osciloscopio. Para realizar el ensayo, todos los nodos fueron colocados en una misma mesa del laboratorio. Los ensayos fueron realizados utilizando la máxima potencia de transmisión.

5.1.2.2. Resultados Experimentales

En este apartado se muestran los resultados obtenidos en cada una de las plataformas de WSN consideradas. En la Figura 5.1, se muestran las formas de onda de corriente consumida en la plataforma XBee. La Figura 5.1a corresponde a la transmisión de 2 bytes y la Figura 5.1b a la transmisión de 20 bytes. Luego, la Figura 5.1c ilustra la forma de onda

Tabla 5.1: Características de Plataformas de Nodos Sensores Disponibles en el Mercado Local.

Fabricante	Modelo	Potencia Máxima [mW]	Tensión de Alimentación [V]	Protocolo Embebido	Entradas Analógicas	Consumo Modo Hibernación [µA]	Sensibilidad [dBm]
Digi	Xbee ZB	2	2,1 - 3,6	ZigBee	4	< 1	-95
Digi	Xbee ZB PRO	10	2,1 - 3,6	ZigBee	4	3,5	-102
Freescale	MC13192	2,29	2,0 - 3,4	IEEE 802.15.4	No posee ^d	1	-92
Freescale	MC13213	2	2,0 - 3,4	IEEE 802.15.4	8	1	-92
Texas Instrument	CC2430	1,14	2,0 - 3,6	IEEE 802.15.4	8	0,5	-92
Texas Instrument	CC2480	1	2,0 - 3,4	ZigBee 2006	2	0,5	-92
Crossbow	TELOS-B (TPR2400)	1	1,8	IEEE 802.15.4	8	5,1	-90
Crossbow	MICAZ	1	1,8 - 3,6	IEEE 802.15.4	8	< 15	-94

^dLa plataforma requiere de un microcontrolador externo, con lo cual posee las entradas analógicas y digitales propias del microcontrolador.

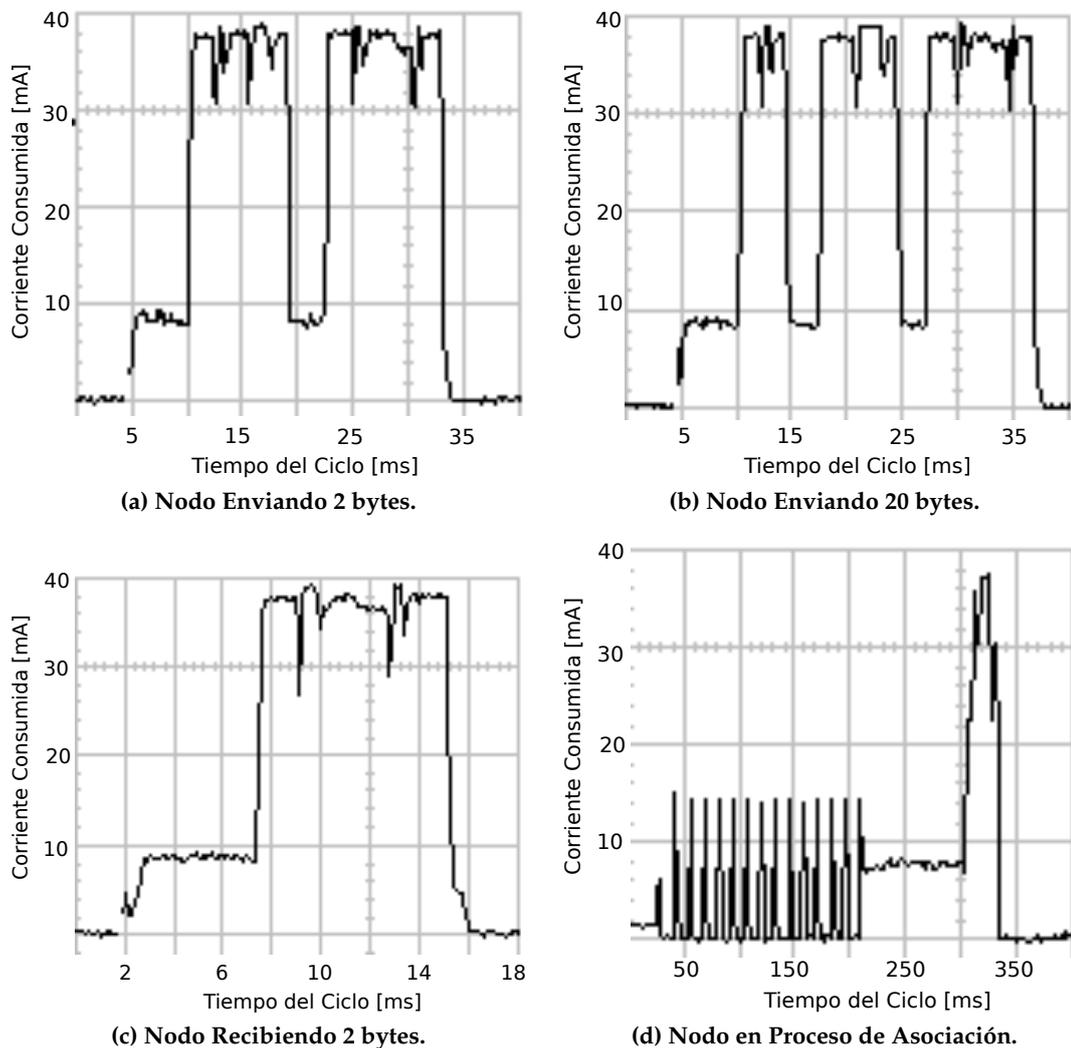


Figura 5.1: Ensayo de Consumo Plataforma XBee.

cuando el nodo recibe 2 bytes desde el Coordinador. Por último, la Figura 5.1d detalla la forma de onda durante el proceso de asociación del nodo con el Coordinador.

Como se puede observar, las Figuras 5.1a, 5.1b, 5.1c y 5.1d muestran picos de corriente de alrededor de 40 mA. Luego, dicho valor será utilizado junto al de corriente en modo hibernación para estimar la corriente promedio consumida por los nodos sensores. Debido a que el consumo máximo se produce durante el proceso de asociación (la duración del proceso de asociación es de aproximadamente 300 ms.), es necesario evitar que se produzcan re-asociaciones.

Finalmente se mide la corriente en modo hibernación. Este valor de corriente se debe adquirir mediante el multímetro por que es menor a 1 μA y no puede registrarse con la resistencia serie y el osciloscopio.

La Figura 5.2, ilustra el ensayo de consumo realizado en la plataforma Freescale. La Figura 5.2a detalla la forma de onda en el nodo para la transmisión de 2 bytes y la Figura 5.2b para la transmisión de 20 bytes. Finalmente, en la Figura 5.2c se muestra la forma de onda cuando el nodo recibe 2 bytes del Coordinador. Cabe destacar que para la plataforma Freescale no se ha podido medir la asociación porque la misma requiere intervención manual y los tiempos de reacción del operador afectan significativamente la medición.

Como se puede observar, las Figuras 5.2a, 5.2b, 5.2c muestran picos de corriente de alrededor de 41 mA. Luego, se mide con el multímetro la corriente en modo hibernación la cual es de aproximadamente 200 μ A. Es necesario aclarar que el fabricante indica que el microcontrolador consume 550 η A en modo hibernación STOP2 [Freescale Semiconductor, 2004] y el transceptor consume 1 μ A en modo hibernación [Freescale Semiconductor, 2008]. Sin embargo, con la aplicación utilizada en el ensayo no se han logrado los valores indicados por el fabricante de corriente en modo hibernación.

Con el objetivo de realizar una estimación del consumo de los nodos, se calcula la corriente promedio consumida por los mismos. Para ello, se considera una transmisión de 20 bytes (como las mostradas en las Figuras 5.1b y 5.2b) cada 5 minutos. Entre dichas transmisiones, el nodo permanece en modo hibernación.

De los tres valores de corriente de bajo consumo declarados por los fabricantes, se considera el valor máximo. Esto es así porque se pretende realizar un cálculo conservador. Dichos valores son de 15,5 μ A para la plataforma Freescale y de 1 μ A para la XBee. Para calcular la corriente promedio se deben aproximar las formas de onda como se muestra en las Figuras 5.3a y 5.3b e integrar el área bajo las curvas resultantes de la aproximación. Finalmente, las corrientes promedios obtenidas resultan 17,5 μ A para Freescale y 4,23 μ A para XBee.

Los resultados muestran que la principal ventaja del nodo Freescale es que posee un microcontrolador de uso general que puede ser utilizado para implementar el software de adquisición de datos requerido por la WSN. Sin embargo, los principales inconvenientes de este nodo son el mayor consumo de energía y que el microcontrolador ocupa gran parte de su capacidad de procesamiento y almacenamiento para implementar el protocolo ZigBee. Además, este nodo requiere mayores tiempos de programación que el nodo XBee y entornos de desarrollo de software pagos para implementar el protocolo ZigBee.

Respecto al módulo XBee, se ha comprobado que tiene menor consumo de energía que el nodo Freescale y permite desarrollar WSNs ZigBee en forma más rápida debido a la facilidad de programación. El principal problema de esta plataforma es que requiere un microcontrolador externo para implementar el software de adquisición de datos de la WSN. Sin embargo, actualmente existen en el mercado sistemas embebidos sencillos, económicos y con librerías de software especialmente diseñadas para trabajar con los módulos XBee que pueden ser utilizados para implementar el software de adquisición

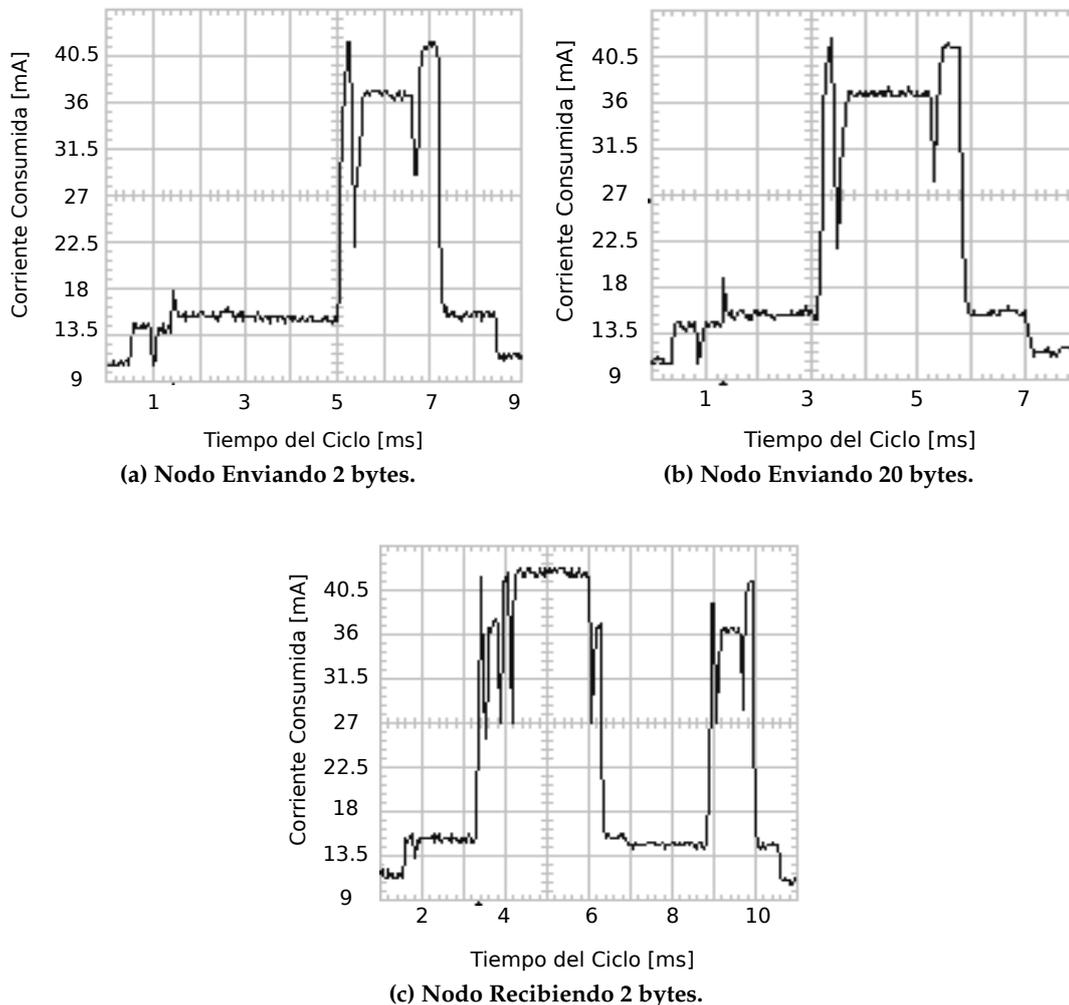


Figura 5.2: Ensayo de la Plataforma Freescale.

de datos. Estos sistemas embebidos son los de la línea Arduino y pueden ser consultados en [Spa, 2014].

5.2. Efecto de la Potencia de Transmisión en la Vida Útil y Distancia de Comunicación Confiable.

En la presente sección se realizan experimentos para medir la influencia de la potencia de transmisión tanto en la vida útil de los nodos sensores como así también en la distancia de comunicación confiable entre dos nodos. Ambos experimentos se aplican a las plataformas XBee Series 2 y Freescale conjunto MC13192 + MC9S08GT32. Cabe destacar que los experimentos detallados en esta sección dieron lugar a la publicación presentada

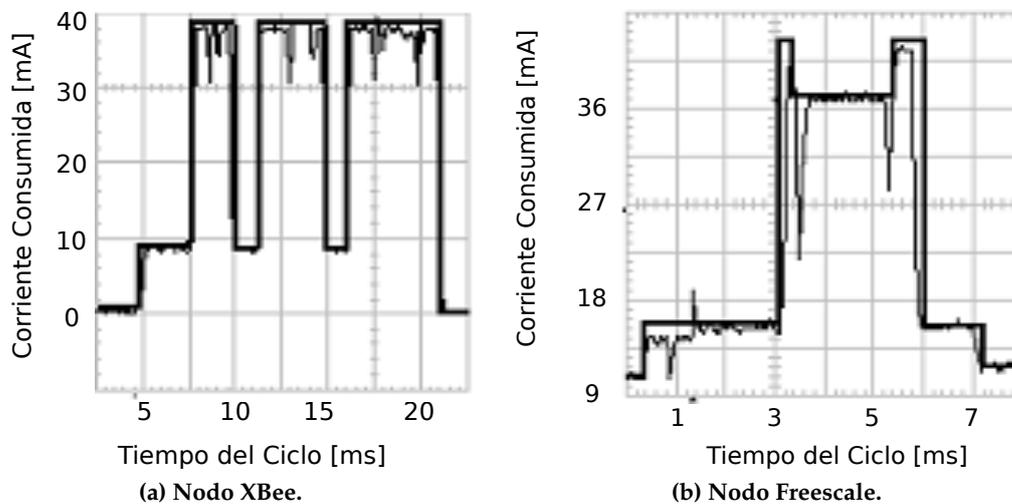


Figura 5.3: Cálculo del Consumo.

en [Godoy et al., 2010].

5.2.1. Ensayo de Consumo y Vida Útil

La variación de la potencia de transmisión en los transceptores de los nodos sensores permite aumentar el alcance del radio enlace. Sin embargo, este aumento de potencia implica mayores consumos de energía y disminución de la vida útil de las baterías de los nodos. En esta subsección, se realizarán experimentos para medir la influencia de la potencia de transmisión en el consumo y la vida útil de la batería de los nodos sensores.

5.2.1.1. Metodología experimental

Para llevar a cabo el ensayo se transmiten datos simulados de temperatura, humedad y radiación solar adquiridos por sensores disponibles en el mercado argentino. La transmisión de datos se ha realizado con distintos niveles de potencias de transmisión: +3 dBm, 1 dBm y -7 dBm. Debido a que cada fabricante define sus propios niveles de potencia en que pueden transmitir los nodos, se utilizan los 3 valores de potencia más cercanos a los propuestos para cada plataforma de hardware.

La WSN ZigBee utilizada para realizar el experimento está compuesta por un Coordinador, un Router y un nodo sensor del tipo End Device. Dicha WSN se despliega en condiciones controladas de laboratorio, es decir con los dispositivos ZigBee ubicados en la misma mesa de trabajo, con línea de visión entre antenas y sin gabinetes de protección contra intemperie.

El ensayo se realiza de la siguiente manera. En primer lugar, se simula la lectura periódica y el envío de los datos de los 3 sensores (temperatura, humedad y radiación solar) mediante la transmisión de 8 bytes encapsulados en una trama ZigBee a través del programa hyperterminal desde el End Device (y vía Router) hacia el Coordinador. Luego, se obtienen las formas de onda de corriente durante un reporte del End Device enviando esos 8 bytes de datos y se verifica la recepción correcta de estos datos en el Coordinador. Finalmente, se mide la corriente en modo bajo consumo mediante un osciloscopio digital Fluke 196C.

Para estudiar los efectos de los distintos niveles de potencia de transmisión en la vida útil del nodo, se realiza la medición de corriente para +3 dBm y -7 dBm. En cada nivel de potencia se calcula la corriente promedio de consumo del nodo I_{AV} (formado por el conjunto microcontrolador-transceptor-sensores) considerando un ciclo de funcionamiento definido por cuatro estados: Hibernación, Sensado, Adquisición y Reporte. El ciclo de funcionamiento dura 10 minutos, dicho tiempo del ciclo es llamado T_M .

La Tabla 5.2 detalla cada uno de los estados del ciclo de funcionamiento. Cada fila corresponde a un estado. La columna uno indica el nombre del estado del ciclo. Las columnas dos y tres muestran el modo de funcionamiento del microcontrolador y de la radio, respectivamente. Luego, la columna cuatro indica si los sensores están apagados ó encendidos para cada estado. Finalmente las columnas cinco y seis muestran la duración de cada estado para los nodos XBee y Freescale, respectivamente.

La corriente I_{AV} , está compuesta por la corriente promedio de cada elemento en el nodo (la cuál varía en cada estado) y se calcula utilizando la ecuación 5.1:

$$I_{AV} = \left[\frac{1}{T_M} \right] * \left\{ \sum_{i=1}^n t_i * \left[\sum_{j=1}^m (n_j * I_{ij}) \right] \right\}, \quad (5.1)$$

donde I_{AV} es la corriente promedio consumida por el nodo en el ciclo correspondiente, n_j la cantidad de elementos del mismo tipo, t_i es el tiempo de duración de cada estado, I_{ij} la corriente típica consumida en el estado i por cada elemento j del nodo, T_M es el tiempo de duración del ciclo, i corresponde al número de estado y j indica el elemento del nodo.

Para hacer el cálculo de la vida útil de un nodo sensor se toma el valor de la corriente promedio I_{AV} y se calcula el siguiente tiempo de vida T_v :

$$T_V = \frac{Q_B}{I_{AV}}, \quad (5.2)$$

donde Q_B es la carga de la fuente de energía del nodo sensor (dos pilas AAA de 1,5 Volts) y su valor es de 1.000 mAh en condiciones ideales. El valor de Q_B se reemplaza en la ecuación 5.2 y se obtiene el resultado de duración del nodo en años.

Tabla 5.2: Estados del Ciclo de Funcionamiento del Nodo.

Estados	Microcontrolador	Radio	Sensores	Duración XBee [s]	Duración Freescale [s]
Reporte	Ejecutar	Ejecutar	Apagados	0,039	0,0068
Sensado	Detener	Hibernación	Encendidos	5	5
Adquisición	Ejecutar	Hibernación	Encendidos	0,210	0,210
Hibernación ^a	Detener	Hibernación	Apagados	594,751	594,784

^aEl nodo se encuentra en modo bajo consumo y se considera que consume la corriente de modo bajo consumo típica declarada por el fabricante.

5.2.1.2. Resultados Experimentales

En esta subsección se muestran los resultados de consumo y vida útil obtenidos para los nodos XBee y Freescale.

Nodo Xbee Los experimentos han sido realizados sobre los nodos XBee Series 2 fabricados por la empresa Digi Digi International Inc. [2012]. Dichos nodos, se configuran con el software X-CTU ¹ provisto por el fabricante y requieren de un microcontrolador adicional para realizar la lectura de sensores que trabajen con protocolo I2C o SPI. En este experimento se ha utilizado un microcontrolador Freescale MC9S08GT32 [Freescale Semiconductor, 2004].

Con el objetivo de realizar la comunicación a través del Router Zigbee, el Coordinador se programa de modo que acepte nuevos End Devices durante solo 6 segundos. Luego, de este tiempo el Coordinador sólo aceptará asociaciones de nodos vía Routers. Motivo por el cual, los End Devices se programan para que se activen después de 6 segundos.

El Coordinador se conecta a una PC en la cuál se utiliza el software X-CTU, para verificar el funcionamiento de la red durante el desarrollo del ensayo. La visualización y medición de la corriente en el Osciloscopio, se realiza mediante una resistencia de 10 Ohm conectada en serie. Respecto al nivel de potencia de transmisión del Router y Coordinador, ambos se programan para transmitir a 3 dBm.

Las Figuras 5.4a y 5.4b muestran las forma de onda de corriente para +3 dBm y -7 dBm, respectivamente. Como se puede observar en ambas figuras, las corrientes consumidas por los nodos sensores son muy similares. Debido a esta observación es que se hará el cálculo de vida útil considerando solamente consumos máximos.

En base a la ecuación 5.1, se calcula la corriente I_{AV} consumida por el nodo XBee, la cual es de 10,1 μ A. Finalmente se reemplaza el valor calculado de la corriente I_{AV} en la ecuación 5.2 y se obtiene el tiempo de vida T_V , el cual es de aproximadamente 11 años.

¹<http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/xctu>

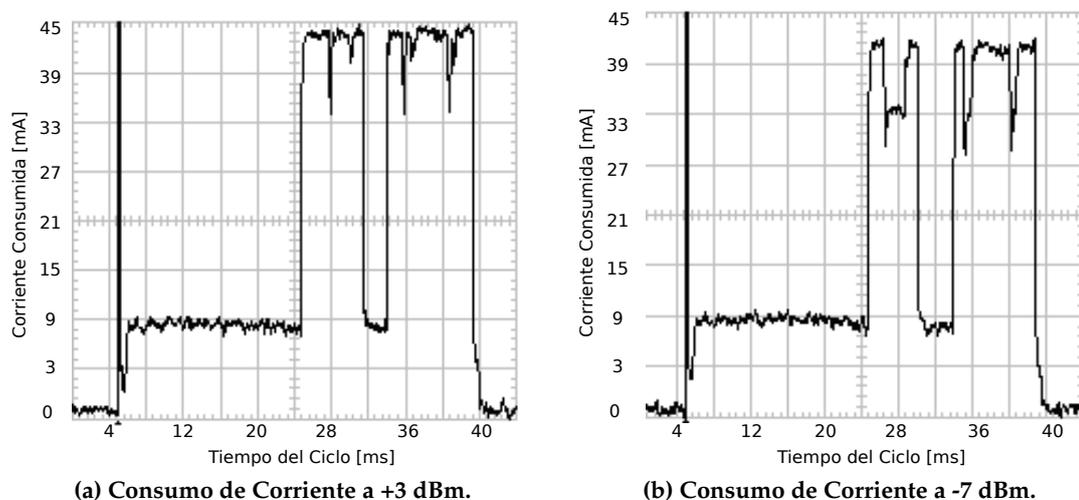


Figura 5.4: Ensayo de Vida Útil XBee.

Nodo Freescale Para llevar a cabo las pruebas con la plataforma Freescale se utilizan las placas Zigbee V 1.0, las cuales están conformadas por un microcontrolador Freescale MC9S08GT32 y un transceptor MC13192. La plataforma es fabricada por la empresa Electrocomponentes S.A. y se configura mediante el software BeeKit ² y el entorno de desarrollo Freescale Codewarrior ³. Además, se utiliza en este experimento una PC con Hyperterminal para visualizar los datos provenientes de los End Devices vía Router y Coordinador Zigbee.

Para asegurar que la comunicación es realizada a través de Routers, primero se asocia el Coordinador con el Router y luego el End Device. Respecto a la potencia de transmisión, el Router y el Coordinador son configurados para trabajar a máxima potencia.

En el ensayo de consumo se probaron los distintos consumos para los dos niveles de potencias más cercanos a los indicados para la plataforma XBee. En el caso de Freescale los niveles de potencia son de +2.9 dBm y -7 dBm.

Las Figura 5.5a muestra la corriente consumida cuando el nodo funciona a +2.9 dBm y la Figura 5.5b ilustra el consumo para el nodo programado a -7 dBm. Si se comparan ambas figuras, se observa que la corriente consumida es similar para cada uno de los niveles de potencia utilizados. Dado que las variaciones en la potencia utilizada para transmitir datos no afectarán significativamente el consumo (y por tanto la vida útil de los nodos), se hará un cálculo de vida útil considerando solamente consumos máximos.

Cabe aclarar que el nodo tiene un consumo de corriente continua permanente de 12 mA, esto es así debido a que la placa utiliza un adaptador de señales MAX232 [Tex, 2015] y

²www.freescale.com/beekit

³www.freescale.com/codewarrior

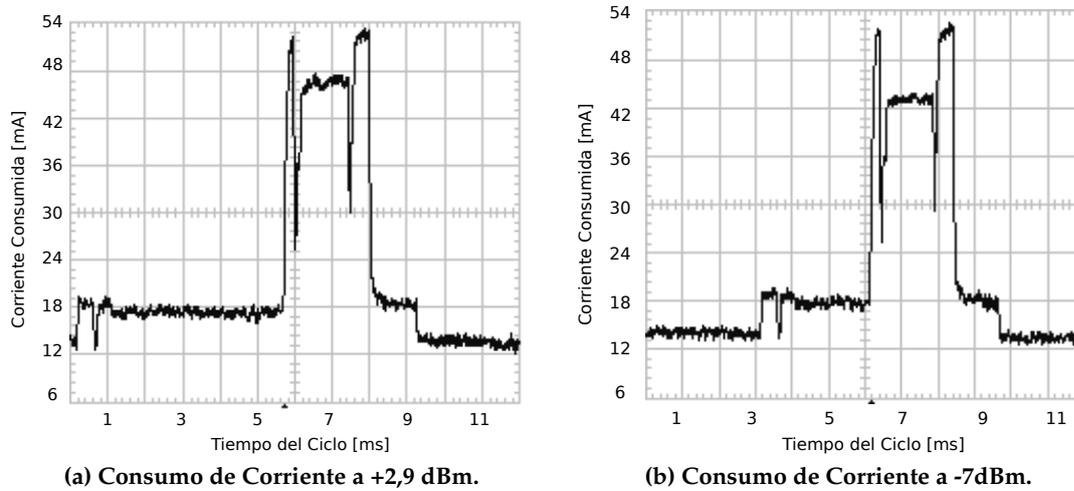


Figura 5.5: Ensayo de Vida Útil Freescale.

dos reguladores de tensión, los cuáles son los responsables de este consumo. Debido a que en caso de trabajo en campo estos elementos no serán necesarios, no serán tomados en cuenta para el cálculo de la vida útil.

Para calcular la vida útil de los nodos, se utilizó el mismo procedimiento que con las plataformas XBee. Se consideraron los tiempos para cada estado del nodo Freescale detallados en la Tabla 5.2 y se realizaron los cálculos con las ecuaciones 5.1 y 5.2. Finalmente, la corriente I_{AV} calculada para el nodo Freescale es de $9,35 \mu A$ y el tiempo de vida T_V es de aproximadamente 12 años.

Los resultados obtenidos en ambas plataformas, muestran que el nodo XBee tendrá una vida útil de aproximadamente 11 años y el nodo Freescale de aproximadamente 12 años. Cabe destacar que estos resultados surgen de considerar una batería ideal y una toma de muestra cada diez minutos, por lo que debe considerarse que en condiciones reales la batería tiene una duración mucho menor. Como referencia a estos resultados, se menciona que la empresa National Instruments declara una duración de baterías de más de tres años para sus nodos NI WSN-3202 [Nat, 2015], a un régimen de toma de una muestra por minuto.

A partir de los ensayos realizados, se concluye que si se varía la potencia de transmisión (manteniendo constante los demás parámetros) no se afecta significativamente la vida útil de los nodos. En cuanto a los sensores, se debe considerar que los sensores de humedad necesitan un tiempo de alrededor de 5 segundos para realizar una medición, lo cual implica una fuente importante de consumo.

5.2.2. Ensayo de Distancia de Comunicación Confiable

La potencia de transmisión influye sobre la máxima distancia a la cual pueden comunicarse dos nodos sensores por que si se aumenta el nivel de potencia de transmisión, se logran mayores distancias de comunicación. Sin embargo, una gran potencia de transmisión repercute en el consumo de energía del nodo y la vida útil de sus baterías, por lo que no siempre es factible utilizar el mayor nivel de potencia en el transceptor. En esta subsección se realizan experimentos para determinar cual es el nivel de potencia de transmisión que permite realizar la comunicación en forma confiable sin generar grandes aumentos de consumo de energía.

5.2.2.1. Metodología Experimental

Para calcular la distancia de comunicación confiable (DCC) se deben considerar dos aspectos, la máxima distancia a la que puede estar el End Device de un Router o Coordinador sin perder la asociación y la tasa de paquetes entregados ó PDR (siglas en inglés de Packet Delivery Rate) en la banda de 2.4 GHz.

En base a los aspectos mencionados, se puede definir a la distancia de comunicación confiable como *la distancia máxima a la cual pueden estar dos nodos, sin perder asociación entre ellos, y que la PDR esté por encima del 80 %* [Li et al., 2010]. Para calcular la PDR para una distancia dada, se transmite un determinado número de paquetes y se utiliza la siguiente ecuación:

$$PDR = \left(\frac{N_r}{N_t} \right) \cdot 100, \quad (5.3)$$

donde N_t es el número de paquetes transmitidos por el transmisor y N_r el número de paquetes recibidos sin error por el receptor.

El ensayo se realiza utilizando un End Device, un Router y un Coordinador. Para ello, se verifica que el End Device esté asociado al Router y no al Coordinador apagando el router y observando que se pierde la comunicación entre el Coordinador y el End Device. La distancia entre el Router y el Coordinador se mantiene igual a 2 metros.

Luego de formar la red, se comienza el experimento aumentando la distancia entre el End Device y el Router, hasta alcanzar la distancia máxima en la cual permanecen asociados. Una vez determinada dicha distancia, se procede a verificar si se cumple el criterio de $PDR > 80 \%$. Si se cumple, se toma esta distancia como distancia de comunicación confiable. Si no se cumple el criterio de $PDR > 80 \%$, entonces se disminuye la distancia hasta lograr que se cumpla el mismo.

Tabla 5.3: Características de los Sensores Simulados.

Modelo de Sensor	Variable Medida	Consumo [μ A]	Tiempo de Sensado [s]
LM75	Temperatura	280	0,3
SHT71	Temperatura y Humedad	550	5
BPW41N	Radiación Solar	No requieren ^a	0

^aNo consume corriente, por lo que el consumo es 0 y el tiempo de adquisición no es relevante.

El ensayo se lleva a cabo en un espacio abierto, con línea de visión entre los nodos y además, la altura del Coordinador es de 1,80 metros y la del End Device 1,5 metros. Los experimentos se realizan sobre las mismas plataformas donde se implementó el ensayo de consumo y de vida útil descrito en la Subsección 5.1 (XBee y Freescale). Para cada plataforma se utiliza la condición más favorable para la antena provista por el fabricante (monopolo externo en el caso de XBee y dipolo impreso en Freescale). En ambas plataformas, se utiliza un microcontrolador externo, el cual a través del puerto UART transmite al End Device paquetes de 100 bytes para que a su vez el End Device los transmita al Coordinador.

Para simular la adquisición de datos, se han considerado los siguientes sensores, (i) sensor de temperatura LM75 [Tex, 2014], (ii) sensor de temperatura - humedad SHT71 [Sen, 2014] y (iii) sensor de radiación solar BPW41N [Vis, 2014]. Las características de los sensores simulados se indican en la Tabla 5.3.

En los experimentos realizados se asume que los sensores se alimentan solamente cuando se debe realizar una medición. Además, durante la medición, se tomará como consumo del sensor el valor típico y como tiempo de sensado el valor máximo. Estos valores están especificados por los fabricantes en las hojas de datos correspondientes a cada sensor.

5.2.2.2. Resultados Experimentales

Por un lado, la Tabla 5.4 muestra los resultados obtenidos en el ensayo para la plataforma XBee. Como se puede observar, los resultados muestran que la máxima distancia de comunicación confiable es de 196 metros y corresponde a +3 dBm.

Tabla 5.4: Resultados Ensayo de Distancia Confiable Nodo XBee.

Potencia [dBm]	Distancia de Comunicación Confiable
-7	138 m
+3	196 m

Por otro lado, la Tabla 5.5 muestra los resultados obtenidos en el ensayo del nodo Freescale. Al igual que en la plataforma XBee, en el caso del nodo Freescale, la máxima distancia de comunicación confiable corresponde al nivel de +3 dBm y es de 128 metros.

Tabla 5.5: Resultados Ensayo de Distancia Confiable Nodo Freescale.

Potencia [dBm]	Distancia de Comunicación Confiable
-7	42 m
+3	128 m

A partir de los resultados obtenidos, se puede observar que la variación de la potencia de transmisión afecta significativamente la distancia de comunicación confiable. Además, si se consideran además los resultados del ensayo de consumo realizado en la Subsección 5.1, se puede concluir que un valor de potencia de transmisión de +3dBm no genera grandes consumos de energía y permite alcanzar DCC adecuadas para WSNs desplegadas en campo.

5.3. Efecto de las Condiciones de Trabajo sobre la Comunicación de Nodos ZigBee

Debido a que la WSN experimental estará desplegada al aire libre, se debe analizar el efecto de las condiciones del entorno donde la WSN será instalada. Para WSNs desplegadas en campo, las condiciones a considerar son las siguientes: presencia de obstáculos (árboles, plantas, terreno irregular, etc), altura del nodo respecto al suelo y posición de la antena. En la presente tesis, dichas condiciones serán denominadas *condiciones de trabajo*. Los experimentos detallados en la presente sección tienen como objetivo el estudio del efecto de las condiciones de trabajo en campo sobre la máxima distancia a la cual los nodos puede comunicarse. Cabe destacar que el resultado de esta investigación dió origen al trabajo presentado en [Godoy et al., 2012].

5.3.1. Metodología Experimental

Para realizar los experimentos, se utilizó el módulo XBee ZB Digi International Inc. [2012]. Dicho nodo también fue empleado en los experimentos realizados en las Secciones 5.1 y 5.2. Como en la Sección 5.2, se implementó una aplicación embebida en un microcontrolador PIC16F873 SMi [2014] conectado vía UART al módulo XBee ZigBee Series 2. La aplicación embebida simula el uso de tres sensores comúnmente utilizados en aplicaciones de sensado remoto de variables ambientales y agrícolas, es decir, sensores de temperatura y humedad relativa SHT71 [Sen, 2014], sensores de temperatura LM75 [Tex,

Tabla 5.6: Escenarios de Prueba.

Escenario	Altura del Nodo	Obstáculo
0 ^a	1,5	Sin obstáculos
1 ^b	0	Sin obstáculos
2	0,15	Sin obstáculos
3	1,5	Arboles
4	0	Arboles
5	1,5	Montículo de tierra
6	0	Montículo de tierra
7	1,5	Antena en mala posición
8	0	Antena en mala posición

^aEscenario de referencia.

^bSe realizan mediciones en tres condiciones distintas de suelo: suelo seco con hierbas y arbustos, suelo seco sin hierba, y suelo muy húmedo con césped.

2014] y sensores de radiación lumínica BPW41N [Vis, 2014]. Respecto a la fuente de energía, el nodo es alimentado mediante una batería recargable de 9 Volts.

En el presente trabajo se utilizaron dos tipos de nodos ZigBee: Coordinador y End Device. Ambos nodos son configurados usando el software X-CTU. La potencia de transmisión usada es de +3dBm, y la sensibilidad del receptor es de -95 dBm. El End Device transmite paquetes de datos cada 300 ms. Para saber si el End Device permanece asociado con el Coordinador, se utilizó un LED que parpadea cada vez que el End Device transmite un paquete de datos y recibe del Coordinador el mensaje de acuse de recibo.

Con el propósito de evaluar el efecto de las condiciones de trabajo sobre el desempeño de la WSN, se han seleccionado algunos escenarios típicos. En cada escenario se considera la influencia de una ó mas de las siguientes condiciones de trabajo:

- Altura del nodo respecto al suelo.
- Presencia de talúdes, árboles y plantas.
- Posición de la antena.

Para comparar los resultados de las mediciones realizadas, es necesario establecer un escenario de referencia. La Tabla 5.6 muestra un resumen de todos los escenarios evaluados. Cada fila de la Tabla 5.6 corresponde a un escenario. En la primer columna, se detallan los nombres de los distintos escenarios mediante un número de referencia entre 0 y 8. Luego, en la segunda columna, se describe la altura del nodo respecto al suelo. Finalmente, en la tercer columna, se muestran los distintos obstáculos presentes en el escenario a evaluar.

Al realizar los experimentos, existen distintos parámetros que pueden ser utilizados para analizar el efecto de las condiciones de trabajo sobre el desempeño de la WSN. En el presente trabajo se ha seleccionado la Distancia Máxima (DM), debido a que presenta un

efecto inmediato sobre como la WSN debe ser desplegada en campo. La DM, es definida como *la máxima distancia a la que el End Device puede, (i) permanecer asociado de manera estable (es decir, sin interrupción de la asociación), (ii) enviar un paquete con 8 bytes de datos cada 300 ms y (iii) re-asociarse en menos de 5 segundos cuando la asociación es manualmente interrumpida.*

Con respecto a la condición (iii), se ha elegido un tiempo de re-asociación menor que 5 segundos debido a que el estado de re-asociación requiere la transmisión (TX) y recepción (RX) de paquetes de datos. Los procesos de TX y RX son los que tienen el mayor consumo de potencia, y como resultado, un tiempo de re-asociación largo implica una disminución de la vida útil de la batería. La condición de *5 segundos* asegura que, si la asociación se pierde debido a un factor externo (personas o animales que bloquean el enlace, eventos climáticos adversos, etc.), la comunicación puede ser recuperada sin grandes consumos de energía. Las tres condiciones que debe cumplir la DM son igualmente importantes para una adecuada operación de la WSN.

En los experimentos realizados, el End Device envía paquetes de 8 bytes de datos (los que simulan una tanda de mediciones de los sensores). Los 8 bytes están encapsulados en una trama ZigBee estándar. En todos los casos, el Coordinador permanece fijo a 1,5 metros sobre el nivel del suelo.

En las pruebas realizadas, el End Device fue posicionado sobre un punto de una circunferencia cuyo centro es el Coordinador. Dicho punto cumple las tres condiciones de la DM. Luego, el End Device fue alejado radialmente, incrementando la distancia entre el Coordinador y el End Device (es decir, el radio de la circunferencia), hasta romper cualquiera de las tres condiciones mencionadas en la DM. En este punto, se considera que la DM ha sido sobrepasada. Una vez que se ha determinado cuando se rompe la DM, se retorna al paso de distancia previo (la máxima distancia sin pérdida de asociación en la dirección seleccionada) y se desplaza angularmente el End Device sobre la circunferencia, buscando una distancia máxima menor.

La MD considerada en el presente trabajo es la menor distancia encontrada con los procedimientos citados anteriormente. Los pasos radiales fueron de 50 cm, y los pasos angulares fueron de 20° aproximadamente. La antena es mantenida en forma vertical en todos los casos excepto en los escenarios 7 y 8.

5.3.2. Resultados Experimentales

A continuación, se detallan los resultados obtenidos para cada uno de los escenarios de prueba. En el escenario de referencia se ubican el End Device y el Coordinador a una altura de 1,5 metros. Además, la comunicación se lleva a cabo bajo condiciones de LOS (Línea de Visión ó *Line Of Sight*) y cespced irregular. La MD obtenida en el escenario de

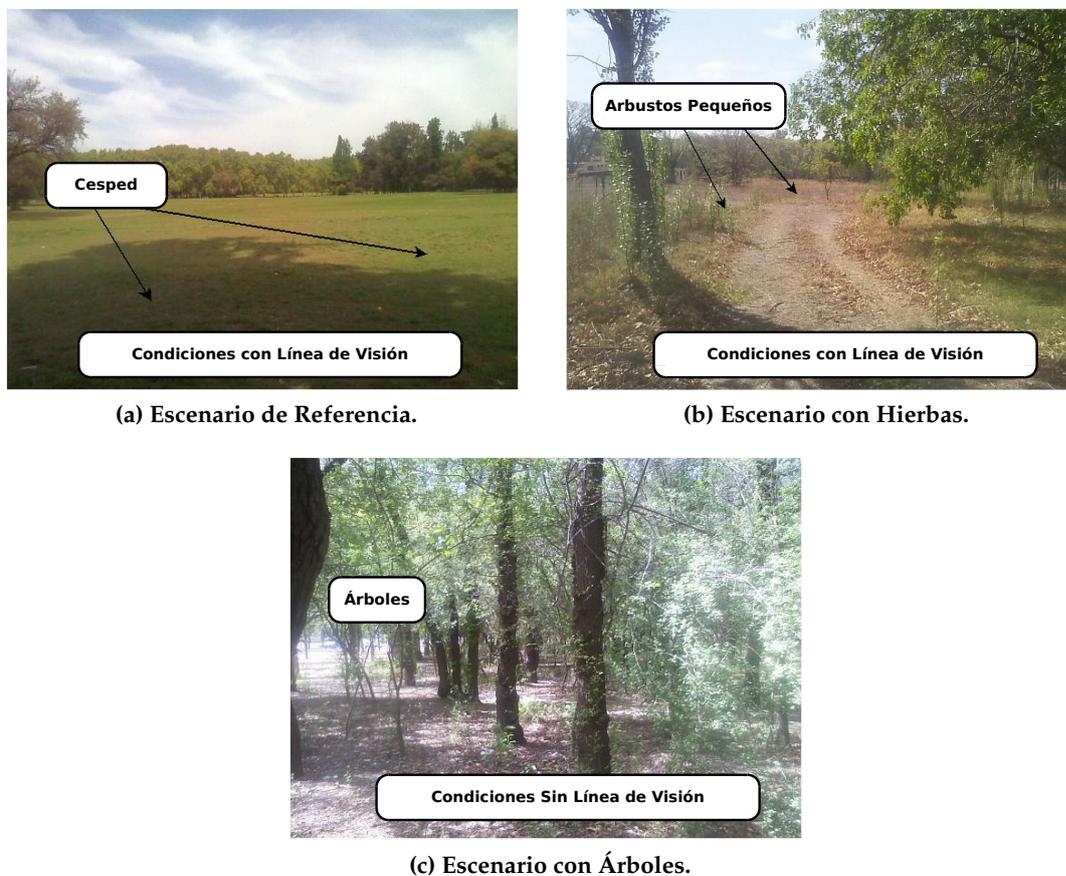


Figura 5.6: Escenarios de Prueba.

referencia fue de 170 metros. Esta distancia es seleccionada como la distancia de referencia y se ilustra en la Figura 5.6a.

A diferencia del escenario de referencia, en el escenario n° 1 el nodo se ubica a nivel del suelo. Además, el experimento se realiza en tres condiciones diferentes de suelo: suelo seco con hierbas y arbustos de pequeño tamaño (Ver Figura 5.6b), suelo seco sin hierba, y suelo muy húmedo con césped de tamaño pequeño. La DM obtenida para este escenario fue de 40 metros en el caso de suelo seco con hierba de pequeño tamaño. Las mediciones mostraron que la DM varía en el orden de 5 metros para las diferentes condiciones de humedad del suelo empleadas (suelo seco sin hierba 45 metros y suelo húmedo 50 metros).

Para el caso del End Device ubicado a 0,15 metros de altura sin obstáculos (escenario n° 2), el experimento fue realizado en las mismas tres condiciones de suelo y presencia de hierba del escenario 1, es decir, suelo seco con hierba y arbustos de pequeño tamaño, suelo seco sin hierba, y suelo muy húmedo con césped de tamaño pequeño. La menor DM registrada en este caso corresponde a la condición de suelo seco con hierbas de pequeño

tamaño y fue de 76 metros.

Las mediciones realizadas en los escenarios 3 y 4 permitieron estudiar el efecto de la presencia de árboles. Para ello, se trasladaron los nodos a un bosque, con árboles ubicados cada 2 ó 3 metros aproximadamente y donde la humedad del suelo fue baja. El nodo se ubica a 1,5 metros de altura (escenario n° 3) y a nivel del suelo (escenario 4). En ambas situaciones se trabajó bajo condiciones de NLOS (*No Line of Sight*, o sin línea de visión) entre los nodos. La Figura 5.6c muestra el bosque, el cual estaba compuesto de árboles de diferentes especies y tamaños. Las DM obtenidas fueron de 75 metros para el End Device ubicado a 1,5 metros sobre el plano de tierra y de 44 metros para el End Device ubicado a nivel del suelo.

Es importante destacar que un montículo de tierra puede bloquear la zona de Fresnel, y como resultado, la DM puede disminuir. El objetivo de los experimentos en los escenarios 5 y 6 consiste en medir la DM en un área con un talud de tierra de aproximadamente 1 metro de altura respecto al nivel del suelo. En particular, en el escenario n° 5, se colocó el End Device sobre el plano de tierra (condiciones de NLOS). Como resultado, la MD arrojó un valor de 40 metros. Luego, en el escenario 6 se posicionó el End Device a 1,5 metros sobre el plano de tierra (condiciones de LOS), donde la DM fue de 66 metros.

Los nodos WSN disponibles en el mercado local tienen antenas muy simples y económicas como antenas monopolo implementadas con un alambre metálico de corta longitud (nodos XBee) ó una antena embebida directamente en el PCB (nodos Freescale V 1.0). Dichas antenas poseen patrones de radiación direccionales, por lo que la posición de la antena afecta el enlace. Debido a que el objetivo de este trabajo consiste en estudiar la performance de nodos disponibles comercialmente, en los escenarios 7 y 8 se estudió la influencia de la posición de la antena en la DM.

De acuerdo al patrón de radiación del módulo XBee [Digi International Inc., 2012], la peor condición para una antena monopolo ocurre cuando es posicionada horizontalmente, por lo que se posicionó la antena en forma horizontal para el End Device y vertical para el Coordinador. Como resultado se obtuvo una DM de 36 metros para el End Device ubicado a nivel del suelo (escenario n° 7) y de 86 metros para el End Device a 1,5 metros sobre el plano de tierra (escenario n° 8).

La Tabla 5.7 muestra la DM obtenida para cada escenario. Los resultados muestran que en una comunicación entre dos nodos ZigBee tanto la presencia de obstáculos (árboles, hierba o montículos de tierra) como la altura de los nodos respecto al piso y la posición de la antena, tienen un importante impacto sobre el enlace. Respecto a los escenarios 1 y 2, los resultados muestran que la altura de los nodos sobre el plano de tierra tiene un efecto muy significativo sobre la DM. Es por este motivo que, cuando la altura de los nodos sobre la tierra disminuye, la DM también disminuye significativamente.

Tabla 5.7: Resultados Obtenidos en cada Escenario de Prueba.

Escenario	Altura del Nodo	Obstáculo	DM [m]
0	1,5	Sin obstáculos	170
1	0	Sin obstáculos	40 ^a
2	0,15	Sin obstáculos	76
3	1,5	Arboles	75
4	0	Arboles	40
5	1,5	Montículo de tierra	66
6	0	Montículo de tierra	40
8	1,5	Antena en mala posición	36

^aEste valor de DM corresponde al peor caso (suelo seco con hierbas y arbustos).

Los experimentos descritos en los escenarios 1 y 4 revelan que la DM a la cual los nodos pueden comunicarse (con el End Device a nivel del suelo) es similar tanto en el interior de un bosque con condiciones de NLOS como así también en campos sin obstáculos. Dichos resultados muestran que la altura sobre la tierra de los nodos ejerce una mayor influencia en la DM que la presencia de árboles.

Finalmente, es importante destacar que la condición de trabajo en campo que más afecta a la DM es en primer lugar la posición de las antenas y luego la altura respecto del suelo, la presencia de árboles, los montículos de tierra, la existencia de hierbas y la humedad del suelo, respectivamente.

5.4. Conclusiones

En este capítulo, se han realizado experimentos para evaluar cual de las plataformas de hardware de WSNs disponibles en el mercado local es la más conveniente en términos de consumo de energía, recursos de hardware, cantidades de interfaces analógicas y digitales, y potencia de transmisión. Para ello, se llevaron a cabo diversos experimentos con el objetivo de estudiar el consumo de energía en los nodos sensores y los efectos de la potencia de transmisión tanto en la vida útil como así también en la distancia de comunicación confiable de los nodos sensores.

En base a las pruebas realizadas se determinó que la plataforma más adecuada para implementar la WSN experimental (la cual será gestionada por Sensor Cirrus) es el transceptor ZigBee XBee Serie 2 [Digi International Inc., 2012]. Para conformar el nodo sensor, se le añadirá al módulo XBee un sistema embebido Arduino Pro 328 [Spa, 2014] y un sensor digital Sensirion SHT71 [Sen, 2014]. El Arduino Pro 328 permitirá adquirir los datos del sensor y enviarlos al transceptor XBee. Todos los elementos que compondrán al nodo sensor están disponibles en el mercado local, tienen bajo costo económico y per-

mitirán desarrollar nodos sensores que cumplan con las condiciones propias del buen funcionamiento de las WSNs descritas en el Capítulo 2.

Una vez que se determinó que plataforma resultó más adecuada para implementar la WSN, se realizó un estudio de la influencia de las condiciones de trabajo en campo (presencia de obstáculos, altura del nodo respecto al suelo y posición de la antena) sobre la máxima distancia a la cual los nodos pueden comunicarse. Los resultados obtenidos permitieron determinar que la altura respecto al suelo ideal para ubicar los nodos sensores en el campo es de 1,5 metros. Además, se determinó cuál es la máxima distancia de comunicación permitida en cada posible escenario de despliegue (escenarios con arboles, taludes, plantas y otros obstáculos).

Finalmente, se puede concluir que los experimentos realizados en este capítulo han dado lugar a una serie de procedimientos a seguir para implementar y desplegar WSNs de monitoreo agrícola. Además los experimentos han permitido diseñar la arquitectura de Sensor Cirrus para que gestione a las WSNs conservando las características propias del buen funcionamiento de las mismas. En el próximo capítulo, se detallarán los experimentos realizados con los dos módulos principales de Sensor Cirrus: el *Módulo WSN - TCP/IP* y el *Módulo TCP/IP - Cloud*.

Experimentos en Sensor Cirrus

En el presente capítulo, se llevan a cabo experimentos para validar que Sensor Cirrus cumpla con la Hipótesis de Trabajo de esta tesis doctoral, donde se plantea que *“Se puede gestionar en forma remota, sencilla y eficiente WSNs conformadas por recursos acotados de hardware y fuente de energía, mediante tecnologías de computación distribuída como Cloud Computing, adecuando las tecnologías disponibles en el mercado local y con un bajo costo”*

En la Sección 6.1 se llevan a cabo experimentos para demostrar que Sensor Cirrus puede gestionar en forma remota, sencilla y eficiente WSNs compuestas por recursos acotados de hardware y fuente de energía. La confiabilidad y eficiencia es determinada en base a la tasa de pérdida de paquetes en los distintos enlaces del proceso de comunicación de Sensor Cirrus.

Dado que Cloud ofrece servicios de bajo costo para la gestión remota de las WSNs, en la Sección 6.2 se desarrollan pruebas con el servicio de predicción de heladas de Sensor Cirrus. Estos experimentos demuestran como se pueden predecir heladas mediante WSNs y servicios Cloud. Además, se presentan modelos teóricos que permiten optimizar el uso de Cloud mediante la selección de la instancia de Amazon EC2 mas adecuada para procesar el servicio de predicción de heladas de Sensor Cirrus. Finalmente, la Sección 6.3 concluye el presente capítulo.

6.1. Pérdida de Paquetes en el Proceso de Comunicación

La presente sección detalla los experimentos realizados en los Módulos WSN - TCP/IP y TCP/IP - Cloud de Sensor Cirrus. El principal objetivo de los experimentos consiste en determinar la eficiencia y confiabilidad de la plataforma mediante el estudio de la tasa de pérdida de paquetes (en Inglés Packet Loss Rate ó simplemente PLR) en el proceso

de comunicación. Este proceso comienza cuando los datos son adquiridos por la WSN y finaliza cuando llegan al Cloud.

6.1.1. Metodología Experimental

La métrica utilizada para estudiar la eficiencia de Sensor Cirrus es la tasa de pérdida de paquetes (packet loss rate ó PLR). La PLR y permite calcular la cantidad de datos provenientes de la WSN que se pierden en cada enlace del proceso de comunicación de Sensor Cirrus. Esta métrica es complementaria a la tasa de paquetes entregados correctamente, la cual fue utilizada en el Capítulo 5, Sección 5.2.2 para calcular la distancia de comunicación confiable.

Actualmente, no hay un criterio único que establezca cual es el valor de PLR más adecuado para WSNs aplicadas a agro-meteorología. En [Humber and Ngai, 2010], los autores desarrollan un servicio de entrega de datos confiable para WSNs y logran un PLR de 7,21 % para una red desplegada en laboratorio compuesta por 9 nodos sensores y de 11,19 % para una WSN simulada e integrada por 18 nodos sensores. Otros autores como [Pierce and Elliott, 2008] logran PLRs de entre 7 y 3 % en una WSN desplegada en campo y compuesta por 21 nodos sensores. Finalmente, en [Nadimi et al., 2008] los autores desarrollan una WSN de 7 nodos sensores para monitoreo de ganado y obtienen una PLR del 8 %. En base a los trabajos mencionados, en la presente tesis se considera que la PLR admitida para considerar al proceso de comunicación eficiente debe ser menor ó igual al 10 %.

El proceso de comunicación de Sensor Cirrus comienza cuando el nodo sensor adquiere los datos en campo y finaliza cuando los datos llegan al Cloud. La Figura 6.1 ilustra la arquitectura general de Sensor Cirrus, la cual ha sido discutida en profundidad en el Capítulo 4. Además, en la Figura 6.1 se detallan los enlaces entre los distintos módulos de Sensor Cirrus que intervienen en el proceso de comunicación. A cada módulo le corresponde un número y a cada enlace se lo identifica con los números de los módulos que conecta.

La Tabla 6.1 muestra los extremos de cada uno de los enlaces del proceso de comunicación y la PLR a calcular para cada uno de ellos. Luego de calcular la PLR en cada uno de los enlaces, se debe calcular la PLR entre los dos extremos del proceso de comunicación (la WSN y el Cloud) para determinar cuántos paquetes se han perdido en total.

En este trabajo, el cálculo de la PLR se realiza mediante la siguiente fórmula:

$$PLR_{tr} = 100 - PDR_{tr}, \quad (6.1)$$

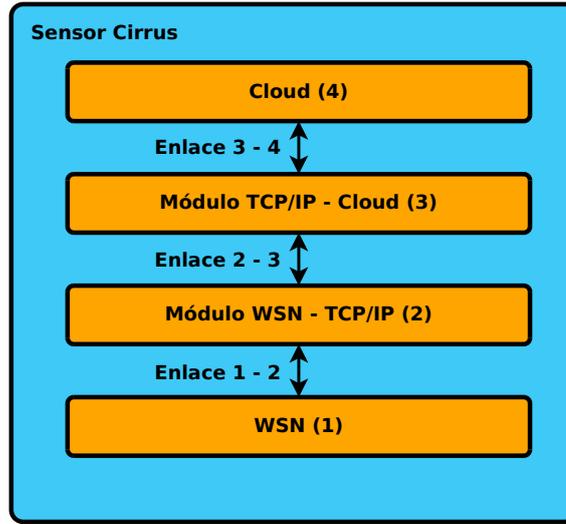


Figura 6.1: Elementos y Enlaces del Proceso de Comunicación.

Tabla 6.1: Enlaces del Proceso y PLR Correspondiente.

Extremos del Enlace	Enlace	Nomenclatura
Cloud y Módulo TCP/IP - Cloud	E ₃₄	PLR ₃₄
Módulo TCP/IP - Cloud y Módulo WSN - TCP/IP	E ₂₃	PLR ₂₃
WSN y Módulo WSN - TCP/IP	E ₁₂	PLR ₁₂

donde PDR es la tasa de paquetes entregados correctamente la cual se calcula con la ecuación (10):

$$PDR_{tr} = \left(\frac{N_r}{N_t} \right) \cdot 100, \quad (6.2)$$

donde t es el extremo del enlace que envía los paquetes, r es el extremo que recibe los paquetes, N_t es el número de paquetes generados en el extremo del enlace t y N_r el número de paquetes recibidos sin error en el extremo del enlace r .

Para calcular las PLR, se utilizó la WSN experimental ZigBee descrita en el Capítulo 4, Sección 4.3.1. La WSN experimental está compuesta por (i) cuatro nodos sensores, (ii) un Router y (iii) una estación base conformada por el Coordinador ZigBee y una máquina tipo PC. Debido a que la WSN se desplegó en laboratorio, para este experimento, no se utilizó el Router. La Figura 6.2 ilustra uno de los nodos sensores utilizados en la WSN para realizar los experimentos.

Se definieron dos escenarios de prueba, uno de media escala compuesto por una WSN de 42 nodos sensores y otro de gran escala, conformado por una WSN de 300 nodos sensores. Debido a que en la actualidad el laboratorio no cuenta con dichas cantidades de nodos físicos, se multiplexó en el tiempo la adquisición de los datos. Esto permitió alcanzar la

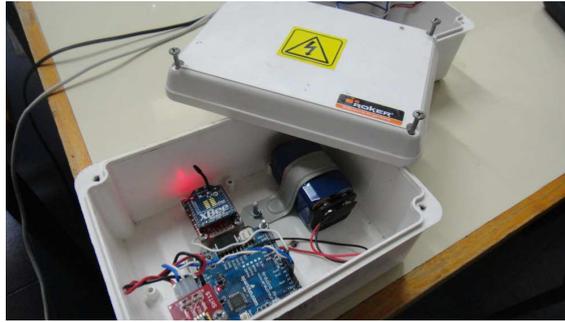


Figura 6.2: Nodo Sensor Utilizado.

cantidad de nodos necesaria para cada uno de los escenarios. La multiplexación consiste en aumentar la frecuencia de muestreo para generar mayores cantidades de datos y procesos de comunicación entre los distintos módulos de Sensor Cirrus.

Para realizar los experimentos, se tomó como base una aplicación de monitoreo de heladas. En este tipo de aplicaciones, generalmente cada nodo de la WSN realiza una adquisición de datos cada 5 [Beckwith et al., 2004] ó 10 minutos [Díaz et al., 2011]. Si se adquieren datos cada 5 minutos se puede trabajar en condiciones similares a las de tiempo real, sin embargo el consumo de energía en los nodos es el doble que para el caso de muestreo cada 10 minutos. Con el objetivo de reducir el consumo de energía, en esta tesis la adquisición de datos será realizada cada 10 minutos.

Cuando el muestreo se realiza cada 10 minutos, se generan 6 paquetes por hora por nodo y 24 paquetes por hora por WSN. Cada trama ZigBee emitida por los nodos sensores contiene los datos de temperatura y humedad reales adquiridos por el sensor digital Sensirion SHT15 [Sen, 2011].

Para el escenario de la WSN de media escala, los nodos sensores realizaron una adquisición de datos cada 57 segundos. Esto permitió generar 63 paquetes por hora por nodo y 252 paquetes por hora por toda la WSN, lo que equivale a una WSN compuesta por 42 nodos muestreando cada 10 minutos. El experimento se realizó durante 9 días de funcionamiento continuo de la WSN, con todos los nodos ubicados en el mismo banco de trabajo, la estación base a 1 metro de los nodos y el equipo que contiene el Módulo TCP/IP - Cloud en la misma LAN Ethernet de la estación base.

Respecto al escenario de la WSN de gran escala, los nodos sensores se programaron para adquirir datos cada 8 segundos. La multiplexación realizada generó 450 paquetes por hora por nodo y 1.800 paquetes por hora por WSN, lo que equivale a una WSN compuesta por 300 nodos muestreando cada 10 minutos. Por motivos de consumo (las baterías de alguno de los nodos comienzan a fallar a los 4 días), el experimento se realizó durante 3 días de funcionamiento continuo de la WSN en iguales condiciones de despliegue que el escenario anterior.

La Tabla 6.2 resume cada uno de los escenarios de prueba utilizados. La segunda columna indica la cantidad de nodos sensores obtenidos gracias a la multiplexación en tiempo, la tercera columna la cantidad de paquetes generados por hora por la WSN y la cuarta columna la duración de cada experimento.

Tabla 6.2: Escenarios Experimentales.

Escenario	Nodos Sensores	Paquetes Generados por Hora	Duración
Media Escala	40	252	9 días
Gran Escala	300	1.800	3 días

6.1.2. Resultados Experimentales

En la presente sección se presentan los resultados obtenidos para cada uno de los escenarios propuestos.

WSN de Media Escala

En la WSN de media escala, se realizaron mediciones durante 9 días. En condiciones ideales la WSN genera unos 252 paquetes por hora lo cual equivale a 54.432 paquetes de datos a lo largo de todo el experimento. Este valor equivale al 100 % de paquetes transmitidos y para que la PLR sea cero, deberían recibirse en el Cloud los 54.432 paquetes. Sin embargo, los resultados experimentales muestran que se pierden datos en algunos de los enlaces del proceso de comunicación. La Tabla 6.3 indica la cantidad de paquetes recibidos y emitidos por cada uno de los módulos de Sensor Cirrus.

Los resultados muestran que la mayor pérdida de datos se produce en el enlace entre el Módulo WSN - TCP/IP (programado en la estación base) y la WSN. En dicho enlace se perdieron 1.730 paquetes. Esto se debe a que la WSN ZigBee utiliza paquetes ó tramas de acuse de recibo [Digi International Inc., 2012]. Las tramas de acuse de recibo son enviadas desde el Coordinador hacia los nodos sensores de la WSN (ó viceversa) para confirmar la recepción correcta de los paquetes de datos y de cambio de configuración. En algunos casos puede ocurrir que el Coordinador esté ocupado comunicandose con un nodo de la WSN y no le envíe el acuse de recibo al nodo. Ante esta situación, el nodo sensor

Tabla 6.3: Paquetes Recibidos y Emitidos por cada Módulo de Sensor Cirrus.

Componente	Paquetes Emitidos	Paquetes Recibidos
Cloud	Sólo recibe paquetes	52.702
Módulo TCP/IP - Cloud	52.702	52.702
Módulo WSN - TCP/IP	52.702	52.702
WSN	54.432	Sólo genera paquetes

Tabla 6.4: Resultados WSN de Media Escala.

Enlace	Componentes	PLR
E_{34}	Cloud y Módulo TCP/IP - Cloud	0,000
E_{23}	Módulo TCP/IP - Cloud y Módulo WSN - TCP/IP	0,000
E_{12}	WSN y Módulo WSN - TCP/IP	3,178

realiza tres intentos más de envío de la trama de datos. Finalmente, si no recibe el acuse de recepción el nodo sensor descarta la trama de datos y se pone en modo reposo para ahorrar batería.

Respecto al enlace entre el Módulo WSN - TCP/IP y Módulo TCP/IP - Cloud, los resultados muestran que no se perdieron paquetes. Esto es así porque el enlace se realiza sobre un socket TCP. De igual forma, en el enlace entre el Cloud y el Módulo Cloud tampoco se pierden datos. Dado que Google Drive sincroniza el 100 % de los datos que llegan al Módulo Cloud - TCP/IP, es esperable que en este enlace no se pierdan paquetes.

Las PLR en cada uno de los enlaces de comunicación se calculan mediante la ecuación 6.1 en base a las paquetes recibidos y emitidos en cada uno de los módulos de Sensor Cirrus. Los resultados son mostrados en la Tabla 6.4.

Se puede observar que la PLR del único enlace donde se pierden datos (E_{12}) es igual a 3,178. Luego, en el enlace E_{23} y en el E_{34} la PLR es igual a cero. Finalmente, se utilizan los paquetes emitidos por la WSN y las recibidas por el Cloud para calcular el PLR correspondiente a todo el proceso de comunicación, el cual es de 3,178 %. Debido a que los resultados muestran que la tasa de pérdida de paquetes es ampliamente menor a 10 % (3,178 %), se puede concluir que la plataforma gestiona la WSNs de media escala eficientemente. Cabe destacar que este experimento ha sido realizado en condiciones de laboratorio, por lo que se espera que la PLR sea mayor para condiciones de trabajo reales.

WSN de Gran Escala

En este apartado se presentan los resultados obtenidos en los experimentos realizados en la WSN de gran escala. Las pruebas consistieron en realizar mediciones de temperatura y humedad en condiciones de laboratorio durante 3 días. En condiciones ideales la WSN genera unos 1.800 paquetes por hora lo cual equivale a 129.600 paquetes durante los 3 días de adquisición de datos. Al igual que en la WSN de media escala, los resultados muestran que se pierden datos en el enlace E_{12} del proceso de comunicación. La Tabla 6.5 detalla la cantidad de paquetes recibidos y emitidos por cada módulo de Sensor Cirrus en las pruebas realizadas en la WSN de gran escala.

Como se puede observar en la Tabla 6.5, el único enlace donde se pierden paquetes es el enlace entre la WSN y el Módulo WSN - TCP/IP. En este enlace se pierden 14.852 tramas

Tabla 6.5: Paquetes Recibidos y Emitidos por cada Módulo de Sensor Cirrus.

Componente	Paquetes Emitidos	Paquetes Recibidos
Cloud	Sólo recibe paquetes	114.748
Módulo TCP/IP - Cloud	114.748	114.748
Módulo WSN - TCP/IP	114.748	114.748
WSN	129.600	Sólo genera paquetes

Tabla 6.6: Resultados WSN de Gran Escala.

Enlace	Componentes	PLR
E_{34}	Cloud y Módulo TCP/IP - Cloud	0,000
E_{23}	Módulo TCP/IP - Cloud y Módulo WSN - TCP/IP	0,072
E_{12}	WSN y Módulo WSN - TCP/IP	11,46

de datos. Luego, en los enlaces E_{23} y E_{34} no se pierden paquetes. La Tabla 6.6 muestra las PLR obtenidos para cada enlace de Sensor Cirrus en el escenario de la WSN de gran escala.

En base a los paquetes emitidos por la WSN y los recibidos por el Cloud, se calcula la PLR correspondiente a todo el proceso de comunicación, la cual tiene un valor de 11,46 %. Se ha observado que la PLR (11,46 %) es mayor que los valores de eficiencia objetivo (10 %). Si bien la PLR obtenida indica que el Módulo WSN - TCP/IP presenta inconvenientes para gestionar WSNs conformadas por 300 nodos sensores ó mas, ésto puede solucionarse reorganizando la cantidad de nodos sensores por estación base. Por ejemplo para gestionar 600 nodos sensores, en vez de 300 nodos por estación base, se recomienda utilizar tres estaciones base con 200 nodos sensores cada una.

6.2. Servicio de Predicción de Heladas

La gestión remota de WSNs permite brindar soluciones a distintas aplicaciones de monitoreo agrícola y ambiental. En la presente tesis doctoral se ha desarrollado un servicio de predicción de heladas que permite evitar pérdidas económicas en producciones agrícolas. El servicio de predicción de heladas de Sensor Cirrus ha sido explicado en profundidad en el Capítulo 4, Sección 4.3.5. Dicho servicio utiliza los datos de temperatura y humedad gestionados por Sensor Cirrus y se ejecuta sobre recursos computacionales provistos por un Cloud público. El servicio puede ser ejecutado en una sola instancia ó en un clúster de máquinas virtuales en el Cloud.

Actualmente, uno de los proveedores líderes de recursos de cómputo es Amazon [Amazon EC2, 2012]. A través del toolkit *Elastic Cloud Computing (EC2)*, Amazon ofrece una gran variedad de máquinas virtuales (instancias) aptas para el procesamiento del servicio de predicción de heladas. Las instancias ofrecidas por Amazon varían en características

como cantidad de procesadores, memoria RAM y espacio de almacenamiento. Debido a la amplia gama de instancias ofrecidas por Amazon, surge la necesidad de determinar cuál es el tipo de instancia que tiene mejor rendimiento (en términos de tiempo de ejecución y coste económico) para procesar el servicio de predicción de heladas propuesto. Para cubrir esta necesidad, se necesitan modelos que permitan determinar cual es la instancia de Amazon más adecuada para procesar el servicio.

En esta sección se presentan los experimentos que han permitido elaborar un conjunto de modelos teóricos destinados a estimar el rendimiento de distinto tipo de instancias de Amazon EC2 aplicadas al procesamiento del servicio de predicción de heladas. Los modelos han permitido determinar cual es la instancia más adecuada cuando es ejecutada en forma aislada ó en paralelo a través de clústeres de máquinas virtuales.

La presente sección está estructurada de la siguiente manera. En la Subsección 6.2.1 se presenta la problemática de las heladas y en la Subsección 6.2.2 se detalla la metodología experimental seguida para llevar a cabo los experimentos. Luego, en la Subsección 6.2.3 se presentan los resultados experimentales obtenidos al ejecutar el servicio en instancias individuales. En la subsección 6.2.4 se discute la precisión de los modelos teóricos desarrollados. Finalmente, en la subsección 6.2.5 se presentan los resultados obtenidos al ejecutar el servicio en paralelo a través de clústeres virtuales.

6.2.1. Heladas en la Región de Cuyo, Argentina

En esta subsección, se presenta el fenómeno de las heladas en cultivos. Este fenómeno agro-meteorológico causa grandes pérdidas económicas en cultivos de la Región de Cuyo en Argentina. Debido al impacto que representan las pérdidas económicas debido a heladas, en esta tesis se estudia este fenómeno y se propone el uso de WSNs gestionadas por tecnologías de Computación Distribuida para dar soluciones tecnológicas reales a los productores de la región.

6.2.1.1. Heladas. Definición y Tipos

Las heladas son un fenómeno agro-meteorológico que produce daño en distinto tipo de cultivos. El fenómeno sucede cuando la temperatura del aire desciende por debajo de cero grado centigrado medidos a una altura de 1,5 metros sobre el nivel del suelo. Existen tres tipos de heladas, (i) advección, (ii) irradiación, (iii) evaporación y (iv) mixtas. Los conceptos teóricos presentados en esta sección han sido extraídos de [Dirección de Agricultura y Contingencias Climáticas Mendoza, 2013] y de [Snyder and Melo-Abreu, 2005].

Heladas por Advección Las heladas por advección son de carácter macroclimático (heladas generales). Este tipo de heladas se debe al ingreso de una masa de aire frío en un área. Además se caracterizan por presencia de nubes, vientos de moderados a fuertes, falta de inversión de temperatura y baja humedad. En algunos casos las heladas de advección se presentan acompañadas por lluvia.

Debido a que la mayoría de los métodos de protección activos funcionan mejor ante la presencia de una inversión de temperatura, las heladas de advección son muy difíciles de combatir.

Heladas por Radiación Este tipo de heladas son de carácter microclimático (heladas parciales). Las heladas por radiación suceden cuando las pérdidas de calor de la superficie del suelo por radiación son superiores a los aportes de calor que el suelo recibe. En este caso, el balance de energía de la superficie del suelo es negativo. La temperatura del aire se ve afectada a nivel del suelo y en algunas capas superiores a la superficie. A diferencia de la situación normal (donde la temperatura del aire disminuye con la altura), en este caso las capas cercanas al suelo tienen temperaturas inferiores a las superiores.

Las heladas por radiación se caracterizan por cielo despejado, muy poco ó nada de viento, inversión de temperatura, temperaturas de punto de rocío bajas, temperatura del aire menor a 0 °C durante la noche y mayor a 0 °C durante el día.

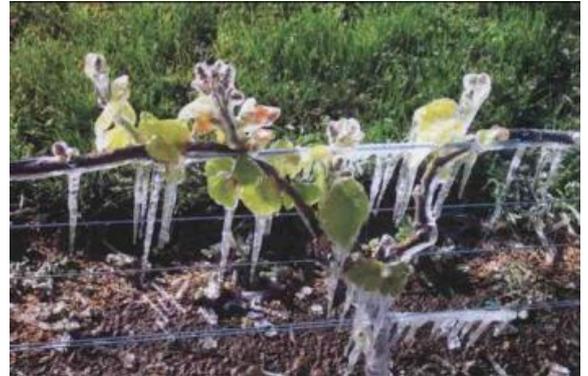
Heladas por Evaporación Se producen debido a ingresos de frentes fríos que causan lluvias. El fenómeno comienza cuando llueve y se deposita agua en la superficie de los cultivos y del suelo. Luego, sigue entrando aire frío y evapora el agua depositada. Finalmente, la evaporación genera pérdidas de calor en el suelo y en las plantas lo que causa daños en los tejidos vegetales. Cabe destacar que este tipo de heladas no tienen gran incidencia en la Región de Cuyo.

Heladas Mixtas Las heladas mixtas se producen por la combinación de heladas por advección, radiación y evaporación. En la Región de Cuyo los daños provocados por este tipo de heladas son los mayores. Las heladas mixtas en Cuyo, se pueden generar en dos situaciones diferentes: presencia o ausencia de viento Zonda antes de la entrada del frente frío. El viento Zonda, es un viento seco y caliente típico de las Provincias de San Juan y Mendoza de la Región de Cuyo.

Para el primer caso (ausencia de viento Zonda antes de la entrada del frente frío), cuando llega el frente frío genera nubosidad y enfría el ambiente. Debido a la presencia de nubosidad, las pérdidas por radiación no son significativas. Sin embargo, cuando el frente



(a) Helada Negra.



(b) Helada Blanca.

Figura 6.3: Daños por Heladas.

se va, la nubosidad se disipa y se produce una fuerte pérdida de calor por radiación que produce la helada.

Respecto al segundo caso (presencia de viento Zonda antes de la entrada del frente frío), cuando se presentan condiciones de viento Zonda, este viento disminuye significativamente la humedad del ambiente. Por lo tanto, no se genera nubosidad cuando entra el frente frío y las pérdidas por radiación son mucho mayores que en el caso anterior.

6.2.1.2. Daños en Cultivos

Cuando ocurre la helada se pueden producir dos tipos de daños: por helada negra ó helada blanca. Por un lado, si cuando sucede la helada no hay suficiente humedad en el ambiente como para formar escarcha, se produce hielo en el interior de las células de la planta. Luego, el hielo crece en forma de agujas y produce ruptura de las membranas celulares. Debido a la ruptura de las membranas, se produce una deshidratación en la planta que provoca la muerte de las células y del tejido vegetal (Ver Figura 6.3a). Este fenómeno es conocido como helada negra.

Por otro lado, si la humedad del ambiente es suficiente para formar escarcha, se produce sólo congelamiento de los tejidos que han estado en contacto con el hielo. Este tipo de daño se denomina helada blanca y es menor que el causado por la helada negra (ver Figura 6.3b).

6.2.1.3. Métodos de Defensa

Si bien es un fenómeno que acontece todos los años, existen distintos métodos de defensa para minimizar los daños. En esta subsección se presentan los distintos métodos

utilizados para defender a los cultivos de las heladas.

Métodos de Defensa Pasivos Según [Snyder and Melo-Abreu, 2005], los métodos de defensa pasivos son aquellos que se han implementado antes de la noche de la helada, lo que puede evitar la necesidad de protección activa. La principal ventaja de los métodos pasivos, es que generalmente son menos costosos que los métodos activos y evitan la necesidad de la protección activa. Sin embargo, no siempre es factible su implementación. La primer columna de la Tabla 6.7, resume los principales métodos pasivos.

En esta tesis doctoral no se hará un análisis de cada uno de estos métodos, los mismos pueden ser consultados en profundidad en [Snyder and Melo-Abreu, 2005] y [Dirección de Agricultura y Contingencias Climáticas Mendoza, 2013].

Métodos de Defensa contra Heladas	
Métodos Pasivos	Métodos Activos
Selección del emplazamiento y de cultivos	Estufas
Manejo del drenaje de aire frío	Ventiladores
Cobertura con árboles y artificial	Aspersores
Gestión de la nutrición de las plantas	Riego de superficie
Poda adecuada	Aislamiento con espumas
Evitar el laboreo del suelo y quitar hierbas	Helicopteros
Riego	Cortinas de humo, niebla ó nubes artificiales
Pintar los troncos y envolturas	Radiadores infrarojos
Control bacteriano	Combinaciones de métodos

Tabla 6.7: Métodos de Defensa.

Métodos de Defensa Activos Los métodos de defensa activos, son aquellos que se aplican durante la noche de ocurrencia de heladas para atenuar los efectos de las temperaturas bajo cero [Snyder and Melo-Abreu, 2005]. La principal función de estos métodos es la de evitar el enfriamiento del entorno de los cultivos aplicando una fuente de energía calórica. En la segunda columna de la Tabla 6.7, se muestran los principales métodos pasivos. A continuación se explicarán algunos de los métodos activos más utilizados.

Estufas. La Figura 6.4a ilustra una estufa para defensa contra heladas. Estas estufas funcionan en base a combustibles derivados del petróleo (diesel, fuel-oil, propano, etc.) y permiten elevar la temperatura del aire frío que rodea a la planta de 3 a 5°C [Dirección de Agricultura y Contingencias Climáticas Mendoza, 2013].

Las estufas son uno de los métodos más comunmente utilizados en la Región de Cuyo. Respecto al uso en Mendoza, la Dirección de Agricultura y Contingencias Climáticas de la Provincia, recomienda encenderlas cuando la temperatura es de 1 grado centígrado y

utilizar un número elevado de estufas pequeñas en remplazo una cantidad más reducida de estufas potentes. Esto es debido a que un número elevado de estufas pequeñas permite una mejor homogeneización del calentamiento y un movimiento más efectivo del aire.



(a) Estufa.



(b) Ventilador Diesel.



(c) Defensa por Aspersión.



(d) Defensa Mediante Helicópteros.

Figura 6.4: Métodos de Defensa contra Heladas.

Ventiladores ó Turbinas Los ventiladores utilizan generalmente motores diesel ó a gas y tienen un costo de uso mucho menor comparado al de las estufas. Los ventiladores utilizan de un 5 % a un 10 % del combustible consumido por un sistema basado en estufas diesel. Sin embargo, requieren una mayor inversión inicial.

Los métodos basados en ventiladores defienden a los cultivos mediante la mezcla del aire más caliente de capas superiores con el aire frío de las capas cercanas a la superficie. Respecto a la cobertura, si se aplica el ventilador en contra del viento, un ventilador protege de 90 a 100 m. En cambio, si se aplica a favor del viento, la cobertura aumenta y protege de 130 a 140 m. Por ejemplo, una máquina diesel de 173 HP puede llegar a proteger hasta 6 hectáreas.

Los sistemas de defensa (Ver Figura 6.4) son activados en base a información provista por sistemas de monitoreo agrícola a gran escala compuestos por instrumentos que registran

datos en campo, como las WSNs, e infraestructuras IT que proveen recursos computacionales *bajo demanda* para almacenar y procesar dichos datos, como Cloud y Grid.

Aspersores En la Figura 6.4c se muestra la defensa de cultivos mediante aspersores. Este método se basa en el uso de la liberación de calor que produce el agua al congelarse (80 cal/g.). Los cultivos son regados por los aspersores y cuando se enfría por la helada el calor liberado permite que la temperatura de la hoja no descienda por debajo de 0 grados centígrados. El uso de microaspersores es preferible debido a que requieren menos presión y gastan menos agua. Con microaspersores, la aspersión se lleva a cabo durante toda la noche hasta que el hielo se haya fundido debido al efecto del sol.

El método de defensa mediante aspersores es muy efectivo porque permite defender a los cultivos contra temperaturas de hasta -7°C . Sin embargo, tiene costos de instalación elevados y puede generar inundación en los cultivos.

Helicópteros Como los ventiladores, los helicópteros mezclan el aire caliente de las capas superiores (de la zona donde está invertido el gradiente de temperatura) con el de las capas más frías cercanas al suelo. El área que cubre un helicóptero depende del tipo y tamaño del vehículo y de las condiciones meteorológicas. Sin embargo, se estima que el área que cubre un único helicóptero varía de 22 a 44 hectáreas [Snyder and Melo-Abreu, 2005].

La Figura 6.4d ilustra un helicóptero en un procedimiento de defensa contra heladas. Generalmente, los helicópteros realizan la defensa volando a una altura de 20 a 30 metros, a velocidades de entre 8 y 40 Km/h y con frecuencia de pasadas que varía de 30 a 60 minutos. Estos parámetros varían de acuerdo a la superficie a defender y las condiciones meteorológicas. Si bien es un método muy efectivo, tiene costos prohibitivos para la mayoría de los productores de la Región de Cuyo.

Riego de Superficie En este método, se inunda el campo y el calor del agua se libera al aire al enfriarse. Si el riego se realiza antes de la helada, se puede elevar la temperatura entre 3 y 4 $^{\circ}\text{C}$. El riego de superficie funciona mejor para defender árboles de crecimiento bajo y vides contra heladas de radiación. La principal ventaja de este método es el bajo costo económico. Mientras que su principal desventaja es que si bien la sumersión parcial o total de las plantas tolerantes es posible, suelen generarse enfermedades y asfixia de las raíces.

6.2.1.4. Sistemas de Alarmas y Procedimiento de Guardias Contra Heladas

Por un lado, los sistemas de alarmas contra heladas indican cual es el momento adecuado para iniciar la defensa en campo. Estos sistemas están compuestos por instrumentos que permiten medir la temperatura, humedad y otros parámetros en finca. Los instrumentos varían desde simples termómetros analógicos ó digitales hasta estaciones meteorológicas ó WSNs.

Debido a que las heladas dependen de la topología del terreno (excepto las generales), es necesario contar con sistemas de alarma dotados con varios puntos de adquisición de datos en finca. Por un lado, si se utilizan varios puntos compuestos por instrumentos tradicionales (como termómetros analógicos o digitales) se requiere monitoreo in-situ de cada uno de los instrumentos, lo que implica una gran cantidad de operarios desplegados en campo. Por otro lado, si se utilizan varias estaciones meteorológicas, se puede contar con múltiples puntos de medición monitoreados en forma remota. Sin embargo esta opción implica altos costos económicos.

A diferencia de los instrumentos tradicionales y las estaciones meteorológicas, las WSNs permiten contar con varios puntos de medición, monitoreados en forma remota y con menores costos económicos. Además, el mantenimiento de las WSNs es menor comparado con las otras alternativas mencionadas. Por tales motivos, se puede concluir que las WSNs son una potencial tecnología para implementar sistemas de alarmas contra heladas.

Por otro lado, los sistemas de alarma contra heladas permiten contar con información para activar el procedimiento de guardia contra heladas. Este procedimiento se realiza de la siguiente manera. En primer lugar, el especialista recibe la alarma de alta probabilidad de helada y envía al personal a la finca. Mientras el personal espera en la finca, se toman mediciones en tiempo real con instrumentos, estaciones meteorológicas y WSNs. Finalmente, si se detectan valores específicos de temperatura, humedad y punto de rocío, el Ingeniero Agrónomo dá la orden y el personal activa los métodos de defensa.

6.2.1.5. Impacto en la Región de Cuyo

En la región de Cuyo, Argentina, la temporada de ocurrencia de heladas se extiende desde marzo hasta noviembre. El período libre de heladas varía entre 130 y 150 días promedio, según zonas. Las heladas que más afectan a los cultivos de la región son las tardías. Estas heladas suceden en primavera y causan daños severos en frutales y en menor medida en la vid ¹. Como ejemplo puede citarse que en la Provincia de Mendoza,

¹http://www.prosap.gov.ar/webDocs/EPSA_MendozayResolucion_2009.pdf

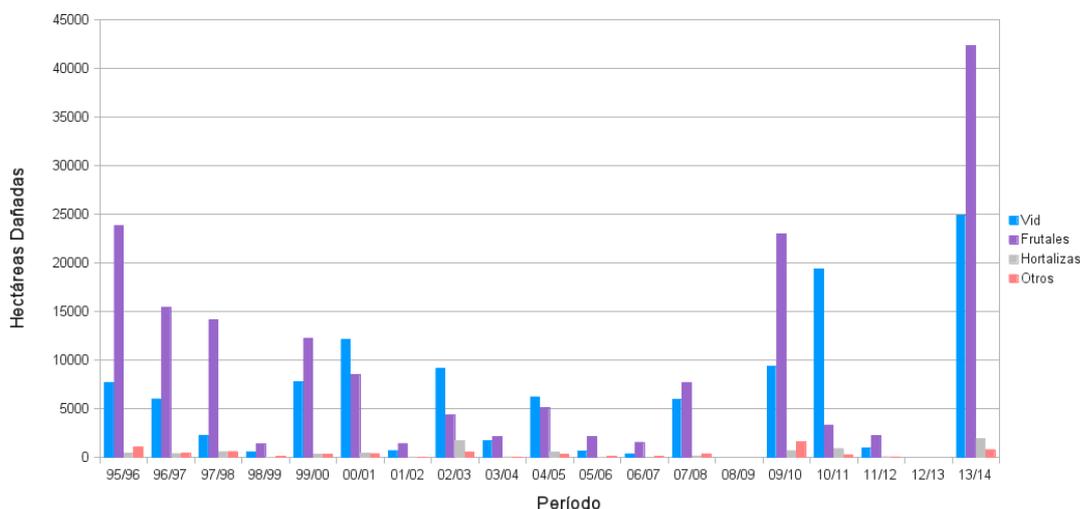


Figura 6.5: Daños por Heladas en Mendoza.

región de Cuyo, las pérdidas económicas debido a heladas alcanzan al 9 % del total del área cultivada bajo riego de la provincia.

En el año 1992, se perdió el 60 % de la cosecha de vid debido a heladas generales. En Octubre del 2002, heladas tardías dañaron 4000 hectareas de viñedos en la zona este de la provincia. Otro caso significativo fue el de Noviembre del 2004, cuando heladas tardías afectaron entre el 80 % y el 100 % de la producción de vid del departamento de San Carlos (uno de los principales departamentos productores de vid en Mendoza). Uno de los años en el que más daños causó la helada fue en el 2013, cuando se decretó la emergencia agropecuaria en la provincia debido a los daños causados por heladas generales. Este fenómeno, ocurrió durante el mes de septiembre y afectó en algunos distritos hasta el 80 % de los cultivos.

La Figura 6.5 ilustra la cantidad de hectareas dañadas a causa de las heladas para cada período desde el año 1995 hasta el año 2014 en la Provincia de Mendoza, Argentina. Los datos utilizados para generar el gráfico de la Figura 6.5, pueden consultarse en el sitio web de la Asociación de Cooperativas Vitivinícolas Argentinas (ACОВI) ².

Con el objetivo de reducir las pérdidas económicas debido a heladas, el Gobierno de la Provincia de Mendoza a través de la Dirección de Agricultura y Contingencias Climáticas (DACC) destina fondos económicos para pronóstico, alerta y prevención de heladas tardías, préstamos para combustibles y para defensa activa de heladas. Además la DACC genera información meteorológica de uso agrícola, la cual se puede consultar en el sitio

²<http://www.acovi.com.ar/observatorio/ambiental/>

web de la DACC ³.

6.2.2. Metodología Experimental

El servicio de predicción de heladas es ejecutado en distintas instancias de Amazon EC2 con el fin de obtener resultados que permitan elaborar los modelos de rendimiento de cada instancia. Las métricas de rendimiento utilizadas son el tiempo de ejecución y costo económico. La ejecución del servicio se realiza para distintas cantidades de nodos sensores (de 10 a 1000) en cada instancia. Debido a que para igual número de nodos (en la misma instancia) los tiempos de ejecución varían en cada ejecución, se deben realizar cuatro ejecuciones por cada experimento para extraer el valor medio del tiempo de ejecución. Luego, se utiliza el promedio del tiempo de ejecución y la lista de precios de Amazon para calcular el costo económico necesario para ejecutar el servicio de predicción de heladas.

Debido a que los modelos se obtienen mediante aproximaciones estadísticas, se debe evaluar la precisión de los mismos para una cantidad específica de nodos sensores, la cual será determinada en base a los requerimientos específicos de un caso típico de uso. Luego de determinar el error de cada modelo, se deben corregir los mismos para poder obtener cual es la instancia más adecuada para procesar el servicio de manera individual. Finalmente, los modelos corregidos permiten determinar cuales son las instancias que permitieran minimizar los tiempos y costos de procesamiento mediante el uso de clústeres virtuales, y cuantas deben utilizarse en el clúster.

6.2.3. Resultados Experimentales en Instancias Individuales

En esta subsección, se presentan los resultados obtenidos que han permitido elaborar los modelos de rendimiento de las instancias de Amazon cuando son ejecutadas de manera individual. El procesamiento del servicio de predicción de heladas permitió obtener resultados de rendimiento empíricos en cada tipo de instancia de Amazon EC2. En los experimentos, se consideraron cinco tipos de instancias. La Tabla 6.8 detalla las características de cada una de las instancias consideradas. Cada fila indica el nombre dado por Amazon a cada instancia. Cada columna de la Tabla 6.8 indica las características de cada una de las instancias, es decir, la cantidad de CPUs virtuales (vCPUs), Unidad de Cómputo de Amazon EC2 (ECU), Memoria (expresada en GBytes) y Precio. El modelo de precios de Amazon utilizado en este trabajo es el "bajo demanda", lo cual significa que el usuario paga por hora de uso de cada instancia ⁴. Además, el precio de las instancias varía según las características de la máquina.

³<http://www.http://contingencias.mendoza.gov.ar/>

⁴<http://aws.amazon.com/es/ec2/purchasing-options/>

Tabla 6.8: Características de las Instancias Utilizadas.

Instancia	vCPUs	ECU	Memoria [GBytes]	Precio [US\$]
t1.micro	1	variable	0,615	0,020
m1.small	1	1	1,7	0,047
m1.large	2	4	7,5	0,190
m1.xlarge	4	8	15	0,379
c3.xlarge	4	14	7,5	0,239

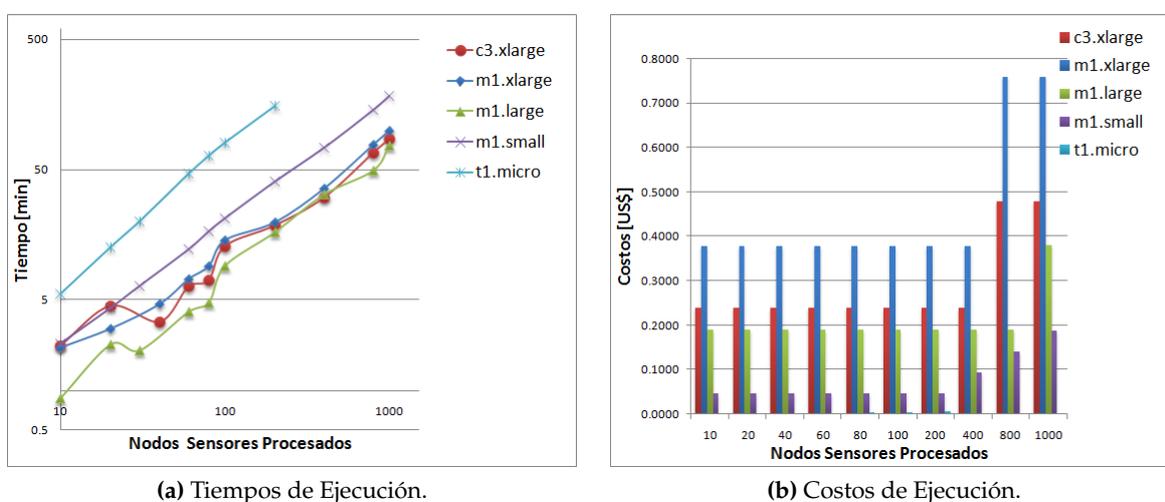


Figura 6.6: Tiempos de Ejecución y Costos Económicos para cada Tipo de Instancia.

La Figura 6.6a ilustra el tiempo de ejecución versus el número de nodos sensores procesados y la Figura 6.6b detalla los costos económicos versus el número de nodos sensores procesados.

Por un lado, en la Figura 6.6a se puede observar que la instancia donde se han logrado menores tiempos de ejecución para el servicio de predicción de heladas es la instancia m1.large. También, se observa que el rendimiento de la instancia m1.large es considerablemente mejor que el de las otras instancias cuando la cantidad de nodos sensores está entre 10 y 200. Luego, el rendimiento de m1.large se vuelve similar al de las instancias m1.xlarge y c3.xlarge.

Además, los resultados muestran que para máquinas multiprocesador como las instancias m1.large y c3.xlarge, los tiempos de procesamiento disminuyen para 30 y 40 nodos sensores respectivamente. Como se puede observar en la Tabla 6.8, las instancias m1.large y c3.xlarge tienen dos y cuatro CPU virtuales respectivamente y la misma memoria RAM (7,5 GBytes). Se puede entonces concluir que la disminución de los tiempos de procesamiento en la Figura 6.6a se debe al balance de carga entre los procesadores y el acceso

Tabla 6.9: Coeficientes de los Modelos de Predicción de Rendimiento Propuestos para cada Tipo de Instancia.

Instancia de Amazon EC2	a	b	c
t1.micro	0	$7,85 \times 10^{-1}$	-1,44
m1.small	$1,84 \times 10^{-6}$	$1,78 \times 10^{-1}$	1,80
m1.large	$6,02 \times 10^{-6}$	$6,50 \times 10^{-2}$	$9,98 \times 10^{-1}$
m1.xlarge	$1,40 \times 10^{-5}$	$8,24 \times 10^{-2}$	2,25
c3.xlarge	$1,66 \times 10^{-5}$	$6,73 \times 10^{-2}$	2,59

a los recursos compartidos de las máquinas (memoria, buses, etc.).

Por otro lado, en la Figura 6.6b se muestran los costos económicos obtenidos al procesar distintas cantidades de nodos sensores. Como se puede observar los costos económicos son los mismos cuando se procesan entre 10 y 400 nodos sensores en cada una de las instancias consideradas. Dicho comportamiento se debe a que Amazon establece los precios de las instancias por hora de uso, por lo que el precio es el mismo si el tiempo de procesamiento es menor o igual a una hora. De igual modo, si el tiempo de procesamiento es superior a una hora (por ejemplo de 800 a 1000 nodos sensores) se duplica el costo.

6.2.3.1. Modelos Teóricos de Predicción de Rendimiento

En esta subsección se presentan los modelos propuestos para estimar el rendimiento de cada instancia. Los modelos se generaron a partir de los datos obtenidos, mediante aproximación por polinomios de hasta segundo grado. Se utilizaron polinomios de este tipo por que son sencillos y se ajustan a los tiempos de ejecución observados en los resultados de la ejecución del servicio de predicción en las instancias consideradas. En general, las expresiones polinómicas utilizadas son de la siguiente la forma:

$$t = ax^2 + bx + c,$$

donde, t es el tiempo de ejecución a calcular con el modelo y x es la cantidad de nodos sensores procesados. Los valores de los coeficientes a , b y c para cada tipo de instancia son detallados en la Tabla 6.9. Estos coeficientes se obtuvieron al aproximar mediante polinomios los resultados obtenidos en la Subsección 6.2.3.

Con el fin de evaluar los modelos de predicción de rendimiento propuestos, se ha calculado con el modelo correspondiente a cada instancia el tiempo de ejecución y coste económico de cada una (Ver Figuras 6.7a y 6.7b respectivamente). Además, se han calculado los indicadores de rendimiento para más de 1000 nodos sensores (hasta 5000).

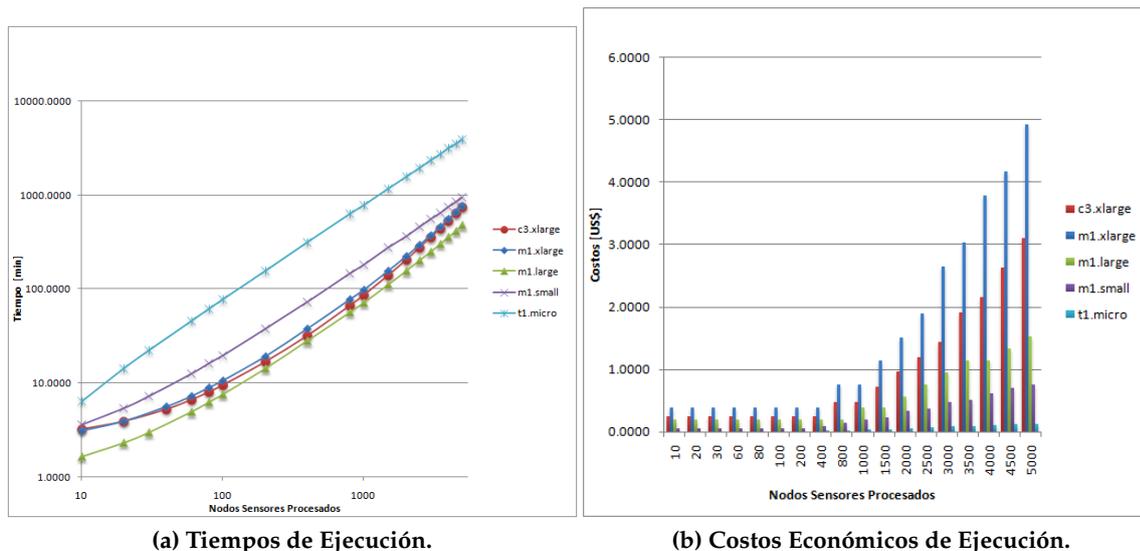


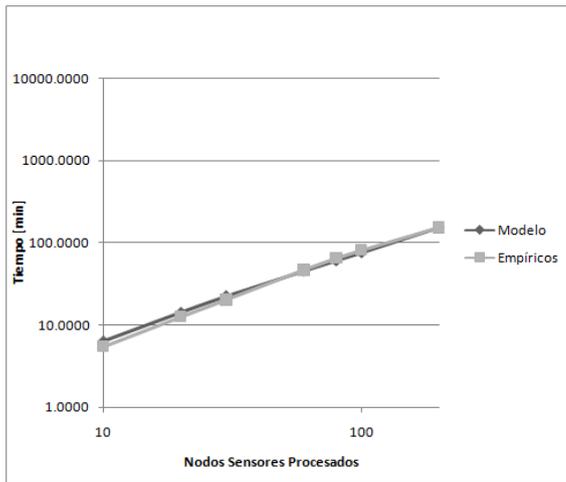
Figura 6.7: Tiempos y Costos de Ejecución Teóricos Calculados con los Modelos de Rendimiento.

Finalmente, como se muestra en las Figuras 6.8 y 6.9 se ha realizado una comparación entre los resultados empíricos y teóricos de cada tipo de instancia. En particular, la Figura 6.8 muestra que los tiempos de ejecución calculados por los modelos propuestos difieren segundos o pocos minutos (dependiendo de la instancia) con respecto a los tiempos obtenidos a través de la ejecución del servicio de predicción de heladas. Por este motivo se puede afirmar que los modelos propuestos son adecuados para estimar los tiempos de ejecución.

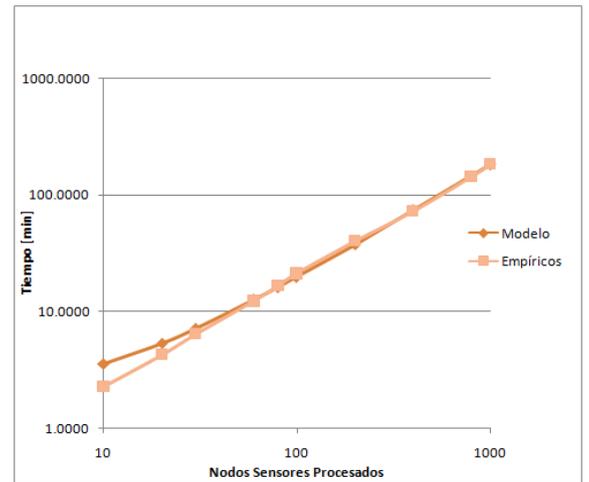
Respecto a los costos económicos, se ha detectado un caso particular cuando los tiempos de ejecución son cercanos a una hora. En esta situación los costos predichos serían el doble que los empíricos si los tiempos calculados por el modelo fueran mayor a una hora. Esto es así debido a que el precio de las instancias de Amazon se fija por hora de uso. De igual forma, el costo calculado por el modelo sería la mitad del costo real si el modelo predice un tiempo de ejecución menor a una hora. No obstante, ninguna de las dos situaciones mencionadas sucede cuando se utilizan los modelos propuestos, por lo que se puede concluir que la precisión de los modelos es adecuada para calcular los costos económicos.

6.2.3.2. Comparación de Instancias de Procesamiento en EC2

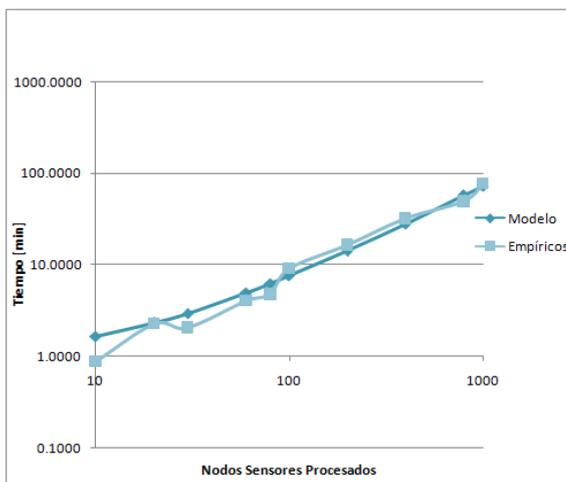
Con el objetivo de determinar cual es la instancia más adecuada para ejecutar el servicio de predicción de heladas, en este apartado se presenta una comparación en un caso de uso, el cual consiste en predecir heladas con datos provenientes de WSNs desplegadas en distintas zonas de la Provincia de Mendoza. La predicción se realiza para un día del mes



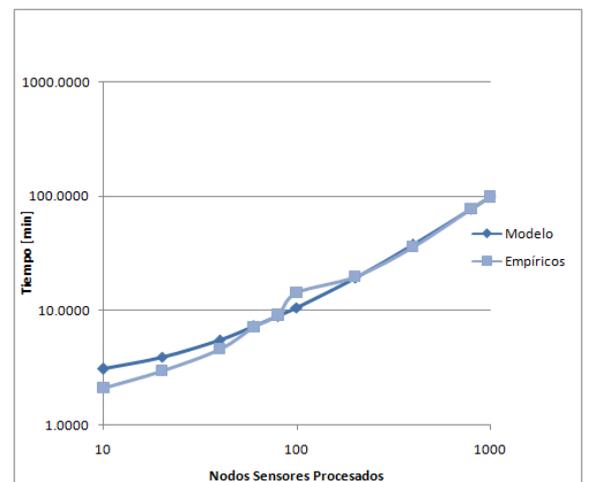
(a) Instancia t1.micro



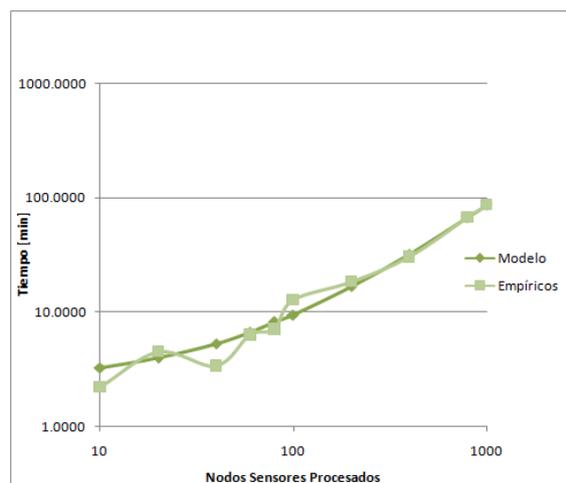
(b) Instancia m1.small



(c) Instancia m1.large

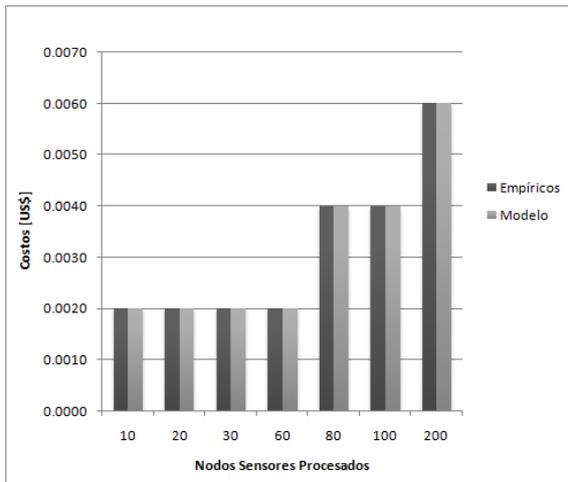


(d) Instancia m1.xlarge

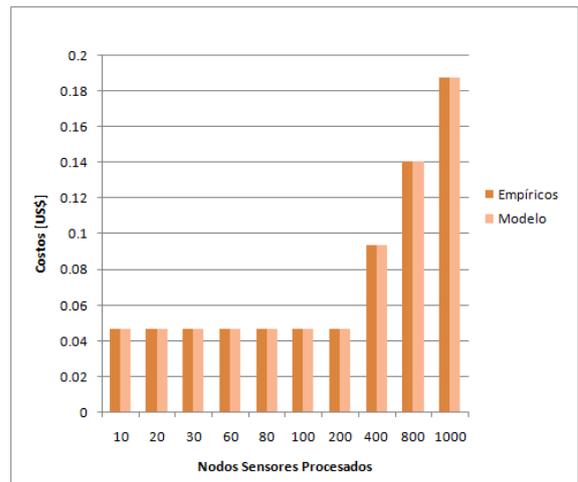


(e) Instancia c3.xlarge

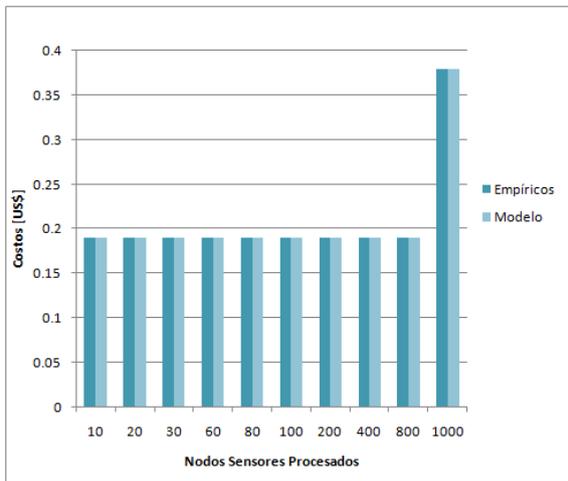
Figura 6.8: Tiempos de Ejecución Empíricos versus Teóricos Calculados con los Modelos de Rendimiento.



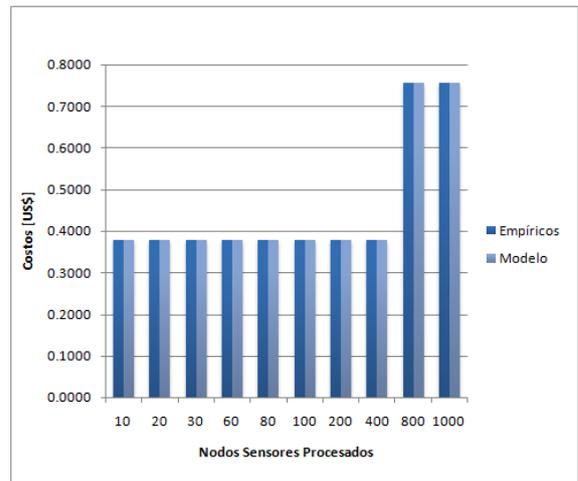
(a) Instancia t1.micro



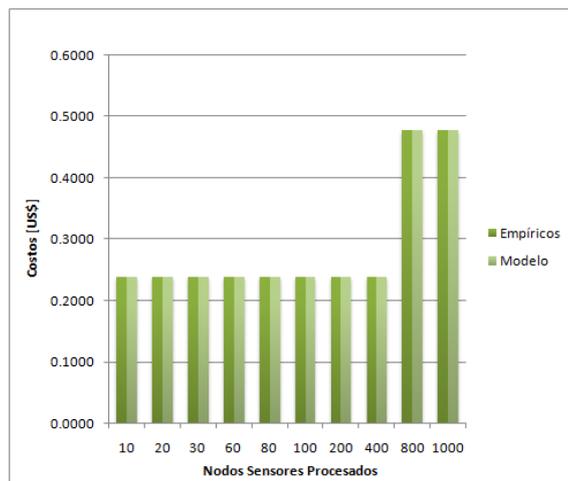
(b) Instancia m1.small



(c) Instancia m1.large



(d) Instancia m1.xlarge



(e) Instancia c3.xlarge

Figura 6.9: Costos Económicos Empíricos versus Teóricos Calculados con los Modelos de Rendimiento.

Tabla 6.10: Nodos Sensores Procesados en una Hora en Cada Tipo de Instancia.

Instancia de Amazon EC2	Nodos Sensores Procesados	Costo Económico [US\$]
t1.micro	78	0,020
m1.small	324	0,047
m1.large	841	0,190
m1.xlarge	632	0,379
c3.xlarge	722	0,239

de Julio. Se ha seleccionado este mes debido a que es uno de los meses de la temporada de heladas.

Se utilizarán los modelos para estimar los tiempos de ejecución y los costos económicos que surgirán cuando el servicio de predicción de heladas se ejecute sobre Cloud y calcule la temperatura mínima que sucederá en la noche. En base al valor predicho de temperatura, se emiten alarmas a los responsables de las fincas. Finalmente, los responsables deciden si se realizará el procedimiento de guardia contra heladas que se explicó en el Capítulo 2, Subsección 6.2.1.4.

Un aspecto a considerar para llevar a cabo la predicción y alertar al especialista, es que hay restricciones de tiempo. Las restricciones surgen por dos motivos, (i) el servicio de predicción de heladas necesita el valor de temperatura T_0 (el cual se adquiere en el mes de Julio en Mendoza a las 21:00 hs aproximadamente) y (ii) la logística del procedimiento de guardia contra heladas requiere que el personal de finca sea alertado antes de las 22:00 hs. En base a los motivos (i) y (ii) se puede concluir que el tiempo máximo de ejecución permitido para el servicio de predicción, debe ser menor o igual a una hora.

En la Tabla 6.10 se muestra la cantidad de nodos sensores procesados en una hora en cada instancia de Amazon EC2 y el costo económico correspondiente.

Como se puede observar, los resultados muestran que la instancia m1.large es la máquina que puede procesar el mayor número de nodos sensores en una hora. Además, el costo económico de dicha instancia es menor que el de las otras dos máquinas que han procesado más de 600 nodos sensores en una hora (m1.xlarge y c3.xlarge).

6.2.4. Precisión de los Modelos Propuestos para el Caso Típico de Uso

Debido a que los tiempos y costos de ejecución son predecidos por los modelos propuestos a través de métodos estadísticos, es necesario llevar a cabo experimentos para estudiar la precisión de los modelos para un número determinado de nodos sensores. En esta sección se discute la exactitud de los modelos teóricos de las instancias t1.micro, m1.small y m1.large para la cantidad de nodos sensores del caso de uso típico, presentado en la Subsección 6.2.3.2

Tabla 6.11: Tiempos de ejecución empíricos obtenidos para el caso típico de uso.

Instancia de Amazon EC2	Nodos Sensores Procesados	Tiempo de Ejecución Promedio [min]
t1.micro	78	12
m1.small	324	48,77
m1.large	841	53,30

La presente sección está organizada de la siguiente manera, en la Subsección 6.2.4.1, se detalla la ejecución del servicio de predicción de heladas para el número de nodos de sensores que pueden ser procesados en una hora, en cada instancia. Luego, en la Subsección 6.2.4.2, se realiza una comparación entre los tiempos de ejecución obtenidos en la Subsección 6.2.4.1 y los predichos a través de los modelos teóricos. El principal objetivo de esta comparación es el de obtener los errores de cada uno de los modelos de las instancias t1.micro, m1.small y m1.large en el caso típico de uso. A continuación, se utiliza el error obtenido para corregir el número de nodos de sensores que pueden procesarse en una hora. Finalmente, en la Subsección 6.2.4.3 se ejecuta el servicio de predicción de heladas para el número corregido de nodos de sensores en cada instancia.

6.2.4.1. Ejecución del Servicio de Predicción de Heladas para las Mejores Instancias del Caso Típico de Uso

En la presente subsección, se detallan los resultados obtenidos mediante la ejecución del servicio de predicción de heladas en las instancias t1.micro, m1.small y m1.large. Estas instancias han sido seleccionadas debido a que procesan más nodos sensores que las instancias m1.xlarge y c3.xlarge a menor costo económico. La reducción de costos y tiempos de ejecución, se logra cuando se ejecuta el servicio de predicción en una sola instancia m1.large ó en varias instancias t1.micro y m1.small ejecutadas en clústeres virtuales.

Los experimentos llevados a cabo en la presente sección, se realizan en cada una de las instancias (ejecutada en forma individual) para el número de nodos de sensores que pueden procesar en una hora predichos por los modelos teóricos de rendimiento. Con el objetivo de extraer el valor promedio del tiempo de ejecución, el servicio predicción de heladas se ejecuta cuatro veces en cada instancia de Amazon EC2.

La Tabla 6.11 presenta los valores promedio de los tiempos de ejecución obtenidos a través de la ejecución del servicio de predicción de heladas en cada instancia.

6.2.4.2. Comparación de Resultados

Con el objetivo de determinar el error de los modelos propuestos, en la Tabla 6.12, se realiza una comparativa entre los tiempos de ejecución obtenidos mediante experimentos y los tiempos predichos por los modelos para las instancias t1.micro, m1.small y m1.large. La columna dos de la Tabla 6.12, detalla el número de nodos sensores procesados en cada instancia. Luego, la columna tres muestra los tiempos de ejecución predichos con los modelos teóricos y la columna cuatro los tiempos obtenidos mediante experimentos. Finalmente, la columna cinco presenta el porcentaje de error observado entre los resultados empíricos y teóricos para cada instancia.

Tabla 6.12: Comparación entre los tiempos de ejecución empíricos y teóricos obtenidos para el caso típico de uso.

Instancia	Nodos Sensores	Tiempo de Ejecución Teórico [min]	Tiempo de Ejecución Empírico [min]	Error [%]
t1.micro	78	60	12	80
m1.small	324	60	48,77	18,83
m1.large	841	60	53,30	11,16

Por un lado, en la segunda fila de la Tabla 6.12, se presentan los resultados obtenidos para la instancia t1.micro. Estos resultados muestran una gran diferencia entre los tiempos empíricos y teóricos (aproximadamente un 80 % mayor el tiempo de ejecución empírico que el teórico). Este error, se debe a que el rendimiento de la instancia t1.micro es variable ⁵.

Por otra parte, en la tercera fila de la Tabla 6.12, se muestran los experimentos realizados en la instancia m1.small. Los resultados obtenidos indican que el errores de 18,83 % entre los tiempos teóricos y empíricos. Se puede observar que el error obtenido en la instancia m1.small es menor que el registrado en el caso de la instancia t1.micro.

Finalmente, la cuarta fila de la Tabla 6.12 detalla los resultados de los experimentos realizados en la instancia m1.large. Los resultados obtenidos en esta instancia, indican un error de 11,16 % entre los tiempos de ejecución empíricos y los teóricos.

6.2.4.3. Ejecución del Servicio de Predicción de Heladas para Resultados Corregidos

En la subsección anterior, se obtuvieron los errores de los modelos teóricos de las instancias t1.micro, m1.small y m1.large. Los errores obtenidos han permitido corregir el número de nodos sensores que puede procesar cada instancia en una hora (caso típico de uso). En la Tabla 6.13 se muestra el número corregido de nodos sensores que pueden procesar las instancias t1.micro, m1.small y m1.large en el caso típico de uso. La segunda

⁵<http://aws.amazon.com/es/ec2/previous-generation/>

Tabla 6.13: Corrección del número de nodos sensores procesados para el caso típico de uso.

Instancia de Amazon EC2	Cantidad de Nodos Sensores Calculadas por el Modelo	Cantidad de Nodos Sensores Corregida	Error [%]
t1.micro	78	140	80
m1.small	324	385	18,83
m1.large	841	934	11,16

Tabla 6.14: Resultados obtenidos para la cantidad de nodos sensores corregidos en el caso típico de uso.

Instancia de Amazon EC2	Nodos Sensores Procesados	Tiempos de Ejecución Empíricos [min]
t1.micro	140	46,23
m1.small	385	54,88
m1.large	934	58,40

columna de la Tabla 6.13 presenta el número de nodos sensores calculado por los modelos teóricos, que pueden ser procesados en el caso típico de uso. Luego, la tercera columna detalla el número corregido de nodos sensores que pueden ser procesados en el caso típico de uso. Finalmente, la cuarta columna muestra los errores utilizados para corregir el número de nodos sensores que pueden ser procesados por cada instancia en una hora.

Una vez corregida la cantidad de nodos sensores que pueden ser procesados en una hora, se debe validar experimentalmente que la cantidad corregida de nodos sensores para cada instancia es correcta. La Tabla 6.14 muestra los tiempos de ejecución obtenidos cuando el servicio de predicción de heladas procesa el número corregido de nodos sensores en cada instancia.

Por un lado, en la fila dos de la Tabla 6.14, se presentan los resultados obtenidos para la instancia t1.micro. Los resultados muestran que la instancia t1.micro puede procesar 140 nodos sensores en menos de una hora (46,23 minutos). Como se mencionó anteriormente, el rendimiento de la instancia t1.micro no es constante, por lo que el tiempo de ejecución varía según la hora a la cual es ejecutado el servicio de predicción de heladas. Sin embargo, debido a que el caso típico de uso determina que la predicción de heladas se debe realizar a las 21:00 hs, se considerará el tiempo de ejecución para el caso en el cual la aplicación se ejecuta a las 21:00 horas.

Por otro lado, en la fila tres de la Tabla 6.14, se detallan los experimentos realizados en la instancia m1.small. Estos resultados muestran que la instancia m1.small puede procesar 385 nodos sensores en 54,88 minutos, por lo que se puede afirmar que la cantidad de nodos sensores corregida es correcta para el caso típico de uso en esta instancia.

Finalmente, la fila cuatro de la Tabla 6.14 muestra los tiempos de ejecución obtenidos en

la instancia m1.large. Como en el caso de la instancia m1.small, el tiempo de ejecución obtenido (58,40 min) permite validar la corrección efectuada en la cantidad de nodos sensores que pueden ser procesados en una hora.

6.2.5. Resultados Experimentales en Clústeres Virtuales

Los experimentos realizados en la Subsección 6.2.4.3, han permitido determinar que la instancia de Amazon EC2 que procesa mayor número de nodos sensores en forma individual es la instancia m1.large. Sin embargo, los experimentos mencionados también sugieren que las instancias t1.micro y m1.small pueden ser utilizadas en paralelo para procesar más nodos sensores que la instancia m1.large con menores costos económicos. En esta subsección, se llevan a cabo experimentos para determinar el número de nodos sensores que pueden ser procesados en las instancias t1.micro y m1.small cuando se ejecutan en paralelo. La ejecución se realiza sólo en los clústeres virtuales en los cuales el costo por hora es inferior al de una instancia m1.large ejecutada en forma individual.

La metodología experimental seguida es la siguiente, en primer lugar, se considera la cantidad de nodos de sensores que puede procesar cada instancia en una hora cuando funciona de manera individual. Luego, se implementa un clúster virtual compuesto por la cantidad de máquinas virtuales cuyo costo total sea menor de 0,19 dólares (el cual corresponde al costo de una hora de la instancia m1.large). A continuación, se ejecuta el servicio de predicción de heladas en cada instancia del clúster virtual, para el número de nodos de sensores que puede procesar cada instancia en una hora. Finalmente, se mide el tiempo de ejecución. El experimento se realiza cuatro veces en cada clúster virtual para obtener valores promedio del tiempo de ejecución.

La Tabla 6.15 indica el número de nodos de sensores procesados en dos clústeres virtuales, uno compuesto por instancias t1.micro y otro por máquinas m1.small. La primera columna detalla cual es la instancia de Amazon EC2 que se utiliza en cada clúster virtual. La segunda columna indica el número de nodos sensores procesados y la tercera columna la cantidad de instancias que se utilizan en paralelo. La cuarta columna presenta el tiempo de ejecución promedio obtenido y finalmente la quinta columna, muestra el costo económico resultante de la ejecución del servicio de predicción de heladas en paralelo mediante clústeres virtuales.

En comparación con los resultados obtenidos para la instancia m1.large (ver Tabla 6.14), los resultados detallados en la Tabla 6.15 indican que el clúster m1.small puede procesar 606 nodos sensores más en una hora que una sola instancia m1.large y a menor costo económico. Este resultado se obtiene mediante el uso de cuatro instancias m1.small ejecutadas en paralelo. Además, el tiempo de ejecución promedio obtenido es de 57 minutos para este caso, lo que cumple con el requisito de tiempo definido en el caso típico de uso.

Tabla 6.15: Tiempos de ejecución en clústeres virtuales para el caso típico de uso.

Instancia de Amazon EC2	Nodos Sensores Procesados	Instancias Ejecutadas en Paralelo	Tiempo de Ejecución Promedio [min]	Costo Económico [US\$]
t1.micro	1260	9	51,44	0,18
m1.small	1540	4	57	0,188

Finalmente, si se compara el número de nodos de sensores procesados en una hora en cada clúster virtual indicado en la columna dos de la Tabla 6.15, se puede concluir que el mejor clúster para procesar el servicio de predicción de heladas es el compuesto por las instancias m1.small. Esto se debe a que el clúster m1.small puede procesar 280 nodos sensores más que el clúster virtual t1.micro (a un costo similar). Además, este clúster no tiene el rendimiento variable propio de las instancias t1.micro.

6.3. Conclusiones

En el presente capítulo, se han realizado distintos experimentos con Sensor Cirrus. Por una parte, se llevaron a cabo pruebas para estudiar la eficiencia de la plataforma desarrollada. El experimento consistió en medir la tasa de pérdida de paquetes (PLR) para una aplicación de monitoreo de heladas en campo. Se plantearon dos escenarios de prueba distintos, uno compuesto por una WSN de media escala (42 nodos sensores por estación base) y otro por una WSN de gran escala (compuesta por 300 nodos sensores por estación base).

Los resultados obtenidos permitieron determinar que Sensor Cirrus presenta valores de PLR de 3,178 % para la WSN de medias escala y de 11,46 % para la de gran escala. Respecto a la PLR de la WSN de media escala, los resultados mostraron que es menor al valor de PLR adecuado para aplicaciones de monitoreo de heladas (menor ó igual al 10 %). En el caso de la WSN de gran escala, la PLR supera al valor objetivo en un 1,46 %. Debido a que en condiciones de trabajo reales este valor puede aumentar considerablemente, se puede concluir que la plataforma realiza la gestión eficiente de WSNs compuestas por hasta 200 nodos sensores por estación base.

Respecto a la pérdida de paquetes en cada enlaces del proceso de comunicación de Sensor Cirrus, los resultados de ambos escenarios mostraron que el único enlace donde se pierden datos es en el enlace entre la WSN y el Módulo WSN - TCP/IP. Las pérdidas en el enlace mencionado se deben a que el Coordinador de la WSN no le envía tramas de acuse de recibo a los nodos por que está ocupado comunicandose con otro nodo. Ante la ausencia de la trama de acuse de recibo el nodo sensor hace tres intentos más de envío de los datos y si no recibe el acuse de recepción descarta la trama de datos para ahorrar

batería.

Por otra parte, se realizaron experimentos con el servicio de predicción de heladas de Sensor Cirrus. Las pruebas realizadas permitieron validar el uso de tecnologías disponibles en el mercado local para dar solución a un problema real de la Región de Cuyo en Argentina, como lo son las pérdidas en cosechas debido a heladas. Además, se pudo construir una serie de modelos teóricos para estimar el rendimiento de instancias de Amazon EC2 destinadas al procesamiento de un servicio de predicción de heladas. Estos modelos permiten seleccionar cuál es la instancia más adecuada para el procesamiento de dicho servicio y optimizar los costos económicos al utilizar recursos de cómputo del Cloud.

En base a los resultados obtenidos, se puede concluir que los modelos propuestos son adecuados para predecir tiempos de ejecución y costos económicos. Sin embargo, se ha observado que los modelos propuestos presentan algunos problemas cuando se utilizan para predecir tiempos de ejecución y costo para un número específico de nodos sensores., por lo que se realizaron nuevos experimentos en un caso típico de uso. Estos experimentos fueron sólo realizados en las instancias que presentan mejor rendimiento en forma individual (m1.large) o paralela (t1.micro y m1.small). Luego, los resultados obtenidos permitieron determinar los errores de los modelos de cada una de las instancias mencionadas para el caso típico de uso.

Los resultados de las pruebas realizadas en el caso típico de uso, mostraron que los errores de los modelos son del 80 %, 18,83 % y 11,16 % para las instancias t1.micro, m1.small y m1.large, respectivamente. El modelo que tiene un mayor error es el de la instancia t1.micro. Esto es debido a la variación del rendimiento propio de esta instancia. Luego de determinar el error de cada modelo, se corrigió el número de nodos de sensores que pueden ser procesados en el caso típico de uso y se realizaron experimentos para validar si el número corregido de nodos sensores cumple con las restricciones de tiempo propias del caso típico de uso. Los resultados mostraron que la cantidad de nodos sensores fue corregida correctamente para cada caso y que las instancias t1.micro, m1.small y m1.large pueden procesar 140, 385 y 934 nodos sensores en una hora, respectivamente.

Los resultados obtenidos permitieron determinar cual es la mejor instancia para el procesamiento del servicio de predicción de heladas en forma individual. Se determinó que para WSNs compuestas por pocos nodos sensores (hasta 140) la instancia más adecuada es la t1.micro. Luego, para WSNs compuestas por mayores cantidades de nodos sensores: 140 - 385, 386 - 934; se deberían utilizar las instancias m1.small y m1.large, respectivamente.

Si bien las instancias m1.xlarge y c3.xlarge son las que tienen mayor rendimiento según Amazon, no se observaron diferencias importantes de rendimiento en comparación con las otras instancias ensayadas. Además, si se consideran los costos elevados de ambas

máquinas se puede concluir que no se recomienda el uso de las instancias m1.xlarge y c3.xlarge para este tipo de aplicaciones.

Los resultados obtenidos para la ejecución del servicio de predicción de heladas en máquinas individuales indicaron que se deben utilizar múltiples instancias de EC2 ejecutadas en paralelo, para ejecutar el servicio en WSNs compuestas por más de 934 nodos sensores. Con el objetivo de determinar cual es la instancia más adecuada para procesar el servicio de predicción en paralelo, se realizaron pruebas en clústeres virtuales. Se utilizaron dos clústeres virtuales, uno compuesto por nueve máquinas t1.micro y otro por cuatro instancias m1.small. Estas instancias fueron seleccionadas porque pueden procesar más de 934 nodos sensores, a un costo menor que el obtenido para la mejor instancia ejecutada en forma individual en el caso típico de uso (m1.large).

Los resultados de las pruebas en clústeres virtuales, demostraron el clúster t1.micro puede procesar 1260 nodos sensores en una hora, y el clúster m1.large 1540 nodos sensores. Se puede entonces concluir que el mejor clúster para procesar el servicio de predicción de heladas en WSNs compuestas por más de 934 nodos de sensores es el conformado por las máquinas m1.small.

Finalmente, se puede concluir que las pruebas realizadas han permitido validar los criterios que componen la Hipótesis de Trabajo de la presente tesis doctoral donde se sostuvo que *“Se puede gestionar en forma remota, sencilla y eficiente WSNs conformadas por recursos acotados de hardware y fuente de energía, mediante tecnologías de computación distribuída como Cloud Computing, adecuando las tecnologías disponibles en el mercado local y con un bajo costo”*.

Conclusiones y Trabajos Futuros

La gestión remota de WSNs con tecnologías cloud ha abierto nuevas oportunidades para el estudio de distintos problemas científicos e industriales. Entre otras aplicaciones de WSNs gestionadas mediante tecnologías cloud, el estudio y prevención de heladas es de especial interés para la Región de Cuyo, Argentina. Las heladas generan grandes pérdidas económicas en la producción agrícola de la región, por lo que se requiere acceso en tiempo real a infraestructuras que permitan recolectar información en campo y proporcionen recursos computacionales para almacenar y procesar dicha información. Gracias a las WSNs y a Cloud, se puede acceder a información que permita activar sistemas de defensa contra heladas y minimizar los daños causados por heladas en los cultivos de la región.

La gestión remota de WSNs mediante tecnologías cloud ha permitido avanzar en el estudio y monitoreo de distintos fenómenos, sin embargo han surgido nuevos desafíos en el área. En primer lugar, surge la necesidad de encontrar soluciones adecuadas a la incompatibilidad de protocolos entre ambas tecnologías. Si bien existen enfoques para solucionar dicha incompatibilidad (gateway y redes overlay), se debe seleccionar el más adecuado para gestionar WSNs con recursos acotados de hardware y fuente de energía, debido a que este tipo de WSNs son las actualmente utilizadas para realizar monitoreo de heladas.

Otro de los desafíos en el área es el de gestionar las WSNs con tecnologías cloud conservando las condiciones propias de su buen funcionamiento. Esto implica gestionar en forma remota las WSNs sin generar consumos extras de energía y de recursos de hardware en los nodos sensores. Además, la gestión remota debe ser realizada en forma eficiente, es decir se deben mantener valores adecuados de tasas de errores en la entrega de datos.

Finalmente, se debe garantizar que la plataforma utilizada para gestionar las WSNs provea recursos computacionales acordes a la demanda y esté disponible las 24 horas del día y

los 365 días del año.

En base a los aspectos discutidos anteriormente, esta tesis doctoral propone una nueva plataforma de gestión remota de WSNs con tecnologías cloud denominada Sensor Cirrus. Dicha plataforma permite: compatibilizar los protocolos de las WSNs y Cloud, acceder a recursos computacionales para almacenar y procesar los datos generados por las WSNs, y finalmente gestionar WSNs con tecnologías cloud manteniendo las condiciones propias del buen funcionamiento de las WSNs.

Este capítulo está organizado de la siguiente manera. En la Sección 7.1, se presentan las contribuciones más importantes de esta tesis. Luego, en la Sección 7.2, se discuten las limitaciones principales de Sensor Cirrus y en la Sección 7.3 se detallan las perspectivas para trabajos futuros. Finalmente, la Sección 7.4 concluye la presente tesis.

7.1. Contribuciones

Esta tesis presenta las siguientes contribuciones al área de Clouds de Sensores:

- Una *nueva plataforma para la gestión remota de Clouds de Sensores*, la cual es adecuada a los recursos y tecnologías disponibles en Argentina. La plataforma utiliza gateways para incorporar WSNs a TCP/IP, y servicios Cloud estándar no programados en la WSN. Esto permite que las WSNs puedan ser gestionadas en forma remota mediante tecnologías Cloud, sin que se generen consumos extras de recursos de hardware y de fuente de energía en los elementos de las WSNs.

La implementación de Sensor Cirrus ha permitido demostrar que la combinación del enfoque de gateways y de servicios Cloud no programados en la WSN no genera cargas extras de software y de fuente de energía en los nodos sensores. Además, se ha validado que Sensor Cirrus puede gestionar datos provenientes de WSNs implementadas con plataformas de hardware de bajo costo económico y disponibles en el mercado local.

- Un *Cloud de Sensores tecnológicamente factible*, compuesto por plataformas de hardware disponibles en el mercado local y de bajo costo económico. Sensor Cirrus permite el desarrollo de Clouds de Sensores de bajo costo económico que permitan ejecutar servicios para activar sistemas de defensa contra heladas y mitigar los daños producidos en los cultivos.

Los experimentos realizados han permitido ejecutar en un Cloud de Sensores un prototipo de servicio de predicción de heladas que utiliza datos provenientes de WSNs, estaciones meteorológicas y recursos computacionales de un Cloud Público.

Los resultados obtenidos fueron logrados con tecnologías WSN y Cloud factibles de ser implementadas en el mercado local y tiene bajo costo económico.

- Una *nueva interfaz* que permite realizar la *gestión remota de WSNs en forma sencilla*, sin que se requiera por parte del usuario conocimientos en la programación de WSNs. Esto se ha logrado mediante el uso de servicios Cloud estándar como Google Drive y tecnologías de redes simples como sockets TCP. Las pruebas llevadas a cabo demostraron que el uso de Google Drive y sockets TCP permiten obtener tasas de pérdida de datos menores al 10 % en el proceso de comunicación.
- Una *prueba de concepto de un servicio Cloud de predicción de heladas*, basado en información proveniente de WSNs y estaciones meteorológicas. El servicio de predicción de heladas de Sensor Cirrus ha permitido realizar pronósticos de heladas de radiación específicos para cada sector de la finca donde está desplegada la WSN.

Se ha logrado implementar y validar el servicio de predicción mediante experimentos en el Cloud de Amazon EC2. Además, se determinó cual es la instancia de Amazon más adecuada para procesarlo, ya sea de manera individual ó paralela mediante clústeres virtuales.

- La *materialización de los aportes mencionados* en un nuevo prototipo de plataforma de gestión de Clouds de Sensores denominado Sensor Cirrus. Los experimentos realizados permitieron validar la capacidad de Sensor Cirrus para gestionar en forma sencilla y eficiente WSNs de bajo costo económico y servicios Cloud. Además, se logró visualizar, almacenar y procesar grandes volúmenes de datos recolectados en campo y cambiar la configuración de las WSNs de acuerdo a las necesidades del usuario.
- Finalmente, cabe destacar que Sensor Cirrus estará al alcance de todos los productores agrícolas de la Región de Cuyo, los cuales contarán con una nueva herramienta para minimizar los daños causados por heladas en los cultivos. En resumen, los aportes materializados en esta tesis permitirán contribuir al desarrollo local de la Región gracias a la disminución del impacto de las heladas en la economía.

7.2. Limitaciones

Sensor Cirrus realiza la gestión de Cloud de Sensores mediante gateways y servicios cloud estándar no embebidos en la WSN. Además, la plataforma permite procesar los datos recolectados en campo a través de recursos de un Cloud público para prevenir los daños causados por las heladas. A continuación se discuten las principales limitaciones de Sensor Cirrus.

Primera Limitación. Surge porque Sensor Cirrus utiliza gateways para incorporar las WSNs a TCP/IP. Por un lado, el uso de gateways implica centralizar el punto de fallo de la WSN en la estación base de la WSN. Esto significa que en caso de presentarse una falla en el gateway, toda la WSN pierde la conexión a TCP/IP. Para solucionar este inconveniente, se deben utilizar técnicas de alta disponibilidad como redundancia. Esta técnica consiste en utilizar dos gateways, uno en modo on-line y otro en modo off-line. Cuando se detecta la falla en el gateway on-line, el que se encuentra en modo off-line es activado y recupera la conexión a TCP/IP.

Por otro lado, otro inconveniente que surge al utilizar gateways es la concentración del tráfico en la estación base de la WSN. Este inconveniente es más notorio en WSNs compuestas por cientos de nodos sensores conectados a una estación base. Este problema se puede solucionar si se aumentan la cantidad de estaciones bases y disminuye la cantidad de nodos sensores por estación base. Sin embargo, esto implica un aumento en los costos económicos por la mayor cantidad de estaciones bases requeridas.

Segunda Limitación. En Sensor Cirrus se ha utilizado Google Drive como servicio cloud estándar para realizar la gestión remota de las WSNs. El principal inconveniente de Google Drive es que sólo está disponible para sistemas operativos Windows, Android y MAC. Esto implica que no se puede utilizar en plataformas que utilizan linux, las cuales pueden ser estaciones bases ó instancias del Cloud. Este inconveniente puede ser solucionado mediante el uso de servicios cloud para linux compatibles con Google Drive (como Grive) ó con servicios cloud compatibles con todos los sistemas operativos como Dropbox.

Tercera Limitación. La tercera limitación de Sensor Cirrus está relacionada al servicio de predicción de heladas. Este servicio funciona de forma monolítica, por lo que no está optimizado para ser ejecutado de forma paralela. Esta limitación puede ser solucionada mediante el uso de distintos hilos dentro de la aplicación. Cada uno de estos hilos correspondería a un nodo sensor a procesar y podría ser ejecutado en un procesador de una de las instancias del clúster virtual utilizado. Esto permitiría finalmente reducir los tiempos y costos de procesamiento.

Con el objetivo de resolver algunas de las limitaciones arriba mencionadas, en la próxima sección se presentan las perspectivas sobre trabajos futuros.

7.3. Trabajos Futuros

7.3.1. Implementación de Estaciones Bases basadas en Android OS

La estación base utilizada en la WSN experimental gestionada por Sensor Cirrus está basada en una PC con sistema operativo linux y el coordinador XBee ZigBee. El principal inconveniente de esta estación base es que no tiene la confiabilidad de funcionamiento y el bajo consumo de energía de las estaciones bases implementadas en sistemas embebidos. Para solucionar este inconveniente, se realizarán nuevos desarrollos en estaciones bases de WSNs. Estas estaciones bases estarán basadas en sistemas embebidos que soporten linux y tengan interprete de python. Se podrá entonces reducir las posibilidades de fallos y el consumo de energía.

Además, se investigarán nuevas posibilidades como el uso de dispositivos Android de primera y segunda generación. Estos dispositivos tienen un bajo costo económico debido a que están siendo reemplazados por teléfonos de nueva generación y pueden transmitir datos a Internet mediante GPRS ó WiFi. Además, pueden conectarse al coordinador de la WSN mediante un gateway Bluetooth - ZigBee, el cual se encuentra en etapa final de desarrollo en el marco de un trabajo final de grado dirigido por el autor de esta tesis.

7.3.2. Paralelización del Servicio de Predicción de Heladas de Sensor Cirrus

Se continuará con la optimización del servicio de predicción de heladas de Sensor Cirrus para su ejecución en máquinas multiprocesados y clústeres virtuales. Los trabajos estarán focalizados en mejorar el código del servicio mediante la incorporación de múltiples hilos y balanceadores de carga entre procesadores e instancias de clústeres virtuales. Además, se deberá estudiar el acceso del servicio a las bases de datos que contienen la información recolectada por las WSNs y las latencias propias de las comunicaciones de los múltiples procesos del servicio, una vez paralelizado.

7.3.3. Validación de la Calidad de la Predicción de Heladas Realizada por Sensor Cirrus

En esta tesis, no se ha profundizado en la evaluación de la calidad de la predicción realizada por el servicio a través del método de Snyder y Melo Abreu. Por una parte, se realizarán pruebas con datos de una temporada de heladas recolectados en la finca de General Alvear donde se desplegó la WSN experimental para evaluar la calidad de la predicción del servicio de Sensor Cirrus.

Por otra parte, se incorporarán nuevos algoritmos basados en aprendizaje de máquinas, los cuales están actualmente en etapa de desarrollo en el marco del proyecto de investigación *PICT-2012-2731 "Predicción localizada de heladas en la provincia de Mendoza mediante técnicas de aprendizaje de máquinas y redes de sensores"*

Finalmente, se incorporarán al servicio de predicción de heladas datos en tiempo real de las WSNs, estaciones meteorológicas y sensores remotos como satélites y radares. Esto permitirá detectar la ocurrencia de heladas de advección, evaporación y mixtas. Además se implementará el servicio de alarma a la comunidad de productores de la provincia a través de sms, correo electrónico y redes sociales.

7.3.4. Utilización de Clouds de otros Proveedores y Selección del más Adecuado para Procesar el Servicio de Predicción de Heladas

Actualmente, hay una amplia oferta de proveedores de servicios cloud de procesamiento (Amazon EC2, Google Cloud, Windows Azure). En los experimentos realizados en esta tesis, sólo se han utilizado instancias de Amazon EC2 para ejecutar el servicio de predicción de heladas de Sensor Cirrus. Se continuará la investigación mediante la ejecución del servicio en otros clouds públicos como los de Google y Microsoft. Esta investigación, incluirá el desarrollo de nuevos modelos teóricos que permitan determinar cual es el cloud más adecuado para procesar el servicio de predicción y cuales serán las instancias de dicho cloud que deberán utilizarse. Además, se trabajará con nuevos algoritmos basados en Meta-heurísticas que permitan seleccionar el cloud más conveniente (en términos de costo económico y tiempos de procesamiento) para procesar el servicio al momento de realizar la predicción de heladas.

7.4. Observaciones Finales

Las WSNs brindan soluciones al estudio y prevención de fenómenos agro-meteorológicos como tormentas de granizo, heladas e inundaciones. Estas redes pueden ser gestionadas en forma remota mediante tecnologías tradicionales de Internet y de computación distribuida como Grid y Cloud Computing. A diferencia de las tecnologías tradicionales de Internet, la gestión remota de WSNs mediante tecnologías cloud permite contar con recursos computacionales acordes a los requerimientos del usuario, las 24 horas al día, los 365 días del año.

El uso de tecnologías cloud para la gestión remota de WSNs ha dado lugar a nuevas infraestructuras a gran escala denominadas Clouds de Sensores, compuestas por WSNs y recursos computacionales de altas prestaciones. Estas infraestructuras deben ser gestion-

adas por plataformas adecuadas al contexto local, es decir plataformas tecnológicamente factibles de ser implementadas en Argentina.

Con el objetivo de desarrollar plataformas de gestión de Clouds de Sensores acordes a los requerimientos del mercado local, se deben utilizar tecnologías cloud que permitan gestionar las WSNs disponibles en el mercado argentino. Estas WSNs tienen plataformas de hardware con recursos acotados de hardware y fuente de energía. Además, se deben utilizar tanto a nivel WSN como a nivel cloud las tecnologías que tengan menor costo económico y que sean sencillas de implementar.

En esta tesis se ha desarrollado una nueva plataforma de gestión de Cloud de Sensores denominada Sensor Cirrus. Dicha plataforma permite el desarrollo de Clouds de Sensores de bajo costo económico y acordes a los requerimientos del mercado local. Si bien Sensor Cirrus puede brindar soluciones a una amplia variedad de aplicaciones agrometeorológicas (tormentas de granizo, viento zonda, etc.), en esta tesis se ha aplicado la plataforma al estudio y prevención de heladas. Este fenómeno causa grandes pérdidas económicas en la Región de Cuyo, Argentina, por lo que se ha desarrollado un servicio de predicción de heladas que utiliza datos de WSNs y servicios de procesamiento de Cloud Computing para predecir la posibilidad de ocurrencia de heladas.

Por un lado, los experimentos realizados en el Capítulo 5 han permitido desarrollar una WSN experimental para el estudio de dicho fenómeno y estudiar el desempeño de la WSN en base a aspectos como consumo de energía, potencia de transmisión y pérdida de datos. Por otro lado, las pruebas realizadas en el Capítulo 6 han permitido validar la eficiencia de Sensor Cirrus para gestionar WSNs y estudiar el desempeño del Cloud de Amazon EC2 para procesar el servicio de predicción de heladas.

En resumen, Sensor Cirrus es un nuevo aporte al área de los Clouds de Sensores, el cual ha abierto nuevas posibilidades de investigación para mejorar los aspectos mencionados en la Sección 7.3. Estas investigaciones permitirán implementar nuevos gateways para WSNs, mejorar la calidad del servicio de predicción de heladas, y finalmente optimizar tanto los tiempos como los costos económicos de procesamiento de dicho servicio.

Publicaciones Realizadas

Publicaciones en Revistas y CCIS

- L. Iacono, P. Godoy, O. Marianetti, C. García Garino, y C. Párraga. Estudio de la Integración entre WSN y redes TCP/IP, Memoria de Trabajos de Difusión Científica y Técnica, Volumen 10, 2012. Universidad de Montevideo, Facultad de Ingeniería. Editor: M. Vega. ISSN 1510- 7450. [Latindex].
- L. Iacono, José Luis Vázquez-Poletti, C. García Garino, and I. M. Llorente. A Model to Calculate Amazon EC2 Instance Performance in Frost Prediction Applications. In First HPCLATAM - CLCAR Joint Conference Latin American High Performance Computing Conference (CARLA) Valparaíso, Chile. In High Performance Computing. Communications in Computer and Information Science (CCIS). Volumen 485, 2014. Springer. ISBN 978-3-662-45482-4.

Publicaciones en Congresos

- L. Iacono, P. Godoy, O. Marianetti y C. García Garino. Estudio de Plataformas de Hardware Empleadas en Redes de Sensores Inalámbricas. En V Workshop Arquitectura, Redes y Sistemas Operativos (WARSO), XVI Congreso Argentino de Ciencias de la Computación (XVI CACIC) 2010. Buenos Aires. Argentina. 2010. Editores: J. Ieroche y H. Padovani. ISBN: 978- 950-9474-49-9.
- P. Godoy, L. Iacono, O. Marianetti y C. García Garino. 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 en-

lace. En Segundo Congreso Virtual Microcontroladores y sus aplicaciones. Paraná. Argentina. 2010. Editores: G. Maggiolo y M. Caballero. ISBN: 978-987-25855-5- 6.

- A. Pra Baldi, P. Godoy, L. Iacono, O. Marianetti y C. García Garino. Utilización de herramientas Freescale de desarrollo de aplicaciones Zigbee sobre entornos GNU/Linux en WSN. En Sección Posters, VI Encuentro de Investigadores y Docentes de Ingeniería, (EnIDI) 2011. San Rafael. Mendoza. Argentina. 2011. ISBN: 978-950-42-0134-2.
- L. Iacono, P. Godoy, R. Cayssials, O. Marianetti and C. García Garino. Programmable Logic Devices in Sensor Networks: A Survey. En Argentine Symposium of Technology (AST), 40 Jornadas Argentina de Informática (40 JAIIO) 2011. Córdoba. Argentina. 2011. Editores: A. Ruiz de Mendarozqueta, M. Marciszack y M. Groppo. ISSN: 1850-2806.
- P. Godoy, L. Iacono, R. Cayssials, C. Párraga and C. García Garino. Effect of working conditions over the performance in ZigBee WSN. En Capítulo Sistemas de Control, Anales de la primera reedición del Congreso de la Sección Argentina del IEEE (Argencon 2012). Córdoba. Argentina. 2012. Editor: J. Sarmiento. ISBN: 987-572-076-3.
- P. Godoy, L. Iacono, R. Cayssials, C. García Garino. A Survey of WSN Testbeds Deployment. En Libro de Trabajos, Artículos de Protocolos y Comunicaciones, Congreso Argentino de Sistemas Embebidos (CASE) 2012. Buenos Aires. Argentina. 2012. pp. 243-248. Editor: D. J. Brengi. ISBN: 978-987-9374-82-5.
- P. Godoy, L. Iacono, R. Cayssials y C. García Garino. Simulación de Redes de Sensores Inalámbricos Mediante Autómatas Celulares. En VII Workshop Arquitectura, Redes y Sistemas Operativos (WARSO), XVIII Congreso Argentino de Ciencias de la Computación (XVIII CACIC) 2012. Bahía Blanca, Buenos Aires. Argentina. 2012. Editores: D. Urribarri, M. L. Ganuza, M. Larrea, D. Flores Choque y N. Gazcón. ISBN 978-987-1648-34-4.
- L. Iacono, C. García Garino, Osvaldo Marianetti and Cristina Párraga. Wireless Sensor Networks: A Software as a Service Approach. En VI Latin American Symposium on High Performance Computing (VI HPCLatAm) 2013. Mendoza. Argentina. 2013. pp. 243-248. Editores: Carlos García Garino y Marcela Printista.

Reportes de Investigación

- L. Iacono, Acceso Remoto a Redes de Sensores Inalámbricas Mediante Tecnologías de Computación Distribuida, Examen de Capacidad Investigadora. Facultad de Ingeniería, Universidad de Mendoza. Mendoza. Argentina. 2013. (Requisito cumplido del Doctorado en Ingeniería, Universidad de Mendoza).

Bibliografía

- Freescale Semiconductor. MC9S08GT32 Data Sheet, 2004. Disponible en: www.freescale.com/.
- A. Z. Abbasi, N. Islam, Z. A. Shaikh, et al. A review of wireless sensors and networks' applications in agriculture. *Computer Standards & Interfaces*, 36(2):263–270, 2014.
- K. Aberer, M. Hauswirth, and A. Salehi. A Middleware for Fast and Flexible Sensor Network Deployment. In *Proceedings of the 32nd International Conference on Very large Data Bases*, pages 1199–1202. VLDB Endowment, 2006.
- K. Ahmed and M. Gregory. Integrating Wireless Sensor Networks with Cloud Computing. In *Seventh International Conference on Mobile Ad-hoc and Sensor Networks (2011 MSN)*, pages 364–366. IEEE, 2011.
- I. Akyildiz and M. C. Vuran. *Wireless Sensor Networks*. John Wiley & Sons, Inc., New York, NY, USA, 2010. ISBN 047003601X, 9780470036013.
- I. Akyildiz, T. Melodia, and K. Chowdhury. A Survey on Wireless Multimedia Sensor Networks. *Computer Networks*, 51(4):921–960, 2007.
- B. Akyol, H. Kirkham, S. Clements, and M. Hadley. A Survey of Wireless Communications for the Electric Power System. *Prepared for the US Department of Energy under Contract DE-AC05-76RL01830*, Enero 2010.
- A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain. A Survey on Sensor-Cloud: Architecture, Applications, and Approaches. *International Journal of Distributed Sensor Networks*, 2013, 2013.
- B. Allen, J. Bresnahan, L. Childers, I. Foster, G. Kandaswamy, R. Kettimuthu, J. Kordas, M. Link, S. Martin, and K. Pickett. Software as a Service for Data Scientists. *Communications of the ACM*, 55(2):81–88, 2012.

- C. C. Allen. A Simplified Equation for Minimum Temperature Prediction. *Monthly Weather Review*, 85:119–120, 1957.
- G. Aloisio, D. Conte, C. Elefante, G. Marra, G. Mastrantonio, and G. Quarta. Globus Monitoring and Discovery Service and SensorML for Grid Sensor Networks. In *15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises 2006 (WETICE'06)*, pages 201–206. IEEE, 2006.
- S. Alsairafi, F. Emmanouil, M. Ghanem, N. Giannadakis, Y. Guo, D. Kalaitzopoulos, M. Osmond, A. Rowe, J. Syed, and P. Wendel. The Design of Discovery Net: Towards Open Grid Services for Knowledge Discovery. *International Journal of High Performance Computing Applications*, 17(3):297–315, 2003.
- Amazon EC2. Amazon Web Services, November 2012. Available in: <http://aws.amazon.com/es/ec2/>.
- D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. SETI@ home: An Experiment in Public-Resource Computing. *Communications of the ACM*, 45(11):56–61, 2002.
- S. Annamraju, S. Gumudavelli, R. Wang, and D. Gurkan. Middleware Design for Measurement Data Exchange in a Smart Sensor Network. In *2010 Instrumentation and Measurement Technology Conference (I2MTC)*, pages 204–207. IEEE, 2010.
- M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A View of Cloud Computing. *Communications of the ACM*, 53(4):50–58, 2010.
- P. Baronti, P. Pillai, V. Chook, S. Chessa, A. Gotta, and Y. Hu. 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, 2007.
- G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli. Wireless sensor networks for environmental monitoring: The sensorscope experience. In *2008 IEEE International Zurich Seminar on Communications*, pages 98–101. IEEE, 2008.
- R. Beckwith, D. Teibel, and P. Bowen. Report from the field: results from an agricultural wireless sensor network. In *29th Annual IEEE International Conference on Local Computer Networks, 2004*, pages 471–478. IEEE, 2004.
- J. Bernat Vercher, S. Perez Marin, A. Gonzalez Lucas, R. Sorribas Mollon, L. Villarrubia Grande, L. Campoy Cervera, and L. Hernández Gómez. Ubiquitous Sensor Networks in IMS: an Ambient Intelligence Telco Platform. In P. Cunningham and M. Cunningham, editors, *Proceedings of the ICT-MobileSummit 2008 Conference*. IIMC International Information Management Corporation, 2008. ISBN: 978-1-905824-08-3.

- D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard. Web Services Architecture. *W3C Working Group Note 11 February 2004*, page 13, 2004. Disponible en: <http://www.w3.org/TR/ws-arch/>.
- C. Bormann, G. Mulligan, J. Arkko, M. Townsley, and C. Schumacher. *IPv6 over Low power WPAN (6lowpan)*. IETF Working Group, 2006. DAvailable in: <http://datatracker.ietf.org/wg/6lowpan/charter/>.
- M. Botts and A. Robin. OpenGIS Sensor Model Language (SensorML) Implementation Specification, 2007. Disponible en: <http://www.opengeospatial.org/standards/sensorml>.
- M. Botts, G. Percivall, C. Reed, and J. Davidson. OGC Sensor Web Enablement: Overview and High Level Architecture. *GeoSensor Networks*, pages 175–190, 2008.
- T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (XML). *World Wide Web Journal*, 2(4):27–66, 1997.
- R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic. 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, 2009.
- C. Catania, C. Careglio, D. Monge, P. Martínez, A. Mirasso, and C. García Garino. Estudios Paramétricos de Mecánica de Sólidos en Entornos de Computación Distribuida. *Mecánica Computacional*, 27:1063–1084, 2008.
- K. Chang, N. Yau, M. Hansen, and D. Estrin. Sensorbase.org-a centralized repository to slog sensor network data. *Center for Embedded Network Sensing*, 2006.
- D. Cook and S. Das. *Smart Environments: Technology, Protocols and Applications*, volume 43. John Wiley & Sons, 2004.
- J. Corchado, J. Bajo, D. Tapia, and A. Abraham. Using Heterogeneous Wireless Sensor Networks in a Telemonitoring System for Healthcare. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):234–240, Marzo 2010. ISSN 1089-7771. doi: 10.1109/TITB.2009.2034369.
- K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke, W. Vambenepe, and S. Weerawarana. The WS-Resource Framework Version 1.0, Mayo 2004. Disponible en: <http://www.globus.org/wsrf/specs/ws-wsrf.pdf>.
- S. E. Díaz, J. C. Pérez, A. C. Mateos, M.-C. Marinescu, and B. B. Guerra. A novel methodology for the monitoring of the agricultural production process based on wireless sensor networks. *Computers and Electronics in Agriculture*, 76(2):252–265, 2011.

- Digi International Inc. XBee ZB Modules Data Sheet, 2012. Disponible en: www.digi.com/.
- Dirección de Agricultura y Contingencias Climáticas Mendoza. Heladas, April 2013. Available in: <http://www.contingencias.mendoza.gov.ar/pdf/heladas.pdf>.
- L. Doolittle and J. Nelson. Boa Webserver, Febrero 2000. Disponible en: <http://www.boa.org/documentation/>.
- A. Dunkels. Full TCP/IP for 8-bit Architectures. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pages 85–98. ACM, 2003.
- A. Dunkels, J. Alonso, and T. Voigt. Making TCP/IP Viable for Wireless Sensor Networks. *Swedish Institute of Computer Science (SICS) Research Report*, 2003.
- A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. In *29th Annual IEEE International Conference on Local Computer Networks, 2004.*, pages 455–462, nov. 2004. doi: 10.1109/LCN.2004.38.
- F. Eady. *Hands-on Zigbee: Implementing 802.15. 4 with microcontrollers*. Newnes, 2007.
- S. Ergen. ZigBee/IEEE 802.15. 4 Summary, 2004. Available in: <http://pages.cs.wisc.edu/>.
- I. Foster. What is the grid? a three point checklist. *GRID today*, 1(6), 2002.
- I. Foster. Globus Online: Accelerating and Democratizing Science Through Cloud-Based Services. *Internet Computing, IEEE*, 15(3):70–73, 2011.
- I. Foster and K. Kesselman. History of the Grid. In I. Foster, W. Gentsch, L. Grandinetti, and G. Joubert, editors, *High Performance Computing: From Grids and Clouds to Exascale*, volume 20 of *Advances in Parallel Computing*, pages 3–30. IOS Press, 2011. ISBN 978-1-60750-802-1. doi: 10.3233/978-1-60750-803-8-94.
- I. Foster and C. Kesselman. The Globus Toolkit. *The grid: Blueprint for a new computing infrastructure*, pages 259–278, 1999.
- I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, 15(3):200, 2001.
- I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. In *Grid Computing*, pages 217–249. Wiley Online Library, 2003.
- I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud Computing and Grid Computing 360-Degree Compared. In *Grid Computing Environments Workshop 2008 (GCE'08)*, pages 1–10. IEEE, 2008.

- I. Foster, S. Tuecke, T. Hauser, R. Kettimuthu, D. Milroy, S. Martin, B. Palen, and J. Braden. Campus Bridging Made Easy via Globus Services. In *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the campus and beyond*, page 50. ACM, 2012.
- Freescale Semiconductor. *MC13192/MC13193 2.4 GHz low power transceiver for the IEEE 802.15. 4 Standar*, 2008. Disponible en: www.freescale.com/.
- Freescale Semiconductor. *Hoja de Datos MC13213*, 2009. Disponible en: www.freescale.com/.
- M. Ghanem, Y. Guo, J. Hassard, M. Osmond, and M. Richards. Sensor Grids for Air Pollution Monitoring. In *Proceedings of the 3rd UK e-Science All Hands Meeting*, Nottingham, UK, 2004.
- B. Gholamzadeh and H. Nabovati. Concepts for Designing Low Power Wireless Sensor Network. *World Academy of Science, Engineering and Technology*, 45, 2008.
- Globus. Globus Plus, July 2014. Available in: <https://www.globus.org/researchers/plus-plans>.
- A. Gluhak and W. Schott. A WSN System Architecture to Capture Context Information for Beyond 3G Communication Systems. In *3rd International Conference on Intelligent Sensors, Sensor Networks and Information, 2007. (ISSNIP 2007)*, pages 49–54. IEEE, 2007.
- P. Godoy, L. Iacono, O. Marianetti, and C. García Garino. 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. In G. Maggiolo and M. Caballero, editors, *Segundo Congreso Virtual Microcontroladores y sus aplicaciones*, Paraná, Argentina, 2010. ISBN: 978-950-42-0134-2.
- P. Godoy, L. Iacono, R. Cayssials, C. Párraga, and C. García Garino. Effect of working conditions over the performance in ZigBee WSN. In *Capítulo Sistemas de Control, Anales de la primera reedición del Congreso de la Sección Argentina del IEEE (Argencon 2012)*., Córdoba, Argentina, 2012. ISBN: 987-572-076-3.
- Google. Google App Engine, February 2013a. Available in: <https://developers.google.com/appengine/>.
- Google. Google Docs, February 2013b. Available in: <https://drive.google.com/>.
- P. Grace, G. Coulson, G. Blair, L. Mathy, W. Yeung, W. Cai, D. Duce, and C. Cooper. Grid-kit: Pluggable Overlay Networks for Grid Computing. *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 1463–1481, 2004.

- J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*, 29 (7):1645–1660, 2013.
- Hart Foundation. WirelessHART Communications Protocol. Disponible en: http://www.hartcomm.org/protocol/wihart/wireless_technology.html, 2011.
- M. Hirafuji, H. Yoichi, T. Kiura, K. Matsumoto, T. Fukatsu, Tanaka, and Others. Creating High-performance/Low-cost Ambient Sensor Cloud System using OpenFS (Open Field Server) for High-throughput Phenotyping. In *Proceedings of 2011 SICE Annual Conference (2011 SICE)*, pages 2090–2092. IEEE, 2011.
- N. Hoeller, C. Reinke, J. Neumann, S. Groppe, D. Boeckmann, and V. Linnemann. Efficient XML Usage Within Wireless Sensor Networks. In *Proceedings of the 4th Annual International Conference on Wireless Internet*, page 74. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering (ICST), 2008. ISBN: 978-963-9799-36-3.
- M. Hori and T. Kawashima, E. and Yamazaki. Application of cloud computing to agriculture and prospects in other fields. *Fujitsu Scientific & Technical Journal*, 46(4):446–454, 2010.
- G. Huang. Casting the Wireless Sensor Net. *Technology Review Manchester NH*, 106(6): 50–57, 2003.
- D. Hughes, P. Greenwood, G. Blair, G. Coulson, P. Grace, F. Pappenberger, P. Smith, and K. Beven. An Experiment with Reflective Middleware to Support Grid-based Flood Monitoring. *Concurrency and Computation: Practice and Experience*, 20(11):1303–1316, 2008.
- G. Humber and E. Ngai. Quality-of-Information Aware Data Delivery for Wireless Sensor Networks: Description and Experiments. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pages 1–6. IEEE, 2010.
- J. Humble, C. Greenhalgh, A. Hamsphire, H. Muller, and S. Egglestone. A Generic Architecture for Sensor Data Integration with the GRID. *Scientific Applications of Grid Computing*, pages 99–107, 2005.
- N. Höller. *Efficient XML Data Management and Query Evaluation in Wireless Sensor Networks*. PhD thesis, Universität zu Lübeck, Lübeck, 2010.
- L. Iacono, P. Godoy, O. Marianetti, and C. García Garino. Estudio de plataformas de hardware empleadas en redes de sensores inalámbricas. In J. Ieroche and H. Padovani, editors, *V Workshop Arquitectura, Redes y Sistemas Operativos (WARSO), Congreso Argentino*

- de Ciencias de la Computación (CACIC) 2010*, pages 784–793, Buenos Aires, Argentina, 2010. ISBN: 978-950-9474-49-9.
- L. Iacono, P. Godoy, O. Marianetti, C. García Garino, and C. Párraga. Estudio de la Integración entre WSN y redes TCP/IP. *Memoria de Trabajos de Difusión Científica y Técnica*, 10, 2012. ISSN 1510-7450.
- L. Iacono, C. García Garino, O. Marianetti, and C. Párraga. Wireless Sensor Networks: A Software as a Service Approach. In C. García Garino and M. Printista, editors, *Prospective and Ongoing Projects, VI Latin American Symposium on High Performance Computing (HPCLatAm 2013)*, pages 184–195, Mendoza, Argentina, 2013.
- L. Iacono, J. L. Vázquez-Poletti, C. García Garino, and I. M. Llorente. A Model to Calculate Amazon EC2 Instance Performance in Frost Prediction Applications. In *High Performance Computing*, pages 68–82. Springer, 2014.
- IEEE. IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pages 1–305, 2006.
- International Society of Automation (ISA). ANSI/ISA-100.11a-2011 Wireless systems for industrial automation: Process control and related applications, 2009. Disponible en: <http://www.isa.org/>.
- M. Islam, M. Hassan, and E. Huh. Sensor Proxy Mobile IPv6 (SPMIPv6)-A Framework of Mobility Supported IP-WSN. In *13th International Conference on Computer and Information Technology (ICCIT) 2010*, pages 295–299. IEEE, 2010.
- H. Jin. ChinaGrid: Making grid computing a reality. *Digital Libraries: International Collaboration and Cross-Fertilization*, pages 13–24, 2005.
- H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. Wiley-Interscience, 2007.
- P. Kinney. Zigbee Technology: Wireless Control that Simply Works. In *Communications design conference*, volume 2, 2003.
- N. Kushalnagar, G. Montenegro, and C. Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals, 2005. Available in: <http://www.hjp.at/doc/rfc/rfc4919.html>.

- M. Lamanna. The LHC computing grid project at CERN. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 534(1):1–6, 2004.
- J. Lee, T. Chung, and S. Gundavelli. A Comparative Signaling Cost Analysis of Hierarchical Mobile IPv6 and Proxy Mobile IPv6. In *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2008)*, pages 1–6. IEEE, 2008.
- K. Lee. IEEE 1451: A Standard in Support of Smart Transducer Networking. In *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference (IMTC2000)*, volume 2, pages 525–528. IEEE, 2000.
- K. Lee and D. Hughes. System Architecture Directions for Tangible Cloud Computing. In *International Workshop on Information Security and Applications (IWISA 2010)*, volume 25, Qinhuangdao, China, Octubre 2010.
- K. Lee, D. Murray, D. Hughes, and W. Joosen. Extending Sensor Networks Into the Cloud Using Amazon Web Services. In *2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications (NESEA)*, pages 1–7. IEEE, 2010.
- Z. Li, T. Hong, N. Wang, and T. Wen. Data Transmission Performance for 2.4 GHz In-field Wireless Sensor Network. In *2010 2nd International Conference on Computer Engineering and Technology (ICCET)*, volume 1, pages 465–469. IEEE, 2010.
- H. Lim, K. Ling, W. Wang, Y. Yao, M. Iqbal, B. Li, X. Yin, and T. Sharma. The National Weather Sensor Grid. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 369–370. ACM, 2007.
- A. Liutkevicius, A. Vrubliauskas, and E. Kazanavicius. A Survey of Wireless Sensor Network Interconnection to External Networks. *Novel Algorithms and Techniques in Telecommunications and Networking*, pages 41–46, 2010.
- X. Ma and W. Luo. The Analysis of 6LoWPAN Technology. In *Pacific-Asia Workshop on Computational Intelligence and Industrial Application 2008 (PACIIA'08)*, volume 1, pages 963–966. IEEE.
- S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks. *ACM SIGOPS Operating Systems Review*, 36(SI):131–146, 2002.
- Microsoft. Windows Azure, Febrero 2013. Disponible en: <http://www.windowsazure.com/es-es/>.
- Mobitrum Corporation. *MobeeNet Data Sheet*, 2010. Disponible en: www.mobitrum.com/MOBEE-NET-Portal.pdf.

- R. Morais, M. Fernandes, S. Matos, C. Serōdio, P. Ferreira, and M. Reis. A ZigBee Multi-powered Wireless Acquisition Device for Remote Sensing Applications in Precision Viticulture. *Computers and Electronics in Agriculture*, 62(2):94–106, 2008.
- E. Nadimi, H. Sogaard, T. Bak, and F. W. Oudshoorn. Zigbee-based wireless sensor networks for monitoring animal presence and pasture time in a strip of new grass. *Computers and electronics in agriculture*, 61(2):79–87, 2008.
- Hoja de Datos NI WSN-3202*. Disponible en: <http://www.ni.com/>. National Instruments, 2015.
- B. Ograph and Y. Morgens. Cloud Computing. *Communications of the ACM*, 51(7), 2008.
- Oracle Microsystems. Sun Spot Owners Manual, Diciembre 2012. Disponible en: <http://www.sunspotworld.com/>.
- T. O'Reilly, K. Headley, J. Graybeal, K. J. Gomes, D. R. Edgington, K. A. Salamy, D. Davis, and A. Chase. MBARI technology for self-configuring interoperable ocean observatories. In *OCEANS 2006*, pages 1–6. IEEE.
- T. O'Reilly, K. Headley, D. Edgington, C. Rueda, K. Lee, E. Song, J. Zedlitz, J. del Rio, D. Toma, A. Manuel, et al. Instrument Interface Standards for Interoperable Ocean Sensor Networks. In *OCEANS 2009-EUROPE*, pages 1–10. IEEE, 2009.
- E. Pacini, M. Ribero, C. Mateos, A. Mirasso, and C. García Garino. Cloud Computing for Parameter Sweep Experiments. In F. Ficarra, K. Veltman, J. Alma, M. Brie, M. Ficarra, J. Carre, and A. Kratky, editors, *New Horizons in Creative Open Software, Multimedia, Human Factors and Software Engineering*, chapter 9, pages 101–118. Blue Herons, Bergamo, Italia, Abril 2012. ISBN: 978-88-96471-01-2.
- F. Pierce and T. Elliott. Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington. *Computers and electronics in agriculture*, 61(1):32–43, 2008.
- J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling Ultra-low Power Wireless Research. In *Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pages 364–369. IEEE, 2005.
- N. Preve. Segedma: Sensor grid enhancement data management system for health care computing. *Expert Systems with Applications*, 38(3):2371–2380, 2011.
- V. Rajesh, J. Gnanasekar, R. Ponmagal, and P. Anbalagan. Integration of Wireless Sensor Network with Cloud. In *International Conference on Recent Trends in Information, Telecommunication and Computing (ITC) 2010*, pages 321–323. IEEE, 2010.

- S. Reddy, G. Chen, B. Fulkerson, S. Kim, U. Park, N. Yau, J. Cho, M. Hansen, and J. Heidemann. Sensor-Internet Share and Search: Enabling Collaboration of Citizen Scientists. In *Proceedings of the ACM Workshop on Data Sharing and Interoperability on the World-wide Sensor Web*, pages 11–16, 2007.
- D. Robinson and K. Coar. The Common Gateway Interface (CGI) Version 1.1, Octubre 2004. Disponible en: <http://tools.ietf.org/html/rfc3875>.
- R. Roman and J. Lopez. Integrating Wireless Sensor Networks and the Internet: a Security Analysis. *Internet Research*, 19(2):246–259, 2009.
- RSS Advisory Board. RSS 2.0 Specification, 2005. Disponible en: <http://www.rssboard.org/rss-2-0-1-rv-6>.
- Sensirion SHT15 Data Sheet. Available in: www.sensirion.com.* Sensirion, 2011.
- Sensirion SHT71 Data Sheet. Available in: www.sensirion.com.* Sensirion, 2014.
- Z. Shelby and C. Bormann. *6LoWPAN: the wireless embedded internet*, volume 33. Wiley, 2010.
- R. Silva, S. Silva, and F. Boavida. A Proxy-based Mobility Solution for Critical WSN Applications. In *2010 IEEE International Conference on Communications Workshops (ICC)*, pages 1–5. IEEE, 2010.
- Microchip PIC16F873 Data Sheet. Available in: www.microchip.com.* SMicrochip, 2014.
- R. L. Snyder and J. P. Melo-Abreu. *Frost protection: Fundamentals, Practice and Economics*, volume 1. Food and Agriculture Organization of the United Nations (FAO), 2005.
- E. Song and K. Lee. Smart transducer web services based on the iee 1451.0 standard. In *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pages 1–6. Ieee, 2007.
- E. Song and K. Lee. Stws: A unified web service for iee 1451 smart transducers. *Instrumentation and Measurement, IEEE Transactions on*, 57(8):1749–1756, 2008.
- E. Song and K. Lee. Integration of iee 1451 smart transducers and ogc-swe using stws. In *Sensors Applications Symposium, 2009. SAS 2009. IEEE*, pages 298–303. IEEE, 2009.
- Arduino Pro. Available in: https://www.sparkfun.com/.* Spark Fun, 2014. URL <https://www.sparkfun.com/products/10915>.
- Hoja de Datos CC2430. Disponible en: http://www.ti.com/product/cc2430.* Texas Instruments, 2007.

- Hoja de Datos CC2480. Disponible en: <http://www.ti.com/product/cc2480a1>. Texas Instruments, 2008.*
- Hoja de Datos LM75. Disponible en: <http://www.ti.com/product/lm75a>. Texas Instruments, 2014.*
- Hoja de Datos MAX232. Disponible en: <http://www.ti.com/>. Texas Instruments, 2015.*
- R. Van Engelen and K. Gallivan. The gSOAP Toolkit for Web Services and Peer-to-Peer Computing Networks. In *2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 128–128. IEEE, 2002.
- Vishay Semiconductors BPW41N Data Sheet. Available in: www.vishay.com. Vishay Semiconductors, 2014.*
- Y. YuJie, W. Shu, and H. Zhao. MPAS: a Connecting Platform for Integrating Wireless Sensor Network with Grid. In *Asia-Pacific Conference on Communications 2005*, pages 1000–1004. IEEE, 2005.
- M. Yuriyama, T. Kushida, and M. Itakura. A New Model of Accelerating Service Innovation with Sensor-Cloud Infrastructure. In *2011 Annual SRII Global Conference (SRII)*, pages 308–314, 29 2011–april 2 2011. doi: 10.1109/SRII.2011.42.
- P. Zhang, M. Chen, and P. He. The Study of Interfacing Wireless Sensor Networks to Grid Computing Based on Web Service. In *2010 Second International Workshop on Education Technology and Computer Science*, pages 439–442. IEEE, 2010.
- X. Zhang. Model Design of Wireless Sensor Network Based on Scale-Free Network Theory. In *5th International Conference on Wireless Communications, Networking and Mobile Computing, 2009 (WiCom'09)*, pages 1–4. IEEE, 2009.
- J. Zheng and A. Jamalipour. *Wireless Sensor Networks: a Networking Perspective*. Wiley-IEEE Press, 2009.
- ZigBee Alliance. ZigBee specification. *ZigBee Document 053474r13*, pages 344–346, 2006.
- M. Zuñiga and B. Krishnamachari. Integrating Future Large-Scale Wireless Sensor Networks with the Internet. USC Computer Science Technical Report CS 03-792, 2003. Available in: <ftp://ftp.usc.edu/pub/csinfo/tech-reports/papers/03-792.pdf>.