UNIVERSIDAD NACIONAL DE COLOMBIA

MÉTODO PARA LA EVALUACIÓN DEL DESEMPEÑO DE LA INTEGRACIÓN DE AGENTES INTELIGENTES & WSN A TRAVÉS DE MÉTRICAS

Tatiana Cristina Muñoz Hernández

Universidad Nacional de Colombia
Facultad de Minas
Departamento de Ciencias de la Computación y de la Decisión
Medellín, Colombia
2012

MÉTODO PARA LA EVALUACIÓN DEL DESEMPEÑO DE LA INTEGRACIÓN DE AGENTES INTELIGENTES & WSN A TRAVÉS DE MÉTRICAS

Tatiana Cristina Muñoz Hernández

Tesis de maestría presentada como requisito parcial para optar al título de:

Magíster en Ingeniería - Ingeniería de Sistemas

Director
Demetrio Arturo Ovalle, Ph.D

Co-Director
Prof. Alcides de Jesús Montoya, Ph.D

Línea de Investigación:
Inteligencia Artificial

Grupos de Investigación:
GIDIA: Grupo de Investigación y Desarrollo en Inteligencia Artificial
GICEI: Grupo de Instrumentación Científica e Industrial

Universidad Nacional de Colombia
Facultad de Minas
Departamento de Ciencias de la Computación y de la Decisión
Medellín, Colombia
2012

Dedicado a:

Luz Ángela Hernández Calderón
Ruth Edith Calderón Cárdenas
María Camila García Hernández
Liliana Hernández Calderón
Mauricio Hernández Calderón
Guillermo Hernández Calderón

Porque definitivamente sin su ayuda y comprensión esto no hubiese sido posible.

Agradecimientos

La autora expresa sus agradecimientos a:

Demetrio Arturo Ovalle Carranza, PhD., profesor titular de la Universidad Nacional de Colombia – Sede Medellín, director de la tesis; por su confianza, paciencia, entrega y todas sus enseñanzas a lo largo de mi maestría. Con su valioso apoyo fue posible culminar este trabajo con éxito.

Alcides de Jesús Montoya Cañola, MSc., profesor de la Universidad Nacional de Colombia – Sede Medellín, co-director de la tesis; por su colaboración.

Diego Luis Aristizábal Ramírez, MSc., profesor de la Universidad Nacional de Colombia – Sede Medellín, Facultad de Ciencias; por su valioso apoyo y porque en él encontré además de un gran maestro un valioso amigo.

Gonzalo Vargas, profesor de la Universidad Nacional de Colombia – Sede Medellín, Facultad de Ciencias; por sus valiosos aportes.

A todos mis compañeros y amigos, con los cuales emprendí este nuevo reto, supere obstáculos y compartí nuevos logros. Especialmente a Alejandro Piedrahita por su valiosa colaboración.

Por último, a mi familia, los cuales siempre han estado presentes de manera incondicional apoyándome, guiándome y alentándome a no desfallecer sin importar la distancia. Especialmente le doy las gracias a mi mamá, Ángela Hernández, quien con su fortaleza me ha impulsado a seguir adelante y siempre me ha brindado su amor incondicional.

Resumen

En esta tesis de maestría se propone un método que permite, a través de la ejecución de un proceso organizado y detallado, realizar el seguimiento y la evaluación del desempeño a través de métricas de una Red de Sensores Inalámbricos o WSN (por su acrónimo en inglés Wireless Sensor Networks) dotada con agentes inteligentes de software y en general de cualquier WSN. Para realizar la selección de las métricas, se sugiere un listado de posibles criterios que podrían ser utilizados, los cuales se han categorizado desde 4 enfoques: energético, prioridad de los mensajes, adaptabilidad de la red y finalmente el número de nodos y de agentes requeridos. El método propuesto en esta tesis de maestría fue validado a través de casos de estudio con dispositivos reales y permite también desplegar estratégicamente una WSN a partir de la identificación de sus principales características. Mediante los diferentes casos de estudio se pudo concluir que cuando se requiere hacer frente a diferentes retos que se presentan en un montaje real, bajo condiciones muchas veces impredecibles, para las cuales la red no fue diseñada, el uso de técnicas de inteligencia artificial, específicamente de agentes inteligentes de software, es una solución promisoria, la cual hace posible mejorar el desempeño de una WSN a través de la modelación de agentes de software que se ajusten a las limitaciones de recursos de éstas sin perder su capacidad de deliberar. De lo contrario, el uso de agentes de software se hace innecesario.

Palabras clave: Redes de Sensores Inalámbricas (WSN), Agentes Inteligentes, Evaluación del Desempeño, Métricas.

Abstract

The aim of this Computer Science master's thesis is to propose a detailed and organized method that allows performing metric-based tracking and performance evaluation of WSN (Wireless Sensor Networks) in general and also when they are equipped with intelligent agents. In order to make the selection of the metrics a list of possible criteria that should be used is proposed by the method taken into consideration four viewpoints: energy consumption, message priority, network adaptability, and finally, the number of nodes and agents required. The method proposed was validated through several case studies with real devices and also strategically allows deploying a WSN using the identification and characterization of its main features. Based on different case studies it can be concluded that when it is required to deal with different challenges presented in a real network, involving unpredictable conditions for which the network wasn´t designed, the use of artificial intelligence techniques and in particular intelligent software agents is a promising solution. Such a solution makes it possible to improve the performance of a WSN through the modeling of software agents that fit the WSN's physical resource constraints without losing their ability to deliberate. Otherwise, the use of software agents becomes unnecessary.

Keywords: WSN (Wireless Sensor Networks), Intelligent agents, Performance evaluation, Metrics.

Contenido

RESUMEN **5**

LISTA DE FIGURAS **9**

LISTA DE TABLAS **10**

LISTA DE ECUACIONES **10**

CAPÍTULO 1 ..... **11**

INTRODUCCIÓN, OBJETIVOS Y METODOLOGÍA **11**

1.1 INTRODUCCIÓN 11
1.2 APORTES 13
1.3 DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN 13
1.4 HIPÓTESIS DE INVESTIGACIÓN 14
1.5 PREGUNTAS DE INVESTIGACIÓN 15
1.6 OBJETIVOS 15
1.6.1 Objetivo general 15
1.6.2 Objetivos específicos 15
1.7 ALCANCE 16
1.8 METODOLOGÍA 17
1.9 ORGANIZACIÓN DEL DOCUMENTO 18
1.10 DIFUSIÓN DE RESULTADOS 18

CAPÍTULO 2 ..... **20**

MARCO TEÓRICO **20**

2.1 INTELIGENCIA ARTIFICIAL E INTELIGENCIA ARTIFICIAL DISTRIBUIDA (IAD) 20
2.1.1 Inteligencia Artificial (IA) 20
2.1.1.1 Agente 20
2.1.1.2 Arquitecturas 21
2.1.2 Inteligencia Artificial Distribuida (IAD) 22
2.1.3 Sistema Multi-Agente (SMA) 23
2.1.4 Metodologías para el desarrollo de SMA 24
2.1.5 Sistemas basados en agentes vs Sistemas Multi-Agente 24
2.1.6 Frameworks 25
2.1.6.1 Frameworks de agentes 25
2.1.6.1.1 Agent Factory (AF) 25
2.1.6.1.2 JADE 25
2.1.6.2 Frameworks para el desarrollo de agentes para sistemas de recursos limitados 26
2.1.6.2.1 LEAP 26
2.1.6.2.2 Agilla 27
2.1.6.2.3 AFME 28
2.2 REDES DE SENSORES INALÁMBRICOS 28
2.2.1 Topología de red 31
2.2.2 Motes 33
2.3 SISTEMAS OPERATIVOS PARA WSN 38

2.3.1 TinyOS 38
2.3.2 Contiki 39
2.3.3 Mantis 40
2.4 MÉTRICAS PARA LA EVALUACIÓN DEL DESEMPEÑO DE UNA WSN 40

CAPÍTULO 3 44

ESTADO DEL ARTE 44

3.1 REVISIÓN DE TRABAJOS RELACIONADOS 44
3.2 MODELOS DE INTEGRACIÓN DE AGENTES INTELIGENTES CON WSN 47
3.3 FACTORES MÁS SIGNIFICATIVOS PARA LA EVALUACIÓN DEL DESEMPEÑO DE UNA WSN IDENTIFICADOS A PARTIR DE LA REVISIÓN DEL ESTADO DEL ARTE 49

CAPÍTULO 4 51

MÉTODO PROPUESTO BASADO EN MÉTRICAS PARA REALIZAR EL SEGUIMIENTO Y LA EVALUACIÓN DE LA INTEGRACIÓN DE AGENTES INTELIGENTES EN WSN 51

4.1 ETAPA 1: CONCEPTUALIZACIÓN 52
4.1.1 Descripción de la aplicación 52
4.1.2 Características de la red 52
4.1.3 Identificación de actores 53
4.1.4 Agentes de software 54
4.2 ETAPA 2: ANÁLISIS Y DISEÑO 54
4.2.1 Selección de la plataforma hardware 55
4.2.2 Herramientas utilizadas para llevar a cabo la implementación 55
4.2.3 Selección de Métricas 55
4.3 ETAPA 3: IMPLEMENTACIÓN 58
4.4 ETAPA 4: EVALUACIÓN DE RESULTADOS 59

CAPÍTULO 5 60

VALIDACIÓN DEL MÉTODO PROPUESTO A TRAVÉS DE CASOS DE ESTUDIO 60

5.1 CASO DE ESTUDIO 1: MEDICIÓN DE POTENCIA DE UN NODO SENSOR 60
5.2 CASO DE ESTUDIO 2: MEDICIÓN DEL TIEMPO DE VIDA DE UNA WSN CON AGENTES DE SOFTWARE 68
5.3 CASO DE ESTUDIO 3: MEDICIÓN DEL TIEMPO DE VIDA DE UNA WSN SIN AGENTES DE SOFTWARE 84

CAPÍTULO 6 90

CONCLUSIONES Y TRABAJO FUTURO 90

REFERENCIAS BIBLIOGRÁFICAS 93

Lista de Figuras

Figura 1. Esquema general de la interacción del agente con el ambiente (tomada de (Vlassis, 2007)) 21
Figura 2. Esquema básico de una WSN (3 motes unidos a un PC a través de una estación base). Las líneas punteadas representan una conexión inalámbrica desde los 3 nodos a la estación base y la línea sólida es una conexión cableada de la estación base al PC 29
Figura 3. Topología de red tipo estrella 32
Figura 4. Topología de red tipo malla 32
Figura 5. Topología de red híbrida 33
Figura 6. Arquitectura de un nodo sensor genérico 34
Figura 7. Notación grafica de los casos de uso (Imagen tomada de (Iglesias Fernández, 1998)) 54
Figura 8. Tmote Sky (Sentilla o formalmente Moteiv) 62
Figura 9. Esquema de conexión para la medición de la potencia consumida por un nodo Tmote Sky. 66
Figura 10. Programa de adquisición de datos realizado en LabView 66
Figura 11. Potencia vs tiempo (Collect View - Instant Contiki) 67
Figura 12. Potencia vs Tiempo (LabView) 68
Figura 13. Esquema de la topología punto a punto a utilizar. Las líneas punteadas representan una conexión inalámbrica desde el nodos a la estación base y la línea sólida es una conexión cableada de la estación base al PC. 69
Figura 14. Esquema de actividades actor del Usuario o Administrador de red (WSN con agentes) 71
Figura 15. Esquema de actividades del actor agente netamente reactivo 71
Figura 16. Esquema de actividades actor del agente reactivo con mayores capacidades 71
Figura 17. MTS310CA 73
Figura 18. Mica2 73
Figura 19. Estación base: MIB510 con nodo Mica2 74
Figura 20. Diagrama de flujo agente1 79
Figura 21. Diagrama de flujo agente2 81
Figura 22. Resultados TL del nodo con el agente1 (Gráfica V vs t) 82
Figura 23. Resultados TL del nodo con el agente2 (Gráfica V vs t) 83
Figura 24. Esquema de actividades del actor Usuario o Administrador de red (WSN sin agentes) 85
Figura 25. Resultados T_L del nodo con la aplicación modificada Oscilloscope (Gráfica V vs t) 88

Lista de Tablas

Tabla 1. Comparación de diversos motes contra su rango direccionable, capacidad de cómputo y capacidad memoria 38
Tabla 2. Sistemas operativos para WSN (Sohraby, et al., 2007) (Lajara , et al., 2010) 38
Tabla 3. Plantilla propuesta para realizar la descripción textual de un caso de uso (Iglesias Fernández, 1998) 53
Tabla 4. Condiciones de operación típicas de un Tmote Sky 63
Tabla 5. Condiciones de operación típicas de un Mica2 74
Tabla 6. Evaluación métrica Tiempo de respuesta 84
Tabla 7. Evaluación métrica Completitud 84
Tabla 8. Resumen de características de algunos nodos sensores 92

Lista de Ecuaciones

Ecuación 1. Potencia recibida por la antena receptora 35
Ecuación 2. Nivel de pérdidas de potencia por trayecto 36
Ecuación 3. Nivel de pérdidas de potencia por trayecto en función de la longitud de onda y la distancia entre T_x y R_x 36
Ecuación 4.Nivel de pérdidas de potencia por trayecto para $\lambda \approx 0{,}34$ m 37
Ecuación 5. Potencia recibida por R_x para una distancia d fija 37
Ecuación 6. Potencia consumida en el mote 65
Ecuación 7. Tiempo de vida de la red 77

Capítulo 1

Introducción, Objetivos y Metodología

Este primer capítulo tiene como objetivo presentar de forma general el contenido de esta tesis de maestría, mostrar los conceptos y enfoques donde se enmarca su desarrollo, tales como son las Redes de Sensores Inalámbricas o Wireless Sensor Networks (WSN), los Sistemas Multi-Agente y la evaluación del desempeño a través de métricas. Además, se presentan los aportes, los objetivos planteados, el alcance, la metodología de trabajo y la difusión de resultados de la misma.

1.1 Introducción

En la actualidad, las tecnologías inalámbricas (Wireless Technologies) se están imponiendo sobre las tecnologías alámbricas, debido al alto costo de los cables y de la mano de obra de estas últimas. Los dispositivos de medida inalámbricos ofrecen la misma calidad y precisión que los sistemas tradicionales de medida cableados pero con mayor flexibilidad y menores costos, permitiendo resolver una amplia variedad de retos donde una solución cableada puede ser poco práctica o no factible (National Instruments, 2009).

Aunque el concepto de una red de sensores no es nuevo (DARPA - Defense Advanced Research Projects Agency- inició el programa Distributed Sensor Networks en 1980), los avances recientes en la fabricación de microprocesadores, de los sistemas microelectromecánicos y de las tecnologías para las comunicaciones inalámbricas, han conducido a una reducción dramática tanto en el tamaño como en la energía consumida por los dispositivos que componen la red, permitiendo desplegarlos rápidamente en un ambiente de interés donde éstos son capaces de realizar tareas de detección y de procesamiento complejas.

Las Redes de Sensores Inalámbricas o Wireless Sensor Networks (WSN), son un área emergente de los sistemas embebidos que tienen el potencial de revolucionar nuestras vidas con aplicaciones de gran escala, incluyendo monitoreo y conservación ambiental, control industrial, administración de negocios, monitoreo estructural y sísmico, transporte, salud y domótica (Zhao & Guibas, 2004). A diferencia de los sistemas de red tradicionales, una WSN está limitada por el consumo de batería, el ancho de banda de la red, la confiabilidad en la transmisión de datos, la tolerancia a fallas, la escalabilidad, limitaciones de hardware (procesamiento y memoria), entre otros. Además, éstas se encuentran más ligadas a la localización geográfica y al ambiente físico que los sistemas centralizados (Cheong, 2007). Estas redes inalámbricas constan de sensores autónomos distribuidos espacialmente (Motes), los cuales se comunican entre ellos a través de transmisiones inalámbricas y actúan de manera conjunta con el fin de monitorear o controlar distintos entornos físicos.

Investigaciones actuales incluyen el uso de técnicas de Inteligencia Artificial Distribuida (IAD), específicamente de agentes inteligentes de software, para hacer frente a los desafíos y limitaciones que éstas traen consigo, entre ellas la administración de la energía, haciendo uso del comportamiento inteligente y proactivo de dichos agentes, además, explotando la solución de problemas en paralelo, la distribución del trabajo y de los recursos. Sin embargo, debido a las limitaciones de recursos disponibles en una WSN, se debe tener en cuenta que los agentes no se podrán desempeñar de la misma forma que lo hacen en máquinas con mayores recursos y de esta forma se hace necesario evaluar el desempeño de dicha integración y así verificar el beneficio de la utilización de agentes inteligentes de software en dichas redes.

Es por esta razón que se propone en este trabajo de tesis un método que permita evaluar el desempeño de la integración de agentes inteligentes en las WSN a través de métricas, donde además, a partir de la evaluación realizada, se brinde la posibilidad a los usuarios de tener parámetros más claros para determinar bajo que escenarios se hace más viable la utilización de los agentes de software dentro de una WSN.

1.2 Aportes

El principal aporte de esta tesis es el desarrollo de un método para evaluar el desempeño de la integración de agentes inteligentes y WSN a través de métricas.

Otros aportes importantes son:

- Caracterización y análisis de plataformas de WSN para analizar la viabilidad en la incorporación de agentes inteligentes
- Identificación y evaluación de las relaciones que existen entre diferentes métricas de evaluación para las WSN
- Un sistema basado en agentes para el mejoramiento de los recursos energéticos de la red a través de la reprogramación de tareas y de la variación tanto de la frecuencia de muestreo como de la frecuencia de transmisión de los nodos sensores

1.3 Definición del Problema de investigación

La integración de agentes inteligentes en las WSN es promisoria en la medida en que los agentes inteligentes permiten hacer frente a los desafíos y limitaciones que las WSN traen consigo, haciendo uso del comportamiento inteligente y proactivo de dichos agentes. Un aspecto crucial en la integración, es lograr evaluar el desempeño de la WSN considerando los diferentes factores que se encuentran íntimamente relacionados, como por ejemplo, el tiempo de vida de la red y la eficiencia energética, la cobertura y el despliegue de nodos, etc.

Cabe señalar que algunos autores ya han analizado el desempeño que ha tenido la integración de agentes inteligentes, específicamente de agentes móviles, en las WSN, cuyos resultados son comparados con el modelo de computación tradicional basado en cliente/servidor, mostrando amplias ventajas respecto a éste. Sin embargo, en estos trabajos se realiza un análisis por separado de las diferentes métricas involucradas en la evaluación del desempeño de dicha integración, sin considerar las relaciones entre ellas, las cuales fueron mencionadas anteriormente.

En general, algunas limitaciones que se pueden encontrar en los trabajos revisados en el estado del arte, que se presentan al momento de evaluar el desempeño en la integración de agentes inteligentes y WSN, son las siguientes:

- No se consideran las relaciones que existen entre muchas de las métricas de evaluación descritas en (Hill, 2003) y solo se aplican algunas de ellas en la evaluación del rendimiento de la WSN.
- No se realiza ningún análisis sobre los efectos en otros atributos de la red que están directamente relacionados con la administración de la energía, como son la cobertura, la latencia, la exactitud y la longevidad.
- No se tienen en cuenta las limitaciones del hardware.
- No se define un método que permita llevar a cabo una evaluación del desempeño de la integración de WSN y Agentes Inteligentes.
- No se evalúa si es posible mejorar el desempeño de las WSN por medio de la integración de técnicas de IAD realizando un análisis de su desempeño basado en métricas, teniendo en cuenta las relaciones existentes entre éstas y de esa forma corroborar el beneficio en la utilización de agentes inteligentes en dichas redes.

1.4 Hipótesis de investigación

Se puede plantear un método para evaluar la integración de agentes inteligentes y WSN a través de métricas tales como tiempo de vida del sistema, cobertura, despliegue de nodos y eficiencia energética, considerando las relaciones existentes entre éstas. Además, se puede analizar y verificar el beneficio de la utilización de agentes inteligentes en dichas redes.

1.5 Preguntas de Investigación

Las siguientes preguntas de investigación fueron establecidas a partir de las limitaciones identificadas en la problemática de investigación a través de la revisión del estado del arte y de la definición del problema:

- ¿Es posible mejorar el desempeño de una WSN por medio de la integración de técnicas de IAD y realizar el análisis de su desempeño a través de un estudio basado en métricas?
- ¿Cuál es la técnica más apropiada para evaluar las relaciones que existen entre varias métricas de evaluación para las WSN?
- ¿Es posible determinar el efecto de cada una de las métricas de evaluación de rendimiento involucradas y determinar cuáles son más significativas y cuáles no para las WSN?
- ¿Qué elementos se deben tener en cuenta al momento de diseñar una WSN para realizar la incorporación de agentes de software inteligentes?
- ¿Qué modelos teóricos de IAD son adecuados para integrarlos en dispositivos reales con recursos limitados, teniendo en cuenta los resultados obtenidos en la evaluación del desempeño?
- ¿Cómo modelar agentes de software que se ajusten a las limitaciones de recursos de una WSN, sin perder su capacidad de deliberar?

1.6 Objetivos

1.6.1 Objetivo general

Proponer un método para evaluar el desempeño de la integración de agentes inteligentes y redes de sensores inalámbricas (WSN) a través de métricas.

1.6.2 Objetivos específicos

- Caracterizar los modelos de integración que han sido propuestos para insertar agentes de software en las WSN.

- Identificar a través de la revisión del estado del arte los factores que son más significativos para las WSN y determinar de esta manera las métricas a emplear en la evaluación de la integración de WSN y agentes.
- Proponer un método basado en métricas para realizar el seguimiento y la evaluación del desempeño de una WSN, con y sin agentes de software.
- Desarrollar y validar el método propuesto a través de casos de estudio con diferentes modelos de integración.

1.7 Alcance

El propósito de esta tesis de maestría es plantear un método, para realizar, a través de métricas, la evaluación de la integración de agentes inteligentes y WSN; además, verificar el beneficio de la utilización de agentes inteligentes en dichas redes. Sin embargo, no se considera incrementar los recursos limitados (procesamiento, memoria, energía, etc.) de los dispositivos que componen una WSN. Tampoco se contempla desarrollar un sistema adaptable a las condiciones físicas presentadas en el entorno, si no que se preestablece e implementa una plataforma de hardware WSN fija y se desarrollan agentes inteligentes de software de diversos tipos con las herramientas encontradas en la literatura.

Finalmente, no se contempla el desarrollo de modelos de integración de agentes y WSN, por el contrario, se revisaron los modelos existentes en la literatura y se escogieron algunos de estos para realizar la validación del método propuesto. Además, tampoco se considera el desarrollo de metodologías para el análisis y diseño de Sistemas Multi-Agente.

1.8 Metodología

FASE	OBJETIVOS	ACTIVIDADES
Fase 1: Conformación del marco teórico y revisión del estado del arte.	Caracterizar los modelos de integración que han sido propuestos para insertar agentes de software en las WSN	1.1 Revisión Bibliográfica sobre Inteligencia artificial e Inteligencia Artificial Distribuida. 1.2 Revisión Bibliográfica sobre WSN y métricas de evaluación de desempeño para dichas redes. 1.3 Revisión Bibliográfica de las diferentes estrategias de integración que han sido propuestas para insertar agentes en WSN. 1.4 Revisión Bibliográfica de diferentes mecanismos para la evaluación del desempeño de la integración de agentes y WSN.
Fase 2: Especificación de requerimientos.	Identificar a través de la revisión del estado del arte los factores que son más significativos para las WSN y determinar de esta manera, las métricas a emplear en la evaluación de la integración de WSN y agentes	2.1 Seleccionar los factores a evaluar con base a la importancia dada por la comunidad científica, identifica mediante la revisión del estado del arte. 2.2 Determinar las posibles métricas a utilizar para realizar la evaluación de los factores seleccionados.
Fase 3: Diseño del método propuesto.	Proponer un método basado en métricas para realizar el seguimiento y la evaluación del desempeño de una WSN, con y sin agentes de software.	3.1 Describir la aplicación que se pretende realizar dejando bien definidos sus alcances y limitaciones. 3.2 Determinar si se requiere el uso de agentes de software, ya sea en un Sistema multi-agente o en un sistema basado en agentes, o si no se requiere el uso de éstos. 3.3 Determinar las características de las plataformas hardware disponibles para la implementación de la red, realizar una revisión detallada de sus características e identificar la posibilidad de la utilización de agentes de software en ellos. 3.4 Seleccionar la plataforma hardware adecuada. 3.5 Identificar las variables relevantes para la evaluación del desempeño de red en la cual se está aplicando el método y con base a ellas seleccionar las métricas a ser usadas para la evaluación del desempeño de la red. 3.6 Desplegar la red.
Fase 4: Implementación y validación del método propuesto.	Desarrollar y validar el método propuesto a través de casos de estudio con diferentes modelos de integración.	4.1 Definir un caso de estudio con el fin de evaluar el rendimiento del método propuesto. 4.2 Seleccionar el modelo adecuado para realizar la integración de agentes de software en la red en caso de ser requeridos. 4.3 Implementar el método propuesto y proveer toda la documentación necesaria como un

		soporte del trabajo realizado 4.4 Validar el sistema.
Fase 5: Producción de artículos y/o informes.	Todos	5.1 Elaborar informes y artículos.

1.9 Organización del documento

El desarrollo de esta tesis por capítulos, con el fin de dar cumplimiento al objetivo general a través de los objetivos específicos, se ha estructurado de la siguiente manera: En el capítulo 2 se presenta una introducción a los conceptos básicos de la Inteligencia Artificial (IA), la Inteligencia Artificial Distribuida (IAD) y las Redes de Sensores Inalámbricas (WSN por su acrónimo en inglés), además, se presentan las diferentes métricas de evaluación de desempeño de las WSN. En el capítulo 3, se realiza la revisión del estado del arte en el tema, donde se describen brevemente algunos trabajos relacionados con WSN y agentes de software. En este capítulo también se realiza la caracterización de los modelos de integración que han sido propuestos para insertar agentes de software en las WSN mediante la presentación de algunas de las estrategias que han sido usadas para realizar dicha integración y además, se exponen los factores más significativos para la evaluación del desempeño de la red identificados a partir de la revisión de los trabajos relacionados. En el capítulo 4 se presentan las etapas que conforman el método propuesto y se describen detalladamente cada una de ellas y en el capítulo 5 se muestra su validación a través de la implementación de casos de estudio, los resultados obtenidos y su análisis. Finalmente, en el capítulo 6, se presentan las conclusiones y el trabajo futuro.

1.10 Difusión de resultados

A continuación se presentan las publicaciones en revistas científicas, memorias en congresos nacionales y algunas publicaciones relacionadas que se han realizado y han permitido la difusión de esta investigación.

- Muñoz T., Ovalle D., Montoya A., "Hacia la evaluación del desempeño de un modelo de integración de agentes inteligentes & WSN a través de métricas". Quinto Congreso Colombiano de Computación (5CCC). Universidad de Cartagena, Cartagena – Colombia, Abril 13-16 de 2010.

- J. Hernández, S. Restrepo, T. Muñoz, D. Ovalle, "Modelo De Un Sistema Híbrido Multi-Agente Para La Detección Y Control De Incendios Ambientales", Quinto Congreso Colombiano de Computación (5CCC). Universidad de Cartagena, Cartagena – Colombia, Abril 13-16 de 2010.

- Muñoz T., Ovalle D., Montoya A. "Hacia la evaluación del desempeño de un modelo de integración de agentes inteligentes & WSN a través de métricas". Revista Colombiana de Computación, ISSN: 16572831, vol.1, no 11, pp. 98-113, junio 2010.

- Muñoz T., Montoya A., Ovalle D. "Análisis de métricas para la Evaluación del Desempeño de WSN enriquecidas con Agentes Inteligentes". Tendencias en Ingeniería de Software e Inteligencia Artificial, ISSN: 978-958-44-8153-5, vol. 4, pp. 139-146, febrero de 2011.

- Aristizábal D., Restrepo R., Ramírez C., Muñoz T., González E., Montoya N. "Enseñanza de las Ciencias Naturales usando las NTIC a través de la Plataforma PhysicsSensor-Arduino". IX Taller internacional "ENFIQUI 2012" "La enseñanza de la Física y la Química" "La enseñanza de las ciencias naturales", ISBN: 978-959-18-0773-1 Matanzas-Cuba. Junio de 2012.

Tesis parcialmente financiada por el proyecto de investigación de la DIME titulado: "Modelo de Sistema Híbrido Inteligente para el Monitoreo de Variables Físicas usando Redes de Sensores Inalámbricos y Sistemas Multi-agente" con código: 20201007312, de la Universidad Nacional de Colombia – Sede Medellín.

Capítulo 2

Marco teórico

En este capítulo se describen los conceptos teóricos básicos de esta tesis, los cuales conformarán la fundamentación base que ayudara finalmente a la comprensión del trabajo propuesto. Temas como Inteligencia Artificial, Inteligencia Artificial Distribuida, Redes de Sensores Inalámbricas y Métricas para la evaluación del desempeño de una red serán presentados.

2.1 Inteligencia Artificial e Inteligencia Artificial Distribuida (IAD)

2.1.1 Inteligencia Artificial (IA)

La Inteligencia Artificial (IA) está relacionada con la construcción de dispositivo dotados de inteligencia, la cual puede ser definida desde el punto de vista de los procesos de pensamiento y razonamiento, donde se considera que los sistemas están dotados de inteligencia cuando piensan y actúan como los humanos, o desde el comportamiento, cuando estos piensan y actúan racionalmente dentro de unos parámetros que son considerados como correctos, comportándose de la mejor manera posible (Russell & Norvig, 2003). Un enfoque moderno de la IA está centrado alrededor del concepto de un agente racional (Vlassis, 2007).

2.1.1.1 Agente

Un agente es definido como una entidad software o hardware que realiza algún conjunto de tareas en beneficio de usuarios con algún grado de autonomía (Yasemin, 1999) (Russell & Norvig, 2003), que puede percibir su ambiente a través de sensores y actuar sobre él a través de actuadores (Figura 1) (Vlassis, 2007).

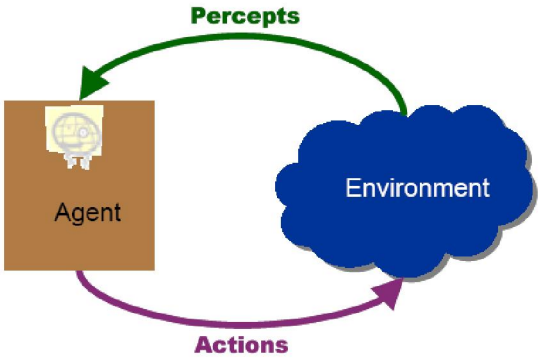

Figura 1. Esquema general de la interacción del agente con el ambiente (tomada de (Vlassis, 2007))

Un agente inteligente o racional es capaz de actuar autónoma y flexiblemente para responsabilizarse de sus objetivos de diseño, es decir, es capaz de seleccionar siempre una acción que optimice una medida de funcionamiento apropiada, según sus conocimientos. La medida de funcionamiento es definida típicamente por el usuario (el diseñador del agente) y refleja lo que espera el usuario del agente en la tarea actual (Vlassis, 2007).

El hecho de que un agente inteligente sea flexible significa que están habilitados para percibir su ambiente y responder de una manera oportuna a los cambios que ocurran en él y de esta manera satisfacer sus objetivos de diseño (**Reactividad**). Además, tienen la capacidad de interactuar con otros agentes y posiblemente seres humanos con la finalidad de ejecutar las tareas para las cuales fueron diseñados (**Habilidad social**) y al mismo tiempo pueden exhibir un comportamiento dirigido a metas tomando la iniciativa para satisfacer sus objetivos de diseño (**Pro-actividad**) (Jennings, et al., 1998). En general, los agentes son: autónomos, orientados a metas, colaborativos, flexibles, comunicativos y adaptativos. Además, pueden ser móviles dándoles la capacidad de migrar entre los nodos de una red para realizar una o varias tareas (Tynan, et al., 2005).

Como se verá más adelante, aunque los agentes son autónomos no son totalmente independientes de otros agentes. Cuando los agentes se comunican y cooperan, llega a ser posible construir organizaciones de agentes conocidas como Sistemas Multi-agente (Yasemin, 1999).

2.1.1.2 Arquitecturas

Las arquitecturas de agentes pueden ser divididas en cuatro grupos: Basadas en la lógica, Reactivas, BDI (Belief-Desire-Intention) y por capas (Wooldridge, 2002). Éstas especifican los

mecanismos fundamentales que soportan el comportamiento del agente para actuar, por ejemplo, ante un estímulo.

Los agentes realizados bajo la **arquitectura basada en la lógica**, toman decisiones a través de deducciones lógicas, en el cual el ambiente es simbólicamente representado y manipulado usando mecanismos de razonamiento. En la **arquitectura reactiva**, la toma de decisiones se basa en un mecanismo estímulo-respuesta, que ofrece respuestas inmediatas a estímulos del entorno sin razonar a cerca de estos (no utiliza un modelo de razonamiento complejo como en el caso de la arquitectura basada en la lógica). Por su parte, la **arquitectura BDI** tiene sus bases en el razonamiento práctico, es decir, el proceso de decidir momento a momento cuales acciones ejecutar en busca de las metas establecidas. Este razonamiento práctico tiene dos procesos sumamente importantes, los cuales son, decidir que metas se quieren lograr y cómo lograrlas. Ha sido el modelo más estudiado e implementado recientemente para el diseño de agentes, ya que combina el modelo psicológico del razonamiento practico humano, numerosas implementaciones, varias aplicaciones y unas semánticas lógicas muy bien definidas, las cuales han sido tomadas en cuenta dentro de toda la comunidad de investigación de agentes (Georgeff, et al., 1999) (Rao & Georgeff, 1995). Las creencias representan las cosas que el agente considera ciertas en la actualidad acerca de sí mismo y de su entorno. Los deseos son los objetivos o metas que se adoptan cuando se cumplen unas pre-condiciones para que las creencia sean adoptadas y las intenciones son los pasos a seguir involucrados en el logro de un deseo adoptado (plan de acciones). Finalmente, en la **arquitectura en capas** la toma de decisiones es realizada a través de diferentes capas de software cada una de las cuales razona acerca del ambiente en diferentes niveles de abstracción. Particularmente pueden existir 2 capas, una para los comportamientos reactivos y otra para los deliberativos (arquitectura híbrida).

2.1.2 Inteligencia Artificial Distribuida (IAD)

La Inteligencia Artificial Distribuida (IAD) es un sub-campo de la IA dedicada al desarrollo de soluciones distribuidas para problemas complejos, explotando la solución de problemas en paralelo, la distribución del trabajo y de los recursos. Ésta se ocupa de los principios y del diseño de los **Sistemas Multi-Agente**, los cuales son sistemas distribuidos compuestos por agentes de software que pueden interactuar potencialmente uno con otro, con el fin de alcanzar sus objetivos individuales y grupales. Esta interacción entre agentes requiere de capacidades de cooperación, coordinación y negociación dinámica (Vlassis, 2007).

2.1.3 Sistema Multi-Agente (SMA)

Como se mencionó anteriormente, un SMA consiste de un grupo de agentes que pueden interactuar potencialmente uno con otro (Vlassis, 2007) a través de lenguajes de comunicación entre agentes; el más utilizado es el lenguaje de comunicación ACL (Agent Communication Languaje) (Anon., 2012).

Los SMA ofrecen modularidad, es decir, si un dominio del problema es particularmente complejo, extenso o imprevisible, este puede ser dividido en varias partes y se usan componentes modulares (agentes) que son especializados en resolver una parte en particular. Esta descomposición permite que cada agente utilice el paradigma más apropiado para solucionar su problema y a través de la cooperación facilitar soluciones cuando se presentan problemas interdependientes ya que la experiencia está distribuida (Sycara, 1998).

El tipo de interacción entre los agentes puede ser descrito como cooperativo cuando comparten metas, self-interested cuando sus metas son totalmente independientes o competitivos cuando tienen metas mutuamente exclusivas.

Algunas características de los SMA, dadas en (Jennings, et al., 1998) (Sycara, 1998), son:

- Cada agente tiene capacidades o información incompleta para solucionar el problema, luego, tiene un punto de vista limitado
- No hay un sistema global de control
- Los datos están descentralizados
- La computación es asíncrona

Además, los SMA son sistemas distribuidos y tienen la capacidad de ofrecer un aumento en la velocidad y eficiencia, ya que ellos pueden funcionar asíncronamente y en paralelo lo cual puede incrementar la velocidad total, la escalabilidad y la confiabilidad, ya que si un agente falla otro agente en el sistema puede asumir el control.

2.1.4 Metodologías para el desarrollo de SMA

Para el análisis y diseño de SMA se han creado varias metodologías, las cuales proporcionan modelos para capturar los principales rasgos del sistema y su entorno y además, definen el conjunto de pasos necesarios para el desarrollo e implementación de un Sistema Multi-Agente.

Entre las metodologías más representativas se plantean MAS-CommonKADS (Iglesias Fernández, 1998), la cual es una metodología de propósito general para el desarrollo de sistemas multi-agente resultante de la extensión del método ***CommonKADS*** (Bayer & Svantesson, 2001); y GAIA (Wooldridge, et al., 1999), especialmente diseñada para el análisis y diseño de la SMA, permitiendo capturar la flexibilidad, la interacción entre los agentes y la complejidad de una estructura organizacional de un sistema de agentes (Bayer & Svantesson, 2001). Gaia se basa en la idea de un SMA como una organización computacional compuesta de varios roles interactuando (obrando recíprocamente) (Wooldridge, et al., 1999).

Otras metodologías desarrolladas son **MaSE** (DeLoach, 2001) (DeLoach, 2006), la cual es una abstracción del paradigma orientado a objetos, donde los agentes son objetos especializados (que pueden o no tener inteligencia) y la **Metodología de Agent Factory (AF)** para la construcción de SMA basado en el modelo BDI (Collier, 2002).

2.1.5 Sistemas basados en agentes vs Sistemas Multi-Agente

Según (Jennings, et al., 1998), un sistema basado en agentes puede ser contextualizado (diseñados e implementados) en términos de agentes, pero puesto en funcionamiento sin ninguna estructura de software correspondiente a éstos. Además, pueden contener uno o más agentes diseñados para interactuar con el ambiente y no con otros agentes específicos, necesitando solo identificar los tipos de agentes del sistema e impacto en el ambiente (H. Van Dyke Parunak, 1999). Por otra parte, los SMA están diseñados e implementados con la idea de que varios agentes estarán cooperando entre sí, con el objetivo de solucionar problemas para los cuales pueden existir múltiples métodos de solución. Por ello, uno de los aspectos básicos en estos sistemas es la interacción entre los agentes que lo conforman y la definición de

modelos concretos de cooperación, coordinación o negociación de los agentes son indispensables.

2.1.6 Frameworks

A continuación se describen algunas de las plataformas de desarrollo de agentes, tanto para sistemas con recursos ilimitados como para aquellos con restricciones.

2.1.6.1 Frameworks de agentes

2.1.6.1.1 Agent Factory (AF)

Agent Factory (AF) (Collier, 2002) es un framework que soporta un enfoque estructurado para el desarrollo y despliegue de aplicaciones orientadas a agentes (Collier, et al., 2003). Específicamente, éste promueve la comunicación entre agentes a través de un Lenguaje de Comunicación de Agente (ACL - Agent Communication Language) y ofrece un amplio soporte para la creación de agentes BDI (Collier, et al., 2003).

AF está organizado en dos ámbitos fundamentales: el Ambiente de Desarrollo AF y el Ambiente de Ejecución AF. El primer ambiente ofrece un conjunto de CASE (Computer-Aided Software Engineering) que apoyen el proceso de fabricación de agentes y la prueba del modelo de agentes inteligentes (Collier & O'Hare, 1999), mientras que la segunda ofrece soporte para el despliegue de aplicaciones orientados a agentes a través de una amplia gama de redes de dispositivos compatibles con Java (Collier, et al., 2003) y permite la entrega de SMA completos (Collier & O'Hare, 1999). AF fue construido usando la versión de Java J2SE (Java 2 Standard Edition) (Muldoon, 2007).

2.1.6.1.2 JADE

JADE (Java Agent Development Framework) (Bellifemine, et al., 1999) es un framework basado en Java desarrollado por Telecom Italy Lab (TILAB), antes llamado CSELT, para el desarrollo de aplicaciones distribuidas multi-agente que cumplen con las especificaciones FIPA para la interoperabilidad de SMA (Bellifemine, et al., 2000).

El framework está compuesto de un número de contenedores de agentes, los cuales se ejecutan en una máquina virtual de Java (JVM) separada. La comunicación dentro de los contenedores se ve facilitada por el control de eventos. Dentro de cada contenedor se soportan los servicios de páginas blancas y páginas amarillas a través del Sistema de Administración de Agentes (AMS) y del Directorio Facilitador (DF) respectivamente (Muldoon, 2007). Además, JADE contiene varias herramientas gráficas para la depuración y pruebas.

En general, JADE proporciona:

- Entorno de desarrollo para la creación de aplicaciones basadas en agentes.
- Entorno de ejecución para que los agentes vivan y se comuniquen.
- Arquitectura peer-to-peer (P2P)
- Interoperabilidad: especificaciones FIPA
- Portabilidad: realizado en JAVA

2.1.6.2 Frameworks para el desarrollo de agentes para sistemas de recursos limitados

Tradicionalmente los SMA estaban destinados básicamente a ambientes de escritorio. Sin embrago, debido al crecimiento exponencial de los dispositivos móviles, el potencial para realizar el despliegue de sistemas distribuidos multi-agente en estos ambientes de recursos limitados es de gran interés. En esta sección se describen algunas de las plataformas para el desarrollo de agentes específicamente para dispositivos móviles de recursos limitados.

2.1.6.2.1 LEAP

LEAP (Light Extensible Agent Platform) es una plataforma de agentes compatible con FIPA capaz de operar tanto en dispositivos fijos y como en dispositivos móviles. LEAP amplía la arquitectura de JADE mediante el uso de un conjunto de perfiles que le permite ser configurado para diferentes Máquinas Virtuales de Java (JVM). La arquitectura es modular y contiene componentes para la administración del ciclo de vida de los agentes y el control de la heterogeneidad de los protocolos de comunicación.

LEAP cuando se combina con JADE sustituye determinados componentes del ambiente de ejecución de JADE para formar un núcleo modificado que se conoce como JADE-LEAP. Esta modificación del núcleo se puede implementar a través de una serie de plataformas y facilita el desarrollo de aplicaciones para máquinas de escritorio, PDAs y teléfonos móviles. JADE-LEAP ofrece un API estándar para el desarrollador, que puede ser utilizado en cualquier entorno (Muldoon, 2007).

2.1.6.2.2 Agilla

Agilla (Fok, et al., 2005) (Fok, et al., 2006) (Fok, et al., 2009) es un middleware para WSN que provee un estilo de programación basado en agentes móviles. Agilla se basa en TinyOS (Hill, 2003), el cual es un entorno de distribución libre diseñado especialmente para WSN. Una red Agilla comprende una familia de agentes móviles que son capaces de migrar tanto su estado y como su código. Cada agente actúa de forma autónoma y hay un número determinado de agentes en cada nodo. El número de agentes es variable y está determinado principalmente por la cantidad de memoria disponible (Muldoon, 2007). Para la coordinación y comunicación se utiliza el método de pizarra mediante recursos como una lista de vecinos y espacios de tuplas.

Mediante Agilla se puede incrementar la utilidad de una WSN, puesto que se pueden tener múltiples agentes embebidos en la red de sensores que ejecuten tareas asociadas a distintas aplicaciones simultáneamente. De esta forma, los agentes podrán compartir los recursos de un mismo dispositivo, y por lo tanto múltiples agentes móviles pueden residir en un mismo nodo de la WSN.

La motivación de Agilla es aumentar la flexibilidad de las WSN, habilitando aplicaciones para adaptarse a los cambios impredecibles y necesidades emergentes de los usuarios. Por ejemplo, debido a que los agentes pueden morir y nuevos agentes pueden ser inyectados a la WSN, existe la posibilidad de reprogramar la red sin necesidad de acceder físicamente a cada uno de los dispositivos. Adicionalmente, se permite a los agentes controlar su destino dentro de la red y mantener tanto su código como su estado a través de toda la red.

Agilla soporta la migración, pero los agentes de Agilla no son inteligentes y no contienen ninguna capacidad de razonamiento (Muldoon, 2007).

2.1.6.2.3 AFME

AFME (Agent Factory Micro Edition) es un framework que soporta el desarrollo de agentes racionales para dispositivos móviles. Éste ha sido específicamente diseñado para hacer frente a los problemas de rendimiento y tamaño de memoria asociados con la ejecución de agentes intencionales en los dispositivos móviles. AFME es una extensión de Agent Factory (AF), que incluye capacidades adicionales para la toma de decisiones racionales con el fin de permitir que los agentes de manera eficaz razonen con recursos computacionales limitados.

Se basa en la plataforma de Java CLDC (Constrained Limited Device Configuration) aumentada con MIDP (Mobile Information Device Profile), los cuales constituyen un subconjunto de la especificación J2ME (*Java 2 Micro Edition*). CLDC /MIDP es de hecho el estándar en la plataforma Java para teléfonos móviles y otros dispositivos con recursos limitados.

Los agentes de AFME adoptan una actitud intencional en el razonamiento acerca de sí mismos y su entorno. El proceso de migración de AFME facilita la migración de los agentes desde fuentes ricas en recursos a ambientes con recursos escasos y viceversa (Muldoon, 2007).

2.2 Redes de sensores Inalámbricos

Las Redes de Sensores Inalámbricas o WSN (por su acrónimo en inglés Wireless Sensor Networks), en general, son redes inalámbricas que se componen de centenares o miles de dispositivos autónomos y compactos llamados nodos sensores (Vieira & Da Silva, 2003), los cuales tienen capacidades de sensado, cómputo y comunicación inalámbrica (Ishibashi & Yano, 2005) y actúan de manera conjunta con el fin de monitorear o controlar distintos entornos físicos. Cada uno de los nodos de una WSN es esparcido en un campo sensor y toman el nombre de “mote”. Las WSN, constituye normalmente una red inalámbrica ad-hoc y soporta un algoritmo de enrutamiento de múltiples saltos (Sankaranarayanan, 2009).

Los principales componentes de una WSN son una estación base (nodo “sink”) y uno o más nodos sensores (ver Figura 2). Cada nodo funciona sin ataduras, tiene un microprocesador y una pequeña cantidad de memoria para el procesamiento de la señal y la

programación de tareas. También, están equipados de uno o más dispositivos de detección tales como arreglos de micrófonos, cámaras de video o fotográficas, infrarrojo (IR), sísmicos o sensores magnéticos (Zhao & Guibas, 2004), los cuales producen una respuesta medible a un cambio en condiciones físicas como la temperatura y la presión (Vieira & Da Silva, 2003). Las funciones deseables para los nodos incluyen: facilidad de la instalación, auto-identificación, auto-diagnóstico, confiabilidad, conocimiento del tiempo para la coordinación con otros nodos (Lewis, 2004), entre otras.

Los nodos retransmiten sus datos sensados directamente a la estación base o por medio de otro nodo (denominado multi-hopping), dependiendo de la escala de la red. A su vez, la estación base envía comandos a los nodos (Tynan, et al., 2005). Las estaciones base son una o más componentes distinguidos de la WSN con muchos más recursos de cómputo, energía y de comunicación. Actúan como un gateway o nodo recolector entre los nodos sensor y el usuario final (Sankaranarayanan, 2009).

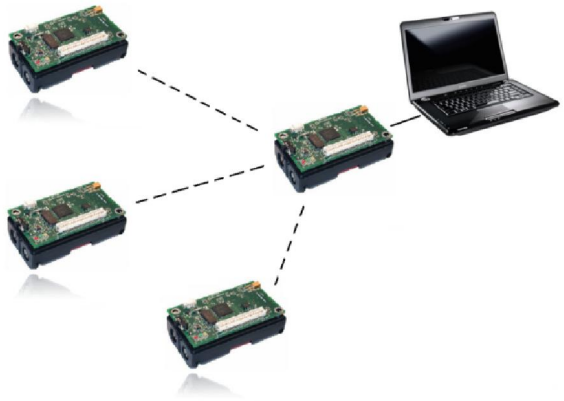

Figura 2. Esquema básico de una WSN (3 motes unidos a un PC a través de una estación base). Las líneas punteadas representan una conexión inalámbrica desde los 3 nodos a la estación base y la línea sólida es una conexión cableada de la estación base al PC

Tomada de **(Tynan, et al., 2005)**

Aunque las WSN son redes de dispositivos de cómputo, ellas son consideradas diferentes de las redes de datos tradicionales. A diferencia de los sistemas de red tradicionales, una WSN está limitada por el consumo de energía, la capacidad de almacenamiento de datos y el ancho de banda de la red. Además, otra diferencia, es el área de aplicación y las implicaciones que estas traen consigo para el tráfico de datos y la interacción de los usuarios. En el caso de las redes tradicionales, un usuario está unido a un nodo o grupo de nodos y si requiere conectarse a otro nodo o un servicio de otro nodo, la red actúa como un medio para unirlos y

el usuario interactúa directamente con otro usuario o servicio en el otro extremo realizando la transmisión de datos. En las WSN, se conectan sistemas distribuidos (nodos), los cuales recopilan datos que en conjunto poseen valiosa información, teniendo en cuenta que los usuarios raramente están interesados en lecturas de uno o 2 nodos en especifico (Sudharsan, et al., 2012).

Los sensores individuales en estas redes deben coordinar típicamente sus acciones de detección con los sensores próximos para alcanzar los objetivos de todo el sistema (por ejemplo, variando sus tiempos de sensado y el tiempo de inactividad con el fin de maximizar la vida de la batería, al mismo tiempo que se reduce la detección redundante de áreas superpuestas). Además, la red de forma autónoma debe adaptar sus respuestas en un entorno en continua evolución de tal manera que puede alcanzar los objetivos a largo plazo de todo el sistema sin intervención humana directa (Rogers, et al., 2009). Asimismo, las WSN tienen capacidad de auto-restauración, es decir, si se avería un nodo, la red encontrará nuevas vías para enrutar los paquetes de datos. De esta forma, la red sobrevivirá en su conjunto, aunque haya nodos individuales que pierdan potencia o se destruyan (Aakvaag & Frey, 2006).

Dado que las WSN a menudo deben recopilar datos durante largos períodos de tiempo y se utilizan en ambientes inhóspitos, donde cambiar las baterías es un inconveniente o un hecho imposible, la mayor parte de la investigación en este ámbito, aborda el reto de reducir al mínimo las necesidades de energía de cada sensor (Rogers, et al., 2009).

Algunas características distintivas de las WSN, tomadas de (Potdar, et al., 2009), son:

a) Topología única de la red
b) Aplicaciones diversas
c) Características únicas de tráfico
d) Limitaciones severas de recursos

Según (Zhao & Guibas, 2004), algunos de los desafíos a los que se les debe hacer frente al momento de diseñar redes de sensores incluyen:

- **Hardware limitado:** Cada nodo tiene capacidades limitadas de procesamiento, almacenamiento y comunicación, igualmente sus fuentes de energía y ancho de banda son limitados.

- **Soporte limitado para el establecimiento de una red (networking):** La red es par-a-par, con una topología mesh y una conectividad dinámica, móvil y no confiable. No existen protocolos de enrutamiento universales o servicios centrales de registro. Cada nodo actúa como un router y como un host de la aplicación.
- **Soporte limitado para el desarrollo del software:** Las tareas son típicamente en tiempo real y distribuidas masivamente, lo cual implica la colaboración dinámica entre nodos y deben manejar eventos de competencia múltiples.

2.2.1 Topología de red

Las WSN pueden ser diseñadas en una variedad de formas para abordar las distintas prioridades y hacer uso de la tecnología apropiada con base en los requisitos de la aplicación. Entre las diferentes arquitecturas que pueden ser usadas para implementar aplicaciones de WSN se incluyen la topología en estrella, mesh (malla) y un hibrido entre la estrella y la mesh. Las topologías se refieren a la configuración de los componentes hardware y como los datos son transmitidos a través de dicha configuración.

La topología de estrella (Figura 3), es un sistema de un único salto en la cual los nodos están dentro del rango de comunicación directo, usualmente entre 30-100 metros, hasta el Gateway. Todos los sensores son idénticos, todos son endpoints, y el Gateway sirve para comunicar los datos y comandos a estos puntos terminales. El Gateway también es usado para transmitir los datos a una estación base. Entre los endpoints no se da el paso de datos o comandos, ellos usan el gateway como un punto de coordinación. Este tipo de topología ofrece el menor consumo general de energía, pero está limitada por la distancia de transmisión de radio desde cada endpoint al gateway. Asimismo, no hay vías de comunicación alternativas a dichos puntos finales. Si la ruta se obstruye, la comunicación con el extremo asociado (endpoint) se perderá (Crossbow Technology, Inc, 2007).

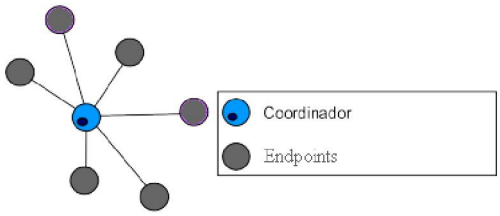

Figura 3. Topología de red tipo estrella

La topología mesh o malla (Figura 4), es un sistema de múltiples saltos en la cual los nodos son idénticos y los dispositivos finales, en este caso, se comportan como routers y tienen la posibilidad de comunicarse unos con otros y con el Gateway. Este tipo de red es altamente tolerante a fallas, ya que cada nodo tiene múltiples trayectorias para comunicarse con el gateway y otros nodos (Crossbow Technology, Inc, 2007).

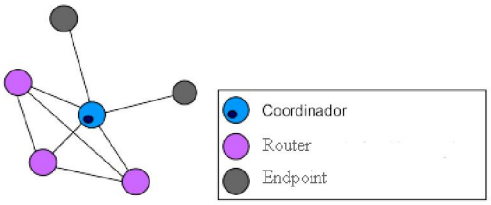

Figura 4. Topología de red tipo malla

Finalmente, como se mencionó anteriormente, también se puede tener una topología híbrida entre una topología estrella y una topología malla (Figura 5). Este tipo de topología, busca aprovechar la baja potencia y simplicidad de la topología en estrella, así como el amplio rango de alcance una topología de malla. Ésta organiza los nodos sensores en una topología de estrella alrededor de routers que, a su vez, se organizan en una red de malla. Los routers sirven tanto para ampliar el alcance de la red como para ofrecer tolerancia a fallos. Puesto que los nodos de sensores inalámbricos se pueden comunicar con varios routers, la red se reconfigura a si misma alrededor de los routers restantes en el caso de que uno falle o si uno de los enlace de radio no tiene el rango apropiado (Crossbow Technology, Inc, 2007).

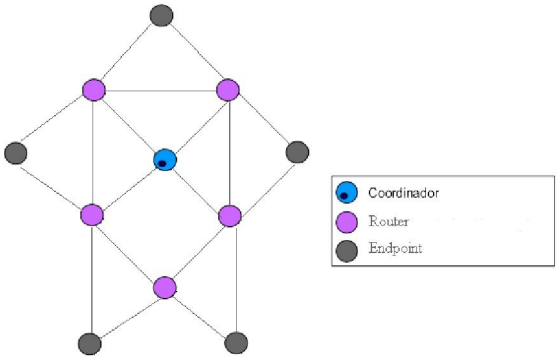

Figura 5. Topología de red híbrida

2.2.2 Motes

Como se había mencionado anteriormente, los motes son unidades autónomas que constan de un microcontrolador, una fuente de energía (casi siempre una batería), un radio transceptor y un elemento sensor (Aakvaag & Frey, 2006).

En la Figura 6 se presenta la arquitectura del sistema de un nodo sensor genérico. Este se compone de cuatro bloques principales: fuente de alimentación, la comunicación, la unidad de procesamiento y el bloque de sensores. El bloque de la fuente de alimentación consiste en una batería y un conversor DC-DC y tiene el propósito de accionar el nodo. El bloque de la comunicación consiste de un canal de comunicación inalámbrico (Transmisor/Receptor de radio). La mayor parte de las plataformas utilizan la radio de corto alcance (radiofrecuencia), otras soluciones incluyen la comunicación óptica (láser) y el infrarrojo. La unidad de procesamiento se compone de memoria para almacenar datos y programas de aplicaciones, un microcontrolador y un conversor análogo-digital para recibir la señal del bloque de sensado. Este último bloque liga el nodo al mundo físico y tiene un grupo de sensores y de actuadores que dependen de la aplicación de la WSN (Vieira & Da Silva, 2003). Entre los diferentes tipos de sensores se tienen, por ejemplo, magnetómetros, acelerómetros, sensores de luz, temperatura, presión, humedad, etc. La tendencia del miniaturización de los componentes del mote ha conducido no solamente a su tamaño pequeño y discreto sino también a su operación de baja potencia (Tynan, et al., 2005).

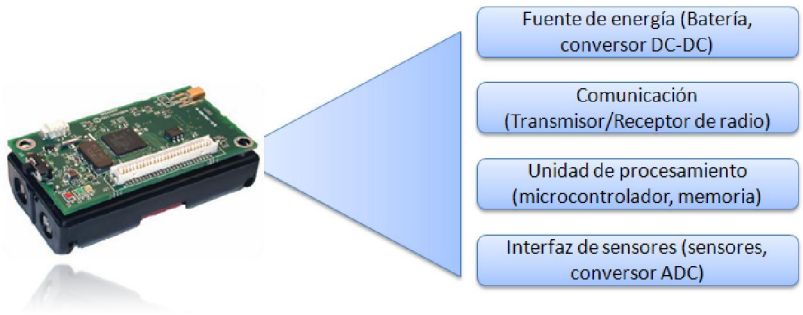

Figura 6. Arquitectura de un nodo sensor genérico

Una de las plataformas mote comúnmente usada es la conformada por la familia de los MICA (Hussain, et al., 2006). Además de las plataforma MICA (Mica2/Micaz), otras plataformas disponibles en el comercio de WSN incluyen las plataformas Telos B/Tmote Sky, IRIS (MEMSIC Inc., 2010), los Sun SPOT (por sus siglas en inglés Small Programmable Object Technology) (Sun SPOT World, 2012), entre otras.

La plataforma MICA Mote (MEMSIC Inc., 2010) está conformada por dispositivos de tercera generación, es totalmente compatible con la plataforma software MoteWorks™ (Crossbow, s.f.) (Crossbow Technology, Inc., 2006) y permite a los usuarios crear redes ad-hoc. El Mica2, desarrollado en el año 2001, opera bajo la frecuencia de 868/916 MHz y posee un radio receptor-transmisor multicanal. Por su parte, el MicaZ, desarrollado en el año 2004, es una versión mejorada del Mica2 y opera en la banda de los 2,4 GHz, además, soporta el estándar IEEE/ZigBee 802.15.4.

Las plataformas mote TelosB y Tmote Sky (MEMSIC Inc., 2010) (Moteiv Corporation, 2006), desarrollados en el año 2004 y 2006 respectivamente, son módulos sensores inalámbricos de baja potencia y de código abierto; se pueden programar por USB, son compatibles con el estándar IEEE 802.15.4 y además, tienen radio, antena y sensores integrados. También, poseen un microcontrolador de bajo consumo, memoria extendida y un conjunto de sensores opcionales. El Tmote Sky es una versión mejorada del diseño de TelosB (Moteiv Corporation, 2006). Este incluye un incremento en el rendimiento, el alcance y en general, de funcionalidad.

El IRIS es un mote que trabaja en la banda de los 2,4 GHz. Se encuentra disponible como un módulo (M2110) o como una plataforma a nivel de board (XM2110). Éste proporciona hasta

tres veces más alcance de radio y el doble de memoria de programa en comparación con generaciones anteriores de MICA motes. Además, facilita la construcción de redes mesh (MEMSIC Inc., 2010). En algunas casos, los IRIS se han provisto de una nueva plataforma de software llamada Mote Runner (Kramp, et al., 2008) (IBM, 2010), la cual se basa en una máquina virtual desarrollada desde cero para los entornos hardware con recursos limitados, introduciendo un nuevo lenguaje de byte-code que, además de ser compacto y eficiente, proporciona soporte nativo para la programación reactiva (IBM, 2010).

Los Sun SPOT (Sun SPOT World, 2012) son dispositivos embebidos desarrollados por la Sun Microsystems, ahora propiedad de Oracle (www.oracle.com), programables en Java y construidos bajo el estándar IEEE 802.15.4. Estos tienen una JVM especial llamada Squawk (Oracle_Java.net, 2010), la cual puede alojar múltiples aplicaciones simultáneamente y no requiere ningún sistema operativo subyacente.

En la Tabla 8, presentada al final de este documento, se resumen las principales características de los motes mencionados anteriormente ((Moteiv Corporation, 2006) (Healy, et al., 2008) (Johnson, et al., 2009) (Potdar, et al., 2009) (Memsic Corporation, 2010) (MEMSIC Inc., 2010) (Sun SPOT World, 2012)). Sin embargo, se debe tener en cuenta que actualmente existen otros alternativas que también permiten el despliegue de WSN, como por ejemplo los módulos inalámbricos Xbee (Digi International Inc., 2013) y la tarjeta de adquisición de datos ARDUINO (Arduino SA, 2013), los cuales acoplados a diferentes sensores permiten el despliegue de una red.

En general, la comunicación entre los diferentes dispositivos se da a través de ondas de radio, las cuales son emitidas desde la antena transmisora T_x hasta la antena receptora R_x. Si P_T es la potencia con la cual se emiten las ondas desde T_x y P_R la potencia con la cual éstas son recibidas en R_x, se puede demostrar bajo la aproximación de Fraunhofer, es decir, de campo lejano, que sin tener en cuentas las pérdidas no relacionadas con la propagación, como por ejemplo las pérdidas por absorción, se cumple la Ecuación 1.

Ecuación 1. Potencia recibida por la antena receptora

$$P_R = P_T G_T G_R \left(\frac{\lambda}{4\pi d} \right)^2$$

Donde G_T es la ganancia de la antena transmisora T_x , G_R la ganancia de la antena receptora R_x , λ la longitud de onda de la onda de radio emitida y d la distancia entre T_x y R_x .

La teoría que predice ésta expresión se conoce como el ***modelo de propagación en el espacio libre*** y supone que entre T_x y R_x se tiene línea de vista (line-of-sight). En términos logarítmicos, se define el nivel de pérdidas de potencia por trayecto (Path Loss), β_{PL} , como se muestra en la Ecuación 2. Este valor es medido en decibeles (dB) y se calcula haciendo en la Ecuación 1 $G_T = G_R = 1$.

Ecuación 2. Nivel de pérdidas de potencia por trayecto

$$\beta_{PL} \equiv 10 \log \frac{P_T}{P_R}$$

Por lo tanto, combinando la Ecuación 1 y la Ecuación 2, se obtiene la Ecuación 3.

Ecuación 3. Nivel de pérdidas de potencia por trayecto en función de la longitud de onda y la distancia entre T_x y R_x

$$\beta_{PL} \equiv -20 \log\left(\frac{\lambda}{4\pi d} \right)$$

Por ejemplo, este modelo de propagación del espacio libre puede ser aplicado en particular a las antenas de los nodos Mica2 MPR400, las cuales son monopolos resonantes de λ/4 y se pueden considerar omnidireccionales, es decir, son antenas emisoras de ondas esféricas que se aplanan a las distancias de Fraunhofer. Esto puede comprobarse cuando se mide el tamaño de éstas ($8\,cm$ aproximadamente) y se compara con el cálculo de la longitud de onda λ de la radiación electromagnética que emiten, la cual corresponde a una frecuencia de transmisión f central entre $868/916\,MHz$, con lo cual $\lambda \cong 34cm$, teniendo en cuenta que $\lambda f = c$, siendo c la velocidad de la luz en el vacío y aproximadamente en el aire.

Para $\lambda \cong 0{,}34m$ la Ecuación 3 toma la forma mostrada en la Ecuación 4 en el sistema internacional de unidades.

Ecuación 4.Nivel de pérdidas de potencia por trayecto para λ ≈ 0,34 m

$$\beta_{PL} \equiv -20 \log\left(\frac{0{,}34}{4\pi d}\right)$$

De la Ecuación 1 se puede concluir adicionalmente que para una distancia fija d entre T_x y R_x, la potencia recibida por R_x , asumiendo de nuevo ganancias unitarias de las antenas, está dada por la Ecuación 5.

Ecuación 5. Potencia recibida por R_x para una distancia d fija

$$P_R = P_T\left(\frac{\lambda}{4\pi d}\right)^2$$

En general, si las baterías de cualquier nodo emisor se van agotando, la P_T disminuye y la P_R captada por el receptor igualmente decae, y si su valor es menor que el que corresponde a la mínima sensibilidad de la antena receptora no habrá la correspondiente comunicación. Al final, habrá que agregar las otras pérdidas de potencia que no tienen que ver con la propagación en sí, como los efectos de absorción, ocasionando que P_T deba ser mayor que el calculado con la Ecuación 5, el cual garantiza el umbral mínimo.

El alcance efectivo de los motes está definido enteramente por el diseño de la antena, la cual, en el caso de comunicaciones de radiofrecuencia, para optimizar la transmisión y la recepción, la antena debe ser por lo menos λ/4, donde λ es la longitud de onda de la frecuencia portadora. El rango promedio está entre 30 m hasta los 100 m.

En la Tabla 1, tomada de (Potdar, et al., 2009), se clasifican los motes basados en el rango direccionable efectivo (AR) definido por el alcance de la antena, la habilidad de cómputo en MHz (CC), dependiendo del tipo de procesador y la energía que éste consume, y la capacidad almacenaje del microcontrolador en KB (SC).

Tabla 1. Comparación de diversos motes contra su rango direccionable, capacidad de cómputo y capacidad memoria

	AR	CC	SC
TmoteSky/TelosB	Medio	Medio	Medio-Alto
Mica2	Amplio	Medio	Alto
Iris	Amplio	Medio	Medio-Alto
Sun SPOT	Medio	Alto	Alto

2.3 Sistemas operativos para WSN

Los Sistemas Operativos (SO) tradicionales son sistemas de software por lo general diseñados para estaciones de trabajo y computadores con recursos ilimitados, el cual no es el caso de las WSN debido a las diferentes restricciones que poseen los dispositivos que las conforman. Los SO diseñados para las WSN buscan ser muy pequeños en términos de los requerimientos de memoria y codificación y juegan un papel fundamental en el desarrollo de la red, controlando la operación del mote, implementando los protocolos de red y administrando el consumo de energía del hardware (Lajara , et al., 2010). Algunos SO para WSN son mostrados en la Tabla 2.

Tabla 2. Sistemas operativos para WSN (Sohraby, et al., 2007) (Lajara , et al., 2010)

Sistema Operativo	Modelo	Tipo de procesos
TinyOS v1 y v2	Eventos	Tareas, comandos y manejadores de eventos
Contiki	Eventos	Proto-hilos
MantisOS	Múltiples hilos	Hilos
SenOS	Máquinas de estado	Procesos
EYE OS	Eventos	Tareas, comandos

A continuación se hará una breve descripción de los SO más utilizados en las WSN, TinyOS, Contiki y Mantis, teniendo en cuenta la selección realizada en (Lajara , et al., 2010).

2.3.1 TinyOS

TinyOS es un sistema operativo de código abierto desarrollado inicialmente por la Universidad de California - Berkeley. Es la plataforma nativa para redes de sensores más ampliamente usada; está basado en componentes y diseñado específicamente para atender las

necesidades de las WSN. Éste usa un modelo de ejecución basado en eventos para proporcionar soporte a la concurrencia de grano fino e incorpora un modelo de componentes altamente eficiente (Hill, 2003). En este contexto, una descripción de grano fino de un sistema, es un modelo detallado de bajo nivel del mismo y la concurrencia se refiere a que las WSN o específicamente los nodos de la red deben soportar y manejar múltiples flujos de información simultáneamente, es decir, diferentes datos se deben mantenerse en movimiento al mismo tiempo (Hill, 2003).

Como se mencionó anteriormente, TinyOS tiene un modelo de programación basado en componentes los cuales son conectados unos con otros a través de interfaces y al igual que otros sistemas operativos, organiza sus componentes de software en capas (Lajara , et al., 2010). Cuanto más baja es la capa más cercana estará al hardware y cuanto mayor sea el componente, más cerca está de la aplicación. Los componentes tienen tres conceptos computacionales: los comandos, los eventos y las tareas. Los comandos y los eventos son los mecanismos de comunicación inter-componentes (entre), mientras que las tareas se utilizan para expresar la concurrencia intra-componente. Un comando es normalmente una solicitud a un componente para realizar un servicio (Crossbow Technology, Inc., 2006). El programador TinyOS utiliza una política de planificación FIFO (First In, First Out), es decir, no se tienen preferencias y lo que llega primero, sale primero (Crossbow Technology, Inc., 2006). En el sistema operativo TinyOS, las librerías y las aplicaciones están todas escritas con su propio lenguaje de programación llamado nesC (network embedded systems C), el cual se extiende del C.

La principal diferencia entre TinyOS v1 y TinyOS v2, desde el punto de vista del programador, es una nueva clase de abstracción llamada componente genérico en la versión 2, la cual puede ser incluida en diferentes componentes, pero cada instanciación es una copia nueva diferente, independiente y privada del resto de ellos (Lajara , et al., 2010).

2.3.2 Contiki

Contiki (Dunkels, et al., 2004), al igual que TinyOS, puede ser considerado un SO dirigido a eventos, pero con la diferencia que Contiki incorpora algunas particularidades que facilitan el desarrollo de nuevas aplicaciones en las cuales hay varias tareas concurrentes involucradas (Lajara , et al., 2010). Fue desarrollado por un grupo de académicos del Swedish Institute of

Computer Science en cabeza de Adam Dunkels (Contiki, 2012). Éste utiliza el lenguaje de programación C, es altamente portable y está construido alrededor de un kernel dirigido a eventos pero proporciona la facilidad de los hilos para procesos individuales. Como se puede apreciar, Contiki sigue una arquitectura hibrida en la cual combina las ventajas de los eventos y los hilos (Farooq, et al., 2010).

Contiki se caracteriza por usar muy poca memoria del stack del microprocesador. La idea de este SO es tener varios hilos de ejecución en paralelo los cuales son activados por eventos y cuando lo hacen, realizan sus tareas y una vez terminadas quedan nuevamente a la espera hasta que otro evento los active. Los eventos se guardan en una cola circular y se procesan por orden cronológico, del más viejo al más nuevo. Cada evento generado tiene un hilo asociado para despertar, al que le puede enviar información cuando lo despierta. Los hilos y los eventos combinados reciben el nombre de protothreads.

2.3.3 Mantis

Mantis (MultimodAl NeTworks of In-situ Sensors) es un SO multi-hilo desarrollado por la Universidad de Colorado, el cual permite que una tarea corta intercale su ejecución con otras tareas más largas y complejas (Lajara , et al., 2010). Mantis soporta un diseño multi-hilo en capas, en donde cada capa actúa como una máquina virtual mejorada para las capas anteriores (Farooq, et al., 2010). Está escrito en C y soporta aplicaciones desarrolladas en C, permitiendo la fácil portabilidad a otras plataformas tanto reales como simuladas.

2.4 Métricas para la Evaluación del desempeño de una WSN

Se define una métrica como un criterio o indicador que evalúa cuantitativamente o cualitativamente el desempeño de un sistema. Dicha evaluación del desempeño, sirve para comparar 2 o más sistemas y determinar el impacto de agregar o eliminar un componente. Algunas de las características de una buena métrica son: linealidad, confiabilidad, repetitividad, facilidad de medir y consistencia.

Hill (Hill, 2003), enuncia que las métricas de evaluación claves para las WSN, como un todo, son principalmente el tiempo de vida, la cobertura, el costo y la facilidad del despliegue, el tiempo de respuesta, la exactitud temporal, la seguridad y la efectividad en las tasas de

muestreo. Muchas de estas métricas de evaluación están interrelacionadas y puede ser necesario disminuir el rendimiento en una de ellas para incrementar el de la otra. A su vez, las métricas de evaluación de un nodo sensor son la energía, la flexibilidad, la robustez, la seguridad, la comunicación, el tiempo de sincronización, el tamaño y costo. A continuación, se explicaran en detalle las métricas de evaluación de una WSN como un todo.

Métricas de evaluación del sistema como un todo

- **Tiempo de vida:** Este término puede ser definido de varias maneras según (Sohraby, et al., 2007): (a) el tiempo de duración hasta que algún nodo agota toda su energía, o (b) el tiempo de duración hasta que la calidad de servicio de las aplicaciones no se pueden garantizar, o (c) el tiempo de duración hasta que la red esté disociada.

  El principal factor limitante para la vida de una red de sensores es el suministro de energía. Cada nodo debe estar diseñado para gestionar su suministro local de energía a fin de maximizar la duración total de la red (Hill, 2003), puesto que es difícil substituir o recargar miles de nodos.

- **Cobertura:** La cobertura se define como la relación entre el espacio monitoreado y todo el espacio de posible monitoreo (Sohraby, et al., 2007). Es importante tener en cuenta que la cobertura de la red no es igual al rango de los enlaces de comunicación inalámbrica que están siendo utilizados. Técnicas de comunicación de múltiples saltos pueden ampliar la cobertura de la red mucho más allá del alcance de la tecnología de radio. Sin embargo, esto puede aumentar el consumo de energía de los nodos y disminuir la vida útil de la red (Hill, 2003).

- **Despliegue:** El despliegue de nodos es el problema básico en las redes de sensores inalámbricas, ya que estas redes, en algunas ocasiones se deben desplegar en ambientes inhóspitos o de difícil acceso, donde una ubicación manual sería un inconveniente o prácticamente imposible. La dispersión de sensores por aeronaves es una posible solución. Sin embargo, usando esta técnica, la posición real de aterrizaje no se puede controlar debido a la existencia de viento y los obstáculos tales como árboles y edificios. En consecuencia, la cobertura puede ser inferior a los requisitos de la aplicación, no importa cuántos

sensores se dejen caer (Gajbhiye & Mahajan, 2008) o en algunas otras circunstancias, se puede presentar información redundante.

- **Tiempo de respuesta:** A pesar del bajo consumo de energía, los nodos deben ser capaces de comunicar mensajes de alta prioridad a través de la red tan rápido como sea posible. Si bien estos eventos serán poco frecuentes, pueden ocurrir en cualquier momento sin previo aviso. El tiempo de respuesta puede ser mejorado mediante la inclusión de los nodos que funcionan todo el tiempo. Estos nodos pueden escuchar los mensajes de alarma y las remitirá por backbone de enrutamiento cuando sea necesario. Esto, sin embargo, reduce la facilidad de la implementación del sistema (Hill, 2003).

- **Exactitud temporal:** Se refiere a la correlación de los datos de diferentes nodos para determinar la naturaleza del fenómeno que está siendo medido por medio de la red. Para lograr una precisión temporal, una red debe ser capaz de construir y mantener una base global de tiempo que puede ser usada para cronológicamente ordenar las muestras y eventos. La información de sincronización debe ser continuamente comunicada entre los nodos (Hill, 2003).

- **Seguridad:** Las WSN deben ser capaces de mantener la información importante y de carácter privado fuera del alcance de intrusos. Una posible solución es el uso de encriptación de la información, sin embargo, esto afecta la energía y el ancho de banda de la red (Hill, 2003). Además, el nodo debe tener capacidades de control de acceso, para prevenir intentos no autorizados, y también capacidades para detectar y prevenir cambios no autorizados en los mensajes (Potdar, et al., 2009).

- **Efectividad en las tasas de muestreo:** Ésta se define como la frecuencia de muestreo en la cual los datos sensados pueden ser tomados por cada nodo individual y comunicarlos a un punto de recolección en la red. Los diferentes tiempos de muestreo se definen dependiendo de la prioridad asignada a las variables que se requieran medir.

- **Eficiencia energética:** se entiende como el número de paquetes que se pueden transmitir con éxito utilizando una unidad de energía. La colisión de

paquetes, los costos del enrutamiento, la pérdida de paquetes y de la retransmisión de estos, reduce la eficiencia energética (Sohraby, et al., 2007).

Otras métricas de rendimiento incluidas en (Sohraby, et al., 2007) son:

- **Confiabilidad:** En WSN, la confiabilidad es usada como una medida para mostrar el grado de confiabilidad del evento sensado puede ser reportado al nodo sink. Para las aplicaciones que pueden tolerar la pérdida de paquetes, la confiabilidad puede ser definida como la proporción de los paquetes recibidos con éxito por el número total de paquetes transmitidos.

- **Conectividad:** Para WSN de múltiples saltos, es posible que la red se desconecte porque algunos nodos han dejado de funcionar. La métrica de conectividad puede ser utilizada para evaluar qué tan bien la red está conectada y/o la cantidad de nodos que han sido aislados.

- **Métricas de calidad de servicio:** Algunas aplicaciones en WSN tienen propiedades en tiempo real. Estas aplicaciones pueden tener requisitos de calidad de servicio tales como retraso, pérdida de relación y ancho de banda.

Capítulo 3

Estado del arte

En este capítulo se presenta la revisión del estado del arte, en la cual se incluye una revisión de los trabajos relacionados en el área de interés de esta tesis, integración de agentes de software en WSN, los métodos que han sido propuestos para tal fin y finalmente, con base a la revisión de los trabajos relacionados, los factores o variables de estudio más significativas para la evaluación del desempeño de una WSN.

3.1 Revisión de trabajos relacionados

Como se mencionó anteriormente, una de las estrategias abordadas para resolver algunos de los diferentes problemas que se presentan en el momento de la implementación de una WSN, son los agentes de software inteligentes. Debido a la flexibilidad, modularidad, aplicabilidad general a una amplia gama de problemas distribuidos, sus capacidades para adaptarse a los cambios dinámicos de la red y además de sus competencias para llevar a cabo tareas a partir de una información parcial y/o imprecisa, los agentes de software se han convertido en una de las alternativas exploradas para hacer frente a los desafíos y limitaciones que las WSN traen consigo.

Los agentes han sido desplegados anteriormente en WSN principalmente para procesar los datos sin tratamiento de forma inteligente con el fin de reducir las transmisiones de datos, el cual es el factor más importante en la determinación de la longevidad de la red. Otras aplicaciones han usado agentes para el enrutamiento de paquetes en la red y otras en el control de iluminación para la conservación inteligente de energía (Tynan, et al., 2005) .

Tynan, et al. (Tynan, et al., 2005), examinan algunas de las decisiones potenciales que un agente de software puede enfrentar en materia de la administración inteligente de la energía a través de la cooperación con otros agentes. Además, muestran cómo la noción más fuerte del

concepto de “agencia”, la cual se caracteriza por la noción de deliberación y es típicamente realizada a través del uso del paradigma BDI (Belief, Desire and Intention), podría emplearse para permitir un razonamiento más amplio respecto a decisiones potenciales. Sin embargo, no se realiza ningún análisis sobre los efectos en otros atributos de la red que están directamente relacionados con la administración de la energía, como son la cobertura, la latencia, la exactitud y la longevidad. Además, no se analiza cómo se sería su aplicación en dispositivos con recursos aún más limitados.

En un trabajo posterior realizado el mismo año, Tynan, et al. (Tynan, et al., 2005) presentan la interpolación como una técnica útil para la conservación de la energía en una WSN, a través de la activación inteligente de los nodos en la red con el uso de agentes. Para determinar si un nodo dado debe estar en modo sleep, se tiene en cuenta si la información que éste está adquiriendo es redundante, es decir, si los datos de este pueden ser determinados a través de una interpolación de los datos obtenidos por los otros nodos de la red, con determinada exactitud. El cálculo computacional de la interpolación, es distribuido entre tantos nodos como sea posible con el uso de los agentes. A pesar de las diferentes ventajas presentadas, solo se hace una presentación formal del esquema de administración de energía basado en agentes (DARWIN) sin incluir los resultados obtenidos o las pruebas de validación.

Otro trabajo presentado por estos autores (Tynan, et al., 2005), donde se propone una metodología para el rápido despliegue de un SMA para WSN, la cual se describe más adelante, no presenta la aplicación de ésta en dispositivos reales y por lo tanto no brinda la posibilidad de evaluar cómo será el comportamiento de un sistema real.

Min Chen, et al. (Chen, et al., 2006), proponen el uso de un paradigma de agentes móviles para el procesamiento, agregación y concatenación de datos en una arquitectura de red plana, eliminando así la redundancia de datos y el costo de la comunicación, especialmente en conexiones con bajo ancho de banda. La arquitectura propuesta es llamada “Mobile Agent based Wireless Sensor Network (MAWSN)”, cuyo rendimiento es comparado, por medio de una simulación, con el de la arquitectura tradicional basada en el paradigma cliente/servidor a través del uso de cuatro métricas de evaluación (consumo de energía, retraso promedio end-to-end, una métrica combinada (energía y retraso) y la proporción de entrega de paquetes). En los resultados presentados se puede ver claramente que MAWSN muestra un mejor rendimiento en términos de consumo de energía y en la proporción de entrega de paquetes,

sin embargo, tiene una mayor latencia end-to-end bajo ciertas condiciones. No obstante, el montaje físico proporcionaría información adicional, permitiendo así mejorar la arquitectura actual, debido a la necesidad de enfrentarse con las limitaciones físicas de los nodos.

En un trabajo posterior, M. Chen, et al. (Chen, et al., 2007), examinan las posibles aplicaciones de los agentes móviles en WSN, se mencionan algunas de las ventajas que estos le proporcionan a las WSN, entre las cuales se destacan la reducción potencial del consumo del ancho de banda y el gasto de energía. Además, se discuten los tópicos claves del diseño para tales aplicaciones. Se descompone la funcionalidad del diseño del agente en cuatro componentes: la arquitectura, la planificación de itinerarios, el diseño de sistemas de middleware y la cooperación de agentes.

Martínez, et al. (Martínez, et al., 2007), describen cómo los conceptos y fundamentos definidos para los sistemas multi-agente pueden ser utilizados en una WSN. Se presenta un modelo arquitectónico donde a través de la cooperación entre agentes y con un procesamiento inteligente de los datos, se reducen el número de transferencias de información necesarias en la red, mejorando el consumo energético. La validación de la arquitectura se realiza mediante una aplicación en un escenario de tracking (rastreo), donde cada nodo tiene asociado y es gestionado por al menos un agente, ya sea, un agentes de información, para la agregación y el filtrado de información; un agente autónomo, para garantizar a la red tolerancia frente a fallos, o agentes móviles, para reprogramar y localizar los nodos en los que se debe ejecutar algún cambio de funcionalidad. Sin embargo, solo se muestra el intercambio de mensajes realizado por los agentes y no como influye este finalmente sobre el tiempo de vida de la red, el ahorro en el consumo energético o el desempeño de la red en general. Además, no se hace claridad sobre el tipo de dispositivo físico (mote) utilizado en dicha validación.

Otros trabajos donde a pesar de no hacer uso de agentes inteligentes de software se realiza un análisis de las características y posibles soluciones para los desafíos de las WSN son presentados a continuación.

Ruzzelli, et al. (Ruzzelli, et al., 2006), muestra una descripción de los principales componentes de las WSN, las ventajas y desventajas de las transmisiones a través de múltiples saltos, las diferentes consideraciones que se deben realizar para un despliegue de nodos, entre las cuales se incluyen la cantidad de nodos requeridos, la administración de su energía y

la calibración de los sensores por los cuales están compuestos. Además, se resalta la capacidad que tienen los sensores para interactuar con PDAs (Asistente Digital Personal) y teléfonos móviles, permitiéndole a las WSN mejorar la supervisión de una actividad y comunicarse con un usuario de una manera más efectiva y eficiente. Sin embargo, se plantean varios retos en los aspectos relacionados con la posibilidad de una interacción fluida y transparente entre las WSN y los dispositivos de computación móvil, entre los cuales están la compatibilidad entre los protocolos de comunicación, el manejo de la confiabilidad de los datos, entre otros.

Gajbhiye, et al. (Gajbhiye & Mahajan, 2008), presenta una revisión bibliográfica de varias arquitecturas y algunas estrategias para el despliegue de nodos en las WSN, las cuales son dependientes de los requerimientos de la aplicación. Se describen los puntos claves de dichas estrategias, entre las cuales se incluyen el área de cobertura de los nodos, la búsqueda de los nodos "muertos", así como los nodos en los límites de la zona de cobertura de otros nodos. Sin embargo, no se presentan las ventajas y limitaciones de las diferentes estrategias abordadas.

3.2 Modelos de integración de agentes inteligentes con WSN

Algunos de los modelos planteados para la incorporación de agentes dentro de redes de sensores son descritos a continuación.

Hairong Qi, et al. (Qi, et al., 2003), presentan un modelo de computación basado en agentes móviles para CSIP (Collaborative Signal and Information Processing) en redes de sensores, donde a diferencia del modelo de computación tradicional basado en cliente/servidor, el código de procesamiento se mueve hasta la localización de los datos a través de agentes móviles. Entre las ventajas resaltadas de un modelo basado en agentes, que pueden suplir los desafíos de una red de sensores, se destacan la escalabilidad, la confiabilidad, la adaptabilidad a las tareas y la "conciencia" en la energía de los nodos, a través de la información obtenida en cada uno de éstos. Para evaluar el rendimiento entre ambos modelos, seleccionaron 2 métricas, tiempo de ejecución y consumo de energía, las cuales son evaluadas a través de una simulación y un estudio analítico, teniendo en cuenta parámetros como el número de nodos, el efecto del uso del radio (tiempo usado para leer y escribir un dato), el efecto del número de agentes móviles y de su tamaño. En los resultados obtenidos resaltan que el uso del modelo basado en agentes móviles es mejor si se tienen en cuenta que

el número de nodos en una WSN es significativamente alto, mejorando la eficiencia energética debido a una disminución en el tiempo de ejecución; además, con el procesamiento colaborativo entre múltiples nodos mejora la tolerancia a fallas y la perdida de información.

Marsh, et al. (Marsh, et al., 2004), plantean como a través de la incorporación de agentes inteligentes se puede dotar a una WSN de características autónomas (Autonomic Wireless sensor Networks - AWSN), en las cuales la meta principal son sistemas de software con el conocimiento necesario para su auto-mantenimiento y auto-administración, donde las características de autonomía y la naturaleza deliberativa de los agentes inteligentes son claves para proporcionar el ciclo deliberativo central requerido para dichos componentes autónomos. Estas características facilitan la administración de la energía de sistemas de cómputo con recursos limitados, como el caso de los nodos en una red, en este caso en particular, a través de la inserción de agentes en la red con diferentes frecuencias de muestreo y uno de ellos dotado con la capacidad de variar su frecuencia de transmisión cuando eventos inesperados ocurren. Teniendo en cuenta que el radio de un nodo sensor es usualmente el componente que más consume energía, se muestra una mejora significativa en este aspecto a través de la implementación de dicho método basado en agentes.

Tynan, et al. (Tynan, et al., 2005) proponen una metodología para el rápido despliegue de un SMA para WSN que permite la verificación (realización de pruebas y depuración) en un computador antes de realizar la inserción de los agentes en los dispositivos de la red que poseen menos recursos. La metodología consiste de 3 fases, con las cuales, a través del uso de agentes inteligentes bajo la noción de agencia débil (agentes netamente reactivos), se pretende tomar ventaja de las capacidades de procesamiento agregado de las WSN y minimizar el número de transmisiones a través del procesamiento inteligente local de los datos. En la primera fase se busca ubicar un único agente sobre la estación base (solución centralizada), donde éste se encarga de recibir los datos sensados de la red, y en los nodos no se realizan ningún tipo de procesamiento inteligente. Además, en esta primera etapa se busca identificar bien el problema. La segunda fase transforma la anterior solución centralizada en una implementación basada en agentes distribuida pero sobre la misma estación base, asignándole un agente a cada nodo de la red (mapeo uno a uno). En este caso la comunicación (envió y recepción de datos y/o comandos) se da entre los nodos y su agente correspondiente sobre la estación base. Finalmente, en la tercera fase, teniendo en cuenta que en las etapas

anteriores se espera que se haya logrado un grado de confianza, debido a la depuración realizada, se inserta el agente correspondiente en su respectivo nodo (mapeo uno a uno).

En (Piedrahita Ospina, et al., 2009), se presenta un modelo de integración de agentes móviles inteligentes en WSN buscando resolver algunas limitaciones presentes en dichas redes, dotándolas de agentes móviles con capacidades como la migración, la clonación, el suicidio, la mutación y la difusión por inundación, característica que es analizada a través de métricas, mostrando un mejoramiento en el tiempo de convergencia de la red cuando el número de nodos de ésta aumenta considerablemente. Sin embargo, se muestra un alto costo en cuanto a la eficiencia del proceso de difusión en relación al consumo de energía de los nodos de la red debido al proceso de clonación. Finalmente, los resultados mostrados indican que la eficiencia es independiente del tamaño de los agentes y en su lugar depende de la topología y densidad de los nodos. Mientras que el tiempo de convergencia si es proporcional al tamaño de los agentes.

3.3 Factores más significativos para la evaluación del desempeño de una WSN identificados a partir de la revisión del estado del arte

Con base a la revisión del estado del arte realizada en la sección 3.1, los criterios más relevantes al momento de evaluar el desempeño de una WSN y en particular de una WSN con agentes inteligentes de software son el consumo de energía de los nodos sensores (Tynan, et al., 2005) (Chen, et al., 2006) (Martínez, et al., 2007) (Ruzzelli, et al., 2006), los costos de la comunicación (Chen, et al., 2007) y la redundancia en la información (Tynan, et al., 2005) (Ruzzelli, et al., 2006) (Gajbhiye & Mahajan, 2008), con base a los cuales se puede proceder a realizar una primera selección de las métricas a utilizar para la evaluación del desempeño, ya que la selección de otras métricas puede depender directamente de la aplicación a realizar como se explicara más adelante.

Para evaluar el consumo de energía de los nodos sensores y el gasto debido a las comunicaciones inalámbricas realizadas entre ellos se podría hacer uso de una métrica de eficiencia energética y a su vez, evaluar la métrica de tiempo de vida de la red que estará directamente relacionada con ambos factores y además, es una variable determinante en el desempeño de una red. Por su parte, la redundancia de la información se debe a la posible

superposición del área de sensado de los diferentes nodos sensores desplegados en el ambiente, especialmente en aquellas áreas donde debido a las condiciones ambientales no es posible realizar un despliegue manual, las métricas de cobertura y de despliegue deberán ser tenidas en cuenta durante el análisis de desempeño de la red. Se debe considerar que el despliegue es una función directa de la posición de cada uno de los nodos y la cobertura es función del despliegue.

Como se mencionó inicialmente, otras métricas de evaluación pueden ser seleccionadas con base a la aplicación a realizar y a las necesidades del usuario.

Capítulo 4

Método propuesto basado en métricas para realizar el seguimiento y la evaluación de la integración de agentes inteligentes en WSN

Como se mencionó anteriormente, la mayor parte de la investigación en el campo de las WSN se enfoca en la optimización de las capacidades energéticas de sus componentes (nodos) debido a que éstos a menudo deben recopilar datos durante largos períodos de tiempo y se utilizan en ambientes inhóspitos, donde cambiar las baterías es un inconveniente o un hecho imposible (Rogers, et al., 2009). Además, de esta forma, se consigue maximizar el tiempo de vida de la red completa.

Diferentes estrategias han sido planteadas con el objetivo de cumplir dicha meta, entre ellas la utilización de agentes inteligentes, como se ha descrito hasta ahora, ya que a pesar de que las WSN se pueden construir para diferentes aplicaciones, la red se diseña con base a aquellos eventos que se conocen de antemano, lo cual no deja espacio para nuevos comportamientos que pudieran solventar futuras eventualidades (Piedrahita Ospina, et al., 2009) y es bajo este contexto donde los agentes se han convertido en una solución promisoria. Sin embargo, se debe tener en cuenta que bajo ciertos escenarios su utilización seria inapropiada, considerando el costo adicional que tiene su utilización sobre la red. A continuación se plantea un método que permite la realización de un proceso organizado y detallado para evaluar el desempeño de dicha integración, agentes inteligentes + WSN, a través de algunas de las métricas definidas en el capítulo 2.4, en el cual se plantean las siguientes 4 etapas para su desarrollo:

- Etapa 1: Conceptualización
- Etapa 2: Análisis y Diseño
- Etapa 3: Implementación y Medición del desempeño
- Etapa 4: Evaluación de resultados

4.1 Etapa 1: Conceptualización

En esta etapa se pretende tener un primer acercamiento al problema y se definirá el alcance de la aplicación a realizar. Se identificaran las características de la red, como topología, frecuencia de muestreo de los nodos sensores, frecuencia de transmisión, etc.; además, el tipo de agentes que intervendrán en dicho proceso, los cuales pueden ser humanos o agentes de software, con el fin de alcanzar una meta en particular. Esto permitirá realizar un primer acercamiento basado en diagramas de casos de uso, los cuales ayudaran a visualizar los pasos que se llevan a cabo.

4.1.1 Descripción de la aplicación

Realizar una descripción detallada, en lenguaje natural (forma textual), del problema o aplicación que se desea abordar y definir de una manera clara y precisa los objetivos de la misma y su alcance. Además, en esta fase es preciso determinar la o las variables físicas a sensar (temperatura, luminosidad, presión, etc.) con base al objetivo de la aplicación, asimismo plantear las posibles limitaciones para su realización y proponer algunas soluciones factibles para las mismas.

4.1.2 Características de la red

Determinar cuál será la mejor topología de red para la aplicación a realizar teniendo en cuenta los definido en la sección 2.2.1 y la descripción detallada efectuada en la fase anterior, además, establecer los parámetros de la adquisición y transmisión de datos (frecuencia de muestreo y frecuencia de transmisión respectivamente), igualmente, definir cuál será el modelo de tráfico de estos, si mediante la entrega basada en eventos, la entrega continua, la entrega por peticiones o la entrega híbrida. También, conociendo los dispositivos mote con los que se cuenta y sus características, ver Tabla 8 y sección 2.2.2, determinar con cuáles de ellos sería factible trabajar. De igual manera, teniendo en cuenta que el alcance de los enlaces de comunicación inalámbrica de los nodos no determinan el rango de cobertura de la red, la cual se puede ampliar mediante el uso de técnicas de múltiples saltos, establecer si esta técnica es requerida o no.

4.1.3 Identificación de actores

Durante esta fase de la etapa de conceptualización, se identifican las entidades que hacen parte del sistema así como sus objetivos, tareas e interacciones, permitiendo de manera gráfica o textual mostrar las entidades, tanto humanas como de software, que intervendrán en él. Una de las técnicas usadas para esto son los casos de uso, los cuales describen las posibles interacciones o usos de un usuario con el sistema. Los usuarios del sistema son denominados actores y representan a las entidades externas al mismo (Iglesias Fernández, 1998). El empleo de casos de uso permite obtener los requisitos de los usuarios y generar casos de prueba.

Las etapas para un análisis mediante casos de uso, descritas en (Iglesias Fernández, 1998), constan de los siguientes pasos:

- Identificación de los actores y descripción de los actores
- Identificación de los casos de uso: Preguntas tales como cuáles son las principales tareas o funciones realizadas por cada actor, cual es la información que éste adquiere del sistema o cual cambia, entre otras, son pertinentes para el desarrollo de este segmento.
- Descripción de los casos de uso: Pueden describirse empleando lenguaje natural a través de plantillas (Ver Tabla 3) o por medio de notación grafica (ver Figura 7).

Tabla 3. Plantilla propuesta para realizar la descripción textual de un caso de uso (Iglesias Fernández, 1998)

CASO DE USO (nombre)	
RESUMEN	
ACTORES QUE INTERVIENEN	
DESCRIPCION	
PRECONDICION	
POSTCONDICIONES	
EXCEPCIONES	

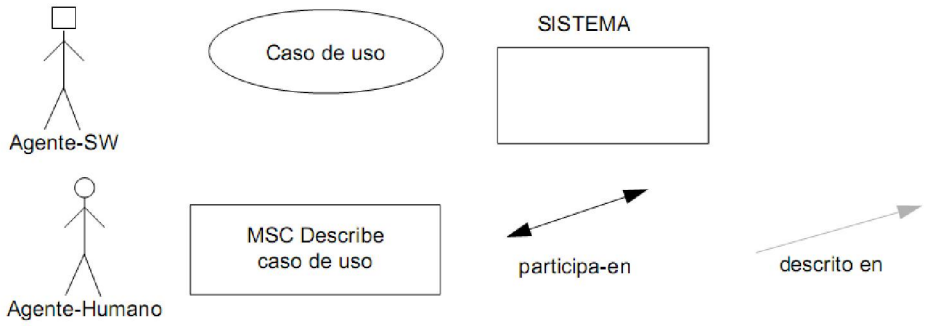

Figura 7. Notación grafica de los casos de uso (Imagen tomada de (Iglesias Fernández, 1998))

En esta fase también se delimitaran las funciones de los agentes dentro de la red, tanto humanos como de software, la interacción entre ellos y sus objetivos específicos.

4.1.4 Agentes de software

En caso de requerir agentes de software se debe definir el número y el tipo de agentes a usar, haciendo claridad sobre su arquitectura ya sea reactiva, deliberativa o híbrida (ver sección 2.1.1.2). Además, determinar si se tendrá un sistema basado en agentes o un SMA (sección 2.1.5). En el caso de un SMA, se debe establecer que metodología se va a emplear para su desarrollo (sección 2.1.4) y para ambos casos, sistema basado en agentes o SMA, se debe seleccionar el framework para el desarrollo de agentes y el lenguaje que utilizaran éstos para la comunicación entre ellos (sección 2.1.6). Se debe tener en cuenta que lo descrito para esta fase depende del tipo de aplicación a realizar y sobre todo del dispositivo sensor que se usara y por lo tanto, no se puede proceder a la creación de dichos agentes si no se han desarrollado en forma correcta las fases anteriores debido a las restricciones ya descritas de estos dispositivos.

4.2 Etapa 2: Análisis y Diseño

Durante esta etapa se determinaran los requisitos del sistema, entre cuales se encuentran la selección de la plataforma hardware a utilizar, el framework para el desarrollo de los agentes de software (en caso de ser requeridos) y en general las herramientas que serán utilizadas

para llevar a cabo la implementación. Además, se especificaran las métricas a ser usadas para el análisis.

4.2.1 Selección de la plataforma hardware

En la etapa de conceptualización se realizó un primer acercamiento general a los requerimientos de la aplicación a realizar, además, se determinaron las variables que se pretenden sensar dentro del entorno y los posibles dispositivos disponibles para ello. Con base a esto, en esta fase, se debe seleccionar el dispositivo mote a utilizar y realizar una búsqueda bibliográfica detallada de sus características (condiciones típicas de operación, sensores incorporados o posibilidades de expansión, alcance de transmisión, sistema operativo, capacidad de memoria, etc.). Además, dicha selección debe estar acorde con la necesidad de usar agentes de software dentro de la red, ya que no en todos los dispositivos esto será posible debido a sus capacidades.

4.2.2 Herramientas utilizadas para llevar a cabo la implementación

Dentro de esta fase se deben determinar cuáles serán las herramientas utilizadas para el desarrollo de la implementación de la red, tanto las requeridas para la programación de los nodos sensores como para el desarrollo de los agentes inteligentes (sección 2.1.6). Además, en esta fase se determinara bajo cual sistema operativo para WSN se realizara la aplicación (TinyOS, Mantis, Contiki, etc.), el cual juega un papel fundamental en el desarrollo de la misma (sección 2.3) y también se debe proceder a la selección de un modelo de integración de agentes inteligentes y WSN (sección 3.2) que permitan la realización de dicha unión y asimismo llevar a cabo la validación y aplicación del método propuesto.

4.2.3 Selección de Métricas

En esta fase se identificaran las variables relevantes del problema, es decir, aquellos factores que se consideren significativos o que requieran ser analizados durante el despliegue de la red y/o durante el tiempo de funcionamiento de la misma, ya que pueden conducir a un mejoramiento de ésta o simplemente porque se requieren estudiar para efectos de la aplicación que se está realizando. Además de la elección de las métricas a utilizar con base a las variables identificadas, en esta fase se deben clarificar las características de cada una de

ellas y con base a esto diseñar un experimento que me permita realizar las mediciones requeridas.

Para evaluar en forma cuantitativa o cualitativa el desempeño del sistema y determinar el impacto por ejemplo de la inserción de agentes en la WSN, que es el objetivo particular de este método propuesto, se suele hacer uso de métricas como un criterio o indicador para realizar dicha evaluación, las cuales se deben seleccionar con base a las variables o factores identificados como relevantes para el estudio de la red, como se mencionó anteriormente. Las métricas para evaluar el desempeño de una WSN son descritas en las secciones 2.4 y 3.3 y pueden ser seleccionadas dependiendo del objetivo de la aplicación y de la o las características que se quieran evaluar dentro de la WSN como se ha descrito hasta ahora. Además, teniendo en cuenta que la utilización de agentes influirá significativamente en el desempeño de la red, la selección de las métricas a emplear también debe realizarse en función de éstos.

Se debe tener en cuenta, además, que dependiendo de la aplicación y del objetivo como tal de ésta, no todas las métricas al momento de ser evaluadas arrojaran un valor "significativamente alto" dentro de la escala fijada, ya que como se ha explicado, cuando se decide maximizar el valor de una de las variables es posible que esto afecte de manera "negativa" a otras, debido a la posible dependencia entre ellas. Por ejemplo, la métrica de cobertura puede ser mejorada implementando técnicas de comunicación de múltiples saltos, sin embargo, esto podría afectar el consumo de energía de los nodos, especialmente de los más cercanos a la estación base, los cuales deben acarrear un mayor número de datos (Muñoz, et al., 2010) y por lo tanto realizar un mayor número de transmisiones.

Otras métricas a tener en cuenta considerando el uso de los agentes de software son:

- Rapidez: Se puede definir como el tiempo empleado en completar una tarea por parte de los agentes en la red; ya sea el tiempo empleado en adquirir los datos de toda la red (desde el nodo más cercano a la estación base hasta el nodo más lejano a ésta) o la reprogramación de tareas todos los nodos. Para la reprogramación, los agentes de software móviles pueden realizarla ya sea migrando a través de la red o por medio de un proceso de difusión por inundación.

- Escalabilidad: La escalabilidad es una métrica que se puede abordar de diferentes maneras, permitiendo en todos los casos encontrar las condiciones mínimas en las que el sistema puede operar. Con ésta se podría determinar el número de nodos mínimo para garantizar la cobertura de la red en una aplicación dada y como es su comportamiento a medida que el número de éstos aumenta. También, se puede hacer uso de esta métrica para estudiar el número de agentes viables en el sistema o como es el comportamiento del mismo a medida que aumenta del número de tareas realizadas por estos agentes, teniendo en cuenta las limitaciones de los nodos de la red.

- Completitud: Se puede definir como la capacidad de un agente de generar un cambio o no en el entorno ante un evento. Esta métrica podría tomar dos valores: 0, cuando no hay cambio ante un evento o 1, cuando hay cambio ante un evento.

- Latencia: La latencia o tiempo de respuesta del sistema en este caso en particular, se define como el tiempo que se demora el agente en realizar un cambio en la frecuencia de transmisión o de muestreo después de detectar un cambio en el estado del ambiente.

Como se puede observar se han presentado un gran número de alternativas en cuanto a las métricas que se podrían utilizar para evaluar el desempeño de la red, lo cual podría acarrear con algunas dudas al momento de su elección. Sin embargo, como se ha enfatizado hasta ahora, su selección depende exclusivamente de la aplicación y/o de las necesidades del usuario, de las características que éste quiera evaluar y/u optimizar dentro del sistema.

A continuación se presenta un listado de posibles criterios que podrían ser utilizados para realizar la selección de las métricas a utilizar para evaluar el desempeño de la red, los cuales se han categorizado desde 4 enfoques: Energético, prioridad de los mensajes, adaptabilidad de la red y finalmente el número de nodos y de agentes requeridos.

a. Enfoque energético: Si se requiere evaluar el consumo de la red y determinar el costo de diferentes actividades realizadas por cada uno de los nodos sensores o del sistema como un todo, se recomienda hacer uso de la métrica de eficiencia de energía, la cual permitirá determinar el número de paquetes que se pueden transmitir con éxito

utilizando una unidad de energía, la cual puede ser afectada por la pérdida o colisión de los mensajes. También, se recomienda evaluar la métrica de tiempo de vida de la red ya que además de ser un factor determinante en el sistema, depende directamente del suministro de energía de los nodos.

b. Prioridad de los mensajes: En el caso de aplicaciones donde se requiera el envío de mensajes de alta prioridad, ya que de éstos, por ejemplo, podrían depender vidas, evaluar la métrica de tiempo de respuesta de la red es vital. Además, la métrica de seguridad debe ser evaluada en caso de que los mensajes contengan información importante y de carácter privado. Otra característica a evaluar es la efectividad en las tasas de muestreo ya que dependiendo de la prioridad de las variables sensadas, se requerirá una tasa de muestro mayor (mayor número de datos sensados por unidad de tiempo), con el fin de garantizar la seguridad. Además, con el objetivo de prevenir falsas alarmas, la métrica de confiabilidad puede ser empleada.
c. Adaptabilidad de la red: Si se requiere de una red que se adapte a las necesidades del usuario, como se verá más adelante, el uso de agentes de software se hace necesario y en este caso, para evaluar la efectividad de la respuesta al cambio ante diferentes eventos, la métrica de completitud podría ser utilizada. Además, la aplicación de las métricas de rapidez y latencia permitirían evaluar el tiempo empleado en la realización de dichas tareas.
d. Número de nodos y de agentes: Si se desea saber el número mínimo o máximo de agentes que pueden estar dentro de cada nodo o en la WSN, o el número máximo de nodos para una aplicación determinada, una métrica de escalabilidad será necesaria.

En esta fase, además de realizar la selección de las métricas a usar, para cada una de ellas deben quedar bien definidos los parámetros bajo las cuales se analizaran, sus unidades, escalas, etc. Además, se sugiere plantear un protocolo para realizar las mediciones, ya que algunos sistemas pueden presentar sensibilidad ante diferentes factores al momento de realizar varias tomas de datos de la misma variable que podrían alterar los resultados.

4.3 Etapa 3: Implementación

Haciendo uso de las características y especificaciones encontradas en las etapas anteriores y además de las métricas seleccionadas con base a los criterios de evaluación establecidos, en

esta etapa se lleva a cabo la implementación de casos de estudio realizando el montaje de la topología de red propuesta, desarrollando los agentes de software necesarios y llevando a cabo las pruebas necesarias para medir el desempeño de la red a través de la utilización de las métricas seleccionadas, cuyos resultados deben ser almacenados para su posterior procesamiento. Asimismo, con base a la fase 4.1.4, se procede a implementar de la metodología para el desarrollo de SMA en el caso de requerir hacer uso de éstos y/o a determinar las características de los agentes, que hacen, sus objetivos, cómo interactúan, etc. cuando se requiere solo un sistema basado en agentes. Para cumplir con todo esto, en esta etapa, se hace necesaria una descripción detallada del prototipo utilizado, sus componentes, etc.

Para el desarrollo del experimento se puede recurrir a diferentes técnicas del diseño de experimentos, el cual estudia como variar deliberadamente las condiciones habituales de un proceso para aumentar la probabilidad de detectar cambios significativos en la respuesta y obtener así un conocimiento más profundo sobre el comportamiento del proceso. Se debe tener en cuenta que se debe garantizar la aleatorización, la réplica del experimento y el bloqueo de los factores que no se deseen analizar, además de los niveles de éstos.

4.4 Etapa 4: Evaluación de resultados

Finalmente, en esta etapa de evaluación, se presenta un resumen de los resultados obtenidos con la aplicación del método propuesto y se procede a realizar el análisis de los datos recolectados para cada una de las métricas empleadas, para lo cual, previamente se deben tabular u organizar en plantillas. Además, haciendo uso de gráficos y/o de diferentes métodos estadísticos para su tratamiento, se puede extraer la máxima cantidad de información relevante contenida en ellos. Entre los métodos estadísticos que se pueden usar se encuentra el proceso de regresión lineal y cuadrática, hasta una simple media y una desviación estándar.

Capítulo 5

Validación del método propuesto a través de casos de estudio

En este capítulo se presentará la validación del método propuesto a través de casos de estudio. Como se describió en el capítulo 4, el método propuesto permite, a través de la realización de un proceso organizado y detallado, evaluar el desempeño de la integración de agentes inteligentes de software y WSN. Sin embargo, este método también podrá ser aplicado en casos de estudio que no requieran de agentes de software en su implementación, como se mostrará en el caso de estudio 1, Medición de la potencia de un nodo sensor, y en el caso de estudio 3, Medición del tiempo de vida de una WSN sin agentes de software.

En el caso de estudio 2 se realizará la Medición del tiempo de vida de una WSN dotada con 2 agentes de software reactivos de características diferentes. Además, se realizará una comparación entre los casos de estudio 2 y 3 con el objetivo de evaluar el beneficio de la utilización de agentes inteligentes de software en una WSN.

5.1 Caso de estudio 1: Medición de potencia de un nodo sensor

- **Etapa de conceptualización**

a) **Descripción de la aplicación**:

Este caso de estudio tiene como objetivo el desarrollo de un método para la medición de la potencia de un nodo sensor recolector durante su tiempo de vida y validarlo mediante la comparación de resultados con un sistema de referencia que permita realizar lo mismo.

b) **Características de la red**:

Con el fin de cumplir dicho objetivo, se podría emplear en un principio cualquiera de los dispositivos con los cuales se cuenta en el laboratorio (mica2, tmote sky o mica2 dot). Además, como el objetivo es medir la potencia consumida por el nodo, la variable que este sensará durante las mediciones realizadas no es relevante, sin embargo, se debe tener en cuenta cuales sensores estarán en uso ya que de ellos también dependerá el consumo del sistema. No se requerirá el uso de agentes de software y se implementará una topología punto a punto (estación base – nodo sensor), donde se tendrá una entrega continua de los datos por parte el nodo sensor a la estación base, con una frecuencia de muestreo fija e igual a la frecuencia de transmisión.

Cabe aclarar que la estación base estará conectada a un computador y por lo tanto tendrá alimentación eléctrica permanentemente y además, le enviará a éste por puerto serial todos los datos recolectados. A través del computador también se dará inicio a la toma de datos y se configuraran las características de la red. La medición de la potencia será para el nodo sensor recolector, el cual se encarga de sensar los datos y enviarlos a la estación base.

c) **Identificación de actores**:

En este caso de estudio con el sistema solo intervendrá un actor llamado usuario y será el encargado de la programación de los nodos, de la recuperación y análisis de la información obtenida en el proceso.

d) **Agentes de software**:

Como se mencionó anteriormente, en este caso de estudio no será requerido el uso de agentes de software. Sin embargo, la metodología propuesta para realizar la medición de la potencia de un nodo sensor sin agentes puede ser empleada también en un nodo que haga uso de ellos.

- **Etapa de análisis y diseño**

a) **Selección de la plataforma hardware**:

Para realizar la medición de la potencia consumida de un nodo sensor se utilizará un un nodo Tmote Sky como referencia, sin embargo, como se especificó anteriormente,

cualquier nodo podría ser usado en principio desde que se pueda encontrar un sistema de referencia o de control que permita posteriormente, a través de una comparación, validar el proceso de medición propuesto.

A continuación se describen más detalladamente las características de la plataforma Tmote Sky.

La plataforma Tmote Sky fue desarrollada originalmente en la Universidad de Berkeley (Moteiv Corporation, 2006) (MEMSIC Inc., 2010) y es un módulo inalámbrico de bajo consumo de potencia que se usa para diferentes aplicaciones de WSN. Éste opera bajo el estándar IEEE 802.15.4/Zigbee, diseñados para aplicaciones que requieren comunicaciones seguras, con baja tasa de envío de datos y maximización de la vida útil de las baterías. El Tmote Sky integra sensores de humedad, temperatura y luminosidad, además, tiene integrado un chip de radio CC2420 para las comunicaciones inalámbricas y la antena; trabaja con un microcontrolador de la Texas Instruments MSP430 a 8MHz y consume 3mW de potencia (0,375 nJ/instrucción) en modo activo y 6μW de potencia en modo sleep (Potdar, et al., 2009) (Baronti, et al., 2007); 10 kB de RAM, 48 kB de flash y 128 B para almacenar información. Tiene un rango máximo de alcance entre 75-100 m; se alimenta por medio de 2 pilas AA (2,1 V – 3,6 V DC) y se puede programar con el sistema operativo TinyOS (Hill, 2003), el cual fue diseñado explícitamente para redes de sensores, como se verá más adelante. Los Tmote Sky tienen un rango de direccionamiento efectivo medio, una capacidad computacional media y una capacidad de almacenamiento media-alta en comparación con otros motes (Potdar, et al., 2009). En la Tabla 4 se muestran las condiciones típicas de operación de un Tmote Sky (Figura 8)

Figura 8. Tmote Sky (Sentilla o formalmente Moteiv)

Tabla 4. Condiciones de operación típicas de un Tmote Sky

	MIN	NOM	MAX	UNIDADES
Voltaje de alimentación	2,1		3,6	V
Voltaje de alimentación durante la programación de la memoria flash	2,7		3,6	V
Temperatura de operación	-40		85	°C
Corriente consumida: MCU on, Radio RX		21,8	23	mA
Corriente consumida: MCU on, Radio TX		19,5	21	mA
Corriente consumida: MCU on, Radio off		1800	2400	µA
Corriente consumida: MCU idle, Radio off		54,5	1200	µA
Corriente consumida: MCU standby		5,1	21,0	µA

b) **Herramientas para llevar a cabo la implementación**:

Como se planteó en la etapa de conceptualización, el objetivo es realizar la medición de la potencia consumida de un nodo sensor durante su tiempo de vida, en este caso particular. Para dar cumplimiento al objetivo propuesto, se hará uso del sistema operativo **Contiki** y de la interfaz de recolección de datos **CollectView** provista por sus desarrolladores dentro de la máquina virtual Instant Contiki. Ésta se utilizará como sistema de referencia o de control para la recolección de datos de un nodo Tmote Sky (Contiki, 2012) y será con los datos adquiridos a través de ella con los cuales se efectuará la validación del proceso de medición propuesto.

CollectView se encuentra diseñado específicamente para nodos Tmote Sky, esta implementado en java y recibe como entrada los datos por serial de la estación base, los procesa y los despliega de manera gráfica. Además, también permite el envío de comandos al nodo sensor que se encuentra recolectando datos del ambiente por consola a través de la estación base.

En el CollectView, como se mencionó anteriormente, es posible visualizar gráficamente la información enviada por los motes (nodos sensores recolectores) a la estación base, tanto la información de cada uno de los sensores por los cuales están conformados como información relativa al consumo de potencia y al estado de sus baterías (pilas). Además, a través del CollectView se dará inicio a la recolección de

datos (se le comunica a los nodos recolectores que comiencen a sensar datos y a enviarlos a la estación base), también se puede detener la recolección de los mismos y configurar el periodo de muestreo. Para el caso específico de la plataforma Tmote Sky, se tendrá acceso a datos de los sensores de temperatura, luminosidad y humedad relativa. Además, en el CollectView se podrán visualizar graficas del consumo promedio e instantáneo, voltaje de las baterías de cada nodo, etc.

Como se ha descrito hasta ahora, como referencia para la validación del proceso de medición propuesto para la potencia consumida de un nodo, se tomaran los datos presentados por la aplicación CollectView de Contiki. Para realizar las mediciones de potencia experimentales se hará uso del entorno de desarrollo **LabView** y de una tarjeta de adquisición de datos de **NI USB-6009**, la cual permite adquirir datos para variables análogas y/o digitales. En LabView se realizó una aplicación que permite manipular los datos obtenidos, desplegarlos gráficamente y almacenarlos en un archivo de texto para su análisis posterior.

Aunque en el Collect View es posible visualizar gráficamente la información recibida por la estación base proveniente de los nodos sensores, a partir de la gráfica obtenida no es posible extraer de manera automática los valores de las abscisas ni de las ordenadas para su análisis y por lo tanto es indispensable su digitalización. Para cumplir con este objetivo se hizo uso del programa **GetData Graph Digitizer** (Anon., 2012), el cual permite de una manera muy sencilla a partir de una imagen su digitalización, obteniendo las coordenadas (x,y) de cada uno de puntos de la gráfica de interés.

Posteriormente, para graficar y analizar los datos obtenidos tanto de LabView como del GetData Graph Digitizer, se utilizó el software **OriginPro** (OriginLab Corporation, 2012), el cual incluye métodos estadísticos, procesamiento de señales y diferentes herramientas matemáticas, desde operaciones básicas hasta integración y diferenciación, para realizar el análisis de datos.

c) **Selección de métricas**:

En este caso particular no se hará el uso de métricas, ya que para el caso de estudio propuesto solo se requiere verificar el proceso de medición de potencia presentado, el

cual sí podrá ser usado posteriormente para ayudar a medir en otros casos de estudio algunas métricas, como por ejemplo, la métrica de eficiencia energética.

- **Etapa de implementación**

Inicialmente, como se describió en la etapa de análisis y diseño, se realizará la medición de potencia en un nodo de prueba Tmote Sky, en el cual se instaló Contiki a través de la interfaz CollectView, la cual además de permitir la programación de los nodos permite la visualización gráfica de la información de cada uno de estos.
Tanto el CollectView como el instrumento virtual desarrollado en LabView se ejecutaron simultáneamente y con el fin de validar la metodología propuesta para la medición de la potencia en un mote, se compararan las áreas bajo la curva de ambas graficas de Potencia vs Tiempo en el software OriginPro.

Para realizar las mediciones de potencia con la tarjeta de adquisición de datos NI USB-6009, es necesario conectar una resistencia en serie entre las baterías del nodo, para lo cual se puede hacer uso de un adaptador de pilas comercial. En la Figura 9 se muestra una representación gráfica de la conexión, donde las líneas rojas representan uniones eléctricas en el sistema y los recuadros negros muestran los puntos donde se deben realizar las mediciones de voltaje, tanto en la resistencia (V_r) como para el sistema total (V). Conociendo el valor de la resistencia, R, es posible calcular la corriente, I, que circula por el sistema y de esta forma encontrar el valor de la potencia consumida por el mote con base a la Ecuación 6.

Ecuación 6. Potencia consumida en el mote

$$P_{mote} = I(V - V_r) = \left(\frac{V_r}{R}\right)(V - V_r)$$

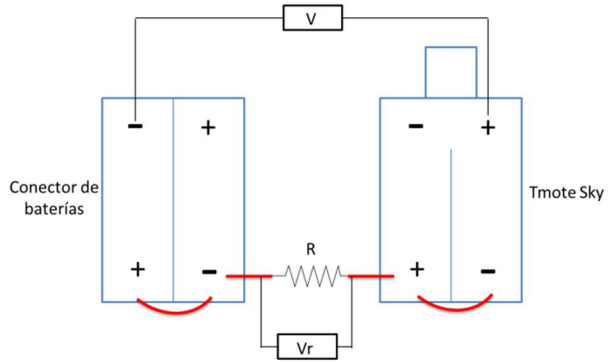

Figura 9. Esquema de conexión para la medición de la potencia consumida por un nodo Tmote Sky.

En la Figura 10 se muestra el esquema de la aplicación realizada en LabView. Los datos adquiridos por la tarjeta de adquisición de datos, V_r y V, son llevados a un instrumento virtual que permite la creación de fórmulas matemáticas, donde es ingresada la Ecuación 6, cuyo resultado, P_{mote}, es graficado y almacenado junto con los valores de voltaje V_r y V, después de ser comprimidos por un método de reducción de media en un factor de 10, debido a la gran cantidad de datos que se obtienen en un intervalo de tiempo reducido. Cabe resaltar que a pesar de realizar una reducción no se perderán datos, ya que en LabView, debido a sus capacidades, se realiza un sobre-muestreo de la potencia consumida por el nodo.

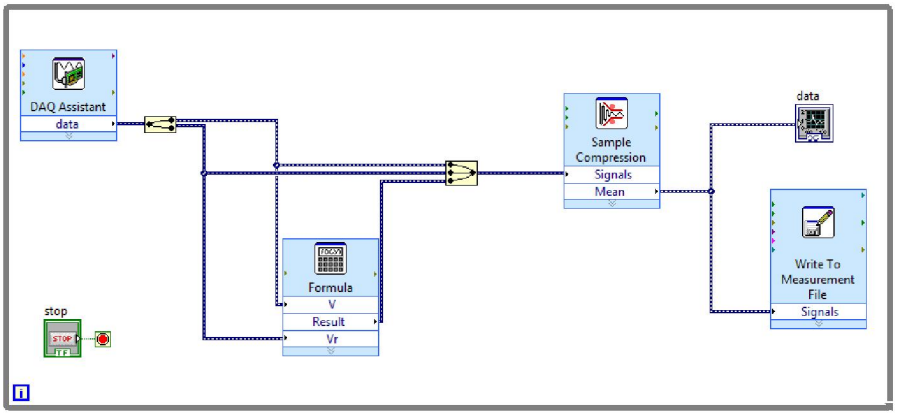

Figura 10. Programa de adquisición de datos realizado en LabView

- **Etapa de evaluación de resultados**

La Figura 11 y Figura 12 muestran las gráficas realizadas en OriginPro de los datos obtenidos tanto en CollectView y en LabView para la potencia consumida por un nodo Tmote Sky en el tiempo (Cantidad intercambiada de energía ΔE ($Joule$)).

Para calcular el área bajo la curva de ambas gráficas se hizo uso de la herramienta de análisis matemático "Integrate" proporcionada por OriginPro, con la cual, para los puntos obtenidos en el Collect View, arroja un área $A_1 = 3{,}2184546471\,\mathrm{J}$ y para los datos obtenidos en LabView un área $A_2 = 3{,}4907536858038\,\mathrm{J}$.

Si consideramos los datos obtenidos en el Collect View como los valores convencionalmente verdaderos, tenemos un $\%E = 8{,}46\,\%$ respecto a la metodología propuesta, el cual, teniendo en cuenta que el Collect View hace un análisis de consumo de potencia con base a una estimación obtenida de una herramienta de software llamada Energest, en la cual algunas características de un nodo real pueden quedar por fuera, los valores obtenidos con las herramientas propuestas se encuentran dentro de los rangos experimentales permitidos y se puede garantizar que el método propuesto para realizar la medición de potencia de un nodo sensor es correcto y se podría implementar para realizar la medición de la métrica de eficiencia energética en otro caso de estudio en caso de ser requerido.

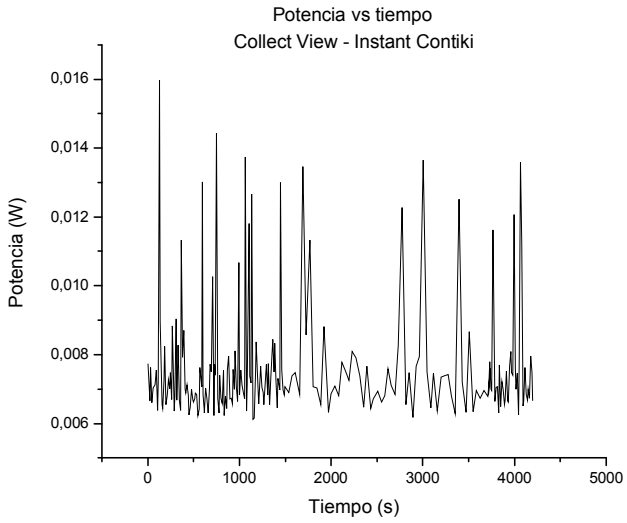

Figura 11. Potencia vs tiempo (Collect View - Instant Contiki)

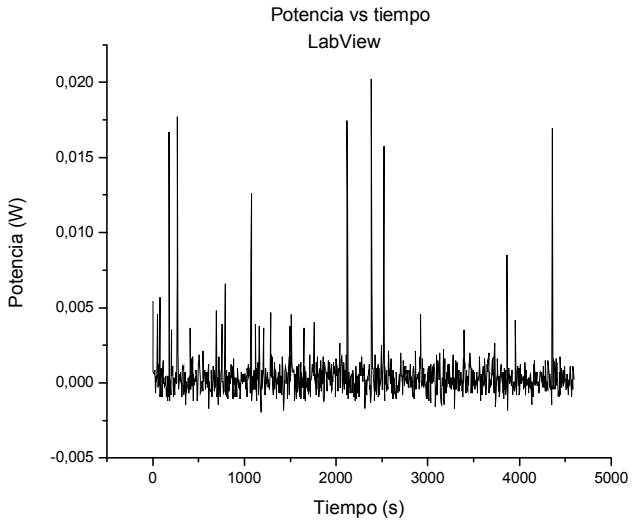

Figura 12. Potencia vs Tiempo (LabView)

5.2 Caso de estudio 2: Medición del tiempo de vida de una WSN con agentes de software

- **Etapa de conceptualización**

a) **Descripción de la aplicación**:

En este caso de estudio se tiene como objetivo evaluar el tiempo de la vida de un nodo sensor comercial con agentes de software reactivos, uno de ellos sin ningún tipo de capacidades deliberativas y otro con la capacidad de variar sus características de transmisión de datos con base a lo sensado del entorno. Se realizaran mediciones de luminosidad en el ambiente y además se obtendrá el valor del voltaje de las baterías de los nodos sensores de la red.

Como se ha explicado anteriormente, los nodos están compuestos por 4 unidades básicas, la unidad de sensado, la unidad de procesamiento, la unidad de energía y la unidad de comunicación, siendo esta última la más significativa en cuanto al consumo de energía en el mote, es por este motivo que la mayoría de esquemas para la administración de la energía se basan en la variación de la frecuencia de transmisión y la activación del nodo (paso del modo activo al modo sleep y viceversa) (Tynan, et al., 2005). Éstas estrategias serán abordadas dentro de la implementación de este caso de

estudio bajo la aplicación del método propuesto en esta tesis con el fin de optimizar los recursos de la red y específicamente mejorar el tiempo de vida del nodo sensor con agentes de software de mayores capacidades, considerando además que la transmisión de 1 bit es más o menos equivalente a la ejecución de 1000 instrucciones (Tynan, et al., 2005).

b) **Características de la red**:

Como se mencionó anteriormente, el objetivo en este caso de estudio es realizar mediciones de luminosidad y obtener el voltaje de las baterías de los nodos sensores con el objetivo final de determinar el tiempo de vida de un nodo sensor con el uso de agentes de software. Ambas mediciones, tanto las de luminosidad como las de voltaje, serán llevas a cabo con una tasa de muestreo y de transmisión de 10 segundos (s) como máximo y para este caso en particular, solo será requerida una estación base y un nodo sensor, por lo tanto con una topología estrella básica (punto a punto) será suficiente (ver Figura 13). El nodo enviara los datos sensados de manera continua a la estación base cada vez que se cumpla la frecuencia de transmisión, donde estos serán almacenados en un documento de texto, al cual se podrá acceder posteriormente para realizar un análisis.

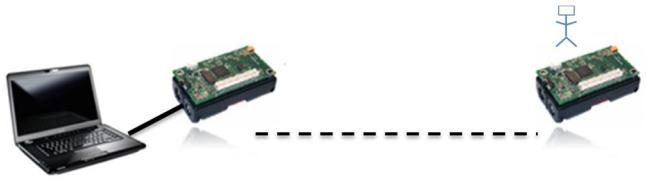

Figura 13. Esquema de la topología punto a punto a utilizar. Las líneas punteadas representan una conexión inalámbrica desde el nodos a la estación base y la línea sólida es una conexión cableada de la estación base al PC.

Para realizar dicha aplicación, teniendo en cuenta que la variable que se desea sensar es luminosidad y la disponibilidad de recursos del laboratorio, se podrían utilizar los dispositivos Mica2 y Tmote Sky. Además, debido a las limitaciones en cuanto a hardware, como se explicara más adelante en la etapa de análisis, en caso de requerir la ampliación de la red, solo será posible trabajar con topologías cuadradas tipo malla.

c) **Identificación de actores**:

En el sistema intervendrán los siguientes actores: Usuario o administrador de red y el agente de software, los cuales poseen las siguientes características:

- Usuario o administrador de red: Será el encargado de realizar la programación de los nodos sensores y el despliegue de la red. Además, es el encargado de iniciar el proceso de adquisición de datos, recuperar la información y el procesamiento de la misma.

- Agente de software: Se propone el uso de un agente de software netamente reactivo, es decir sin ningún tipo de capacidad deliberativa, cuyo ciclo en este caso es básico y consiste en sensar y actuar enviando un dato al agente Usuario (Ver Figura 15) y un agente de software que además de realizar el ciclo básico, tiene la capacidad de procesar la información sensada y actuar, ya sea enviar un dato al agente Usuario o actuar sobre el nodo en el cual se encuentra variando las características de muestreo y transmisión de datos (ver Figura 16).

A continuación se presentan de forma gráfica la descripción de los actores dada anteriormente, donde se muestra la secuencia de pasos o actividades que éstos realizaran (Figura 14, Figura 15 y Figura 16). Cabe aclarar que estas actividades son secuenciales.

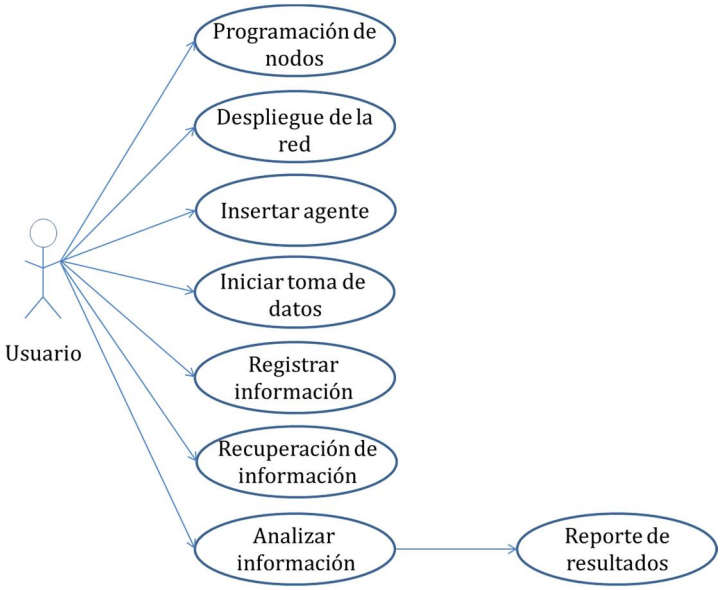

Figura 14. Esquema de actividades actor del Usuario o Administrador de red (WSN con agentes)

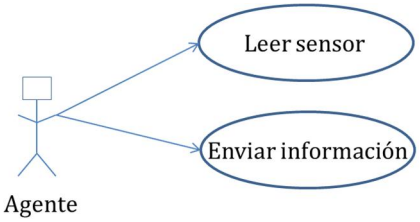

Figura 15. Esquema de actividades del actor agente netamente reactivo

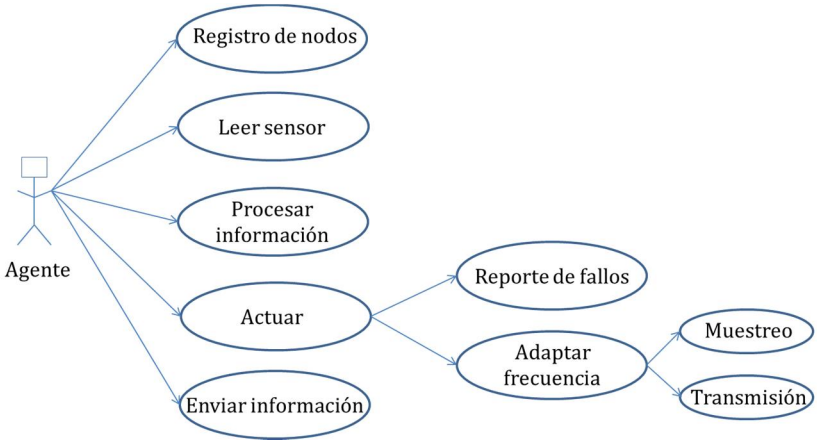

Figura 16. Esquema de actividades actor del agente reactivo con mayores capacidades

d) **Agentes de software**:

En este caso de estudio solo se hará uso de 2 agentes de software tipo reactivo dentro de un nodo sensor y como se mencionó anteriormente, la tarea de uno de ellos, **agente1**, será adquirir un dato de luminosidad cada 10 s y enviarlo de manera inmediata a la estación base, dato que adquiere dato que envía. Mientras que el segundo agente, **agente2**, tendrá la capacidad de actuar con base a lo percibido a través de los sensores del nodo en el cual se encuentra. Para su desarrollo, se utilizará el framework para dispositivos con recursos limitados Agilla.

- **Etapa de análisis y diseño**

a) **Selección de la plataforma hardware**:

Como se planteó en la etapa de conceptualización, los 2 dispositivos disponibles y que cumplen con los requerimientos de la aplicación son los nodos Mica2 y el Tmote Sky. Entre ambas plataformas, la seleccionada para la implementación de dicho caso de estudio, es la plataforma Mica2, ya que solo en ésta, debido a sus características, fue posible realizar la inserción de agentes y de esta forma se adapta a los requerimientos planteados en la primera etapa de este método.

La plataforma Mica2 (MPR400) fue desarrollada por Crossbow y es utilizada también para el despliegue de WSN de bajo consumo de potencia. Opera en el rango de frecuencias de 868/916 MHz y a diferencia del Tmote Sky, la plataforma Mica2 no posee sensores integrados, en su lugar posee un conector expansor de 51 pines que soporta entradas análogas digitales (I/O) y a través de éste permite la adaptación de diferentes tarjetas de adquisición de datos con diferentes sensores adaptados. La utilizada para la realización de las pruebas de validación del método propuesto es una **MTS310CA** (Crossbow Technology, Inc., 2007), la cual posee sensores de temperatura (-40 °C – 70 °C), luminosidad, un acelerómetro de 2 ejes, un micrófono, un buzzer y un magnetómetro de 2 ejes (Figura 17).

Figura 17. MTS310CA

Para la comunicación inalámbrica, el Mica2, tiene integrado un chip de radio CC1000 el cual consume 29 mW en modo de recepción, 42 mW en modo de transmisión y 0,2 µA en modo sleep (Potdar, et al., 2009); trabaja con un microcontrolador Atmel ATmega128L a 8MHz y consume 33 mW de potencia (4,459 nJ/instrucción) en modo activo y 75 µW de potencia en modo sleep (Potdar, et al., 2009) (Baronti, et al., 2007); 4 kB de RAM, 512 kB de flash. Tiene un rango máximo de alcance entre 150-300 m; se alimenta por medio de 2 pilas AA y al igual que los Tmote Sky se puede programar con el sistema operativo TinyOS (Hill, 2003). En la Tabla 5 se muestran las condiciones típicas de operación de un Mica2 (Figura 18).

Figura 18. Mica2

Tabla 5. Condiciones de operación típicas de un Mica2

	MIN	NOM	MAX	UNIDADES
Voltaje de alimentación	2,1		3,6	V
Voltaje de alimentación durante la programación de la memoria flash	2,7		3,6	V
Temperatura de operación	-40		70	°C
Corriente consumida: MCU on, Radio RX		7,4	10	mA
Corriente consumida: MCU on, Radio TX		10,4	27	mA
Corriente consumida: MCU on, Radio off			8	mA
Corriente consumida: MCU sleep, Radio off			15	µA
Board de sensores (operativa totalmente)			5	mA

Para la estación base, cualquier nodo Mica2 debe ser conectado a una MIB510 (Serial Interface Board) (Crossbow Technology, Inc, 2007), la cual suministra energía al dispositivo a través de un adaptador de energía externo y le proporciona una interfaz para establecer una conexión a un computador a través de un puerto serial RS-232, por medio del cual se pueden programar los diferentes motes y se entablara la conexión entre el usuario y la red, para el envío de comandos y la recepción de datos a través del puerto (Figura 19).

Figura 19. Estación base: MIB510 con nodo Mica2

b) **Herramientas para llevar a cabo la implementación**:

Para llevar a cabo la inserción de agentes de software dentro de los nodos sensores en este caso de estudio se hará uso del modelo descrito en (Marsh, et al., 2004), ver sección 3.2. A continuación se describen las herramientas necesarias tanto para la programación de los nodos sensores como para la creación de los agentes de software.

- **Java Communications API**: Para la programación de los nodos Mica2 es necesario conectarlos a la MIB510, la cual le proporciona una interfaz para establecer una conexión a un computador a través de un puerto serial RS-232, permitiendo realizar la programación de los nodos y la conexión con el usuario como se mencionó anteriormente. Para acceder al hardware a través de un puerto serial, es necesario la configuración de determinadas herramientas que permitan entablar dicha comunicación. Java Communications API (Oracle, 2012), también conocida como javax.comm, es una extensión de Java que facilita el desarrollo de aplicaciones independientes para la comunicación de diferentes tecnologías, entre ellas los sistemas embebidos o motes, a través del acceso al hardware por puertos seriales, permitiendo la transferencia de datos a través de ellos. Se debe tener en cuenta que ahora Java es propiedad de Oracle y para sistemas operativos de 64 bits se desarrolló otra librería con el mismo objetivo llamada RXTX, basada en javax.comm. La implementación del caso de estudio se realizó en una máquina de 32 bits y por lo tanto se utilizó javax.comm.

- **Agilla**: Como se mostró en la sección 2.1.6, existen varios frameworks para el desarrollo de agentes y específicamente para el desarrollo de agentes en dispositivos con recursos limitados, como es el caso de una WSN, tales como Jade-LEAP, AFME y Agilla. Como se señaló allí, Jade-LEAP se extiende de la arquitectura de JADE y la amplía mediante el uso de un conjunto de perfiles que le permite ser configurado para diferentes Máquinas Virtuales de Java (JVM), del cual se han implementado 3 versiones que son dependientes de las ediciones de Java. Las plataformas soportadas son J2SE, usado para el desarrollo de aplicaciones de escritorio; PersonalJava, para aplicaciones PDA; y J2ME (Java 2 Micro Edition), para trabajar con dispositivos móviles. Por su parte, AFME se basa en la plataforma de Java CLDC (Constrained Limited Device Configuration) aumentada con MIDP (Mobile Information Device Profile), los cuales constituyen un subconjunto de la especificación J2ME. CLDC /MIDP es de hecho el estándar en la plataforma Java para teléfonos móviles y otros dispositivos con recursos limitados (Muldoon, 2007). Por las

características descritas anteriormente, para la creación e inserción de agentes con LEAP y AFME, es necesario que los motes tengan JVM, como es el caso de los Sun SPOT.

A diferencia de los 2 frameworks mencionados anteriormente, Agilla, el primer middleware de agentes móviles para WSN, puede ser instalado sobre nodos sensores que soporten el sistema operativo TinyOS, como por ejemplo los motes Mica2, MicaZ, TelosB y Tmote Sky, permitiendo de esta manera la creación de agentes en estos dispositivos. Sin embargo, se debe tener en cuenta que los agentes creados en Agilla son de tipo reactivo, mientras que los agentes creados en los otros 2 frameworks, LEAP y AFME, son de tipo deliberativo, esto debido a las capacidades intrínsecas de los motes.

Por lo descrito anteriormente, para el desarrollo de los agentes se hará uso del framework para dispositivos de recursos limitados agilla 2.1.6.2.2)

- **Agilla Injector**: Interfaz gráfica proporcionada por Agilla Middleware que permite ingresar y administrar los agentes móviles en la WSN, además de diseñarlos y programarlos. Éste está construido en Java y posee capacidades para la comunicación a puertos seriales, paralelos y USB en el host local.

c) **Selección de métricas**:

Como se ha descrito hasta ahora, el factor más significativo en el estudio de las WSN es el consumo de energía, el cual limita de manera directa el tiempo de vida del sistema. Debido a que los nodos son desplegados en muchos casos en zonas inaccesibles o peligrosas, donde el mantenimiento frecuente, como el cambio de las baterías no es deseable y en algunos casos es imposible, una de las métricas más importantes en el análisis de las WSN en general es ésta, el tiempo de vida del sistema. Por este motivo, para evaluar en un principio el desempeño de la integración de agentes inteligentes y WSN a través de la implementación del método propuesto en este caso de estudio, será utilizada la métrica de tiempo de vida de la red y además, se evaluará el tiempo de respuesta ante eventos inesperados para la WSN con el agente1 y para la WSN con el agente2 respectivamente. Asimismo, se tendrá en cuenta el grado de completitud de ambas.

Recordemos que el **tiempo de vida** puede ser definido de varias maneras según (Sohraby, et al., 2007): (a) como el tiempo de duración hasta que algún nodo agota toda su energía, o (b) el tiempo de duración hasta que la calidad de servicio de las aplicaciones no se pueden garantizar, o (c) el tiempo de duración hasta que la red esté disociada. También se debe tener en cuenta que otros atributos de la red están directamente relacionados con la administración de la energía y por ende con el tiempo de vida del sistema, como son la cobertura, la latencia, la exactitud y la longevidad.

Para mejorar el tiempo de vida de la red, en la literatura se han propuesto múltiples soluciones, como por ejemplo, mejorarlo a través de la variación de parámetros como la longitud del paquete, otros mediante la reducción del número de saltos recorridos por un paquete para llegar a su destino (técnicas de comunicación de múltiples saltos), etc. Sin embargo, se debe tener en cuenta que las decisiones tomadas para el mejoramiento de la red como tal deben ser realizadas a conciencia y dependiendo de la aplicación a realizar, ya que la variación en un parámetro en busca del mejoramiento de uno de los atributos de la red, puede influir significativamente en otros. Por ejemplo, la variación de la longitud de un paquete puede afectar directamente el número de paquetes requeridos para la transmisión de datos y a su vez la probabilidad de encontrar errores en dichos paquetes y por ende el número promedio de retrasmisiones que serían requeridas.

Para realizar la medición del tiempo de vida, durante la implementación de este método en el presente caso de estudio, se considerara el tiempo de vida como el tiempo de duración hasta que algún nodo agota toda su energía, por lo tanto tenemos que el tiempo de vida de la red, T_L, será directamente proporcional a la energía de cada nodo sensor, E, (en Joule), la cual a sus vez es directamente proporcional a la cantidad de voltaje de las baterías conectadas a la fuente de energía que alimenta el nodo sensor en determinado instante del tiempo, V. Por lo tanto, finalmente tendremos que:

Ecuación 7. Tiempo de vida de la red

$$T_L \propto V$$

Para cada una de las medidas realizadas, con el objetivo de garantizar las mismas condiciones iniciales de descarga y además con el fin de garantizar la réplica y aleatorización del experimento, siempre se hizo uso de las mismas baterías (pilas), las cuales se dejaron cargar durante el mismo número de horas y con el mismo cargador, ya que debido al uso y manipulación de las baterías cada una adquiere una curva de descarga diferente lo cual podría afectar el experimento realizado.

Por otra parte, los nodos deben ser capaces de comunicar mensajes de alta prioridad a través de la red tan rápido como sea posible y por este motivo se evaluó la métrica de **Tiempo de respuesta** para ambas WSN, para lo cual se midió el tiempo que tarda el agente en enviar un mensaje donde se refleje el cambio en el estado de luminosidad del ambiente a la estación base. Recordemos que éste puede ser mejorado mediante la inclusión de nodos que funcionen todo el tiempo, sin embargo esto afectaría el tiempo de vida de la red conforme a los parámetros de evaluación seleccionados en este caso de estudio, ya que se aumentaría el consumo de energía por parte de los nodos sensores y por ende se reduciría el tiempo de vida de éstos.
Además, se evalúa la **Completitud** del sistema como la capacidad de un agente de generar un cambio o no en el entorno ante un evento, como se describió en la sección 4.2.3.

- **Etapa de implementación**

En esta etapa de implementación se aplicaran todas las características, parámetros y especificaciones encontradas en las etapas anteriores del método propuesto.
Con el objetivo de mejorar el tiempo de vida de la red, a través de la optimización del consumo de energía de cada nodo, se ha planteado una estrategia donde mediante el uso de agentes inteligentes de software se realicen variaciones en la frecuencia de transmisión y de muestreo de los nodos sensores, agente2, cuyos resultados en primera instancia serán comparados con los obtenidos con un agente netamente reactivo, agente1. Como se describió en la etapa de análisis y diseño, con Agilla solo es posible desarrollar agentes netamente reactivos y por lo tanto no se exploraron otras opciones, además, también debido a los recursos limitados de la plataforma mote utilizada no es posible realizar la inserción de agentes deliberativos. Sin

embargo, como se verá a continuación, estos agentes "simples" pueden facilitar la administración de los recursos de los nodos de la red.

Para la implementación, la configuración original de Agilla fue modificada con el fin de sensar voltaje mediante el comando SENSE y también de adquirir los datos enviados por los agentes de software inteligente en un documento en formato .txt, ya que ambas opciones no están por defecto definidas dentro de la configuración inicial de Agilla.

Como se ha descrito hasta ahora, el **Agente1** es netamente reactivo y enviará datos a la estación base cada 10 s de luminosidad y de voltaje. En la Figura 20 se muestra el diagrama de flujo para este agente.

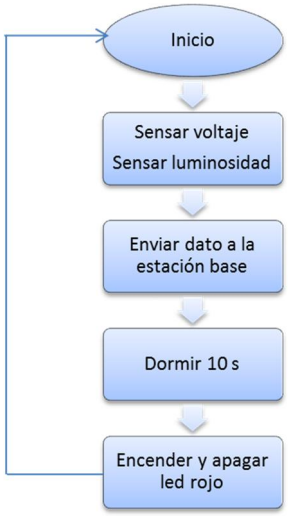

Figura 20. Diagrama de flujo agente1

Por su parte, el **Agente2** tiene la capacidad de variar la frecuencia de transmisión de datos dependiendo de lo que percibe de su ambiente a través de los sensores del mote en el cual reside, manteniendo la frecuencia de muestreo constante, la cual en caso de ser requerido, también se podría variar. El agente2 realiza un procesamiento de los datos que recibe y toma decisiones con base a un conjunto de reglas lógicas para alterar su comportamiento. Con base a los resultados obtenidos del proceso de deliberación que ejecutó, este agente varia la frecuencia de transmisión de datos, como se había indicado anteriormente, reduciendo de esta forma el consumo de la batería.

El agente durante su estado de inactividad entra en periodos de sueño donde el consumo de energía es hasta de 0,001875 %, lo que es equivalente a decir que, en estado de sueño el dispositivo consume 533 veces menos energía que en estado activo según (Piedrahita Ospina, et al., 2009). Por ejemplo, si planteamos como espacio para realizar la aplicación un invernadero, las plantas de éste deben recibir en promedio la misma intensidad de luz durante su desarrollo para un óptimo crecimiento. Por lo tanto, si en promedio la cantidad de luz se mantiene estable a lo largo de un intervalo de tiempo, no será necesario tener con mucha frecuencia su valor y reportarlo cada 30 minutos, por ejemplo, será más que suficiente. Sin embargo, si el promedio de la intensidad de luz varia en forma inusual, se deberá modificar tanto la frecuencia de transmisión para dar parte del evento y de esta forma se pueda actuar si es preciso y evitar pérdidas.

En la Figura 21 se muestra el diagrama de flujo de este agente, el cual inicialmente comienza con una frecuencia de transmisión de 10 s, la cual en cada ciclo se va incrementando en 10 s hasta llegar a una frecuencia mínima de transmisión de 900 s (15 minutos), para este caso particular. La frecuencia de muestreo en este caso es de 5 s y permanecerá constante a lo largo del experimento. Si se genera un cambio repentino en las condiciones del ambiente, el agente enviará datos a la estación base de manera inmediata. Los eventos inesperados en este caso son detectados realizando una comparación entre una muestra de luminosidad tomada en un instante de tiempo t_{i-1} (luminosidad 1) y la inmediatamente siguiente tomada en el instante t_i (luminosidad 2), la cual es adquirida 5 s después según lo descrito anteriormente. Si la diferencia entre ambos datos sensados supera cierto umbral, en este caso 10 lumen, el sistema reiniciará todas sus variables regresando nuevamente a la máxima frecuencia de transmisión, permitiéndole al usuario identificar rápidamente algún cambio.

Se debe tener en cuenta, que en el caso del agente2, la variación de la frecuencia de transmisión de datos podría resultar, por ejemplo, en una carencia en la correlación de tiempo entre los eventos, lo cual afectaría la exactitud temporal y disminuiría el tiempo de vida del sistema. Sin embargo, estos factores no serán tenidos en cuenta para el análisis realizado.

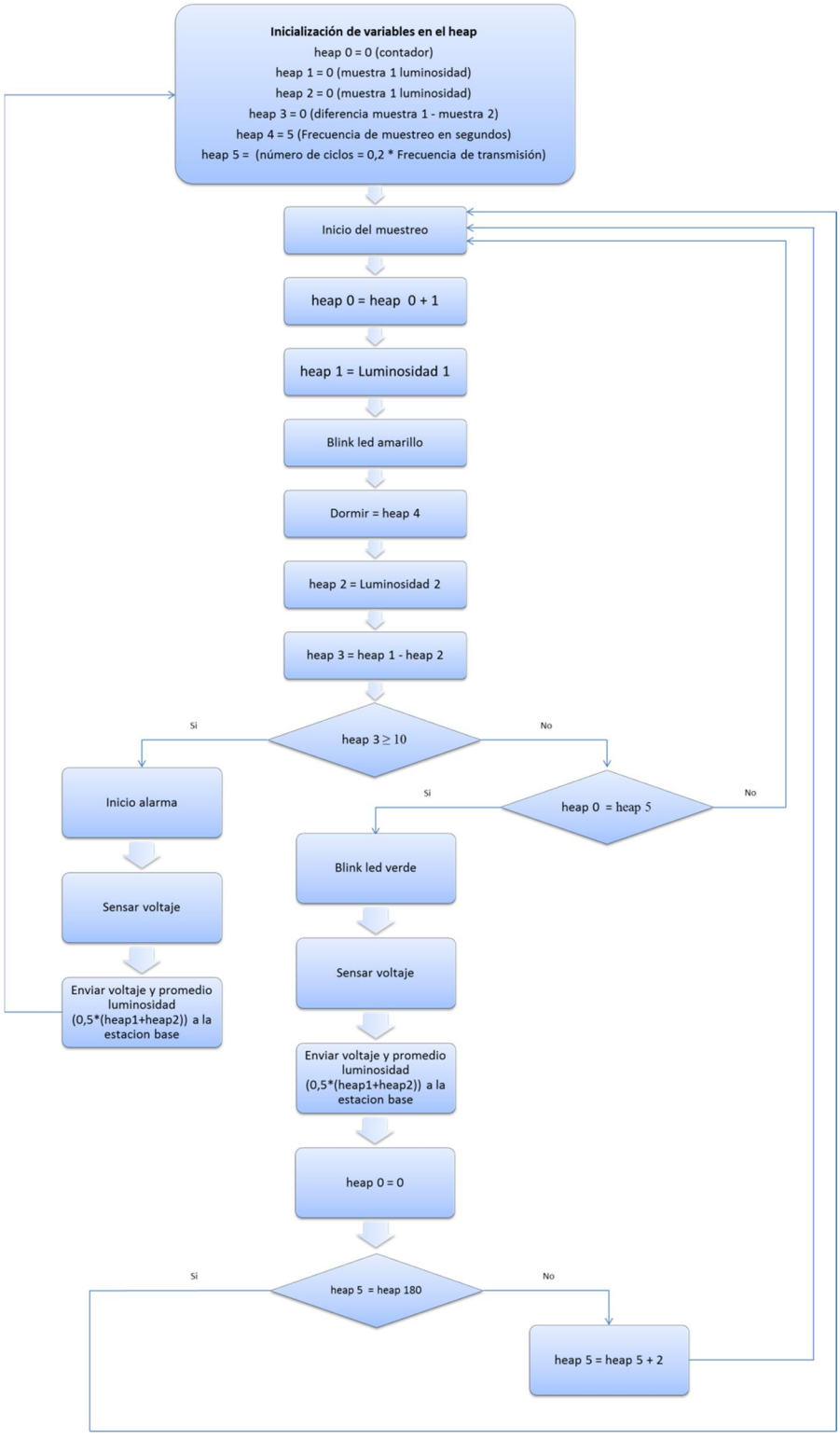

Figura 21. Diagrama de flujo agente2

- **Etapa de evaluación de resultados**

En la Figura 22 y la Figura 23 se muestran los resultados obtenidos para el tiempo de vida de una red conformada por 2 nodos sensores mica2 haciendo uso de agentes de software. Como se puede observar de la Figura 22, resultados del tiempo de vida de la red con la integración del agente1, el tiempo de vida promedio fue de 47 horas aproximadamente, mientras que con la integración del agente2 en la red, Figura 23, se obtuvo un tiempo de vida aproximado de 58 horas.

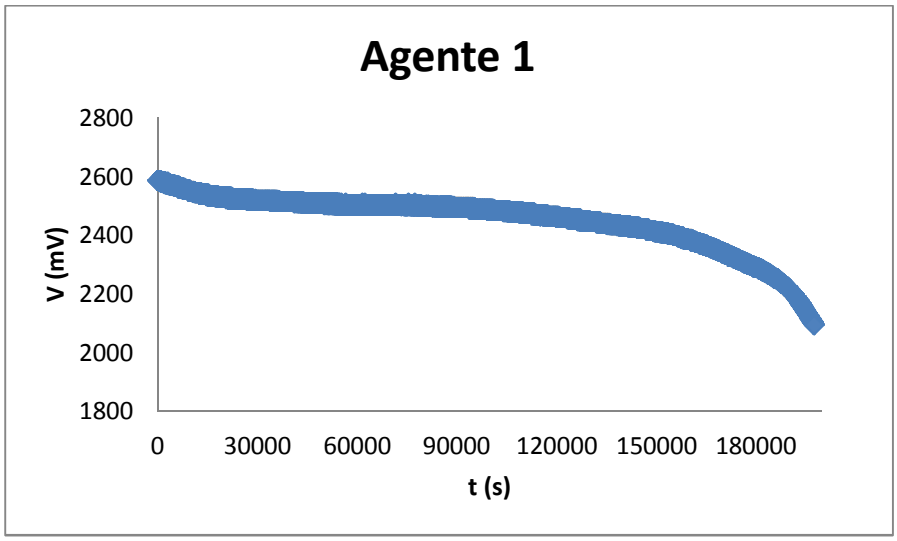

Figura 22. Resultados TL del nodo con el agente1 (Gráfica V vs t)

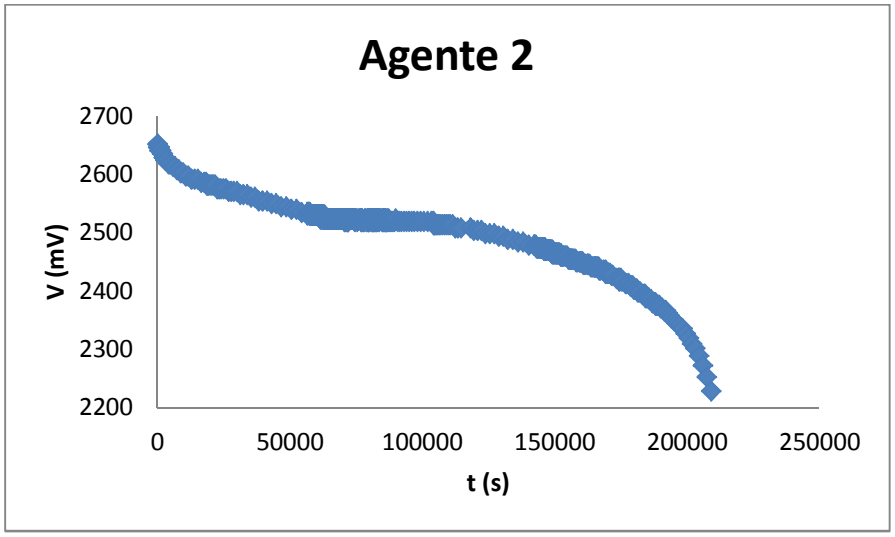

Figura 23. Resultados TL del nodo con el agente2 (Gráfica V vs t)

Aunque la diferencia no es considerablemente significativa, si consideramos el tiempo máximo de vida de la red como el obtenido con la integración del agente 1, al mejorar las capacidades deliberativas de los agentes de software utilizados, tenemos una mejora hasta del 19 % en el tiempo de vida de la red aproximadamente.

En cuanto a la evaluación del tiempo de respuesta de ambas redes ante eventos inesperados, WSN+agente1 y WSN+agente2, según lo planteado en la sección 2.4 y lo descrito en la fase de análisis y diseño para la selección de métricas, mediante la utilización de agentes de mayores capacidades es posible tener una frecuencia de muestreo alta, mayor número de muestras por unidad de tiempo, permitiendo detectar eventos inesperados en menor tiempo y una frecuencia de transmisión variable, dependiendo de las características de la aplicación, permitiendo una disminución de las comunicaciones inalámbricas y de esta forma evitando afectar de manera significativa el tiempo de vida de la red. Los resultados de la métrica tiempo de respuesta son presentados en la Tabla 6 y en ellos se puede apreciar que el tiempo de respuesta de la WSN+agente2 es 2 veces menor que el tiempo de respuesta de la WSN+agente1 de acuerdo a las características dadas a cada uno de los agentes. Según esto, se podría pensar en disminuir la frecuencia de transmisión del agente1 de tal manera que iguale el tiempo de respuesta del agente2, sin embargo, esto afectaría aún más el tiempo de vida de la red evaluado anteriormente.

Tabla 6. Evaluación métrica Tiempo de respuesta

	Tiempo de respuesta (s)	Frecuencia de transmisión (s)
WSN + Agente1	10	10
WSN + Agente2	5	10 – 900 (variable cada 10 s)

En la Tabla 7 se muestran los resultados de la métrica de completitud, la cual se definió como la capacidad de un agente de generar un cambio o no en el entorno ante un evento, en la cual se evidencian las ventajas de la integración del agente con mayores capacidades de procesamiento sobre la red que con la integración del agente netamente reactivo.

Tabla 7. Evaluación métrica Completitud

Evento	Red	Cambio	Completitud
Cambio repentino en la intensidad lumínica del ambiente	WSN + Agente1	Ninguno	0
	WSN + Agente2	Variación de la frecuencia de transmisión (aumenta)	1
Intensidad lumínica del ambiente estable durante determinado intervalo de tiempo	WSN + Agente1	Ninguno	0
	WSN + Agente2	Variación de la frecuencia de transmisión (disminuye)	1

5.3 Caso de estudio 3: Medición del tiempo de vida de una WSN sin agentes de software

- **Etapa de conceptualización**

a) **Descripción de la aplicación**:

Este caso de estudio tiene como objetivo evaluar la eficiencia de la utilización de agentes de software dotados con mayores capacidades para el mejoramiento del tiempo de vida de una WSN mediante una comparación con una red sin agentes de software y además, determinar bajo que parámetros es más factible su utilización. Con

el fin de cumplir dicho objetivo, al igual que en el caso de estudio 2, se realizaran mediciones de luminosidad en el ambiente y además se obtendrá el valor del voltaje de las baterías de los nodos sensores de la red.

b) **Características de la red**:

Para realizar dicha aplicación, teniendo en cuenta que el objetivo de este caso de estudio es realizar una comparación con el caso de estudio anterior, las variables a medir, la frecuencia de muestreo y la frecuencia de transmisión serán las mismas, es decir, se sensará luminosidad y voltaje con una tasa de muestreo y de transmisión de 10 segundos (s) en una topología estrella básica (punto a punto) con nodos mica2.

c) **Identificación de actores**:

En este caso de estudio solo intervendrá un actor Usuario o administrador de red y es el encargado de realizar la programación del nodo sensor y fijar tanto la frecuencia de transmisión como la frecuencia de muestreo de los datos. También es el encargado de iniciar el proceso de adquisición, de realizar la recuperación de la información y el procesamiento de la misma. A continuación se presentan de forma gráfica las actividades del actor identificado (Figura 24).

Figura 24. Esquema de actividades del actor Usuario o Administrador de red (WSN sin agentes)

d) **Agentes de software**:

En este caso de estudio no se requerirá el uso de agentes de software.

- **Etapa de análisis y diseño**

a) **Selección de la plataforma hardware**:
Como se describió en la etapa de conceptualización, para el desarrollo de este caso de estudio se hará uso de la plataforma mote mica2, cuyas características detalladas fueron dadas en el caso de estudio anterior.

b) **Herramientas para llevar a cabo la implementación**:
Para llevar a cabo la implementación de este caso de estudio y para el desarrollo de la WSN sin agentes de software, se hizo uso del Java Communications API, que al igual que en el caso de estudio anterior permitirá establecer una comunicación vía puerto serial entre la estación base conformada por la MIB510 y uno de los nodos mica2 con un computador, permitiendo la programación de los nodos sensores y la interacción del usuario con la red. Además, se hizo uso del sistema operativo **TinyOS v1** para la programación de los nodos, el cual, como se especificó en el sección 2.3.1, es la plataforma nativa para redes de sensores más ampliamente usada y usa un modelo de ejecución basado en eventos y tanto sus librerías como sus aplicaciones están escritas en su propio lenguaje de programación llamado nesC.

c) **Selección de métricas**:
Con el fin de cumplir el objetivo planteado para este caso de estudio, será utilizada la métrica de tiempo de vida de la red y además, se evaluará su grado de completitud. Para realizar la medición de las métricas mencionadas, se seguirán los mismos parámetros establecidos para el caso de estudio anterior.

- **Etapa de implementación**

Se debe tener en cuenta que para una red SIN agentes de software solo es posible pre-establecer una configuración determinada con base a la meta de la aplicación a realizar y por este motivo, todos los cambios en la programación de los nodos deben ser establecidos con

anterioridad, ya que en caso de requerir un cambio en la red se debe hacer la programación nodo por nodo. Para sensar luminosidad en el ambiente con los nodos mica2 en una red sin agentes de software, se ha realizado una modificación de la aplicación Oscilloscope encontrada dentro de las herramientas de TinyOS v1, la cual finalmente permitirá realizar el muestreo de la luminosidad en el ambiente cada 10 segundos (s) y además medir el voltaje de las baterías del nodo sensor al mismo tiempo, y asimismo, crear un documento en formato .txt donde se recopilan todos los datos enviados por los nodos sensores. Estas 2 últimas características no están por defecto en la aplicación Oscilloscope y fueron configuradas con el fin de cumplir el objetivo propuesto. La aplicación usada en este caso de estudio para la estación base es llamada TOSBase y también se encuentra disponible en los recursos de TinyOS v1 y no se han realizado ningún tipo de modificaciones en ella.

Como se especificó anteriormente, tanto la frecuencia de muestreo como la de transmisión son iguales a 10 s. Una vez el nodo sensor adquiere un dato del ambiente de luminosidad, lo envía de forma inmediata a la estación base junto con el valor del voltaje de las baterías en ese mismo instante de tiempo. Para el almacenamiento de los datos en tiempo real, se realizó la implementación en java de un Listen del puerto serial, el cual permite almacenar en un documento de texto los datos recibidos por él.

Para cada una de las medidas realizadas, con el objetivo de garantizar las mismas condiciones iniciales de descarga y demás características del caso de estudio 2, se hizo uso de las mismas baterías (pilas) empleadas en éste y del mismo protocolo de manipulación.

- **Etapa de evaluación de resultados**

La Figura 25 muestra los resultados obtenidos para el tiempo de vida de una red conformada por 2 nodos sensores mica2 haciendo uso de la aplicación modificada Oscilloscope de TinyOS. En este caso el tiempo de vida de la red fue de aproximadamente 82 horas y si consideramos éste como el tiempo de vida máximo, al compararlo con el tiempo de vida de 58 horas de la red con el agente2 integrado, se tiene una disminución del 29 % en el tiempo de vida a pesar de tener una frecuencia de transmisión variable. De la misma forma, si se evalúa el tiempo de vida de la red con el agente1 integrado, se tiene una mayor disminución del tiempo de vida de la red como es de esperarse, la cual alcanza hasta un 42 % de disminución. En ambos casos la

disminución del tiempo de vida de la red es debido al gasto extra realizado por el agente de software.

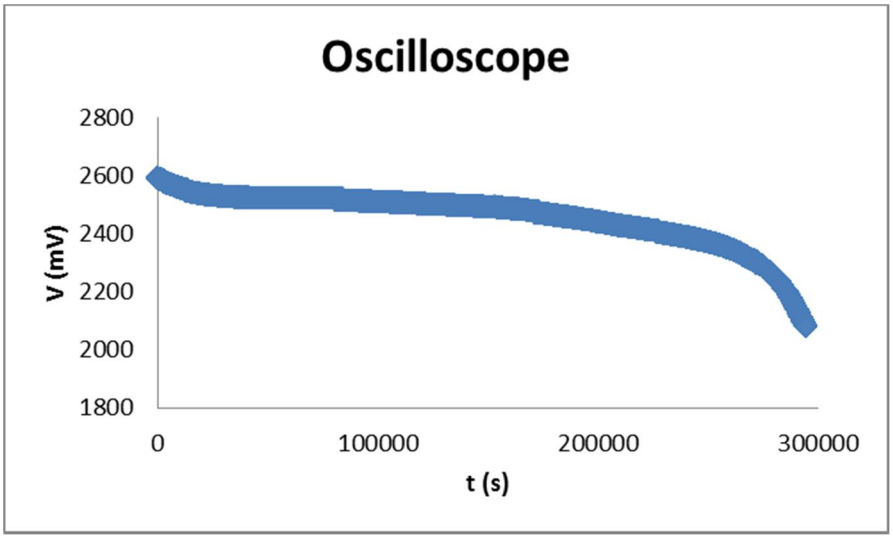

Figura 25. Resultados T_L del nodo con la aplicación modificada Oscilloscope (Gráfica V vs t)

Si se evalúa la **completitud** de la red, al igual que la WSN integrada con el agente1 del caso de estudio 2, ante cualquier variación en las condiciones de luminosidad del ambiente, la red permanecerá inalterada (completitud = 0), a diferencia de la red integrada con el agente2 donde se adapta a las características del medio (completitud = 1).

Con base a los resultados obtenidos, se debe tener en cuenta que en el caso de no requerir de una red que se adapte a las características y necesidades del medio en el cual se encuentra desplegada y además, en la cual se necesite tener el máximo tiempo de vida posible, el uso de agentes de software se hace innecesario y traería consigo un gasto en energía adicional sin aparente justificación. Mientras si se busca hacer frente a diferentes retos que se presentan diariamente en un montaje real, bajo condiciones muchas veces impredecibles, para las cuales la red no fue diseñada, el uso de las técnicas de inteligencia artificial, específicamente de los agentes inteligentes de software, es una solución promisoria permitiendo además hacer frente a otros desafíos que la red trae consigo.

Cuando se requiera de la utilización de éstos, se sugiere hacer uso del mayor número posible de las capacidades que poseen, las cuales fueron descritas en la sección 2.1, de tal manera que ayuden a mejorar el rendimiento de la red en otros aspectos, tales como enrutamiento de paquetes, envío de información sobre la disociación de la red, el estado de las baterías de los

nodos, etc. justificando aún más su utilización y el aumento en el rendimiento del sistema en general.

Capítulo 6

Conclusiones y Trabajo Futuro

La integración de agentes inteligentes en las WSN es promisoria en la medida en que los agentes inteligentes permiten hacer frente a los desafíos y limitaciones que las WSN traen consigo, haciendo uso del comportamiento inteligente y proactivo de dichos agentes. Sin embargo, un aspecto crucial para dicha integración, es lograr evaluar el desempeño de ésta considerando los diferentes factores o variables que se encuentran íntimamente relacionados. Para ello se propuso en esta tesis un método que permite, a través de la ejecución de un proceso organizado y detallado, realizar la evaluación del desempeño de dicha integración a través de métricas una vez identificados los factores más relevantes para el análisis de la red, asimismo, se identificaron diferentes modelos de integración que han sido propuestos para insertar agentes de software en las WSN.
El método propuesto en esta tesis, permite además de evaluar el desempeño de la integración de agentes inteligentes de software y WSN, evaluar casos de estudio que no requieran de agentes de software en su implementación.

En el desarrollo de esta tesis también se lograron resumir las características generales de los nodos más comerciales para WSN, la identificación e implementación de diferentes herramientas para su programación y despliegue, y además, los frameworks que permiten la inserción de agentes de software en estos dispositivos de recursos limitados. Gracias a esto, se puedo realizar la mejor selección de la plataforma mote, dentro de las plataformas con las que se contaba, para la implementación del método propuesto y por ende para la inserción de agentes en ellos. Cabe resaltar que la validación del método se realizó en dispositivos de hardware reales, tratando con sus limitaciones a escala real, lo cual da un valor agregado ya que en la mayoría de la literatura se realizan análisis a partir de simulaciones donde podrían quedar por fuera algunas de las características de un nodo real.

Se debe tener en cuenta que en el caso de no requerir de una red que se adapte a las características y necesidades del medio en el cual se encuentra desplegada, el uso de agentes de software se hace innecesario y traería consigo un gasto en energía adicional sin aparente justificación. Pero si por el contrario, se busca hacer frente a diferentes retos que se presentan diariamente en un montaje real, bajo condiciones muchas veces impredecibles para las cuales la red no fue diseñada, el uso de técnicas de inteligencia artificial, específicamente de agentes inteligentes de software, es una solución promisoria, la cual de una manera bien encaminada hace posible mejorar el desempeño de una WSN a través de la modelación de agentes de software que se ajusten a las limitaciones de recursos de éstas sin perder su capacidad de deliberar, como se mostró a lo largo del desarrollo de esta tesis.

Como trabajo futuro, se pretende ampliar el método propuesto y realizar su implementación en dispositivos de mayores recursos, como los SunSpot, en los cuales sea posible explotar al máximo las características de los agentes inteligentes y aplicar otras técnicas de IA con el fin de mejorar el razonamiento en la red, como por ejemplo redes bayesianas, redes neuronales, etc. Además, fortalecer los agentes desarrollados y ampliarlos a nuevas escalas, de modo que permitan mejorar las características de las redes conformadas por dispositivos con recursos más limitados.

Asimismo, también se busca explorar nuevas alternativas como el uso de múltiples estaciones base con amplios recursos, como los SheevaPlug (Globalscale Technologies Inc., 2008), que faciliten una mayor cobertura de la red y la interoperabilidad de diferentes dispositivos mote, ampliando de esta forma el número de aplicaciones a realizar.

A pesar de que se pudo verificar el método propuesto a través de la implementación de los casos de estudio presentados, comprobando su utilidad al momento de evaluar el desempeño de la integración de agentes de software en las WSN y además de redes sin agentes, se pretende evaluar el desempeño conforme cambia la topología de red haciendo uso además de las demás métricas expuestas en esta tesis de maestría, evaluando la relación entre ellas.

Tabla 8. Resumen de características de algunos nodos sensores

Plataforma mote	Compañía	SO	Características del microprocesador y Especificaciones de memoria								RF transceiver						Características electro-mecánicas				
			Tipo de procesador	Bus	Reloj	Consumo de corriente (en los diferentes modos de operación) activo sleep	RAM	EEPROM	Memoria flash	ADC	Banda de operación (GHz)	Estandar de comunicación	Tasa de transmisión de datos	Alcance máximo (out door w indoor)	Consumo de corriente	Plataforma de radio (chip RF)	Alimentación	Interfaz de usuario	Sensores integrados	Tamaño (mm) sin baterías ni antena	Peso sin baterías (g)
MICA2 MPR400 (2002)	Crossbow (2da generación)	MoteWorks™ (TinyOS)	Atmel ATmega128L	8 bit	8 MHz	**Modo activo:** 8mA (33mW) **Modo Sleep (total)**: <15μA (75μW)	4K	4Kbytes	Atmel AT45DB41B 512K	10 bit ADC, 8 canales, 0-3V input	0.868-0.916 **Radio multicanal, programable entre 4/50 y es específico para el país		hasta 76,8 Kbps	**Outdoor:** 152.4m **Indoor**:	**Modo de recepción:** 7,4 μA-7,4 mA (29mW) **Modo de transmisión:** (10,4) mA (42mW) **Modo Sleep:** (<1)μA	Chipcon CC1000	2 pilas AA	3 Leds	Ninguno, en su lugar tiene un conector expansor de 51 pines que soporta entradas análogas digitales (I/O)	58x32x7	18
MICAz MPR2400 (versión mejora del MICA2, acepta el IEEE 802.15.4) (2004)	Crossbow (3cer generación)	MoteWorks™ (TinyOS)	Atmel ATmega128L	8 bit	8 MHz	**Modo activo:** 8mA **Modo Sleep (total)**: <15μA	4K	4Kbytes	Atmel AT45DB41B 512K	10 bit ADC, 8 canales, 0-3V input	2.4-2.48 Banda ISM	IEEE 802.15.4	250Kbps	**Outdoor:** (75-100)m **Indoor**: (20-30)m	**Modo de recepción:** (19.7)mA **Modo de transmisión:** 11mA (-10dBm), 14mA (-5dBm) o 17.4 mA (0dBm) **Modo Sleep:** 1μA **Modo inactivo (idle):** 20μA	Chipcon CC2420	2 pilas AA	3 Leds	Ninguno, en su lugar tiene un conector expansor de 51 pines que soporta entradas análogas digitales (I/O)	58x32x7	18
TelosB (2005)	Crossbow (UC Berkeley)	TinyOS	MSP430 F1611	16 bit	8 MHz	**Modo activo:** (1.8)mA (3,24μW) **Modo Sleep (total)**: (5.1)μA (15μW)	10K	16K	48K	12 bit ADC (8 canales, 0-3V) y DAC (2 puertos) respectivamente	2.4-2.484	IEEE 802.15.4	250 Kbps	**Outdoor:** (75-100)m **Indoor**: (20-30)m	**Modo de recepción:** (23)mA **Modo de Idle:** (21)mA **Modo Sleep:** (1)μA	Chipcon CC2420	2 pilas AA		Si, Humedad, temperatura y luminosidad. Además posee un conector de expansor	65x31x6	23
Tmote Sky (2005)	Sentilla (UC Berkeley)	TinyOS Contiki	MSP430 F1611	16 bit	8 MHz	**Modo activo:** (1800-2400)μA (3mW) **Modo idle (total)**: (54.5-1200)μA (15μW) **Modo standby:** (5.1-21.0) μA	10K	1M	48K	8 canales	2.4-2.484	IEEE 802.15.4	250 Kbps	**Outdoor:** 125m **Indoor**: 50m	**Modo de recepción:** (21.8 - 23)mA **Modo de transmisión:** (19.5-21)mA **Modo Sleep:** (0.02-426)μA	Chipcon CC2420	2 pilas AA		Si, Humedad, temperatura y luminosidad. Además posee un conector de expansor	33x66x7	14.93
IRIS XM2110CA (2007)	Crossbow	TinyOS	ATmega 1281	8 bit	8 MHz	**Modo activo:** 8mA **Modo Sleep (total)**: 8μA	8 K	4Kbytes	512K	10 bit ADC, 8 canales, 0-3V input	2.405 – 2.48 Banda ISM	IEEE 802.15.4	250Kbps	**Outdoor:** 500m **Indoor**: >50m	**Modo de recepción:** 16mA **Modo de transmisión:** 10mA (-17dBm), 13mA (-3dBm) o 17 mA (3dBm) **Modo Sleep:** 20nA	Atmel AT86 RF230	2 pilas AA	3 Leds (rojo, verde y amarillo)	Ninguno, en su lugar tiene un conector expansor de 51 pines que soporta entradas análogas digitales (I/O)	58x32x7	18
Sun Spot (2007)	Oracle	**Squawk VM (Java)**	Atmel AT91RM9200 - ARM920T	32 bit	400MHz	**Modo activo:** 24.4mA **Modo standby**: 520μA **Modo Sleep: 65**μA	1Mb	Ninguna	8Mb		2.4-2.4835	IEEE 802.15.4	250Kbps	**≈100m**	**Modo de recepción:** (18.8)mA **Modo de transmisión:** (17.4)mA **Modo Sleep:** (0.02-426)μA	Chipcon CC2420	Batería recargable onboard (3.7V). La capacidad de la batería incorporada es de 770mA/h		Tienen sensores como acelerómetros 3D, sensores de luz, de temperatura, 8 led, 2 switches, 5 pines I/O de propósito general y 4 pines de salida de alta corriente. Además, tiene la posibilidad de conectar otras boards a la board principal, las cuales también poseen los sensores mencionados anteriormente.	41x23x70	54

Referencias bibliográficas

Aakvaag, N. & Frey, J.-E., 2006. Redes de sensores inalámbricos. *Revista ABB 2,* pp. 39-42.

Allen, R., 2010. *Self - the power of simplicity.* [En línea]
Available at: http://selflanguage.org/
[Último acceso: 27 Noviembre 2010].

Anon., 2012. *Foundation for Intelligent Physical Agents.* [En línea]
Available at: www.fipa.org
[Último acceso: 12 Noviembre 2012].

Anon., 2012. *GetData Graph Digitizer.* [En línea]
Available at: http://getdata-graph-digitizer.com/
[Último acceso: Diciembre 2012].

Arduino SA, 2013. *Arduino.* [En línea]
Available at: http://www.arduino.cc/
[Último acceso: Agosto 2013].

Asti Automation S.R.L., 2010. *MoteWorks Software Platform.* [En línea]
Available at: http://astiautomation.ro/produse/Memsic/swplatform_en.html
[Último acceso: 23 Noviembre 2010].

Baronti, P. y otros, 2007. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. *Computer Communications,* 26 Mayo, 30(7), pp. 1655-1695.

Bayer, P. & Svantesson, M., 2001. Comparison of Agent-Oriented Methodologies Analysis and Design MAS-CommonKADS versus Gaia. *Department of Software Engineering and Computer Science, Blekinge Institute of Technology.*

Bellifemine, F., Caire, G. & Rimassa, G., 2000. *Jade.* [En línea]
Available at: http://jade.tilab.com/
[Último acceso: 10 05 2010].

Bellifemine, F., Poggi, A. & Rimass, G., 1999. JADE - A FIPA-compliant agent framework. *Proceedings of PAAM'99,* Abril.pp. 97-108.

Buonadonna, P., Hill, J. & Culler, D., 2007. *TinyOS papers.* [En línea]
Available at: http://www.tinyos.net/papers/ammote.pdf
[Último acceso: 24 Noviembre 2010].

Carretero, N. & Bermejo, A., 2005. INTELIGENCIA AMBIENTAL.

Chaparro, D., 2003. *Computación ubicua,* España: Universidad Rey Juan Carlos.

Chen, M., Gonzalez, S. & Leung, V., 2007. Applications and Design issues for Mobile Agents in Wireless Sensor Networks. *IEEE Wireless Communications Magazine, Special Issue on Wireless Sensor Networking,* pp. 20-26.

Chen, M., Kwon, T., Yuan, Y. & Leung, V. C., 2006. Mobile Agent Based Wireless Sensor Networks. *Journal of Computers,* pp. 14-21.

Cheong, E., 2007. *Actor-Oriented Programming for Wireless Sensor Networks.* University of California, Berkeley: s.n.

Cheong, E., Liebman, J., Liu, J. & Zhao, F., 2003. *TinyGALS: A Programming Model for Event-Driven Embedded Systems.* Melbourne, FL: The 18th ACM Symposium on Applied Computing (SAC'03).

Collier, R., 2002. *Agent Factory: A Framework for the Engineering of Agent-Oriented Applications.* Department of Computer Science, Faculty of Science, University College Dublin: s.n.

Collier, R. & O'Hare, G., 1999. *Agent Factory: A Revised Agent Prototyping Environment.* Ireland, s.n.

Collier, R., O'Hare, G. & Rooney, C., 2004. *A UML-based Software Engineering Methodology for Agent Factory.* University College Dublin: s.n.

Collier, R., O'Hare, G., Lillis, D. & Dragone, M., 2007. *Agent Factory.* [En línea]
Available at: http://www.agentfactory.com/index.php/Main_Page
[Último acceso: 10 05 2010].

Collier, R., O'Hare, G., Lowen, T. & Rooney, C., 2003. Beyond Prototyping in the Factory of Agents. En: *Multi-Agent Systems and Applications III.* s.l.:Springer Berlin / Heidelberg, pp. 383-393.

Contiki, 2012. *Contiki: The Open Source OS for the Internet of Things.* [En línea]
Available at: http://www.contiki-os.org/
[Último acceso: Diciembre 2012].

Crossbow Technology, Inc., 2006. *MoteWorks Getting Started Guide.* [En línea]
Available at: www.xbow.com

Crossbow Technology, Inc., 2007. *MTS/MDA Sensor Board Users Manual.* [En línea]
Available at: http://bullseye.xbow.com
[Último acceso: 20 Noviembre 2012].

Crossbow Technology, Inc, 2007. *MPR-MIB Users Manual.* [En línea]
Available at: http://bullseye.xbow.com:81/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf
[Último acceso: 20 Noviembre 2012].

Crossbow Technology, Inc, 2007. *XMesh User's Manual.* s.l.:s.n.

Crossbow, s.f. *MoteWorks™.* [En línea]
Available at: http://www.willow.co.uk/MoteWorks_OEM_Edition.pdf
[Último acceso: Diciembre 2012].

Cygwin, 2010. [En línea]
Available at: http://www.cygwin.com/
[Último acceso: 10 Octubre 2010].

DeLoach, S., 2001. *Analysis and Design using MaSE and agentTool.* Oxford, Ohio, s.n.

DeLoach, S., 2006. Engineering Organization-Based Multiagent Systems. En: *Software Engineering for Multi-Agent Systems IV.* s.l.:Springer Berlin / Heidelberg, pp. 109-125.

Digi International Inc., 2013. *Xbee: Wireless RF Modules.* [En línea]
Available at: http://www.digi.com/xbee/
[Último acceso: Agosto 2013].

Dunkels, A., 2007. *Contiki: The Operating System for Connecting the Next Billion Devices - the Internet of Things.* [En línea]
[Último acceso: 1 Diciembre 2010].

Dunkels, A., Gronvall, B. & Voigt, T., 2004. Contiki - a lightweight and flexible operating system for tiny networked sensors. *29th Annual IEEE International Conference on Local Computer Networks,* 16-18 Noviembre.pp. 455-462.

Ergen, S. C., 2004. *ZigBee/IEEE 802.15.4 Summary.* [En línea]
Available at: www.sinemergen.com/zigbee.pdf
[Último acceso: 27 Noviembre 2010].

Farooq, M., Aziz, S. & Dogar, A., 2010. State of the Art in Wireless Sensor Networks Operating Systems: A Survey. En: *Future Generation Information Technology.* s.l.:s.n., pp. 616-631.

Fok, C.-L., 2008. *Lesson 1: Agilla Installation.* [En línea]
Available at: http://mobilab.cse.wustl.edu/projects/agilla/docs/tutorials/1_installation.html

Fok, C., Roman, G. & Lu, C., 2005. Mobile agent middleware for sensor networks: An application case study. *IEEE Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005,* 15 Abril.pp. 382-387.

Fok, C.-L., Roman, G.-C. & Lu, C., 2006. *Agilla: A Mobile Agent Middleware for Sensor Networks.* [En línea]
Available at:
http://cse.wustl.edu/Research/Lists/Technical%20Reports/Attachments/722/504_main.pdf
[Último acceso: 18 Diciembre 2012].

Fok, C.-L., Roman, G.-C. & Lu, C., 2009. Agilla: A mobile agent middleware for self-adaptive wireless sensor networks. *ACM Transactions on Autonomous and Adaptive Systems (TAAS),* Julio, 4(3), p. 26.

Gajbhiye, P. & Mahajan, A., 2008. A Survey of Architecture and Node deployment in Wireless Sensor Network. *First International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2008),* pp. 426-430.

Garavito, O., Ocampo, J. & Torres, M., 2008. *Metodología de Pruebas para Sistemas Multi-Agentes (SMA) integrada a AOPOA.* Medellín, s.n.

Georgeff, M. y otros, 1999. The Belief-Desire-Intention Model of Agency. *Proceedings of the 5th International Workshop on Intelligent Agents V:Agent Theories, Architectures, and Languages (ATAL-98).*

Globalscale Technologies Inc., 2008. *SheevaPlug Development Kit.* [En línea]
Available at: http://www.globalscaletechnologies.com/t-sheevaplugdetails.aspx#features
[Último acceso: 1 Diciembre 2010].

H. Van Dyke Parunak, 1999. Industrial and Practical Applications of DAI. En: G. Weiss, ed. *Multiagent Systems: A Modern Approach to Distributed Modern Approach to Artificial Intelligence.* London, England: s.n.

Healy, M., Newe, T. & Lewis, E., 2008. Wireless Sensor Node hardware: A review. *IEEE Sensors,* 26-29 Octubre.pp. 621-624.

Hill, J., 2003. *System Architecture for Wireless Sensor Networks.* s.l.:UC Berkeley.

Huang, M.-C., Huang, J.-C., You, J.-C. & Jong, G.-J., 2007. The Wireless Sensor Network for Home-Care System Using ZigBee. *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing,* 26-28 Noviembre, Volumen 1, pp. 643-646.

Hussain, S., Shakshuki, E. & Matin, A., 2006. *Agent-based system architecture for wireless sensor networks.* s.l., IEEE, pp. 296-302.

IBM, 2010. *IBM Mote Runner.* [En línea]
Available at: http://www.zurich.ibm.com/moterunner/
[Último acceso: 1 Noviembre 2010].

Iglesias Fernández, C. Á., 1998. *Definición de una metodología para el desarrollo de sistemas multiagente.* s.l.:Departamento de Ingeniería de Sistemas Telemáticos Universidad Politécnica de Madrid.

Iglesias, C., Garijo, M., Gonzalez, J. & Velasco, J., s.f. A methodological proposal for multiagent systems development extending CommonKADS. *in Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop.*

Ishibashi, K. & Yano, M., 2005. A Proposal of Forwarding Method for Urgent Messages on an Ubiquitous Wireless Sensor Network. *6th Asia-Pacific Symposium on Information and Telecommunication Technologies,* pp. 293-298.

Jennings, . N. R., Sycara, K. & Wooldridge, M., 1998. A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems,* 01 03, 1(1), pp. 7-38.

Johnson, M., Heal, M., van de Ven, P. & Hayes, M., 2009. A comparative review of wireless sensor network mote technologies. *IEEE Sensors,* Octubre.pp. 1439-1442.

Kramp, T. y otros, 2008. *Mote Runner.* [En línea]
Available at: http://www.zurich.ibm.com/moterunner/publications.html
[Último acceso: 27 Noviembre 2010].

Lajara , R., Pelegrí-Sebastiá, J. & Perez Solano, J. J., 2010. Power Consumption Analysis of Operating Systems for Wireless Sensor Networks. *Sensors,* p. 18.

Lewis, F., 2004. Wireless Sensor Networs. En: *Smart Environments: Technologies, Protocols, and Applicationsed.* New York: D.J. Cook and S.K. Das, John Wiley, p. Capítulo 2.

Marsh, D., Tynan, R., O'Kane, D. & O'Hare, G. M. P., 2004. Autonomic wireless sensor networks. *Engineering Applications of Artificial Intelligence,* Octubre, 17(7), pp. 741-748.

Martínez, J. y otros, 2007. An approach for applying multi-agent technology into wireless sensor networks. *In Proceedings of the 2007 Euro American Conference on Telematics and information Systems EATIS '07. ACM,* 14-17 Mayo.pp. 1-8.

Massaguer, D., 2005. *Multi Mobile Agent Deployment in Wireless Sensor Networks.* Irvine: University of California.

Memsic Corporation, 2010. *IRIS.* [En línea]
Available at: http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=179%3Airis_moterunner
[Último acceso: 1 Noviembre 2010].

MEMSIC Inc., 2010. *Wireless Modules.* [En línea]
Available at: http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html
[Último acceso: 1 Noviembre 2012].

Moteiv Corporation, 2006. *Tmote Sky: Datasheet.* s.l.:http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf.

Muldoon, C., 2007. *An Agent Framework for Ubiquitous Services.* s.l.:University College Dublin for the degree of Doctor of Philosophy in the College of Engineering, Mathematical & Physical Sciences.

Muñoz, T., 2010. *Proyecto de tesis de maestría.* Medellín: s.n.

Muñoz, T., Montoya, A. & Ovalle, D., 2010. Análisis de Métricas para la Evaluación del Desempeño de WSNs enriquecidas con Agentes Inteligentes. En: *Tendencias en Ingeniería de Software e Inteligencia artificial.* s.l.:Universidad Nacional de Colombia-Sede Medellín.

National Instruments, 2009. [En línea]
Available at: http://zone.ni.com/devzone/cda/tut/p/id/8707
[Último acceso: 28 Septiembre 2010].

O'Hare, G., 2010. *Curso de Multi-Agent Systems(MAS) & Distributed Artificial Intelligence(DAI).* [En línea]
Available at: http://www.csi.ucd.ie/content/comp-30240

Oracle_Java.net, 2010. *Squawk.* [En línea]
Available at: https://squawk.dev.java.net/
[Último acceso: 27 Noviembre 2010].

Oracle, 2012. *Java Communications API.* [En línea]
Available at: http://www.oracle.com/technetwork/java/index-jsp-141752.html
[Último acceso: 20 Noviembre 2012].

OriginLab Corporation, 2012. *OriginLab: Data Analysis and Graphing Software.* [En línea]
Available at: http://www.originlab.com/
[Último acceso: Diciembre 2012].

Pastrana, M. & Ovalle, D., 2008. Método de comparación utilizando métricas de calidad para protocolos de negociación electrónica en sistemas Multi-Agente. *Dyna,* pp. 231-240.

Piedrahita Ospina, A., Montoya , A. & Ovalle Carranza, D., 2009. Integration model of mobile intelligent agents within Wireless Sensor Networks. *IEEE Latin-American Conference on Communications, LATINCOM '09,* 10-11 Septiembre.pp. 1-6.

Potdar, V., Sharif, A. & Chang, E., 2009. Wireless Sensor Networks: A Survey. *International Conference on Advanced Information Networking and Applications Workshops,* pp. 636-641.

Qi, H., Xu, Y. & Wang, X., 2003. Mobile-Agent-Based Collaborative Signal and Information Processing in Sensor Networks. *Proceeding of the IEEE,* pp. 1172-1183.

Rao, A. & Georgeff, M., 1995. BDI Agents: From Theory to Practice.

Restrepo, D., Montoya, A. & Ovalle, D., 2010. *A Multi-Agent Model to simulate Intelligent Wireless Sensor Networks.* s.l.:s.n.

Rodríguez, J., Torres, M. & González, E., 2007. La Metodología AOPOA. *Revista Avances en Sistemas e Informática,* 4(2), pp. 71-78.

Rogers, A., Jennings, N. & Corkill, D., 2009. Agent Technologies for Sensor Networks. *IEEE Intelligent Systems Journal, IEEE Computer Society,* pp. 13-17.

Russell, S. J. & Norvig, P., 2003. *Artificial Intelligence: A Modern Approach.* Segunda edición ed. New Jersey: Pearson Education, Inc..

Ruzzelli, A., Tynan, R., O'Grady, M. & O'Hare, G., 2006. Advances in Wireless Sensor Networks. En: *Encyclopedia of Mobile Computing & Commerce, (EMCC).* s.l.:s.n., p. Vol 1.

Safaric, S. & Malaric, K., 2006. ZigBee wireless standard. *48th International Symposium ELMAR-2006 focused on Multimedia Signal Processing and Communications,* Junio.pp. 259-262.

Sankaranarayanan, S., 2009. Intelligent agent based information routing in wireless body sensor Mesh Networks. *Conference on Wireless and Optical Communications Networks_IEEE,* pp. 1-5.

Sentilla Corporation, 2010. *Moteiv Hardware Product Transition.* [En línea]
Available at: http://www.sentilla.com/moteiv-transition.html
[Último acceso: 10 10 2010].

Silicon Laboratories Inc, 2010. *Silicon Laboratories Inc.* [En línea]
Available at:
http://www.silabs.com/support/pages/batterylifeestimator.aspx?cm_mmc=GlobalSpec-_-Newsletter-_-Specs+&+Techs-_-Battery+Life+Estimator
[Último acceso: 1 Diciembre 2010].

Sohraby, K., Minoli, D. & Znati, T., 2007. Wireless Sensor Networks – Technology, Protocols, and Applications. En: Canadá: WILEY-INTERSCIENCE, John Wiley & Sons, pp. 283-301.

Spot, S., 2010. [En línea]
Available at: http://www.sunspotworld.com

Squeak, 2010. . [En línea]
Available at: http://www.squeak.org/
[Último acceso: 27 Noviembre 2010].

Sudharsan, D., Adinarayana, J., Tripathy, A. & et al., 2012. GeoSense: A Multimode Information and Communication System. *ISRN Sensor Networks,* pp. 1-13.

Sun SPOT World, 2012. *Oracle Corporation.* [En línea]
Available at: http://www.sunspotworld.com
[Último acceso: 13 Noviembre 2012].

Sycara, K., 1998. *Multiagent Systems.* [En línea]
Available at: http://www.perada.eu/documents/articles-perspectives/multi-agent-systems.pdf
[Último acceso: 1 Abril 2009].

Thant, K. P. & Naing, T. T., 2005. A Migration Framework for Ubiquitous Computing Applied in Mobile Applications. *6th Asia-Pacific Symposium on Information and Telecommunication Technologies, APSITT 2005 Proceedings.,* pp. 213-218.

Tynan, R., Marsh, D., O'Kane, D. & O'Hare, G., 2005. Agents for Wireless Sensor Network Power Management. *International Conference Workshops on Parallel Processing (ICPPW'05),* pp. 413-418.

Tynan, R., Marsh, D., O'Kane, D. & O'Hare, G., 2005. Intelligent Agents for Wireless Sensor Networks. *International Conference on Autonomous Agents. Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems. AAMAS'05. ACM,* 25-29 July.pp. 1179-1180.

Tynan, R., O'Hare, G., Marsh, D. & O'Kane, D., 2005. Multi-Agente System Architectures for Wireless Sensor Networks. En: *Computational Science – ICCS 2005.* s.l.:Springer Berlin / Heidelberg, pp. 687-694.

Tynan, R., Ruzzelli, A. & O'Hare, G., 2005. A Methodology for the Deployment of Multi-Agent Systems on Wireless Sensor Networks. *Agent-Oriented Software Development Methodology (AOSDM'2005) Workshop, In Proceedings of 17th International Conference on Software Engineering and Knowledge Engineering (SEKE'05),* p. Julio.

Vieira, M. A. & Da Silva, D. C., 2003. Survey on Wireless Sensor Network Devices. *IEEE* , pp. 537-544.

Vlassis, N., 2007. A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning,* 1(1), pp. 1-71.

Weiser, M., 1991. The Computer for the 21st Century.

Wooldridge, M., 2002. *An Introduction to MultiAgent systems.* s.l.:John Wiley & Sons.

Wooldridge, M., Jennings, N. & Kinny, D., 1999. A methodology for agent-oriented analysis and design. *In Proceedings of the Third International Conference on Autonomous Agents (Agents 99),* p. 69–76.

Wooldridge, M., Jennings, N. & Kinny, D., 1999. The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Autonomous Agents and Multi-Agent Systems,* pp. 285-312.

Yasemin, A., 1999. INTELLIGENT AGENTS: A DATA MINING PERSPECTIVE. *Dallas.*

Zhao, F. & Guibas, L., 2004. *Wireless Sensor Networks: An Information Processing Approach.* s.l.:Elsevier/Morgan-Kaufmann.