



# **ESPE**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y  
TELECOMUNICACIONES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO EN ELECTRÓNICA Y  
TELECOMUNICACIONES**

**TEMA: DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE  
SISTEMA DE RADIOLOCALIZACIÓN MEDIANTE UNA RED DE  
SENSORES INALÁMBRICOS (WSN).**

**AUTOR: GALARZA HERRERA, ALEJANDRA ELIZABETH**

**DIRECTOR: ING. VIZCAINO, PATRICIO MSc.  
CODIRECTOR: ING. GUALSAQUÍ, MARCO MSc.**

**SANGOLQUÍ  
2016**

**CERTIFICADO****DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA****CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES****CERTIFICACIÓN**

Certifico que el trabajo de titulación, “**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE RADIOLOCALIZACIÓN MEDIANTE UNA RED DE SENSORES INALÁMBRICOS (WSN)**” realizado por la señorita **ALEJANDRA ELIZABETH GALARZA HERRERA**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a la señorita **ALEJANDRA ELIZABETH GALARZA HERRERA** para que lo sustente públicamente.

**Sangolquí, 8 de enero del 2016**

ING. VIZCAINO, PATRICIO MSc.

**DIRECTOR**

ING. GUALSAQUÍ, MARCO MSc.

**CODIRECTOR**

## AUTORÍA DE RESPONSABILIDAD



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

### AUTORÍA DE RESPONSABILIDAD

Yo, **ALEJANDRA ELIZABETH GALARZA HERRERA**, con cédula de identidad N° 1721833430, declaro que este trabajo de titulación "**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE RADIOLOCALIZACIÓN MEDIANTE UNA RED DE SENSORES INALÁMBRICOS (WSN)**" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 8 de enero del 2016

-----  
ALEJANDRA ELIZABETH GALARZA HERRERA  
C.C: 1721833430

**AUTORIZACIÓN (PUBLICACIÓN BIBLIOTECA VIRTUAL)****DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA****CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES****AUTORIZACIÓN**

Yo, **ALEJANDRA ELIZABETH GALARZA HERRERA**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación "**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE RADIOLOCALIZACIÓN MEDIANTE UNA RED DE SENSORES INALÁMBRICOS (WSN)**" cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

**Sangolquí, 8 de enero del 2016**

-----  
**ALEJANDRA ELIZABETH GALARZA HERRERA**  
C.C: 1721833430

## **DEDICATORIA**

Dedico este trabajo principalmente a Dios porque siempre ha estado conmigo, a mi mamá porque siempre me ha apoyado aun cuando no tenía la posibilidad de hacerlo y a mi familia porque siempre han creído en mí.

## **AGRADECIMIENTO**

Agradezco primeramente a Dios, porque me ha dado la vida, salud y todo lo que he necesitado para salir adelante. A mi mamá María Elena y a mis hermanas Diana y Fernanda, por toda su paciencia, sacrificio y por los valores inculcados que me han ayudado a crecer personal y profesionalmente. A mis amigos William, Caro, Alexis y Jenny por ser los mejores amigos que he podido encontrar y porque siempre han estado en los momentos más difíciles. A mi director Ing. Patricio Vizcaíno y codirector Marco Gualsaqui, por su paciencia, ayuda y orientación en la consecución de este trabajo.

## ÍNDICE DE CONTENIDO

### CARÁTULA

CERTIFICADO.....	ii
AUTORÍA DE RESPONSABILIDAD .....	iii
AUTORIZACIÓN (PUBLICACIÓN BIBLIOTECA VIRTUAL) .....	iv
DEDICATORIA .....	v
AGRADECIMIENTO .....	vi
ÍNDICE DE CONTENIDO .....	vii
ÍNDICE DE TABLAS .....	x
ÍNDICE DE FIGURAS.....	xii
ABSTRACT.....	xvi
GLOSARIO .....	xvii
CAPITULO 1	
PRESENTACIÓN.....	1
1.1. Antecedentes.....	1
1.2. Justificación e Importancia.....	3
1.3. Alcance del Proyecto .....	4
1.4. Objetivos.....	5
1.4.1. General .....	5
1.4.2. Específicos.....	5
CAPITULO 2	
MARCO TEÓRICO .....	6
2.1. Redes de Sensores Inalámbricas (WSN) .....	6
2.1.1. Introducción.....	6
2.1.2. Estándar IEEE 802.15.4.....	7
2.1.3. Estándar ZigBee.....	15
2.1.4. Aplicaciones .....	21
2.2. Radiolocalización con WSN.....	25

2.2.1. Introducción.....	25
2.2.2. Algoritmos de Radiolocalización con WSN .....	25
CAPITULO 3	
MÉTODOS Y MATERIALES.....	28
3.1. Análisis del Kit de Sensores Inalámbricos MEMSIC 2.4ghz, ZigBee (MEMSIC, 2006) .....	28
3.1.1. Hardware MEMSIC .....	28
3.2. Sistema Operativo TinyOS .....	31
3.3. Lenguaje de Programación NesC.....	35
3.4. Técnicas de Estimación de la Posición (Polo, 2013) .....	36
3.4.1. Trilateración .....	36
3.4.2. Triangulación.....	38
3.4.3. Multilateración .....	39
3.4.4. Proximidad .....	39
3.5. Algoritmos de Radiolocalización con WSN Basado en Rango (Poellabauer, 2010) .....	40
3.5.1. Variable RSSI.....	40
3.5.2. Algoritmo ToA .....	42
3.5.3. Algoritmo AoA. ....	42
3.6. Algoritmos de Radiolocalización con WSN Libre de Rango .....	44
3.6.1. Algoritmo DV-HOP .....	44
3.6.2. Algoritmo MDS – MAP .....	44
3.6.3. Algoritmo ELA. ....	45
3.7. Selección del Algoritmo para Implementación del Prototipo de Sistema de Radiolocalización WSN.....	46
3.8. Modelamiento y Metodología de Localización .....	48
CAPITULO 4	
DISEÑO E IMPLEMENTACIÓN.....	52
4.1. Criterios de Diseño .....	52
4.2. Implementación del Prototipo de Sistema de Radiolocalización WSN .....	56

4.2.1.	Descripción del Algoritmo en Código NESC .....	58
4.2.2.	Medición RSSI (ATMEL, 2009) .....	71
4.3.	Aplicación de Monitorización de la Red .....	73
4.4.	Calibración Escenarios .....	79
4.5.	Definición de Variables .....	81
CAPITULO 5		
ANÁLISIS DE RESULTADOS.....		85
5.1.	Análisis Primer Cuadrante .....	86
5.2.	Análisis Segundo Cuadrante .....	94
5.3.	Análisis Tercer cuadrante .....	102
5.4.	Análisis Cuarto Cuadrante .....	109
CONCLUSIONES Y RECOMENDACIONES .....		121
6.1.	Conclusiones .....	121
6.2.	Recomendaciones .....	123
6.3.	Descripción de Utilidad del Proyecto para Trabajos Futuros .....	124
Trabajos Citados.....		126

## ÍNDICE DE TABLAS

Tabla 1 Bandas de Frecuencia 802.15.4 .....	10
Tabla 2 Parametros Técnicos 802.15.4 .....	11
Tabla 3 Exponente de Pérdidas.....	54
Tabla 4 Toma de Muestras .....	78
Tabla 5 Tiempo de Espera del Sistema .....	79
Tabla 6 Coordenadas Motas Fijas .....	82
Tabla 7 Desconocida .....	83
Tabla 8 RSSI a un metro de Distancia [m].....	83
Tabla 9 Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 1- 1er Cuadrante) .....	86
Tabla 10 Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 2-1er Cuadrante) .....	87
Tabla 11 Error de Posición Mínimo y Máximo (1er Cuadrante) .....	88
Tabla 12 Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 1-1er Cuadrante).....	89
Tabla 13 Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 2-1er Cuadrante).....	90
Tabla 14 Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 1-2do Cuadrante) .....	95
Tabla 15 Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 2-2do Cuadrante) .....	95
Tabla 16 Error de Posición Mínimo y Máximo (2do Cuadrante) .....	96
Tabla 17 Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 1-2do Cuadrante).....	97
Tabla 18 Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 2-2do Cuadrante) .....	98
Tabla 19 Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 1- 3er Cuadrante).....	102

Tabla 20 Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 2- 3er Cuadrante) .....	103
Tabla 21 Error de Posición Mínimo y Máximo (3er Cuadrante)) .....	103
Tabla 22 Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 1-3er Cuadrante).....	105
Tabla 23 Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 2-3er Cuadrante).....	105
Tabla 24 Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 1- 4to Cuadrante) .....	109
Tabla 25 Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 2- 4to Cuadrante) .....	110
Tabla 26 Error de Posición Mínimo y Máximo (4to Cuadrante).....	110
Tabla 27 Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 1-4to Cuadrante).....	112
Tabla 28 Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 2-4to Cuadrante).....	112
Tabla 29 Condición máximo error (Escenario 1).....	116
Tabla 30 Condición máximo error (Escenario 2).....	116
Tabla 31 Condición mínimo error (Escenario 1).....	117
Tabla 32 Condición mínimo error (Escenario 2).....	117

## ÍNDICE DE FIGURAS

Figura 1 Estructura de Canales .....	9
Figura 2 Relación de la IEEE 802.15.4 con OSI .....	13
Figura 3 Dispositivos ZigBee según su papel en la Red.....	18
Figura 4 Topologías de Redes ZigBee .....	20
Figura 5 Aplicaciones de las WSN.....	24
Figura 6 Mota Iris .....	29
Figura 7 Diagrama de Bloque Mota Iris .....	29
Figura 8 Conector de expansión de 51 pines .....	30
Figura 9 Gateway MIB520 .....	31
Figura 10 Modelo de componentes de TinyOS.....	34
Figura 11 Ejemplo de código NesC .....	36
Figura 12 Escenario de Trilateración .....	37
Figura 13 Escenario de Triangulación dos dimensiones.....	39
Figura 14 Esquema de Comunicaciones .....	53
Figura 15 Aplicación Mota Desconocida.....	61
Figura 16 Aplicación Mota Fija.....	61
Figura 17 Aplicación Mota Base .....	62
Figura 18 Interfaces .....	64
Figura 19 Ventana Principal.....	74
Figura 20 Cargar Datos .....	75
Figura 21 Ej. Distancia Euclidiana entre Posición Estimada.....	76
Figura 22 Ej. Trayectoria.....	77
Figura 23 Ej. Error en X e Y.....	77
Figura 24 Escenario 1 (sobre el césped, .....	80
Figura 25 Escenario 2 (sobre el piso) .....	80
Figura 26 Eje de Coordenadas .....	81
Figura 27 Posición Motas .....	82

Figura 28 X[m] (RSSI a un metro de Distancia) .....	84
Figura 29 Error Eje X y Y (1er Cuadrante) .....	88
Figura 30 Distancia[m]- RSSI [dBm] (Escenario 1-1er Cuadrante) .....	91
Figura 31 Distancia[m]-RSSI [dBm] (Escenario 2-1er Cuadrante) .....	91
Figura 32 Error de Posición al utilizar 3 y 4 motas (Escenario 1-1er Cuadrante).....	93
Figura 33 Error de Posición al utilizar 3 y 4 motas (Escenario 2-1er Cuadrante).....	94
Figura 34 Error Eje X y Y (2do Cuadrante) .....	96
Figura 35 Distancia[m]-RSSI [dBm] (Escenario 1-2do Cuadrante) .....	99
Figura 36 Distancia[m]-RSSI [dBm] (Escenario 2-2do Cuadrante) .....	99
Figura 37 Error de Posición al utilizar 3 y 4 motas (Escenario 1-2do Cuadrante).....	100
Figura 38 Error de Posición al utilizar 3 y 4 motas (Escenario 2-2do Cuadrante).....	101
Figura 39 Error Eje X y Y (3er Cuadrante) .....	104
Figura 40 Distancia[m]-RSSI[dBm] (Escenario 1-3er Cuadrante) .....	106
Figura 41 Distancia[m]-RSSI[dBm] (Escenario 2-3er Cuadrante) .....	106
Figura 42 Error de Posición al utilizar 3 y 4 motas (Escenario 1-3er Cuadrante).....	107
Figura 43 Error de Posición al utilizar 3 y 4 motas (Escenario 2-3er Cuadrante).....	108
Figura 44 Error Eje X y Y (4to Cuadrante) .....	111
Figura 45 Distancia[m]-RSSI [dBm] (Escenario 1-4to Cuadrante) .....	113
Figura 46 Distancia[m]-RSSI [dBm] (Escenario 2-4to Cuadrante) .....	113
Figura 47 Error de Posición al utilizar 3 y 4 motas (Escenario 1-4to Cuadrante).....	114
Figura 48 Error de Posición al utilizar 3 y 4 motas (Escenario 2 -4to Cuadrante).....	115
Figura 49 Error de Posición en función de n .....	118

Figura 50 Condición Mínimo Error de Posición.....119  
Figura 51 Condición Máximo Error de Posición .....120

## RESUMEN

La radiolocalización es una de las aplicaciones de las redes de sensores inalámbricos (WSN), más requeridas por la industria, porque se basa en dispositivos de bajo costo que poseen facilidades de instalación, seguridad y precisión en las mediciones, por lo que son capaces de reemplazar redes de alto costo y complejidad. En el presente proyecto se realizó el diseño y la implementación de un prototipo de sistema de radiolocalización, utilizando los equipos MEMSIC que trabajan en la banda de 2.4GHz. Con el conocimiento de la posición se podrán desarrollar aplicaciones como monitoreo del hábitat de los animales, resguardo de la salud humana, entre otros. En primer lugar se realizó el estudio del marco teórico de las WSN y su potencial para el diseño de aplicaciones como radiolocalización. Además se realizó la verificación de la técnica Trilateración-RSSI y se implementó la WSN utilizando el lenguaje *NesC*, que corre bajo *TinyOS*, para la adquisición-procesamiento de datos y la configuración de la interfaz de usuario se utilizó Matlab. Finalmente se realizaron pruebas al prototipo de sistema y se analizaron los resultados obtenidos del error, media y varianza, en escenarios reales, el primer escenario a nivel de piso y el segundo sobre el césped a 0.75 m de altura. Se determinó que el primer escenario presentó mejores resultados que el segundo, debido a que los nodos de referencia y el nodo que se desconoce su posición no estuvieron a la misma altura que el gateway lo que ocasionó mayores errores en la determinación de la posición en el plano X e Y.

### **PALABRAS CLAVE:**

- WSN
- NESC/TINYOS
- RADIOLOCALIZACIÓN

## ABSTRACT

The radiolocation is one of the applications of wireless sensor networks (WSN), most required by industry, because they are based on low cost devices, that have facilities installation, safety and accuracy in measurements, so they are able to replace high cost and complexity networks. This project involves the design and implementation of a radiolocation prototype system, using MEMSIC equipment that works in the 2.4GHz band. With knowledge of the position can be developed monitoring applications, such as animal habitats, protection of human health, among others. First was studied the theoretical framework of the WSN and its potential for radiolocation application design. In addition was performed the verification of Trilateration-RSSI technique and the WSN was implemented using the language NesC, which runs on TinyOS, for the acquisition of data processing and configuration of the user interface was used Matlab. Finally, the tests of prototype of system were performed to obtained results of error, mean and variance, on real scenarios. The first scenario was at floor level and the second was on the lawn to 0.75 m in height. Because the reference nodes and node position is unknown, were not up to the same height as the gateway, caused inconsistencies in the data submitted.

### KEYWORDS:

- WSN
- NESC/TINYOS
- RADIOLOCATION

## GLOSARIO

### B

**Beacon** Es un dispositivo de bajo consumo, que emite una señal broadcast.

**BoudRate** Unidades de señal por segundo.

**BPSK** Binary Phase Shift Keying, modulación de desplazamiento de fase de 2 símbolos.

**Buffer** Instrumento digital reservado para el almacenamiento temporal de información digital.

### C

**COM** Puerto USB, es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos.

**CPU** Central Processing Unit, Unidad Central de Procesamiento.

**CSMA/CA** Carrier Sense Multiple Access with Collision Avoidance, acceso múltiple con escucha de portadora y evasión de colisiones.

### D

**dBm** Decibelio Milivatio, es una unidad de medida de potencia expresada en decibelios (dB) relativa a un milivatio (mW).

**Dispersión** Es el fenómeno de separación de las ondas de distinta frecuencia al atravesar un material. Todos los medios

materiales son más o menos dispersivos, y la dispersión afecta a todas las ondas.

- DLL** Data Link Layer, Capa de Enlace de Datos.
- DSSS** Direct Sequence Spread Spectrum, espectro ensanchado por secuencia directa.
- F**
- FHSS** Frequency Hopping Spread Spectrum, espectro ensanchado por salto de frecuencia.
- FIFO** First In, First Out, es un concepto utilizado en estructuras de datos y teoría de colas. Guarda analogía con las personas que esperan en una cola y van siendo atendidas en el orden en que llegaron, es decir, que 'la primera persona que entra es la primera persona que sale'.
- Frame** Es una trama de datos.
- G**
- Gateway** Puerta de enlace, equipo para interconectar redes.
- GPS** Global Position System, Sistema de Posicionamiento Global.
- GUI** Graphical User Interface, Interfaz de Usuario.

**H**

**Half Dúplex** Es un modo de envío de información es bidireccional pero no simultáneo.

**I**

**I2C** Es un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de "inteligencia", sólo requiere de dos líneas de señal y un común o masa.

**IEEE 802.2** Estándar que define el control de enlace lógico (LLC), que es la parte superior de la capa enlace en las redes de área local.

**ISM** Banda industrial, médica y científica (Industrial Scientific and Medical).

**L**

**LLC** Logical Link Control.

**LOS** Line of Sight, Línea de Visión Directa.

**M**

**MAC** Medium Access Control.

**MCPS-SAP** MAC common part sublayer, para datos de la MAC.

**Mesh** Topología de Malla, los nodos se pueden conectar a múltiples nodos en el sistema y pasar los datos por el camino disponible de mayor confiabilidad.

**MLME-SAP** MAC layer management entity, para servicios de administración.

**Modem** Modulator demodulator, es un dispositivo que convierte las señales digitales en analógicas y analógicas en digitales.

**MSK** Minimum Shift Keying, modulación por desplazamiento mínimo.

**Multitrayecto** Es un fenómeno consistente en la propagación de una onda por varios caminos diferentes

## N

**NesC** Es un lenguaje de programación basado en C, orientado a sistemas embebidos que incorporan el manejo de red.

**NLOS** Non Line of Sight, Sin Línea de Visión.

## O

**Open source** Es la expresión con la que se conoce al software o hardware distribuido y desarrollado libremente.

**OSI** Open System Interconnection, modelo de interconexión de sistemas abiertos.

**P**

**Pathloss** Pérdida en la trayectoria, es causado por la disipación de potencia, radiada por el transmisor, debido a los efectos del canal de propagación.

**PHY** Physical layer of the OSI model, es la interfaz entre la MAC y el medio físico.

**PN** Pseudo Noise, secuencia de ruido falso.

**Q**

**QPSK** Quadrature Phase Shift Keying modulation, modulación por desplazamiento de fase en cuadratura escalonada.

**R**

**Reflexión** Es el cambio de dirección de una onda, que al entrar en contacto con la superficie de separación entre dos medios cambiantes, regresa al punto donde se originó.

**S**

**SFD** Start of Frame Delimiter, detecta los bits de inicio de trama.

**Shadowing** Es un fenómeno causado por los obstáculos que se encuentran entre el transmisor y el receptor y produce una atenuación de la señal a causa de la absorción y reflexión.

<b>SPI</b>	Serial Peripheral Interface, es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.
<b>Stack</b>	Es una lista ordenada o estructura de datos en la que el modo de acceso a sus elementos es de tipo LIFO (del inglés Last In First Out, último en entrar, primero en salir) que permite almacenar y recuperar datos.
<b>T</b>	
<b>TIC</b>	Tecnologías de la información y la comunicación.
<b>TinyOS</b>	Es un sistema operativo libre y de código abierto (open source), basado en un modelo de programación característico de los sistemas embebidos (modelo de componentes), diseñado para redes de sensores inalámbricos.
<b>U</b>	
<b>UART</b>	Universal Asynchronous Receiver-Transmitter, es el dispositivo que controla los puertos y dispositivos serie.
<b>UWB</b>	Ultra Wideband, Banda Ultra Ancha.
<b>W</b>	
<b>WiFi</b>	Es un mecanismo de conexión de dispositivos electrónicos de forma inalámbrica.
<b>WPAN</b>	Wireless Personal Área Network, Red Inalámbrica de Área Personal o red de área personal inalámbrica.

**WSN** Wireless Sensors Network, Red de Sensores Inalámbricos.

## Z

**ZigBee** Es un estándar inalámbrico normalizado para la conexión de una red WPAN, que conlleva un conjunto de protocolos de alto nivel de comunicación inalámbrica, diseñado para soportar aplicaciones que requieren bajo costo y consumo de energía.

# CAPITULO 1

## PRESENTACIÓN

### 1.1. Antecedentes

El aparecimiento de nuevas tecnologías de miniaturización ha permitido el desarrollo de equipos de mejor portabilidad como son microprocesadores y microcontroladores de menor tamaño. Esto ha permitido el desarrollo de hardware de mejores capacidades tanto en velocidad como en rendimiento que se ajustan de manera versátil a los diferentes requerimientos que pueden existir para demandas específicas en el mercado. Un ejemplo de esto ha sido el aparecimiento de una amplia variedad de sensores que se comunican de manera inalámbrica conformando a lo que se conoce actualmente como Redes de Sensores Inalámbricas o WSN (Wireless Sensors Network) (Figuera, 2009).

Las WSN presentan una gran ventaja de versatilidad en su funcionamiento, además de ser fácilmente transportables a diferentes puntos de un escenario cualquiera, tanto interior como exterior. El aparecimiento de esta tecnología también aprovecha las bondades de las técnicas digitales empleadas en los sistemas de comunicación inalámbrica.

Las primeras aplicaciones con las WSN nacieron en organizaciones militares y desde entonces su aparecimiento en el mercado ha ido en un marcado ascenso por lo que las puede considerar como una tecnología emergente con un gran número de aplicaciones potenciales en muchos campos tales como: medicina, seguridad, agricultura, radiolocalización, entre otros. Las principales ventajas que presentan las WSN son: su tiempo de vida, el rango de cobertura de los sensores, costos, facilidad de instalación, buenos tiempos de

respuesta, bajo consumo de potencia, precisión y frecuencia de las mediciones (Guidas, 2005).

Una de las aplicaciones con resultados más prometedores en las WSN es la *radiolocalización*, en muchos casos es útil e incluso necesario que un nodo conozca su posición o la ubicación de un objetivo, ya sea móvil o fijo, dentro del rango de alcance, gracias a esto se puede realizar un rastreo de un objetivo que puede proveer información variada.

Hasta el momento el sistema más usado para posicionamiento es el GPS (Global Position System), sin embargo las redes WSN ofrecen una amplia variedad de funcionalidades y ventajas como sus bajos costos y su mayor precisión en ambientes cerrados (Tseng, 2007). Además los equipos de sensores inalámbricos pueden medir diferentes variables ambientales facilitando el estudio de distintas variables en el entorno.

Los sistemas de *radiolocalización* han sido estudiados de manera amplia en el campo teórico por lo que se han desarrollado un gran número de algoritmos que podrían implementarse en un ambiente real, el cual es el objetivo final de este proyecto. Existen dos esquemas de algoritmos de *radiolocalización* que se pueden utilizar para estimar las posiciones de los nodos a partir de la medición de las distancias de los mismos. El esquema *libre de rango* y el esquema *basado rango*. Los algoritmos basados en rango miden la distancia mediante hardware especializado para estimar sus posiciones. Por otro lado, los esquemas libres de rango se basan enteramente en la información de conectividad. Así, los algoritmos de *radiolocalización* deben ser analizados de manera minuciosa para escoger un modelo adecuado para el desarrollo del prototipo de sistema (Fidan & Mao, 2006).

## 1.2. Justificación e Importancia

Desde tiempos remotos el hombre se interesó por orientarse correctamente durante incursiones por tierra o mar. Posteriormente aumentaron las necesidades de localización, por lo que esta tecnología impulsada principalmente por la industria militar ha desarrollado varios diseños y creaciones de sofisticados sistemas que calculan las coordenadas de un objetivo específico las cuales también son imprescindibles para obtener la posición y realizar un rastreo de distintos móviles como: personas, automóviles, barcos, aviones, fauna animal y demás puntos de interés.

Actualmente en el mundo existe una diversidad de sistemas enfocados a la *radiolocalización* como el sistema de posicionamiento global GPS, sistema satelital Argos, sistema de posicionamiento Cambridge, entre otros (Fidan & Mao, 2006). Sin embargo, la *radiolocalización* mediante WSN provee distintas ventajas frente a estos sistemas como su bajo costo, facilidades de instalación, seguridad, precisión en las mediciones, entre otros, lo que le convierte en una red sencilla y eficiente, capaz de reemplazar redes de alto costo y con alta complejidad. Además las WSN pueden de brindar información de distintas variables acerca del entorno donde se localizan, como por ejemplo: temperatura, humedad, entre otros. Proveyendo mayores prestaciones para aplicaciones específicas.

La mayor parte de los algoritmos de *radiolocalización* no han sido comparados ni puestos a prueba en equipos reales (Fidan & Mao, 2006). Por lo tanto, este proyecto se enfocará en los resultados obtenidos de la implementación, tratando de corroborar los resultados concluidos en estudios teóricos previos, esperando que los resultados del proyecto sirvan de ayuda para futuras investigaciones que conlleven aplicaciones de *radiolocalización* con WSN en el Departamento de Eléctrica y Electrónica.

### 1.3. Alcance del Proyecto

En el presente proyecto se realizará el diseño e implementación de un prototipo de sistema de *radiolocalización* mediante una red de sensores Inalámbricos (WSN) para objetivos fijos o móviles. En primer lugar se propone realizar el análisis de las características, funcionalidades, ventajas y desventajas de los esquemas y algoritmos basados en *radiolocalización*, para verificar cuales son los más convenientes y cuáles nos podrán brindar las mejores prestaciones a nivel de aplicación.

Una vez hecho el análisis correspondiente, se procederá a efectuar la implementación del prototipo de sistema de *radiolocalización* en ambiente de laboratorio, tomando en cuenta las características técnicas de los equipos MEMSIC disponibles en el laboratorio de Radiofrecuencia. El lenguaje de programación usado en las redes de sensores WSN es *NesC*, que es una variante del popular lenguaje de programación *C*, pero diseñado para trabajar con las especificaciones de hardware de los sensores inalámbricos que corren bajo el sistema *TinyOS*.

Se procederá a realizar los respectivos estudios del error de posicionamiento de los valores tomados de la implementación que ayudarán a corroborar si el prototipo del sistema de *radiolocalización* es adecuado y cumple con las expectativas de diseño. Finalmente las conclusiones obtenidas en el proyecto, serán expuestas para ayudar a investigaciones futuras referentes al tema.

## 1.4. Objetivos

### 1.4.1. General

- Diseño e implementación de un prototipo de sistema de radiolocalización mediante una Red de Sensores Inalámbricos (WSN).

### 1.4.2. Específicos

- Determinar el marco teórico de las redes de sensores inalámbricos (WSN), la importancia y su potencial para el diseño de varias aplicaciones como *radiolocalización*.
- Realizar el estudio de los algoritmos más importantes de *radiolocalización* (Trilateración-RSSI, ToA, AoA ELA, entre otros), con la finalidad de seleccionar uno de ellos para la implementación.
- Realizar las pruebas necesarias para evaluar el comportamiento del prototipo de sistema de *radiolocalización* en escenarios reales.
- Realizar el análisis de los resultados obtenidos de la implementación con la finalidad de generar propuestas para el mejoramiento del prototipo de sistema y tener una base para realizar aplicaciones similares en un futuro.

## **CAPITULO 2**

### **MARCO TEÓRICO**

#### **2.1. Redes de Sensores Inalámbricas (WSN)**

##### **2.1.1. Introducción**

Hoy en día, los sensores pueden encontrarse en un gran número de sistemas y dispositivos electrónicos aplicados a diferentes áreas como la industria automotriz, robótica, industria aeroespacial, medicina, industria de manufactura, entre otros. La mayor parte de estos sensores carecen de la capacidad de procesar y analizar datos, limitándose a funcionar solo como dispositivos que pueden detectar magnitudes físicas o químicas, llamadas variables de instrumentación y transformarlas en variables eléctricas (García A. , 2009).

Sin embargo, está surgiendo una nueva generación de sensores a larga escala, capaces de organizarse a sí mismos, coordinarse e interconectarse de forma inalámbrica para llevar a cabo distintas funciones, de esta manera nacen las WSN que son una tecnología emergente de redes de tipo PAN (Personal Área Network) que pueden ser utilizadas en una amplia gama de aplicaciones gracias a las prestaciones que presentan.

Las WSN han evolucionado para ser capaces de resolver problemas relacionados con el consumo de potencia, tamaño de equipos, alto costo de implementación y mantenimiento. Esta clase de redes se caracterizan por su facilidad de despliegue y por ser auto configurables, pudiendo convertirse en todo momento en emisor-receptor, ofrecer servicios de encaminamiento entre nodos sin visión directa, así como registrar datos referentes a los sensores locales de cada nodo. Otra de sus características es su gestión eficiente de la

energía, que les permite obtener una alta tasa de autonomía que las hacen plenamente operativas.

Las WSN están conformadas por un gateway que provee conectividad inalámbrica y nodos distribuidos que intercambian información entre sí, sin necesidad de cables y mediante un protocolo de comunicación preestablecido. Cada nodo está compuesto por una unidad central de proceso (CPU), memoria, dispositivos de radiofrecuencia y un sensor capaz de monitorear alguna variable física dentro de su área de operación que recolectar datos de su ambiente como como cantidad de luz, temperatura, humedad, vibraciones y otros factores ambientales. Los nodos presentan capacidades limitadas de almacenamiento, procesamiento y comunicación, además disponen de soporte limitado para desarrollo de software (Maroto, 2010) , todas estas limitaciones se deben tomar en cuenta para el diseño e implementación del prototipo de sistema.

#### **2.1.2. Estándar IEEE 802.15.4**

IEEE 802.15.4 es un estándar que define el nivel físico (Physical Layer, PHY) y el control de acceso al medio (Medium Access Control, MAC), para dar servicio a un tipo específico de red inalámbrica de área personal (Wireless Personal Área Network, WPAN). El estándar está destinado a la interconexión de dispositivos inalámbricos en ambientes de oficinas, laboratorios y hogares a bajas tasas de transmisión, donde se enfatiza el bajo costo en la comunicación entre nodos cercanos y sin infraestructura (Adams, 2006).

En el año 2000, dos grupos especialistas en estándares (ZigBee y el grupo 15 de trabajo IEEE 802) realizaron varios estudios, con el objetivo de estandarizar redes inalámbricas de bajo consumo de energía y bajos costos, para ser utilizadas en ambientes industriales y caseros (Freescale, 2007). Para

este tipo de aplicaciones, se descartó Wi-Fi por ser demasiado complejo y por soportar anchos de banda mayores del que es necesario para ser utilizados en sensores. La tecnología Bluetooth fue inicialmente considerada como una posibilidad, sin embargo los sistemas infrarrojos requieren línea de visión directa, lo que no es siempre posible y fue considerada como tecnología demasiado complicada y costosa para satisfacer los requerimientos de conexión.

Esto abrió la puerta a un nuevo estándar IEEE 802.15.4 que puede operar en la banda industrial, médica y científica (Industrial Scientific and Medical, ISM), disponible a nivel mundial, soportando tasas de transmisión de datos de hasta 250 kbps en radios de 10 a 100 metros. Diseñada para ser un complemento de tecnologías inalámbricas como Bluetooth, Wi-Fi y ultra-wideband (UWB), dirigido comercialmente a aplicaciones de medición donde las conexiones cableadas no son posibles, y donde el bajo costo y consumo de energía son requeridos (Jiménez, 2007).

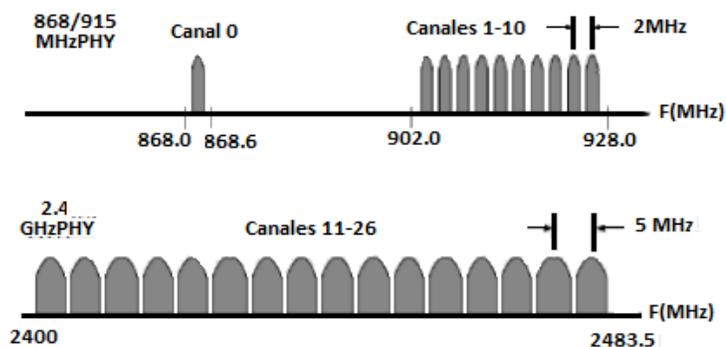
Entre los aspectos más importantes de este estándar se encuentran, la flexibilidad que posee la red, adecuación de su uso para tiempo real por medio de slots de tiempo garantizados, evasión de colisiones por CSMA/CA (Adams, 2006), soporte integrado a las comunicaciones seguras y disminución en la complejidad de la instalación porque posee gran capacidad de integrar componentes inalámbricos de una forma menos costosa. También se incluyen funciones de control del consumo de energía como calidad del enlace y detección.

Los dispositivos se relacionan entre sí a través de una red inalámbrica sencilla. La definición de los niveles se basa en el modelo OSI, aunque los niveles inferiores se definen en el estándar. Se prevé la interacción con el resto de niveles, posiblemente por medio de un subnivel de control de enlace lógico basado en IEEE 802.2, que acceda a MAC a través de un

subnivel de convergencia. La implementación puede basarse en dispositivos externos o integrarlo todo en dispositivos autónomos.

## Capa Física

La capa física o PHY provee servicios de transmisión de datos sobre el medio físico, así como la interfaz con la entidad de gestión, por medio de la cual se puede acceder a todos los servicios de gestión de nivel y mantener una base de datos con información de redes PAN relacionadas. De esta forma, la capa física controla el transceptor de radiofrecuencia y realiza la selección de canales junto con el control de consumo y de la señal. Como se muestra en la Figura 1, la capa física opera en una de tres posibles bandas de frecuencia de uso no regulado y se definen 27 canales entre las tres bandas, numeradas desde 0 a 26.



**Figura 1 Estructura de Canales**

Fuente: (González, 2009)

La PHY de los 868/915 MHz soporta un Canal (Canal 0) entre los 868 y los 868.6 MHz, y diez canales (Canales 1-10) entre los 902.0 y 928.0 MHz. Existe una separación entre las frecuencias centrales de cada Canal de 2 MHz por lo que se puede utilizar el mismo hardware para ambas bandas, reduciendo los costos en manufactura. La PHY de los 2.4 GHz soporta 16 canales

(Canales 11 – 26) entre los 2.4 y los 2.4835 GHz. La separación entre las frecuencias centrales de cada Canal es de 5 MHz, permitiendo facilitar los requerimientos de filtrado en la transmisión y recepción.

Como se muestra en la Tabla 1, la PHY de los 868/915 MHz especifica operaciones en la banda de 865 MHz en Europa y 915 MHz en la banda ISM en Estados Unidos, ofreciendo una alternativa a la congestión creciente asociadas a la banda de 2.4 GHz permitiendo mayores rangos por enlace debido a que existen menores pérdidas de propagación. Sin embargo la PHY de los 2.4 GHz, especifica operación en la banda ISM disponible a nivel mundial proporcionando ventajas en términos más amplios de mercado y costos de manufactura más bajos.

**Tabla 1**

**Bandas de Frecuencia 802.15.4**

PHY	Banda (MHz)	Lugar	Canales
<b>868/915 MHz</b>	868-868.8	Europa	Permite un canal de comunicación en versión 2003, extendido a tres en la revisión de 2006.
	902-928	Estados Unidos	Permite hasta diez canales en versión 2003, extendidos a treinta en 2006.
<b>2.4 GHz</b>	2400-2483.5	Todo el mundo	Uso en todo el mundo, hasta dieciséis canales en las versiones 2003 - 2006.

Fuente: (Adams, 2006)

Como se muestra en la Tabla 2, la PHY de los 868/915 MHz ofrece velocidades de transmisión de 20 kb/s y 40 kb/s respectivamente. En cuanto a modulación se basa en Espectro Ensanchado por Secuencia Directa (Direct Sequence Spread Spectrum, DSSS), donde cada bit transmitido se representa por un pulso de código DSSS de máxima longitud de secuencia (secuencia  $m$ ). Los datos binarios son codificados al multiplicar cada secuencia  $m$  por +1 o -1, y la secuencia del pulso de código resultante se modula dentro de la portadora

utilizando BPSK (binary phase shift keying). Antes de la modulación se utiliza una codificación de datos diferencial para permitir una recepción coherente de baja complejidad.

**Tabla 2**  
**Parámetros Técnicos 802.15.4**

PHY	Parámetros de los Datos			Parámetros del Chip		
	Banda (MHz)	Velocidad de bits (kbps)	Velocidad de símbolo (kbsp)	Modulación	Velocidad de Chip (kchips/s)	Modulación
868/915 MHz	868-868.8	20	20	BPSK	300	BPSK
	902-928	40	40	BPSK	600	BPSK
2.4 GHz	2400-2483.5	250	62.5	16-Semi orthogonal	2000	O-QPSK

Fuente: (Adams, 2006)

La PHY de los 2.4 GHz presenta una velocidad de transmisión de 250 kb/s, y se lo atribuye a un mayor orden en la modulación, en la cual cada símbolo representa múltiples bits. Utiliza una técnica semi-ortogonal basada en DSSS. Los datos binarios son agrupados en símbolos de 4 bits, y cada símbolo especifica una de las 16 secuencias de transmisión semi-ortogonales de código de pseudo-ruído (PN). Las secuencias de PN son concatenadas para que sean datos de símbolos exitosos, y la secuencia agregada al chip es modulada en la portadora utilizando MSK (minimum shift keying). El uso de símbolos casi ortogonales simplifica la implementación a cambio de un desempeño ligeramente menor (< 0.5 dB).

En términos de energía requerida por bit, la señalización ortogonal mejora su desempeño en 2 dB que BPSK diferencial. Sin embargo, en términos de sensibilidad de recepción, la PHY 868/915 PHY tiene una ventaja de 6 a 8 dB debido a que tiene velocidades de transmisión más bajas. En ambos casos

las pérdidas de implementación debido a la sincronización, forma del pulso, simplificaciones en el detector, y demás resultan en desviaciones en sus curvas óptimas de detección.

Las diferentes velocidades de transmisión se pueden usar para una gran variedad de aplicaciones como por ejemplo, en la PHY de los 868/915 MHz se puede ocupar para lograr mayor sensibilidad y mayores áreas de cobertura, con lo que se reduce el número de nodos requeridos para cubrir un área geográfica, mientras que la velocidad de transmisión en la PHY de los 2.4 GHz se puede utilizar para conseguir salidas superiores y de poca latencia.

Las especificaciones actuales de sensibilidad del estándar IEEE 802.15.4 son de -92 dBm para la PHY de los 868-915 MHz y -85 dBm para la PHY de los 2.4 GHz. Los valores de sensibilidad contienen suficiente margen para las tolerancias que se solicitan debido a las imperfecciones en la fabricación, de la misma manera que permite implementar aplicaciones de bajo costo.

La sensibilidad del receptor y la potencia del transmisor están en función del rango deseado. El estándar IEEE 802.15.4 establece que cada dispositivo debe de ser capaz de transmitir a una potencia de transmisión de al menos 1 mW, pero esta puede ser mayor o menor dependiendo de las necesidades de la aplicación para aprovechar de mejor forma la energía.

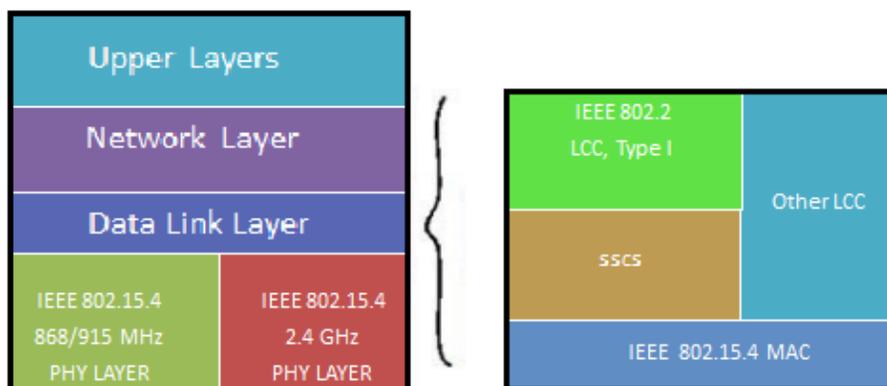
Para dispositivos que transmiten a una potencia de 1mW se esperan que cubran un rango de entre 10 a 20 m; sin embargo, con una buena sensibilidad y un incremento moderado en la potencia de transmisión, una red con topología tipo estrella puede proporcionar una cobertura total para toda una casa. Para aplicaciones que requieran mayor tiempo de latencia, la topología tipo mesh ofrece una opción atractiva para buenas coberturas del hogar, dado

que cada dispositivo solo necesita suficiente energía para comunicarse con su vecino más cercano.

Sin embargo en el estándar IEEE 802.15.4, uno de los principales requerimientos es la larga duración en las baterías, esto se logra con poca energía en la transmisión y muy pocos ciclos de servicio. Los dispositivos se pasan dormidos el 99.9 % del tiempo, y ocupan transmisiones de baja energía en el espectro extendido.

### Capa de Enlace de Datos

El estándar IEEE 802 divide a la capa de enlace de datos (Data Link Layer, DLL) en dos sub-capas, la sub-capa control de acceso al medio (Medium Access Control, MAC) y la sub-capa de control de enlaces lógicos (Logical Link Control, LLC). La MAC depende del hardware y varía respecto a la implementación física de esta capa y la sub-capa LLC es igual para todos los estándares 802 (Freescale, 2007). El estándar IEEE 802.15.4 se basa en la organización internacional para la estandarización (ISO) del modelo de referencia para la interconexión de sistemas abiertos (OSI) como se muestra en la Figura 2.



**Figura 2 Relación de la IEEE 802.15.4 con OSI**

Fuente: (Adams, 2006)

La sub-capa MAC envía tramas MAC mediante el canal físico, proporciona servicios de datos, ofrece una interfaz de control y regula el acceso al canal físico y al balizado de la red. También se encarga de la validación de las tramas y las asociaciones/disociaciones entre nodos, garantizando slots de tiempo. La MAC proporciona dos tipos de servicios hacia las capas superiores que se acceden a través de dos puntos de acceso a servicios (Service Access Point, SAPs). La LLC también puede acceder a la MAC a través de los SAPs.

- (MAC common part sublayer, MCPS-SAP) para datos de la MAC.
- (MAC layer management entity, MLME-SAP) para servicios de administración.

Los servicios de datos se acceden por medio del punto de acceso MCPS-SAP y el manejo de servicios por MLME-SAP. Los dos servicios proporcionan una interface entre las sub-capas de convergencia de servicios específicos (SSCS) u otro LLC y las capas físicas. El administrador de servicios MAC tiene 26 primitivas, comparadas con el 802.15.1 (bluetooth), que tiene alrededor de 131 primitivas en 32 eventos.

La MAC 802.15.4 es muy versátil para aplicaciones hacia las que fue orientado, aunque se paga el costo de tener un estándar con características menores a las del 802.15.1, por ejemplo el 802.15.4 no soporta enlaces sincronizados de voz. La capa MAC define tiempo, trama y administración de canales RF, destinada a coexistir en dominio de frecuencia con otros servicios. Los mecanismos de acceso al canal que se implementan en la MAC 802.15.4 son los siguientes:

- Para una red 'sin beacon' se utiliza un estándar CSMA-CA enviando reconocimientos positivos para paquetes recibidos correctamente.

- En una red 'con beacon' se utiliza una estructura de supertrama, en este modo, a los dispositivos se les asigna una de las 16 ranuras de tiempo entre beacons cada cierto intervalo de 15 mseg, hasta 245 seg.

### **2.1.3. Estándar ZigBee**

ZigBee es un estándar inalámbrico normalizado para la conexión de una red WPAN, que conlleva un conjunto de protocolos de alto nivel de comunicación inalámbrica, diseñado para soportar aplicaciones que requieren bajo costo y consumo de energía. ZigBee está orientado al segmento del mercado no atendido por estándares existentes, con baja transmisión de datos, bajo ciclo de servicio de conectividad (Autores, 2004).

La alianza ZigBee es un consorcio no lucrativo de más de 70 compañías incluyendo Invensys, Mitsubishi Electric, y Motorola (Autores, 2004); donde el objetivo principal fue diseñar redes ad hoc de auto organización para radios digitales y además desarrollar dispositivos para aplicaciones diferentes las cuales proveerán al usuario soluciones inalámbricas rentables, fáciles de utilizar, altamente confiables y seguras.

El estándar IEEE 802.15.4, en el que se basa ZigBee, fue determinado en Mayo de 2003. Philips, el promotor más importante, formó parte de importantes investigaciones y pasó a ser un miembro promotor de la junta directiva de la alianza ZigBee. Sus especificaciones fueron ratificadas el 14 de diciembre de 2004 por la IEEE, pero el 13 de Junio de 2005 se hicieron públicas y disponibles para su uso en universidades y centros de desarrollo. El estándar IEEE 802.15.4 define el hardware y el software, descrito en términos de conexión de redes, como la PHY y la MAC. La alianza ZigBee ha añadido las

especificaciones de las capas de Red (Network Layer, NWK) y de Aplicación (APL) para completar lo que se llama la pila o stack de ZigBee (Autores, 2004).

### **Características** (Vidri, 2012)

- ZigBee utiliza las bandas de 2.4 GHz, 868 MHz y 915 MHz, especificadas en el estándar IEEE802.15.4. Sin embargo, a la hora de diseñar dispositivos, las empresas optan por la banda de 2.4 GHz, por ser libre en todo el mundo.
- Trabaja a una velocidad de transmisión entre 25 y 250 kbps.
- A pesar de que ZigBee coexiste en la misma frecuencia que otras redes inalámbricas semejantes de la familia WPAN, como por ejemplo Bluetooth, su desempeño no se ve afectado, debido a la baja tasa a la que transmite información y además el desarrollo de la tecnología se centra en la sencillez y bajo costo.
- Utiliza un protocolo asíncrono, half duplex y estandarizado, permitiendo la interoperabilidad entre dispositivos fabricados por compañías diferentes.
- Tiene capacidad de operar en redes de gran densidad, esta característica ayuda a aumentar la confiabilidad de la comunicación, ya que entre más nodos existan dentro de una red, mayor será el número de rutas alternas para garantizar que un paquete llegue a su destino, además su topología de malla MESH permite a la red auto recuperarse de problemas en la comunicación.

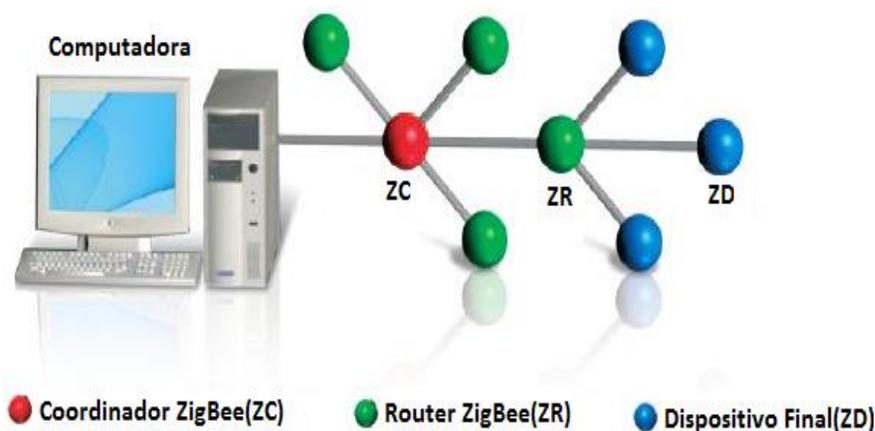
- Cada red ZigBee tiene un identificador de red único, lo que permita que coexistan varias redes en un mismo canal de comunicación sin ningún problema. Teóricamente pueden existir hasta 16 000 redes diferentes en un mismo canal y cada red puede estar constituida por hasta 65 000 nodos, obviamente estos límites se ven truncados por algunas restricciones físicas como la memoria disponible, ancho de banda, entre otros.
- ZigBee es un estándar de comunicación segura ya que se puede implementar encriptación, autenticación y multi-salto, es decir, que se puede establecer comunicación entre dos nodos aun cuando estos se encuentren fuera del rango de transmisión, siempre y cuando existan otros nodos intermedios que los interconecten.

### **Tipos de dispositivos**

Como se muestra en la Figura 3, se definen tres tipos de dispositivos ZigBee según su papel en la red (Autores, 2004):

- **Coordinador ZigBee (ZC):** Es el dispositivo más completo e importante, su función es almacenar información sobre la red y actuar como su centro de confianza en la distribución de claves de cifrado. Puede actuar como director de una red en árbol así como servir de enlace a otras redes. Debe existir uno por cada red.
- **Router ZigBee (ZR):** Como su nombre indica actúa como router interconectando dispositivos separados en la red y limitados debido a su rango. Además ofrece un nivel de aplicación para la ejecución de código de usuario.

- **Dispositivo final (ZED):** Este dispositivo puede comunicarse con el coordinador o un router, pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Un ZED tiene requerimientos mínimos de memoria y por lo tanto, es significativamente más barato.



**Figura 3 Dispositivos ZigBee según su papel en la Red**

Fuente: (Autores, 2004)

Aunque esta clasificación está muy extendida y aceptada, basándose en su funcionalidad puede plantearse una segunda clasificación:

- **Dispositivo de funcionalidad completa (FFD):** Es un dispositivo de función completa que se puede usar en cualquier topología. También conocido como nodo activo. Dispositivo capaz de recibir mensajes en formato 802.15.4. Gracias a la memoria adicional y a la capacidad de computar, puede funcionar como Coordinador o Router ZigBee, o puede ser usado en dispositivos de red que actúen de interfaz con los usuarios.
- **Dispositivo de funcionalidad reducida (RFD):** Es un dispositivo de función reducida, también conocido como nodo pasivo. Limitado a una

topología en estrella. Tiene capacidad y funcionalidad limitadas con el objetivo de conseguir un bajo costo y una gran simplicidad. Básicamente, son los sensores/actuadores de la red.

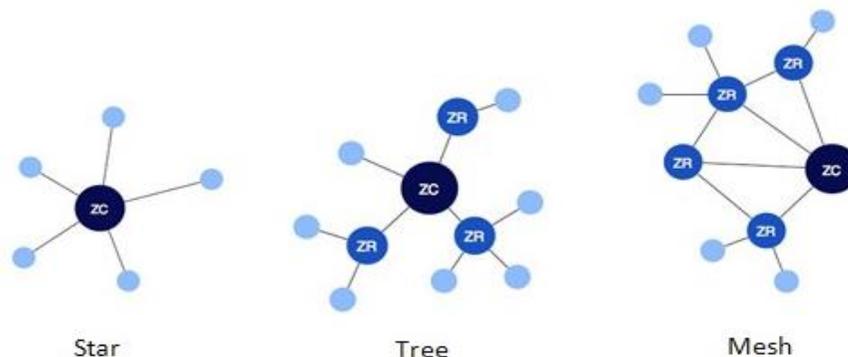
Un nodo ZigBee tanto activo como pasivo, puede reducir su consumo de energía gracias a que permanece dormido la mayor parte del tiempo, esto quiere decir en modo de ahorro de energía (sleep), hasta que ocurre un evento o recibe una interrupción externa. Cuando el nodo ZigBee se despierta aproximadamente en 15 ms, se demora un tiempo ínfimo para volverse a dormir si deja de ser requerido. Entre otras medidas de tiempo se tiene que la enumeración por parte del ZC hacia los nodos esclavos tarda aproximadamente 30 ms y el acceso al canal entre un nodo activo y uno pasivo tarda aproximadamente 15 ms.

### **Topologías de Red**

Los nodos WSN están organizados en uno de tres tipos de topologías de red (Daboín & Verde, 2012):

- **Topología de Estrella (Star):** Cada nodo se conecta directamente al gateway.
- **Topología de Árbol (Tree):** Cada nodo se conecta a un nodo de mayor jerarquía en el árbol y después al gateway. Los datos son ruteados desde el nodo de menor jerarquía en el árbol hasta el gateway.
- **Topología de Malla (Mesh):** Los nodos se pueden conectar a múltiples nodos en el sistema y pasar los datos por el camino disponible de mayor confiabilidad. El enlace malla es referido como un ruteador. Mesh es la

topología que se va a utilizar en el proyecto porque que ofrece mayor confiabilidad, que las otras topologías vistas anteriormente.



**Figura 4 Topologías de Redes ZigBee**

Fuente: (Adams, 2006)

### **Estrategias de conexión de los dispositivos en una red ZigBee**

Como ya se mencionó anteriormente una de las principales características que presentan los nodos esclavos de una red ZigBee, es que son capaces de conservar energía. Los nodos esclavos permanecen mucho tiempo en modo dormido y solo se despiertan por una fracción de segundo para confirmar que están vivos en la red de dispositivos de la que forman parte. En las redes ZigBee, se pueden usar dos tipos de entornos o sistemas (Adams, 2006):

- **Con balizas:** Mediante este entorno las balizas sincronizan a todos los dispositivos que conforman una red, y les permiten conocer cuando pueden transmitir información. Los intervalos de las balizas son asignados por el ZC y pueden variar desde los 15 ms hasta los 4 minutos. Los dispositivos, escuchan al ZC durante el balizamiento. Si un dispositivo desea intervenir en la conversación, se registra en el ZC y verifica si hay mensajes para él. Si existen mensajes, actúa; si no es así, el dispositivo vuelve a dormir, y se despierta de acuerdo a un horario que ha establecido previamente el ZC. En cuanto el ZC termina el

balizamiento, vuelve a dormir. De esta manera el consumo de energía se reduce drásticamente.

- **Sin balizas:** Este entorno se utiliza principalmente en sistemas de seguridad, donde sus dispositivos (sensores, detectores de movimiento o de rotura de cristales), duermen usualmente todo el tiempo. Estos dispositivos se despiertan de forma regular para anunciar que siguen en la red. Cuando se produce un evento, en este caso cuando se detecta algo, el sensor despierta instantáneamente y transmite un mensaje al ZC. Cuando el ZC, recibe el mensaje enviado por el sensor, activa la alarma correspondiente.

#### 2.1.4. Aplicaciones

En WSN se puede encontrar una diversidad de sensores incluyendo visuales, magnéticos, térmicos, acústicos, sísmicos entre otros, que son capaces de monitorizar cualquier variación de las condiciones ambientales, generalmente temperatura, humedad, presión, velocidad, presencia de objetos o seres vivos, lo que proporciona el potencial para revolucionar complejos sistemas de control y observación donde se puede obtener un amplio rango de aplicaciones en diferentes escenarios (Maroto, 2010).

- **Entornos de alta seguridad:** Las WSN pueden ser utilizadas en lugares que se requiera una alta seguridad como los aeropuertos para medición de ruido acústico, centrales químicas y nucleares para medición de radiaciones ionizantes, en edificios gubernamentales que requieran seguridad donde las WSN son provistos con cámaras que envían imágenes a través de la red sin necesidad de conexión directa con cable, para evitar posibles sabotajes y ataques.

- **Entornos militares:** Inicialmente las WSN fueron creadas para usos militares y desde entonces esta tecnología sigue siendo utilizada en este campo. Se pueden realizar funciones como: Monitorización de fuerzas y equipos, vigilancia de zonas estratégicas, reconocimiento del terreno, detección de ataques biológicos, químicos o nucleares entre otros.
- **Entornos ambientales:** En escenarios ambientales como bosques, ríos y océanos, las WSN son utilizadas para el control de parámetros como temperatura, humedad, actividad sísmica entre otros. De esta manera se obtiene un mejor conocimiento acerca de los contaminantes y cambios geológicos que pueden ocurrir en este tipo de entornos, previniendo desastres medioambientales que pueden ser irremediables.
- **Entornos agrícolas:** En este tipo de entornos las WSN pueden ser utilizadas para el control remoto de la cantidad de agua, fertilizante y pesticida que las plantas necesitan, optimizando la producción y la calidad en diversos aspectos de la agricultura. También pueden ser utilizadas para alertar problemas que pueden afectar las cosechas, tales como granizo, plagas o intrusiones de animales. De esta manera se optimiza las explotaciones agrarias tanto de invernadero como las de campo abierto.
- **Entornos industriales:** Las WSN se pueden aplicar para monitoreo y diagnóstico de maquinarias, medidas de calidad, telemetría, funcionamiento de motores eléctricos y de combustión interna entre otros. Las características que presentan estos sensores permiten determinar los posibles problemas en escenarios industriales. Para producción industrial los sensores son utilizados para controlar el tiempo entre procesos y para etiquetar los productos que son

procesados, para conocer el lugar en donde se encuentran, en cualquier periodo de tiempo.

- **Medicina:** La monitorización continua en tiempo real que las WSN ofrecen, permite mejorar el tiempo de atención a pacientes que necesitan tener controladas sus constantes vitales tales como pulsaciones, presión arterial, nivel de azúcar, entre otros. Además se puede realizar un cuidado exhaustivo de los pacientes crónicos recogiendo continuamente datos y enviándose a su médico, reemplazando sistemas de telemetría vía cable que pueden resultar incómodos e incluso peligrosos.
- **Domótica:** El precio y la facilidad de instalación en edificios y hogares hacen factible el uso de las WSN para domotización, permitiendo la gestión total del espacio, controlando el gasto energético, la seguridad y el confort. La iluminación, el riego de jardines, la climatización y la detección de presencia son unas de las funciones que pueden realizar las WSN.
- **Tráfico:** Mediante las WSN se puede determinar la presencia o ausencia de vehículos para tener un control sobre las vías de circulación y de la situación del tráfico en ángulos muertos que no son cubiertos por cámaras. En caso de accidente se puede informar a los conductores la situación, permitiendo a los usuarios tomar rutas alternativas. También se utilizan en gestión de aparcamientos y avisos en cruces entre semáforos y vehículos.



**Figura 5 Aplicaciones de las WSN**

Las WSN pueden ser utilizadas en diversos entornos satisfaciendo las necesidades requeridas gracias a las características que este tipo de redes presentan. La aplicación que se va a desarrollar en este proyecto es *radiolocalización*, la estimación de la posición de cualquier objetivo es fundamental en la mayoría de las aplicaciones con WSN.

En las aplicaciones como monitoreo ambiental, tales como el monitoreo del hábitat de los animales, vigilancia de incendios de matorrales, monitoreo de la calidad del agua y la agricultura de precisión; los datos de medición no tendrían sentido sin un conocimiento exacto de la ubicación, de donde estos son obtenidos. Por otra parte, la disponibilidad de información sobre la ubicación puede permitir una gran variedad de aplicaciones, tales como la gestión de inventarios, detección de intrusos, monitoreo de tráfico, atención de la salud, vigilancia y reconocimiento.

## **2.2. Radiolocalización con WSN**

### **2.2.1. Introducción**

Hoy en día, existen sistemas de *radiolocalización* que basan su funcionamiento en láser o ultrasonido y brindan una buena precisión de la ubicación, sin embargo, el uso de este tipo de dispositivos aumenta el costo, tamaño y el consumo de potencia (Díaz & Tabernilla, 2010). Gracias a la utilización de las WSN, se está alcanzando un mayor nivel de control y monitorización, lo cual conlleva a una mejora del manejo del entorno de trabajo y de respuesta frente a problemas que puedan presentarse.

La utilización de las WSN para determinar la ubicación de un objetivo, permite alcanzar un mejor rendimiento y funcionamiento debido a las ventajas que estos presentan frente a otros sistemas de *radiolocalización*. La *radiolocalización* mediante las WSN, han atraído el interés de la investigación, se espera que siga creciendo con los avances en las técnicas de comunicación inalámbricas y el aumento de aplicaciones que involucren las WSN (Santana, 2010).

### **2.2.2. Algoritmos de Radiolocalización con WSN**

Las WSN son redes ad-hoc, caracterizadas por constar de un gran número de nodos con capacidades de medición incorporadas en el propio nodo o en placas de sensores acoplables y se basan en estas mediciones para el desarrollo de muchas aplicaciones como por ejemplo *radiolocalización*. Hoy en día, las técnicas de *radiolocalización* en WSN se pueden dividir en dos grandes grupos, esquemas basados en rango y esquemas libres de rango (Moreno,

2011). Estos esquemas difieren en la manera en que se obtienen las posiciones de los nodos.

### **Esquema basado en Rango**

El esquema basado en rango se basa en mediciones físicas de las distancias entre los nodos que corresponden a una red de sensores o bien los ángulos entre ellos. En este esquema se utilizan técnicas para realizar las respectivas mediciones dependiendo del intercambio de señales entre los nodos. La recepción de la señal puede ser función de la distancia entre los nodos, de la posición relativa y la línea de vista entre el transmisor y el receptor. En estas técnicas se requiere un modelo de propagación de la señal electromagnética; hardware especializado basado en señales acústicas o, incluso, arreglos de antenas para estimar el ángulo de arribo de la señal de radio frecuencia.

### **Esquema Libre de Rango**

El esquema libre de rango no necesita estimar las distancias entre nodos a diferencia del otro esquema, si no que se basa únicamente en la información de conectividad entre nodos adyacentes. Este esquema utiliza técnicas de posicionamiento, para no tener la necesidad de realizar mediciones de distancias.

Los algoritmos que corresponden a este esquema se basan en la medición de la conectividad, donde se puede identificar los vecinos de una mota. Si dos motas son capaces de transmitir información entre ellos, entonces hay una gran probabilidad de que la distancia entre estas, sea menor o igual al radio de comunicación. El hecho de que el esquema libre de rango no se base en mediciones de distancias, no es una limitante para que pueda

estimar su propio rango, es decir, en este esquema la distancia entre motas se mide en saltos.

Por ejemplo, una técnica factible es emplear el algoritmo de vector distancia (DV), para estimar la mejor ruta entre una mota que desconoce su posición y una mota fija, donde cada mota propaga las distancias que conoce a diferentes puntos de la red. La distancia entre cada par de motas se estima mediante el promedio del número de motas acumuladas a lo largo de la trayectoria, por lo que conociendo el radio de alcance de la señal de radio frecuencia, es posible estimar la distancia física entre dos motas.

## **CAPÍTULO 3**

### **MÉTODOS Y MATERIALES**

#### **3.1. Análisis del Kit de Sensores Inalámbricos MEMSIC 2.4ghz, ZigBee (MEMSIC, 2006)**

La plataforma MEMSIC está demostrada en numerosos despliegues y es ideal para aplicaciones donde se requiera la utilización de múltiples nodos-sensores. Conecta el medio físico con sistemas de gestión de información para proporcionar soluciones avanzadas de monitorización, automatización y control. MEMSIC ofrece una amplia gama de productos para WSN para satisfacer las necesidades de distintas aplicaciones. El modelo disponible para el proyecto corresponde a IRIS (MEMSIC Inc., 2010) y esta conformada por dispositivos de última generación que son compatibles con la plataforma de software MoteWorks™ (Crossbow Technology, Inc., 2006) que permite a los usuarios crear redes ad-hoc.

##### **3.1.1. Hardware MEMSIC**

###### **Nodo Inalámbrico**

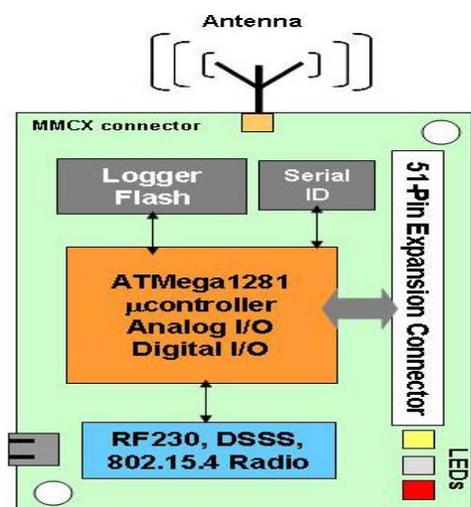
Los nodos inalámbricos son comúnmente conocidos como motas, son dispositivos capaces de realizar mediciones de un determinado fenómeno con ayuda de un sensor y propagarlas vía radio. Sus principales características son su reducido tamaño y consumo de potencia. La mota IRIS (ver Figura 6) está compuesta de un microcontrolador de bajo consumo de potencia en ATmega1281 que corre bajo MoteWorks desde su flash interna de memoria (AT45DB041) y un radio transceiver (AT86RF230) de Atmel que emplea O-QPSK (Modulación por desplazamiento de fase en cuadratura escalonada).



**Figura 6 Mota Iris**

Fuente: (MEMSIC, 2006)

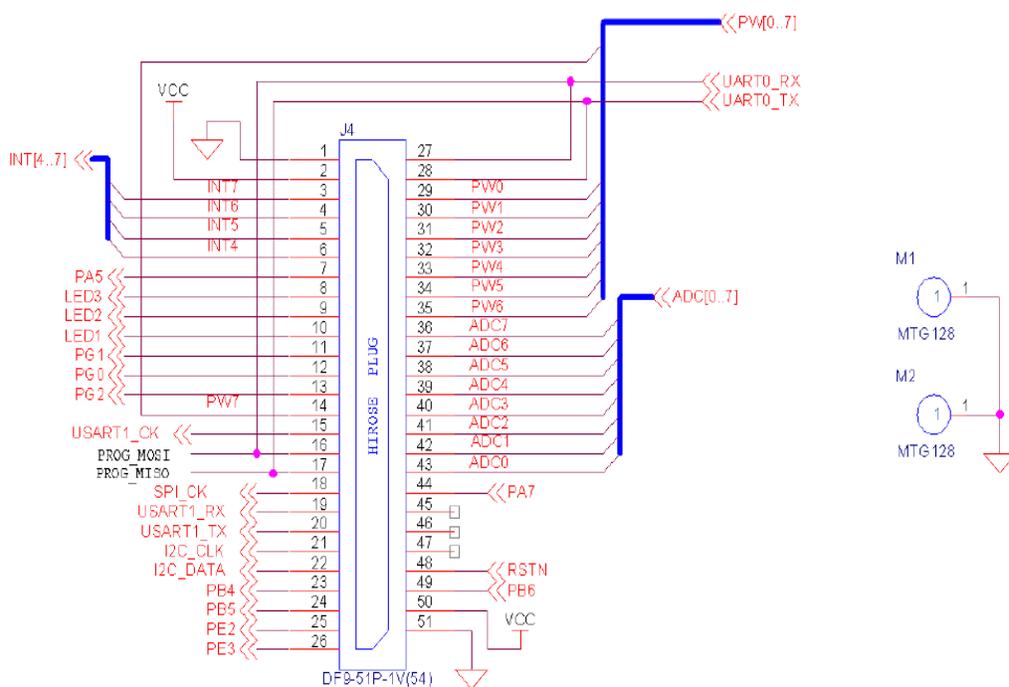
La radio incluye un módem de banda base DSSS (espectro ensanchado por secuencia directa), que proporciona una ganancia de difusión de 9 dB y una tasa de datos de 250 kbps. La radio es una solución altamente integrada para la comunicación inalámbrica en la banda ISM sin licencia de 2,4 GHz, que puede ser sintonizada en los canales de 802.15.4 que son numerados desde 11 (2.405 GHz) a 26 (2.480 GHz) cada uno separado por 5 MHz. La potencia de transmisión puede ser programable de -17,2 dBm a 3 dBm (ver Figura 7).



**Figura 7 Diagrama de Bloque Mota Iris**

Fuente: (MEMSIC, 2006)

La tarjeta del procesador (XM2110) se la puede configurar para ejecutar a la vez su sensor de aplicación/procesamiento y su radio de comunicación. La mota posee un conector de expansión de 51 pines (ver Figura 7) que soporta entradas analógicas, entradas y salidas digitales, I2C, SPI e interfaces UART. Estas interfaces hacen que sea fácil la conexión a una amplia variedad de periféricos externos. Además la Mota incorpora una interfaz formada por tres diodos LEDs de color rojo, amarillo y verde, que permiten conocer el funcionamiento de la mota.



**Figura 8 Conector de expansión de 51 pines**

Fuente: (MEMSIC, 2006)

El consumo de corriente en modo activo es de 8mA y 8µA para modo hibernación. Las dimensiones de la Mota IRIS son 58x32x7mm y tiene un peso de 18gr excluyendo las baterías que son dos pilas AA. Las baterías deben proporcionar un voltaje entre 2.7 y -3.3 V para que la mota funcione de manera adecuada.

## Gateway

El gateway que se va utilizar en la aplicación es MIB520 (ver Figura 9), que proporciona conectividad mediante USB entre las motas IRIS con el ordenador. Además de la transferencia de datos, MIB520 también proporciona una interfaz de programación y la alimentación del bus USB elimina la necesidad de una fuente de alimentación externa. Cuando se instalan los drivers para el gateway se genera 2 COM de comunicaciones, uno para la programación de las motas y otro para escuchar los paquetes provenientes de la red.



**Figura 9 Gateway MIB520**

Fuente: (MEMSIC, 2006)

### 3.2. Sistema Operativo TinyOS

*TinyOS* es un sistema operativo libre y de código abierto (open source), basado en un modelo de programación característico de los sistemas embebidos (modelo de componentes), diseñado para redes WSN. Está escrito en lenguaje *NesC* y su diseño de comunicación centrada y el modelo de software modular se adaptan a las necesidades específicas de las WSN, tales como reducido tamaño de memoria, bajo consumo de energía, operaciones de concurrencia intensiva, entre otros.

*TinyOS* comenzó como una colaboración entre la Universidad de California, Berkeley, en cooperación con Intel Research and Technology Ballesta, y desde entonces ha crecido para ser un consorcio internacional llamado Alianza *TinyOS* (García J. , 2008). El diseño de Kernel de *TinyOS* está basado en una estructura con los siguientes niveles de planificación (Suárez, 2009):

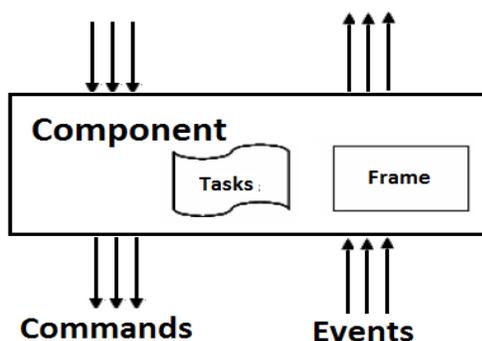
- **Eventos:** Los eventos son diseñados para realizar procesos pequeños, por ejemplo cuando el contador del timer se interrumpe, o atender las interrupciones de un conversor análogo-digital. Además pueden interrumpir las tareas que se están ejecutando. Este enfoque es la solución ideal para alcanzar un alto rendimiento en aplicaciones de concurrencia intensiva. Adicionalmente utiliza las capacidades de la CPU de manera eficiente gastando el mínimo de energía.
- **Tareas:** Las tareas son diseñadas para un mayor de procesamiento que los eventos y no son críticas en tiempo, por ejemplo al calcular el promedio en un arreglo. Las tareas se ejecutan en su totalidad, pero la solicitud de iniciar una tarea, y el término de ella son funciones separadas, esta característica es propia de la programación orientada a componentes. Con este diseño se permite que los eventos que son rápidamente ejecutables puedan ser realizados inmediatamente, pudiendo interrumpir las tareas que presentan mayor complejidad.

*TinyOS* se encuentra construido sobre un conjunto de componentes de sistema, los cuales proveen la base para la creación de aplicaciones. En la Figura 10, se muestra el modelo de un componente. Una aplicación se define conectando los componentes y usando un conjunto de especificaciones detalladas. Este modelo es esencial en sistemas embebidos para incrementar la confiabilidad, sin dejar a lado el desempeño. Un componente provee y usa

interfaces que son el único punto de acceso al componente. Además está compuesto de un espacio de memoria y un conjunto de tareas. Los cuatro elementos que conforman un componente son los siguientes:

- Manejador de comandos.
- Manejador de eventos.
- Un frame de tamaño fijo y estáticamente asignado, en el cual se representa el estado interno del componente.
- Un bloque con tareas simples.

Como se muestra en la Figura 10, las dos fuentes de concurrencia en *TinyOS* son las tareas y los eventos. Los componentes entregan tareas al planificador, siendo el retorno de éste de forma inmediata, aplazando el cómputo hasta que el planificador ejecute la tarea. Para asegurar que el tiempo de espera no sea largo, se recomienda programar tareas cortas, y en caso de necesitar procesamientos mayores, se recomienda dividirlo en múltiples tareas. Las tareas se ejecutan en su totalidad, y no tiene prioridad sobre otras tareas o eventos. Así también los eventos hasta completarse, pero estos sí pueden interrumpir otros eventos o tareas, con el objetivo de cumplir de la mejor forma los requerimientos de tiempo real.



**Figura 10 Modelo de componentes de TinyOS**

Fuente: (Suárez, 2009)

Los comandos son peticiones hechas a componentes de capas inferiores. Estos generalmente son solicitados para ejecutar alguna operación. Existen dos posibles casos de uso de los comandos. El primero es para operaciones bifase, donde los comandos retornan inmediatamente, no generando bloqueos por la espera de la ejecución, es decir, una vez que realizan la petición, el planificador será el encargado de ejecutar lo solicitado, generando un evento que indique el fin de la operación. El otro es el caso de realizar una operación no bifase, donde ésta se realizará completamente y por tanto, no habrá evento retornado.

Los manejadores de eventos son invocados por eventos de componentes de capas inferiores, o por interrupciones cuando se está directamente conectado al hardware. Similar a los comandos, el frame será modificado y las tareas agregadas. Los eventos son capaces de interrumpir tareas, no así viceversa. Las tareas son entregadas a un planificador (task scheduler) que en este caso está implementado con método FIFO. Debido a esta implementación, las tareas se ejecutan secuencialmente y no deben ser excesivamente largas. Alternativamente al planificador de tareas FIFO, puede ser reemplazado por planificadores basados en prioridades (priority-based) o

por planificadores basados en plazos (deadline-based), los cuales pueden ser implementados sobre *TinyOS*.

### 3.3. Lenguaje de Programación NesC

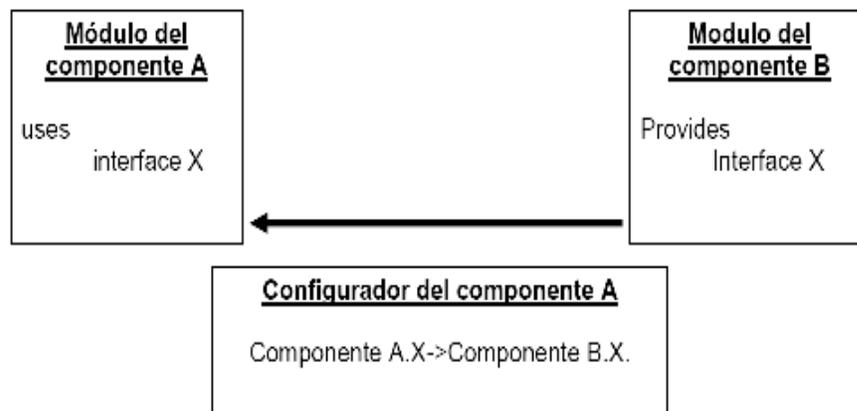
*NesC* es un lenguaje de programación basado en C, orientado a sistemas embebidos que incorporan el manejo de red. Además soporta un modelo de programación que integra el manejo de comunicaciones, las concurrencias que provocan las tareas, eventos y la capacidad de reaccionar frente a sucesos que puedan ocurrir en los ambientes donde se desempeña (García J. , 2008).

*NesC* utiliza Interfaces bidireccionales, como ya se comentó anteriormente las interfaces son los accesos a los componentes que contienen comandos y eventos, los cuales son los que implementan las funciones. El proveedor de una interfaz implementa los comandos, mientras que el que las utiliza implementa eventos.

La unión estática de componentes, mediante interfaces., aumenta la eficiencia en tiempos de ejecución, incrementando la robustez del diseño y permitiendo un mejor análisis del programa. Adicionalmente *NesC* presenta herramientas que optimizan la generación de códigos. *NesC* está compuesto por dos componentes (Gay, 2005):

- **Módulos:** Proporcionan el código de la aplicación, implementando una o más interfaces.
- **Configuraciones:** Son utilizadas para que los componentes se puedan unir entre sí, conectando las interfaces que algunos componentes proveen con las interfaces que otras usan. Es el encargado de ensamblar todos los componentes de la aplicación.

En la Figura 11 se muestra un diagrama de bloques que representa un ejemplo de código NesC, en la que se puede observar como el componente B proporciona la interfaz X al componente A para que esta la utilice. Toda aplicación en *TinyOS* está compuesta por al menos un componente módulo, un componente de configuración y un *Makefile*. El *Makefile*, al igual que en el lenguaje C, es un archivo utilizado para compilar.



**Figura 11 Ejemplo de código NesC**

Fuente: (Gay, 2005)

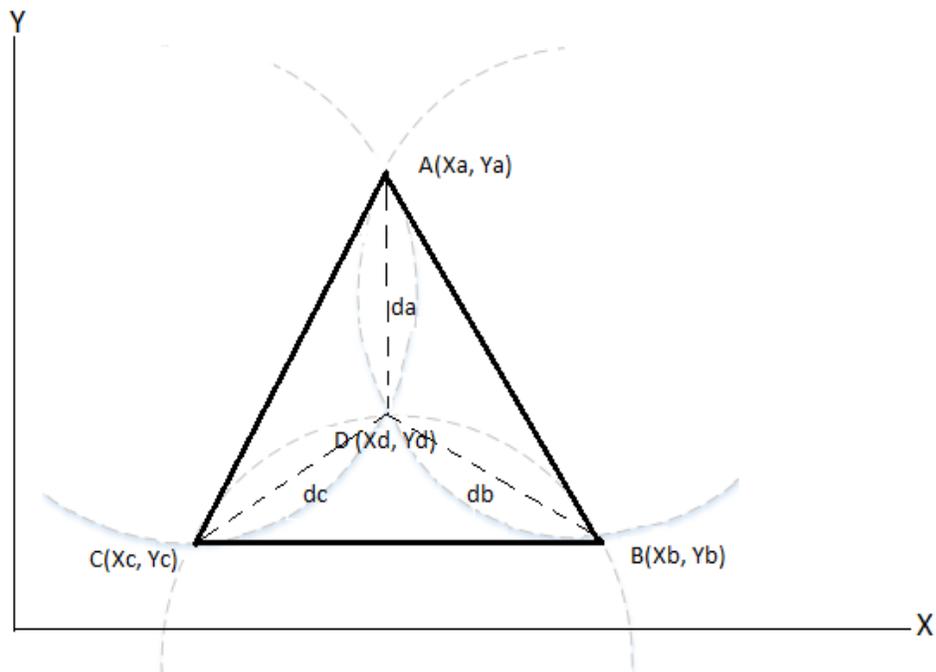
### 3.4. Técnicas de Estimación de la Posición (Polo, 2013)

Las principales técnicas de estimación son: Trilateración, Triangulación, Multilateración, Análisis de la escena y Proximidad. Los sistemas de *radiolocalización* actuales las utilizan individualmente o combinándolas. A continuación se detallará cada una de esas técnicas.

#### 3.4.1. Trilateración

La Trilateración es un método para determinar la posición de un objeto en función del alcance simultáneo (distancia), mediante las mediciones de tres o

más puntos de referencia en ubicaciones conocidas. Considerando en un caso ideal, los 3 círculos obtenidos a partir de los puntos de referencia  $A(X_a, Y_a)$ ,  $B(X_b, Y_b)$  y  $C(X_c, Y_c)$  y distancias  $d_a$ ,  $d_b$  y  $d_c$  se cortan en un punto exactamente, pero en mediciones con ruido, los círculos pueden intersectar en un área o no se cortan en absoluto, en este caso la solución que da el mínimo error deberá ser considerado.



**Figura 12 Escenario de Trilateración**

Las expresiones de la Figura 12 son las siguientes:

$$(x_d - x_a)^2 + (y_d - y_a)^2 = d_a^2 \quad (1)$$

$$(x_d - x_b)^2 + (y_d - y_b)^2 = d_b^2 \quad (2)$$

$$(x_d - x_c)^2 + (y_d - y_c)^2 = d_c^2 \quad (3)$$

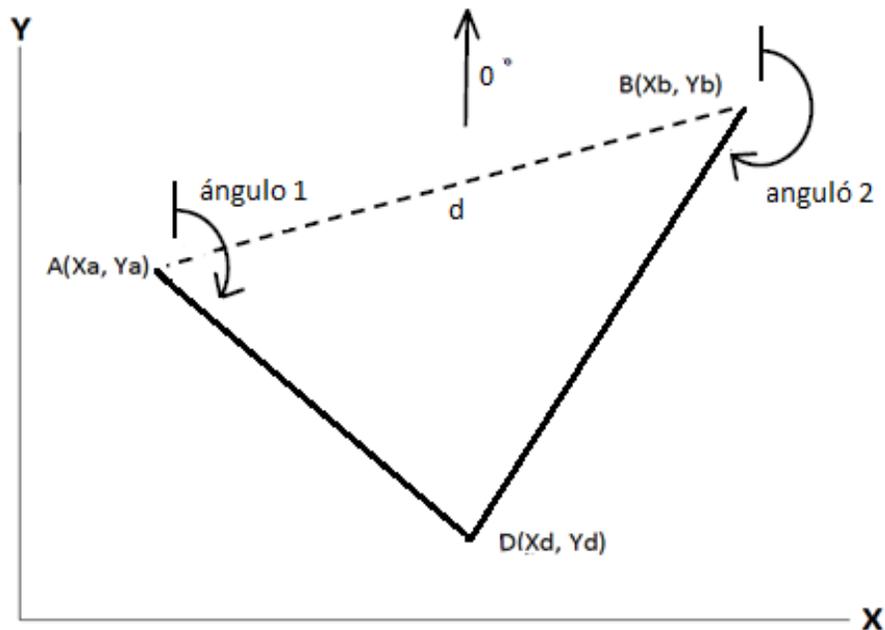
$$(x_d - x_i)^2 + (y_d - y_i)^2 = d_i^2, \{i = 1, \dots, n\} \quad (4)$$

La expresión (4) se puede pasar a un sistema lineal de ecuaciones, para que pueda ser resuelta.

### **3.4.2. Triangulación**

La Triangulación es similar a la Trilateración, solo que en lugar de distancias, se utilizan los vértices y ángulos que se encuentran dentro del área del triángulo. Se requieren dos ángulos y la distancia entre dos puntos de referencia en un entorno bidimensional (ver Figura 13). Para tres dimensiones son necesarios dos ángulos, la distancia entre dos nodos de referencia y un azimut para especificar una posición precisa. Normalmente se utiliza un vector de referencia constante con valor  $0^\circ$  (por ejemplo, el norte magnético).

El uso de una red de antenas sincronizadas (phased antenna array) es una excelente opción para la técnica de triangulación. Varias antenas con una distancia conocida entre ellas, miden el tiempo de llegada de una señal. Dadas las diferencias entre tiempos de llegada y la geometría de la red de antenas, es posible calcular el ángulo desde el cual se originó la emisión. Si hay suficientes elementos en la red de antenas y suficientes distancias, se puede realizar el cálculo de la triangulación.



**Figura 13 Escenario de Triangulación dos dimensiones**

Fuente: (Polo, 2013)

### 3.4.3. Multilateración

Al igual que en el método de Triangulación, para la Multilateración se necesitan las distancias de un número determinado de nodos fijas, para calcular la posición de un nodo, solo que en este caso, las distancias se determinan utilizando TDoA.

### 3.4.4. Proximidad

La técnica de proximidad determina cuando un objeto está cerca de una posición conocida. Existen tres métodos generales para determinar proximidad entre dos puntos.

- **Detectando el contacto físico:** Es el método más básico de proximidad y se suelen utilizar sensores de presión, sensores de tacto y detectores de capacitancia.
- **Monitorizando puntos de acceso celulares inalámbricos:** La monitorización de un dispositivo móvil, para determinar si está en el rango de uno o más puntos de acceso en una red celular inalámbrica.
- **Observando sistemas automáticos de identificación:** Algunos ejemplos de este sistema son los terminales de puntos de venta con tarjeta de crédito, los historiales de *login* en computadores y los registros de llamadas de teléfono.

### 3.5. Algoritmos de Radiolocalización con WSN Basado en Rango (Poellabauer, 2010)

#### 3.5.1. Variable RSSI

El indicador de fuerza de la señal recibida (Received Signal Strength Indicator, RSSI), está basada en la potencia medida de la señal electromagnética que es recibida por un receptor y cuantificada, para poder estimar la distancia que separa los nodos. Sin embargo la relación distancia/potencia de la señal recibida no es un valor estable y lineal, porque esta técnica se ve afectada por las reflexiones, difracciones y multitrayectos que se presentan en las transmisiones.

En la práctica, el efecto del multitrayecto puede ser mitigado usando métodos de espectro ensanchado como DSSS (Direct Sequence Spread Spectrum) y FHSS (Frequency Hopping Spread Spectrum), que promedian la

potencia recibida sobre un amplio rango de frecuencias, evitando así el desvanecimiento selectivo en frecuencia causado por el multitrayecto. Además el empleo de técnicas de espectro ensanchado también reduce las interferencias ocasionadas en la banda libre, que es la usada comúnmente por las WSN (Pozuelo, 2009).

La potencia de la señal recibida en un receptor, es inversamente proporcional a la distancia hacia el transmisor. El modelo de propagación de la señal más común en las WSN es el modelo de espacio libre, de acuerdo con la ecuación de Friis, la potencia decrece de manera cuadrática cuando la distancia aumenta (Oguejiofor O.S, 2013).

$$p_{rx} = p_{tx} * g_{tx} * g_{rx} \left( \frac{\lambda}{4\pi d} \right)^2 \quad (5)$$

Dónde:

$p_{rx}$ : Potencia de Recepción.

$p_{tx}$ : Potencia de Transmisión.

$g_{tx}$ : Ganancia de la antena de Transmisor

$g_{rx}$ : Ganancia de la antena de Recepción.

$d$ : Distancia entre el transmisor y el Receptor.

$\lambda$ : Longitud de Onda.

En dispositivos embebidos, la potencia de recepción es convertida a RSSI definida como la proporción de la potencia recibida con respecto a la potencia de referencia cuyo valor es de  $P_{ref} = 1\text{mW}$ . Como se muestra en la ecuación (6), un aumento en la potencia de recepción incrementa el valor de RSSI (AKAR, 2010).

$$\text{RSSI} = 10 * \log \frac{P_{\text{Rx}}}{P_{\text{ref}}} [\text{dBm}]$$

6)

En la mayoría de hardware de comunicaciones el RSSI ya está integrado, por lo que es factible la experimentación con este algoritmo y obteniendo el modelo y los parámetros de desempeño se puede estimar las distancias entre motas.

### 3.5.2. Algoritmo ToA

El algoritmo ToA (Time of Arrival) estima la distancia en base al tiempo que se demora la señal en llegar del transmisor al receptor. Esta técnica requiere que el transmisor y receptor estén sincronizados, la distancia puede calcularse multiplicando este tiempo a la velocidad de propagación de la señal, que típicamente será la velocidad de la luz.

El receptor debe ser capaz de estimar con precisión el tiempo de llegada de la señal entre los nodos, obtener esa precisión es complicado, debido al ruido aditivo y las señales multitrayecto que llegan al receptor. Dependiendo del tipo de señal empleada para estimar la distancia, como una señal RF, el tiempo de llegada requiere de relojes con una resolución alta para producir una precisión aceptable. También la sincronización va a depender de la velocidad de transmisión, cuanto más lento viaje la señal en el aire, el error se volverá menos significativo (Moreno, 2011).

### 3.5.3. Algoritmo AoA.

El algoritmo AoA (Angle of Arrival, AoA) se basa en las mediciones del ángulo de llegada, que se define como el ángulo que forma la dirección de propagación

de una onda incidente y una determinada dirección de referencia. Mediante la técnica de triangulación vista anteriormente, se puede estimar la posición de una mota desconocida, para esto se emplea el uso de dos técnicas: La que utiliza la respuesta en amplitud de la antena del receptor [d (m)] y la que utiliza la respuesta en fase de la antena del receptor [ $\theta^\circ$ ] (Rugeles, 2010).

- **Respuesta en amplitud de la antena del receptor:** Para esta técnica es necesario el uso del patrón de recepción de la antena para realizar mediciones del ángulo. La precisión de la medición depende de la sensibilidad de la antena. Si la antena es capaz de girar sobre su propio eje, la dirección de la potencia máxima de la señal se representa como la dirección del transmisor.
- **Respuesta en fase de la antena del receptor:** Esta técnica se basa en las mediciones de las diferencias de fase mientras ocurre la llegada del frente de onda. Es necesario el uso de un arreglo de antenas o de un receptor que tenga una antena de mayor tamaño, en comparación con la longitud de onda de la señal del transmisor.

Estas técnicas pueden ser precisas en un medio donde no se produzcan multitrayectos porque cuando un receptor recibe una señal puede pensar que la señal proviene de otro lugar al medir el ángulo de la señal reflejada, por lo tanto, estas técnicas están limitadas por la direccionalidad de la antena y las multitrayectorias que se presentan.

### 3.6. Algoritmos de Radiolocalización con WSN Libre de Rango

#### 3.6.1. Algoritmo DV-HOP

Es una técnica en la cual la mota que desconoce su posición estima el número de saltos hacia las motas fijas. Luego las motas fijas estiman la longitud promedio de un salto y se transmite una corrección mediante un flujo de datos hacia toda la red (Grilo, 2009). La longitud del salto puede estimarse como:

$$c_i = \frac{\Sigma \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\Sigma h_i}, i \neq j \quad (7)$$

Dónde:

$(x_i - x_j)$  y  $(y_i - y_j)$ : Son las coordenadas de la mota fija  $i$  y la mota fija  $j$ .

$h_i$ : Número de saltos para ir de la mota  $i$  a la mota  $j$ .

Cuando las motas que desconocen su posición reciben la corrección, pueden determinar la distancia hacia cada una de las motas fijas y mediante la técnica de triangulación, la mota desconocida puede estimar su posición. DV-HOP es una técnica que se caracteriza por ser simple y por no depender del rango de error de medición, sin embargo solo se puede trabajar con topologías uniformes y sin obstáculos.

#### 3.6.2. Algoritmo MDS – MAP

Es una técnica que estima las posiciones de las motas desconocidas utilizando únicamente información de conectividad, identificando las motas que se encuentran en el radio de comunicaciones. Inicialmente se organiza en una matriz las trayectorias más cortas previamente calculadas, entre cada par de

motas empleando algoritmos como Bellman-Ford o Dijkstra, posteriormente se utiliza el escalamiento multi-dimensional (MultiDimensional Scaling, MDS) para estimar las posiciones de las motas que se ajusten a las distancias estimadas, finalmente se obtiene un mapa, basado en motas fijas (Dargie & Poellabauer, 2010).

La ventaja de este sistema es que para obtener la posición de una mota en dos dimensiones es suficiente contar con 3 motas fijas, mientras que otros necesitan que se distribuyan uniformemente. Sin embargo, el inconveniente que presenta esta técnica, es que una sola mota realiza todos los cálculos necesarios para generar la salida MDS, por ejemplo, si una red está compuesta por un gran número de motas, solo una debe realizar la descomposición de una matriz relativamente grande. Dado que MDS-MAP emplea la trayectoria más corta entre dos motas para aproximar la distancia Euclidiana, este estimado es apropiado sólo si la red es densa y uniforme, pero no lo es para redes irregulares.

Existe una versión perfeccionada de MDS-MAP, denominada MDS-MAP (P). Esta idea mejorada de la técnica anterior nace por la importante desventaja que MDS-MAP guarda frente a redes irregulares. La idea principal es crear un mapa local, utilizando MDS para motas cercanas y después combinar estos mapas locales para construir un mapa global. Además de operar bien tanto en redes regulares como irregulares, esta técnica posee la ventaja de poder trabajar de manera distribuida, lo que la hace apropiada para ser utilizada en redes muy grandes.

### **3.6.3. Algoritmo ELA.**

El algoritmo de Localización Elástica (Elastic Localization Algorithm, ELA) utiliza un modelo físico de oscilaciones y relajaciones de un resorte. Primero busca la

gráfica empotrada que se parezca lo más posible a la configuración real. Luego se modifica el empotrado empleando el método de masa-resorte que es una técnica de optimización distribuida para corregir errores.

Se asume que la longitud del resorte, en estado de equilibrio, es igual a la distancia que existe entre dos motas. Si la distancia estimada es menor que la distancia medida, el resorte empuja ambos nodos en dirección contraria. Por otro lado, si la distancia estimada es mayor que la distancia medida, el resorte tira de las masas para que estas se acerquen. En cada paso, los nodos son colocados de acuerdo a las fuerzas que actúan sobre ellos. El proceso finaliza cuando cada nodo alcanza el equilibrio, es decir cuando la fuerza resultante sobre toda la red es cero.

### **3.7. Selección del Algoritmo para Implementación del Prototipo de Sistema de Radiolocalización WSN**

Para seleccionar el algoritmo de *radiolocalización* más adecuado, se debe tomar en cuenta, factores importantes como el hardware-software disponible y las ventajas-desventajas que estos algoritmos presentan frente al entorno de trabajo. Los algoritmos ToA y TDoA realizan mediciones de distancias basándose en el tiempo de propagación de la señal que se demora en llegar del transmisor al receptor. El receptor debe ser capaz de estimar con precisión el tiempo de llegada de la señal correspondiente a la línea de visión directa entre las motas. Obtener precisión en estas medidas es complicado, debido al ruido aditivo y las señales multitrayecto que llegan al receptor.

Los dos mayores problemas que presentan estos algoritmos, son los retardos producidos, debido al hardware y al software, que se añaden a los tiempos medidos y la gran atenuación que suele sufrir la señal correspondiente al trayecto principal, pudiendo ser confundida con niveles de ruido o ser

suplantada en el receptor por alguna de las señales provenientes del multitrayecto.

La atenuación de la señal correspondiente a la línea de visión directa, puede ser rectificadas desplegando la red de forma que la separación entre las motas no sea muy grande, el escaso retardo que sufren las señales multitrayecto es muy difícil de aminorar, por lo que es necesario el uso de sistemas de mayor resolución temporal y lo por tanto, de mayor ancho de banda, que requieren necesariamente la utilización de hardware costoso, elevando de esta manera el consumo de energía.

El algoritmo AoA, al igual que los anteriores, sufre una gran degradación en las medidas debido al multitrayecto, lo que conlleva la posibilidad de medir ángulos de señales reflejadas y ruido aditivo, provocando oscilaciones en las medidas. Por ello, esta técnica se emplea a menudo en combinación con las anteriores y no de forma individual. Los algoritmos basados en el esquema libre de rango a pesar de las ventajas que estos presentan, solo pueden trabajar en topologías uniformes y sin obstáculos. Este esquema no es viable en topologías conformadas por gran número de motas ya que estos se basan en un proceso iterativo de refinamiento e intercambio de estimaciones, por lo que no son directamente aplicables, debido a que introducen una alta sobrecarga de tráfico. Adicionalmente la mayoría de estos algoritmos como por ejemplo, el algoritmo ELA necesitan de hardware adicional para poder llevar a cabo la implementación.

El chip de radio incorporado en las motas de la plataforma MEMSIC, tiene hardware necesario para medir y reportar la intensidad de la señal de paquetes recibidos, por lo que la técnica de trilateración mediante variable RSSI, se convierte en una opción viable para este proyecto. Esta técnica proporciona ventajas referentes al costo de implementación, porque no va a ser

necesario utilizar hardware adicional, a diferencia de otros algoritmos, en los que necesariamente se debería añadir algún tipo de hardware en las motas, que les permita conseguir, entre otras características, una sincronización aceptable.

La principal desventaja de presenta esta técnica, es que las medidas realizadas son impredecibles por la dependencia directa con el entorno. Además, existen desafíos adicionales como, la construcción de un mapa de potencia recibida fiable o la variabilidad de la señal RSSI en función del tipo de antena empleado en los receptores, fabricantes e incluso del estado de carga de la batería en cada momento. La precisión de la medición va a depender de la técnica para construir el modelo.

A pesar de estos inconvenientes, esta técnica es válida en entornos con visión directa (LOS) o sin ella (NLOS), por lo que se puede inferir que la medición de RSSI para estimar distancias, exige requisitos mínimos entre todos los algoritmos del esquema basado en rango, además ofrece la posibilidad de determinar distancias con un esfuerzo mínimo, porque la velocidad de transmisión, la modulación y la precisión en tiempo del sistema, no son relevantes para la toma de medidas, por lo que tampoco se requiere sincronización ni coordinación entre transmisor y receptor. Por las ventajas mencionadas anteriormente con respecto a otros algoritmos de *radiolocalización*, la técnica de trilateración mediante variable RSSI, es la más adecuada para este proyecto.

### **3.8. Modelamiento y Metodología de Localización**

La precisión en la ubicación es importante para determinar la eficiencia del sistema de *radiolocalización* y dadas las condiciones variables en un entorno inalámbrico, es importante considerar modelos propagación que permitan tener

en cuenta todos los elementos adversos que pudieran presentarse como obstáculos que reflejan, absorben u obstruyen la señal electromagnética. A diferencia de un sistema alámbrico que es estacionario y predecible, los sistemas inalámbricos son aleatorios, variantes en el tiempo y difíciles de modelar porque en cada caso se presentan diferentes variables que dificultan la comunicación.

Hoy en día no existe un modelo que sea ideal y que pueda aplicarse en todos los ambientes ni siquiera uno que sea funcional para un ambiente en todos los casos, por lo que en cada caso se debe tomar precauciones respectivas. Los modelos de propagación predicen la intensidad de la señal recibida en una antena a una cierta distancia del transmisor, así como la variabilidad de la intensidad de la señal en su proximidad (Fernandez, 2006).

Por la distancia entre el transmisor y receptor los modelos de propagación se pueden dividir en dos grupos: los modelos de pequeña escala que se enfocan en entornos interiores y tienen rangos de distancia de unos cuantos metros hasta unos cien metros aproximadamente y los de gran escala se usan en entornos externos, donde la distancia entre emisor y receptor va desde cientos de metros hasta algunos kilómetros. Los modelos pueden tomarse a partir de experiencias previas o generar nuevos en función de las características del ambiente en el que se va a instalar el prototipo de sistema. Estos modelos de propagación se pueden clasificar en empíricos-estadísticos, teóricos-deterministas, o una combinación de los anteriores (Fernandez, 2006).

Mientras que los modelos empíricos se basan en la extrapolación estadística de resultados a partir de medidas del fenómeno realizadas sobre el terreno, los modelos determinísticos se basan en los principios fundamentales de los fenómenos de propagación de ondas de radio (Fernandez, 2006). Ya sea mediante el uso modelos de propagación simples o modelos más complejos

como los empíricos, la distancia se puede obtener mediante el Indicador de la Fuerza de la señal recibida (RSSI), como se muestra en la ecuación (5).

Cuando la transmisión se la realiza en el espacio libre, es decir, en un caso ideal en el que no encontramos ningún tipo de obstáculo, la potencia decrece de manera cuadrática cuando la distancia aumenta, sin embargo existen reflexiones que pueden afectar la señal. El siguiente modelo predice las pérdidas en la trayectoria (pathloss) (Oguejiofor O.S, 2013):

$$PL(di)[dB] = PL(do)[dB] + 10n \log_{10} \left( \frac{di}{do} \right) \quad (8)$$

Donde:

$PL(di)[dB]$ : Pathloss.

$PL(do)[dB]$ : Pathloss a una distancia de referencia conocida  $do$ .

$n$ : Constante de Pathloss.

$di$ : Distancia entre el transmisor y receptor.

El Pathloss se lo puede modelar como una variable aleatoria al azar y una distribución logarítmica normal con una distancia dependiente del valor medio (Oguejiofor O.S, 2013):

$$PL(di) [dB] = PL(d0) [dB] + 10n \log_{10} \left( \frac{di}{do} \right) + S \quad (9)$$

Donde  $S$ , es el factor de shadowing que es una variable aleatoria gaussiana (en dB) con una desviación estándar  $\sigma$  [dB]. La constante de pathloss ( $n$ ) que es una constante empírica que depende del entorno de propagación se la puede calcular como (Oguejiofor O.S, 2013):

$$n = \frac{PL(di) - PL(d0)}{10 \log_{10} \left( \frac{di}{d0} \right)} \quad (10)$$

Determinar el valor de " n ", hace que sea posible obtener el valor real del RSSI durante las mediciones empíricas y este valor se relaciona con la distancia mediante la siguiente ecuación (Oguejiofor O.S, 2013):

$$RSSI [dBm] = -10n \log_{10} (d) + X [dBm] \quad (11)$$

Donde X, corresponde al RSSI [dBm] medido a un metro de distancia.

## CAPÍTULO 4

### DISEÑO E IMPLEMENTACIÓN

#### 4.1. Criterios de Diseño

El prototipo de sistema de *radiolocalización* estará conformado por una mota que se desconoce su posición, la cual se llamará desconocida, esta mota se va a encargar de enviar paquetes en difusión a las motas que se conocen sus posiciones, estas se llamarán fijas y van a servir como puntos de referencia para poder estimar la posición de la mota desconocida. Las motas fijas van a obtener del paquete el valor del RSSI y se encargarán de enviarlo a la mota base que va a estar conectado al gateway MIB520 que servirá de puente entre la PC y la red inalámbrica. Los datos enviados al gateway van a ser transmitidos a la PC a través del cable USB conectado a esta. En la Figura 15 se muestra el esquema de las comunicaciones utilizado para este trabajo.

Para poder identificar los valores de RSSI obtenidos de las transmisiones entre la mota desconocida y las motas fijas, se les deben asignar a las motas un identificador único, para conocer que motas están enviando y recibiendo los paquetes. Se realizarán varias mediciones para calcular la media de los valores obtenidos, porque como se comentó anteriormente puede existir algún tipo de reflexión, obstrucción o absorción que pueda afectar la señal electromagnética, estos efectos son dependientes del medio ambiente (en interiores, al aire libre, lluvia, edificios, entre otros.).

Una vez obtenidos los promedios se realizará una conversión a dBm y se procederá a despejar de la expresión (11) la distancia, por lo que quedaría en función del RSSI y de los valores  $n$  y  $X$  (expresión 12) que se adaptan a los cambios de potencia de transmisión y al entorno de comunicación.

$$d = 10^{\frac{X-RSSI}{10*n}} \quad (12)$$

Como se mencionó anteriormente el parámetro  $n$  se refiere a la constante que describe como la fuerza de la señal decrece cuando la distancia hacia el transmisor aumenta (ver Tabla 3). El modelo en el espacio libre ( $n=2$ ) es una idealización, por lo que se deben realizar previamente pruebas con diversos valores de  $n$ , hasta encontrar el que mejor se ajuste al entorno de trabajo.

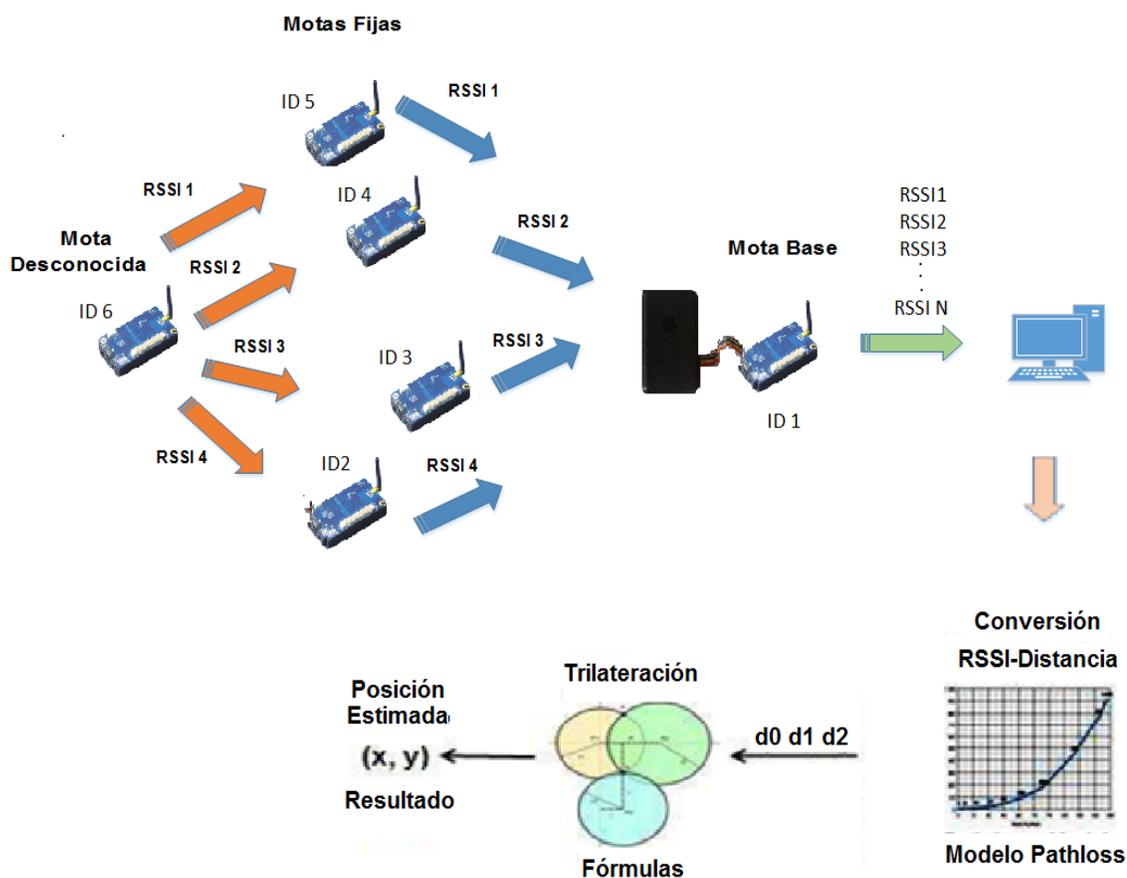


Figura 14 Esquema de Comunicaciones

El parámetro  $X$  representa la fuerza de la señal a un metro de distancia y debería ser igual en todas las direcciones, sin embargo en la práctica no se

cumple, porque este valor es calculado de modo diferente en cada uno de los receptores, por lo que al igual que los valores de RSSI obtenidos de la transmisión de la mota desconocida hacia las motas fijas, se debería utilizar un valor promedio de  $X$  realizando varios envíos.

**Tabla 3**

**Exponente de Pérdidas**

Ambiente	Exponente de Pérdidas [n]
Espacio Libre	2
Área Urbana con cobertura Celular	2.7 a 3.5
Shadowed Área Urbana con cobertura Celular	3 a 5
En edificio con línea de vista	1.6 a 1.8
En edificio sin línea de vista	4 a 6
En fabrica con obstáculos	2 a 3

Fuente: (Laaraiedh, 2009)

De la Expresión (12) se podrá obtener las distancias que separan a la mota desconocida de cada una de las motas fijas, las cuales serán empleadas en el sistema de ecuaciones propuesto por el algoritmo de Trilateración comentando anteriormente, donde las únicas incógnitas serán las coordenadas de la mota desconocida.

La resolución del sistema (4) comúnmente se suele hacer por el método de sustitución, sin embargo para este proyecto se va a utilizar el método LS (Least Squares), para minimizar el error y encontrar las coordenadas  $x_d$  y  $Y_d$  de una manera sencilla (Poellabauer, 2010). De las ecuaciones (1,2 y 3), se procede hacer restas entre las distancias para deshacerse de los términos cuadráticos:

$$d_b^2 - d_a^2 = (x_b - x_d)^2 + (y_b - y_d)^2 - (x_a - x_d)^2 - (y_a - y_d)^2 \quad (13)$$

$$d_c^2 - d_a^2 = (x_c - x_d)^2 + (y_c - y_d)^2 - (x_a - x_d)^2 - (y_a - y_d)^2 \quad (14)$$

$$d_c^2 - d_b^2 = (x_c - x_d)^2 + (y_c - y_d)^2 - (x_b - x_d)^2 - (y_b - y_d)^2 \quad (15)$$

Reordenando los términos se muestra una ecuación lineal en  $(x_d, y_d)$ :

$$2x_d(x_a - x_b) + 2y_d(y_a - y_b) = x_b^2 + y_b^2 - x_a^2 - y_a^2 + d_a^2 - d_b^2 \quad (16)$$

$$2x_d(x_a - x_c) + 2y_d(y_a - y_c) = x_c^2 + y_c^2 - x_a^2 - y_a^2 + d_a^2 - d_c^2 \quad (17)$$

$$2x_d(x_b - x_c) + 2y_d(y_b - y_c) = x_c^2 + y_c^2 - x_b^2 - y_b^2 + d_b^2 - d_c^2 \quad (18)$$

Las ecuaciones linealizadas en forma de matriz se convierten en:

$$\mathbf{A}\mathbf{p} = \mathbf{b} \quad (19)$$

Donde:

$$\mathbf{A} = \begin{pmatrix} 2x_d(x_a - x_b) & 2y_d(y_a - y_b) \\ 2x_d(x_a - x_c) & 2y_d(y_a - y_c) \\ 2x_d(x_b - x_c) & 2y_d(y_b - y_c) \end{pmatrix} \quad (20)$$

$$\mathbf{p} = \begin{pmatrix} x_d \\ y_d \end{pmatrix} \quad (21)$$

$$\mathbf{b} = \begin{pmatrix} x_b^2 + y_b^2 - x_a^2 - y_a^2 + d_a^2 - d_b^2 \\ x_c^2 + y_c^2 - x_a^2 - y_a^2 + d_a^2 - d_c^2 \\ x_c^2 + y_c^2 - x_b^2 - y_b^2 + d_b^2 - d_c^2 \end{pmatrix} \quad (22)$$

Si se desean incorporar más motas fijas de las necesarias para calcular la posición, las expresiones (20, 21 y 22) se adecuarían de la siguiente manera:

$$\mathbf{A} = \begin{pmatrix} 2x_d(x_a - x_b) & 2y_d(y_a - y_b) \\ 2x_d(x_a - x_c) & 2y_d(y_a - y_c) \\ 2x_d(x_b - x_c) & 2y_d(y_b - y_c) \\ \dots & \dots \\ 2x_d(x_{n-1} - x_n) & 2y_d(y_{n-1} - y_n) \end{pmatrix} \quad (23)$$

$$\mathbf{p} = \begin{pmatrix} x_d \\ y_d \end{pmatrix} \quad (24)$$

$$\mathbf{b} = \begin{pmatrix} x_b^2 + y_b^2 - x_a^2 - y_a^2 + d_a^2 - d_b^2 \\ x_c^2 + y_c^2 - x_a^2 - y_a^2 + d_a^2 - d_c^2 \\ x_c^2 + y_c^2 - x_b^2 - y_b^2 + d_b^2 - d_c^2 \\ \dots \\ x_n^2 + y_n^2 - x_{n-1}^2 - y_{n-1}^2 + d_{n-1}^2 - d_n^2 \end{pmatrix} \quad (25)$$

Por lo tanto, la posición del nodo desconocido se puede calcular minimizando  $\|A \cdot p - b\|^2$ . Al utilizar el método LS (Least Squares) se obtiene la solución de esta ecuación como:

$$p = (A^T A)^{-1} A^T b \quad (26)$$

#### 4.2. Implementación del Prototipo de Sistema de Radiolocalización WSN

Una vez realizado el diseño del prototipo de sistema de *radiolocalización* se deben tratar los aspectos relacionados con la instalación, programación, configuración y software necesario para alcanzar el funcionamiento deseado de la WSN utilizada para desarrollar el prototipo de sistema, entre estos aspectos se encuentran:

- **Programación de las motas:** Cada mota utilizada en la aplicación tiene una funcionalidad propia, por lo que es necesario disponer un programa en *NesC* para cada mota (desconocida, fija, base).
- **Adquisición de Datos:** Para la adquisición de los datos provenientes de la red inalámbrica se desarrolló una aplicación en Matlab.
- **Interfaz de Usuario y Procesamiento de Datos:** Se desarrolló una aplicación en GUI de Matlab para el procesamiento de la información y la configuración de la interfaz del usuario.

Para implementar la red inalámbrica se utilizó el kit de la plataforma de desarrollo Mote Works proporcionada por Crossbow que está dotado de un conjunto de herramientas de desarrollo de software para aplicaciones con WSN, incluidos los drivers para placa de sensores, acondicionamiento de la señal, procesamiento y manejo de mensajes. MoteWorks incluye un compilador optimizado para la tarjeta de la mota y un editor avanzado para el desarrollo de aplicaciones. Además instala y configura automáticamente estas herramientas de desarrollo para una rápida configuración.

Se hubiera podido realizar un programa en *NesC* para que las motas resolvieran las ecuaciones del sistema, no obstante se decidió adquirir y procesar los datos obtenidos de la red inalámbrica mediante GUI de Matlab por ser una herramienta de software que resuelve problemas matemáticos de manera rápida y sencilla. Además se desarrolló una interfaz de usuario que contiene diferentes elementos gráficos tales como: botones, campos de texto, gráficos, entre otros. Para que el usuario pueda de una manera rápida e intuitiva manejar el sistema de *radiolocalización*.

Una vez desarrolladas las distintas fases de programación que van a permitir la implementación del prototipo de sistema se verá necesaria una fase de pruebas para obtener ciertos parámetros necesarios para la puesta en marcha del sistema y que permitan el correcto funcionamiento en los escenarios que se va a trabajar.

Todo el proceso de instalación del software necesario para la implementación del proyecto se muestra en el Anexo A, y el código de la programación en MATLAB y en NesC se muestra en el Anexo B.

#### 4.2.1. Descripción del Algoritmo en Código NES C

La aplicación en NesC va a estar compuesta de un componente tipo módulo donde se va alojar el programa principal, un componente de tipo configuración para el ensamble entre las interfaces provistas y usadas por los componentes utilizados en la aplicación y un *Makefile* que es un archivo utilizado para compilar (Gay, 2005). El sistema operativo *TinyOS 2.x* proporciona un conjunto de interfaces que permite la comunicación vía radio entre las motas. Cuando un paquete es recibido por una mota, este se almacena en una estructura de datos llamada *message\_t*, que reemplaza a la estructura *TOS\_Msg* de *TinyOS 1.x*. Esta estructura está definida en */tos/types/message.h* y se compone de los siguientes campos (Levis, 2012):

- **header[sizeof(message\_header\_t):** el campo cabecera es un array de 11 bytes. Debido a que los stacks de radio a menudo prefieren los paquetes para ser almacenado de forma contigua, el diseño de un paquete en la memoria no refleja necesariamente el diseño de su estructura NesC.

- **Data [TOSH\_DATA\_LENGTH]:** el campo de datos de *message\_t* almacena la carga útil de un paquete de un solo salto y el tamaño de TOSH\_DATA\_LENGTH predeterminado es de 28 bytes, en cualquier aplicación de *TinyOS* se puede redefinir la longitud. Si se recibe un paquete cuyo tamaño de carga útil es mayor que la longitud de TOSH\_DATA\_LENGTH se descarta el paquete.
- **Footer [sizeof(message\_footer\_t):** el campo footer asegura que *message\_t*, tenga suficiente espacio para almacenar los footer de todos las capas de enlaces subyacentes.
- **Metadata [sizeof(message\_metadata\_t):** el campo metadata es un array de 7 bytes, almacena los datos que un stack de un solo salto.

No se puede acceder a los campos *header*, *footer* y *metadata* directamente, sin embargo la capa de enlace de datos proporciona acceso a los campos a través de las interfaces *NesC*. A diferencia de *TinyOS* 1.x, donde *TOS\_Msg* que es un paquete de mensajería activo, *message\_t* es más un búffer de enlace de datos. En la práctica las capas de enlace de datos en *TinyOs* 2.x proporcionan mensajería activa. Como se muestra en la Figura 15, la WSN va estar compuesta por una mota desconocida, mínimo 3 motas fijas y una mota base. Para la aplicación de *radiolocalización* las motas van a tener las siguientes funcionalidades:

- **Mota Desconocida:** Es la mota cuya posición es desconocida y va a transmitir mensajes cada cierto tiempo (50 ms) a las motas fijas en función de un temporizador. Al iniciar la transmisión se toma la parte correspondiente de la carga útil del mensaje en *message\_t* y se realiza una conversión a la estructura que se definió *MensajeRssi*, de esta manera se puede acceder al mensaje e inicializar los valores de las

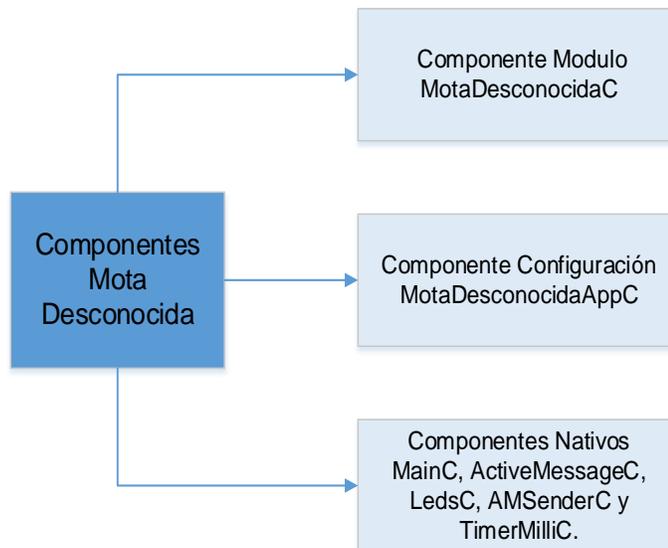
variables que se desean transmitir mediante un puntero. La mota desconocida va a escribir en el mensaje su propio ID de nodo, el valor del temporizador para que en los leds de las motas fijas se pueda visualizar cuando se están recibiendo los paquetes y una bandera para que la mota base pueda identificar los mensajes provenientes de la mota desconocida y descartarlos, porque solo nos va a interesar la información proveniente de las motas fijas.

- **Mota fijas:** Son motas cuya posición es conocida y van a servir como eje de referencia para poder calcular la posición de la mota desconocida. Estas motas reciben el mensaje, comprueban que el tamaño del paquete es el esperado y el payload del mensaje sufre una conversión a un puntero de estructura del tipo *MensajeRssi* para poder definir las variables que se van a enviar a la mota base. Al puntero se va asociar el valor RSSI medido, el ID de la mota fija y una bandera para que la mota base pueda identificar los mensajes provenientes de las motas fijas.
- **Mota Estación Base** Es la mota que se encuentra acoplada a la gateway, va a recolectar los mensajes provenientes de la red inalámbrica y comunicarlos a la PC.

### **Estructura de las Aplicaciones de las Motas** (Tinyos-2.x, 2008)

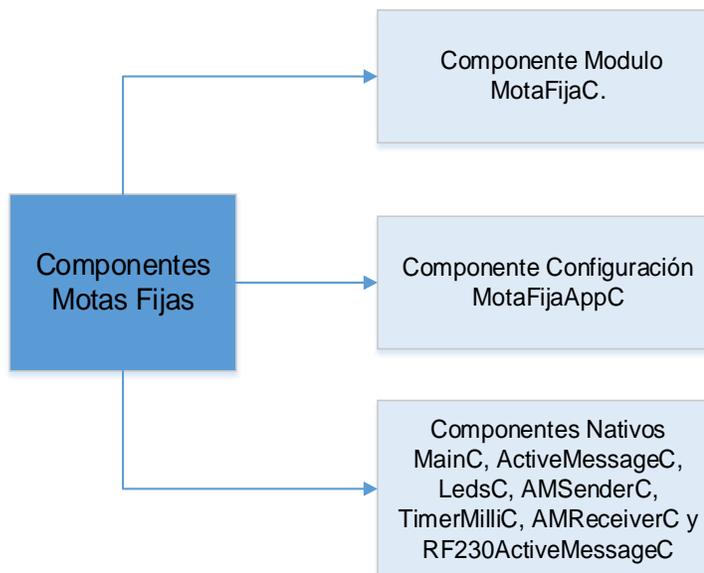
Se requieren un mínimo de 4 archivos para poder generar una aplicación en NesC. Según la funcionalidad que se implemente se puede variar el número de módulos que va a componer la aplicación. El esqueleto de la aplicación de *radiolocalización* en *TinyOS* consiste en las siguientes aplicaciones (ver Figura 16, 17 y 18):

- **Aplicación mota desconocida:**



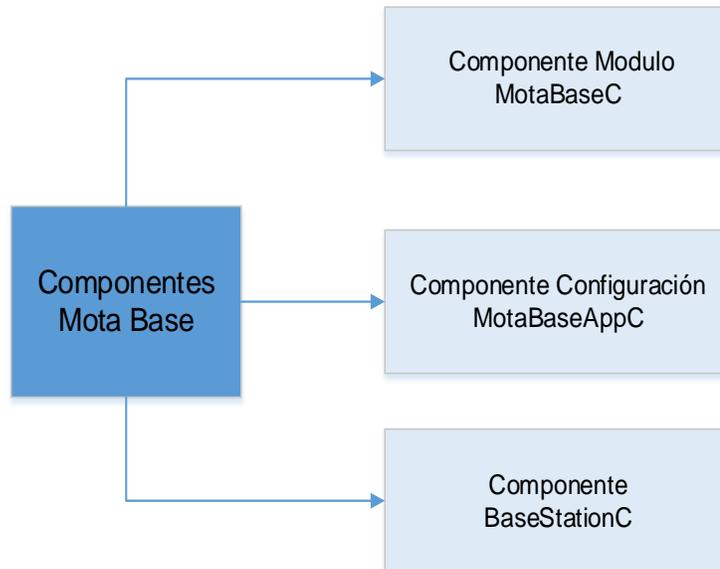
**Figura 15 Aplicación Mota Desconocida**

- **Aplicación mota fija:**



**Figura 16 Aplicación Mota Fija**

- **Aplicación mota base:**



**Figura 17 Aplicación Mota Base**

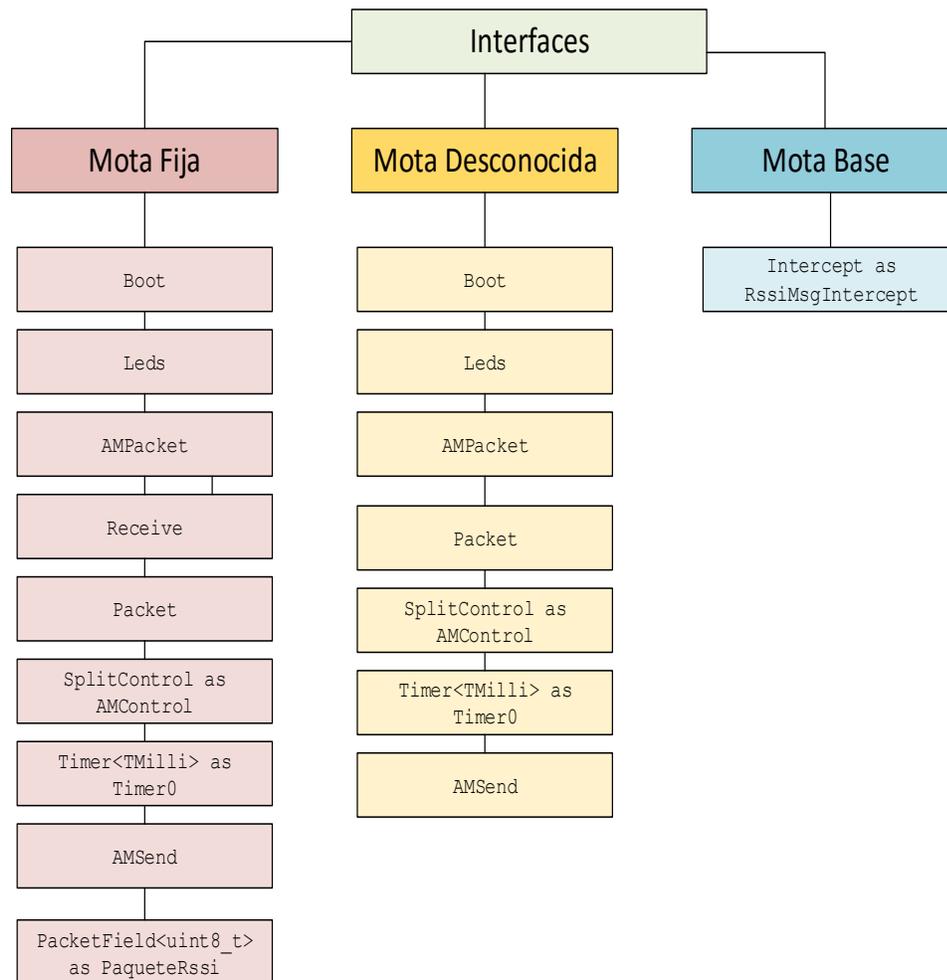
Los componentes nativos que implementan las interfaces de comunicación y las interfaces (Active Message), se encuentran en la carpeta /tos/system. Algunos de estos componentes son:

- **MainC:** implementa la interfaz *boot*.
- **TimerMilliC:** implementa la interfaz *timer*.
- **LedsC:** implementa la interfaz *Led*.
- **AMSenderC:** implementa interfaces como *AMSend*, *Packet*, *AMPacket*, entre otros.
- **AMReceiverC:** implementa las interfaces *Receive*, *Packet* y *AMPacket*.

- **ActiveMessageC:** implementa la interfaz *SplitControl*, es un componente que se utiliza para compatibilizar las interfaces antes expuestas con cada una de las arquitecturas que maneja *TinyOS*. Para la implementación del prototipo de sistema de *radiolocalización* se está trabajando con motas iris por lo que el componente *ActiveMessageC* necesario es implementado por *RF230ActiveMessageC*.

### **Interfaces** (Tinyos-2.x, 2006)

Para la programación de las motas es necesario el uso de interfaces que proporcionan comandos y eventos que implementan funciones para el desarrollo de la aplicación (ver Figura 19). No se puede acceder a ninguna función del módulo si antes no ha sido definida en una interfaz. La aplicación requiere timers (interfaz *Timer*), leds (interfaz *Leds*), interfaz *boot* que notifica a los componentes cuando *TinyOS* ha arrancado (inicializado todos sus componentes), interfaz *PacketField* para obtener el valor de RSSI del paquete e interfaces necesarios para la comunicación entre las motas. Estas interfaces se encuentran en la carpeta */tos/interfaces*.



**Figura 18 Interfaces**

### Interfaces para la comunicación (Tinyos-2.x, 2006)

- **Packet:** Interfaz que incluye métodos de acceso básicos para *message\_t*, proporciona comandos para borrar el contenido de un mensaje, obtener el tamaño de la carga útil del mensaje y establecer punteros.
- **Send:** interfaz que proporciona comandos para el envío y cancelación de mensajes (envía mensajes en difusión) y eventos para indicar si el mensaje fue enviado o no con éxito.

- **Receive:** interfaz que proporciona comandos para la recepción de mensajes y eventos para indicar si el mensaje fue recibido o no con éxito.

Puesto que es muy común tener múltiples servicios que utilizan el mismo radio para comunicarse, *TinyOS* proporciona la capa de mensajes activos Active Message (AM), para multiplexar el acceso al radio. El término "tipo AM" se refiere al campo utilizado para la multiplexación. Los tipos AM son similares al tipo de campo de la trama Ethernet, protocolo IP y el puerto UDP, porque todos ellos se utilizan para multiplexar el acceso a un servicio de comunicación. En el paquete AM se incluye un campo de destino, que almacena una "dirección AM" para direccionar paquetes a motas particulares. Las interfaces AM necesarios para la aplicación son las siguientes:

- **AMPacket:** incorpora comandos para obtener la dirección AM de una mota, el destino AM de un paquete y el tipo de paquete AM. La dirección AM de un nodo es el identificador que se asigna al programar la mota, este identificador puede cambiarse en tiempo de ejecución con el componente *ActiveMessageAddressC*.
- **AMSend:** es una interfaz con funcionalidades similares a send pero que necesita una dirección AM para identificar al destino.
- **SplitControl:** es la interfaz encargada de inicializar el transceptor radio.

### **Librerías** (Tinyos-2.x, 2006)

Para declarar las librerías que se van a utilizar en las aplicaciones de las motas se utiliza una directiva *include* que indica al compilador que busca el contenido

del archivos *Timer.h*, *message.h* y *mensajesrssi.h* en los sitios por defecto (subdirectorios de la carpeta /tos) y considere su contenido parte del programa.

Las comillas ordenan al compilador que busque la librería en la carpeta donde se encuentra el programa antes de buscar en los subdirectorios por defecto. En lugar de obtener y guardar los datos en *message\_t*, se utilizará una estructura *MensajeRssi* para almacenarlos y posteriormente copiarlos en el área de payload del mensaje. La utilización de estructuras para guardar datos en lugar de copiarlos directamente en *message\_t*, evita el tener que usar de índices para ubicar los datos en *message\_t*. En la librería *mensajesrssi.h* se va a encontrar el formato del mensaje que va a contener la información que se va a enviar vía radio (Levis, 2012):

```
typedef nx_struct MensajeRssi{
    nx_int16_t rssi;
    nx_uint16_t motadesconocida_id;
    nx_uint16_t motafija_id;
    nx_bool bandera;
    nx_uint16_t contador;
} MensajeRssi;
```

El formato contiene el prefijo *nx* que indica que la estructura contiene variables externas que se pueden utilizar en todas las aplicaciones e informa al compilador que se está usando el formato big-endian que permite trabajar con la información sin importar la arquitectura de la plataforma con la que se está trabajando.

Para evitar recuperar, guardar y ubicar los datos dentro de *message\_t*, es necesario definir estructuras para almacenarlos y copiarlos en el área de carga útil del mensaje. La estructura va a incluir el valor de RSSI, el Identificador de la mota emisora y de la mota receptora, el valor del contador y

una bandera para que la mota base pueda identificar los datos provenientes de las motas.

### **Envío y recepción** (Tinyos-2.x, 2008)- (Tinyos-2.x, 2006)

Para el envío de mensajes se utiliza el componente *AMSenderC* que implementa la interfaz *AMSend* necesaria para que las motas fijas y la mota desconocida puedan enviar los paquetes vía radio y las interfaces *AMPacket/Packet* para acceder al tipo abstracto de *message\_t*. También se utiliza el componente *ActiveMessageC* que llama a la interfaz *SplitControl* encargada de inicializar el transceptor de radio.

En la aplicación de las motas fijas se utiliza el componente *AMReceiverC* que implementa la interfaz *Receive* necesaria para poder recibir los mensajes que envía la mota desconocida. Se definió un contador que está en función de un temporizador de envío de paquetes (*Timer0.startPeriodic*) para que se pueda visualizar por los leds de las motas fijas si se están recibiendo los mensajes. Además se necesita una bandera para identificar si el transmisor está ocupado y un tipo de *message\_t* para llevar a cabo la transmisión.

```
uint16_t contador;
message_t pkt;
bool busy = FALSE;
```

Para que las motas fijas puedan obtener el valor de RSSI del paquete se define una variable *ObtenerRssi* que es un tipo de *message\_t*

```
uint16_t ObtenerRssi(message_t *msg);
```

Para la visualización del envío de los mensajes por parte de la mota desconocida se define un contador *encenderLeds* para que se muestren los tres bits menos significativos del contador en los leds de las motas.

```
void encenderLeds(uint16_t encender) {
    if (encender & 0x01)
        call Leds.led0On();
    else
        call Leds.led0Off();
    if (encender & 0x02)
        call Leds.led1On();
    else
        call Leds.led1Off();
    if (encender & 0x04)
        call Leds.led2On();
    else
        call Leds.led2Off();
}
```

Cuando se prenda la mota se va a inicializar la radio por lo que en el evento *Boot.booted* se va a llamar al evento *AMControl.start*. Se procede a arrancar el temporizador si la radio se ha inicializado con éxito (*error\_t = SUCCESS*), caso contrario se intenta inicializar nuevamente hasta que la radio pueda arrancar correctamente. Cuando se incrementa el temporizador (cada 50 ms) se envían los paquetes por el canal de radio. El evento *AMControl.stopDone* no se va a utilizar pero obligatoriamente debe implementarse por lo que se va a dejar vacío.

```
event void Boot.booted() {
    call AMControl.start();
}

event void AMControl.startDone(error_t err) {
    if (err == SUCCESS) {
        call Timer0.startPeriodic(50);
    }
    else {
        call AMControl.start();
    }
}

event void AMControl.stopDone(error_t err) {
}
```

En la aplicación de la mota desconocida cada vez que el temporizador se incrementa en el evento Timer0.fired se realiza lo siguiente:

- Se verifica que la bandera del transmisor esté libre y se procede a obtener la carga útil del mensaje.
- Se realiza una conversión a la estructura que se definió *MensajeRssi*, de esta forma se puede acceder al mensaje e iniciar los valores de las variables utilizando un puntero (mensaje). Las variables corresponden al ID de la mota desconocida, el contador para que se pueda visualizar el envío de los paquetes por los leds en las motas fijas y una bandera para que la mota base pueda descartar los paquetes provenientes de la mota desconocida.
- Una vez que la capa AM ha aceptado el mensaje para transmitir se cambia el estado de la bandera a ocupado.

```

event void Timer0.fired() {
    contador++;
    if (!busy) {
        MensajeRssi* mensaje = (MensajeRssi*)(call
Packet.getPayload(&pkt, sizeof(MensajeRssi)));
        if (mensaje == NULL) {
            return;
        }
        mensaje->motadesconocida_id = TOS_NODE_ID;
        mensaje->contador = contador;
        mensaje->bandera = FALSE;
        if (call AMSend.send(AM_BROADCAST_ADDR, &pkt,
sizeof(MensajeRssi)) == SUCCESS) {
            busy = TRUE;
        }
    }
}

```

El evento *AMSend.sendDone* se encarga de cambiar el estado de la bandera cuando se ha verificado que el paquete que se ha enviado es el mismo que el que se colocó en el buffer, esto es así porque si dos componentes ensamblan la misma interfaz *AMSend*, ambos recibirían un evento *sendDone* si alguno de los dos hubiese llamado al comando *send*.

```

event void AMSend.sendDone(message_t *msg, error_t error){
    if (&pkt == msg) {
        busy = FALSE;
    }
}

```

Para la recepción de paquetes provenientes de la mota desconocida se realiza en el evento *Receive.receive* lo siguiente:

- Se comprueba que el tamaño del mensaje es el esperado.
- La carga del mensaje sufre una conversión a un puntero a estructura del tipo *MensajeRssi*.

- En el mensaje la mota escribe su propio ID del nodo, se cambia la bandera a *true* para que la mota base procese los datos provenientes solo de la motas fijas y almacene el RSSI obtenido.
- Se llama al evento *encenderLeds* para mostrar el valor del contador en los leds.

```

event message_t* Receive.receive(message_t* msg, void*
payload, uint8_t len){
    MensajeRssi* mensaje = (MensajeRssi*)payload;
    return msg;

    mensaje->motafija_id = TOS_NODE_ID;
    mensaje->bandera = TRUE;
    mensaje->rssi = ObtenerRssi(msg);
    encenderLeds(mensaje->contador);
}

```

El evento *ObtenerRssi* implementa la interfaz *PaqueteRssi* que llama a un comando *get* para obtener el valor de RSSI del paquete, la definición de este evento va a depender de la arquitectura que trabajan las motas.

```

uint16_t ObtenerRssi(message_t *msg){
    if(call PaqueteRssi.isSet(msg))
        return (uint16_t) call PaqueteRssi.get(msg);
    else
        return 0xFFFF;
}

```

#### 4.2.2. Medición RSSI (ATMEL, 2009)

Para realizar la medición del valor del RSSI, se utilizó el valor de energía que es medido por el módulo Energy Detection (ED) del transceiver AT86RF230 y es almacenado en el registro PHY\_ED\_LEVEL. El modulo ED, forma parte del hardware de soporte especial que es parte del transceiver AT86RF230 y que

se utiliza para cumplir con el estándar IEEE 802.15.4-2003. El módulo ED tiene las siguientes características:

- 85 niveles de energía definidos.
- Resolución de 1 dB.

El valor medido por el módulo ED, es una estimación del nivel de la potencia de señal recibida dentro del ancho de banda del canal definido en IEEE 802.15.4-2003, y es calculado como un promedio de los valores de RSSI de ocho símbolos recibidos, en un tiempo estimado de 128 us. El valor calculado es almacenado en el registro llamado PHY\_ED\_LEVEL, y estará disponible 140 us después de su inicialización. Existen dos formas para poder iniciar una medición de ED:

- De forma manual, realizando una petición de lectura del registro PHY\_ED\_LEVEL.
- De forma automática, detectando los bits de inicio de trama (Start of Frame Delimiter -SFD) de un mensaje entrante.

En el presente trabajo para obtener los valores de RSSI, se utiliza la forma automática, ya que se realiza el envío de paquetes de datos entre las diferentes motas y de los paquetes de datos recibidos se obtiene el valor del RSSI. Para poder realizar esto se utiliza la siguiente bandera DRF230\_RSSI\_ENERGY, la cual se define en el *Makefile* de la aplicación de la mota fija.

```
ifneq ($(filter iris,$(MAKECMDGOALS)),)
    CFLAGS += -DRF230_RSSI_ENERGY
endif
```

### **Cálculo del RSSI a partir del valor obtenido del registro PHY\_ED\_LEVEL**

PHY\_ED\_LEVEL es un registro de 8 bits de longitud, en el cual se registran los valores de ED entre 0 a 84 dB con una resolución de 1 dB, un valor de 0 indica que el valor de energía medido es menor que -91 dBm. Para poder obtener los valores de RSSI se utiliza la siguiente relación, obtenida del datasheet del módulo AT86RF230:

$$RSSI [dBm] = PHY\_ED\_LEVEL - RSSI_{offset} \quad (27)$$

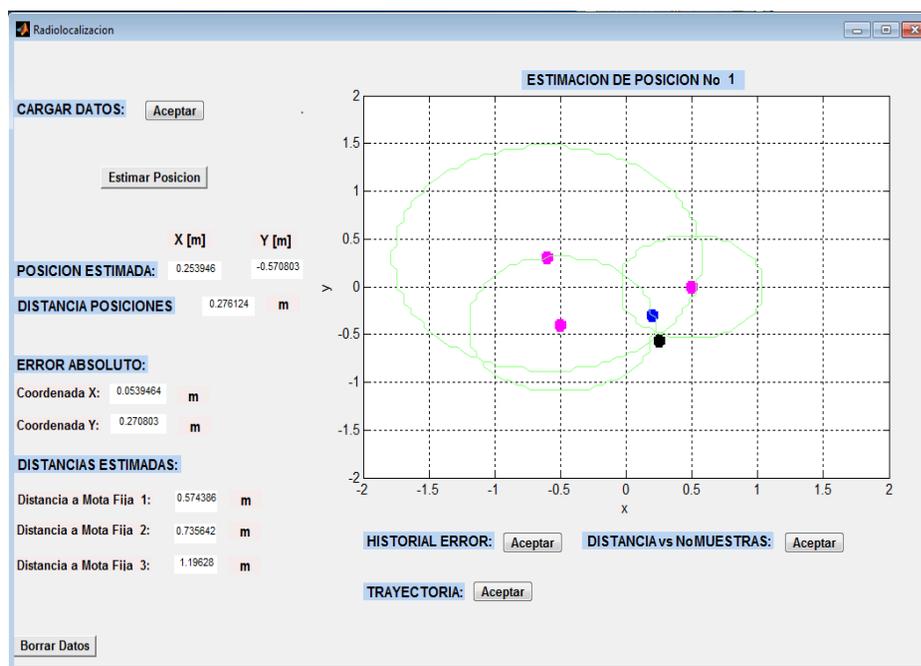
Donde:

$$RSSI_{offset} = -91 [dBm]$$

Si se tiene un valor de PHY\_ED\_LEVEL de 50, el Valor de RSSI será -41 [dBm]. Debido a factores de entorno (temperatura, voltaje, parámetros de los semiconductores, entre otros), el valor de energía calculado tiene una exactitud de  $\pm 5$  dB, este valor se considera como un offset sobre el rango de medida.

### **4.3. Aplicación de Monitorización de la Red**

La aplicación de monitorización cuenta con una interfaz desarrollada en GUI de Matlab y está compuesta por una parte de la adquisición de datos de la red y por otra de la resolución del algoritmo de *radiolocalización* y visualización de los resultados obtenidos. Se hubiera podido realizar un programa en NesC para la mota fija que resolviera las ecuaciones de las distancias, no obstante se decidió procesar los datos mediante GUI MATLAB, al ser una herramienta de software matemática que permite una fácil aplicación de los conceptos de *radiolocalización*.



**Figura 19 Ventana Principal**

En la Figura 20 se puede observar que la interfaz está compuesta por un componente gráfico donde se puede visualizar la red inalámbrica implementada, por un lado el color violeta está ligado a las motas que actúan como referencia, el color azul está asociado a la posición real de la mota desconocida y el color negro está ligado a la posición calculada por el algoritmo de trilateración tras la recepción de los RSSI. La ventana principal de la aplicación cuenta con 6 botones y cada uno tiene su propia funcionalidad:

**Botón (Cargar Datos):** Se carga otra ventana en donde se deben ingresar las coordenadas reales de las motas fijas y la mota desconocida en metros, el número de lecturas que se desea realizar, los valores de X (RSSI a un metro de distancia dBm) y n (constante de pathloss) (ver Figura 21). Una vez ingresados los datos se debe presionar el botón Aceptar y automáticamente se cargan en el programa principal para que puedan ser utilizados en los cálculos.

INGRESO DE COORDENADAS:		
	X [m]	Y [m]
Mota Fija 1:	0.5	0
Mota Fija 2:	-0.5	-0.4
Mota Fija 3:	-0.6	0.3
Mota Desconocida:	0.4	0.3

**Figura 20 Cargar Datos**

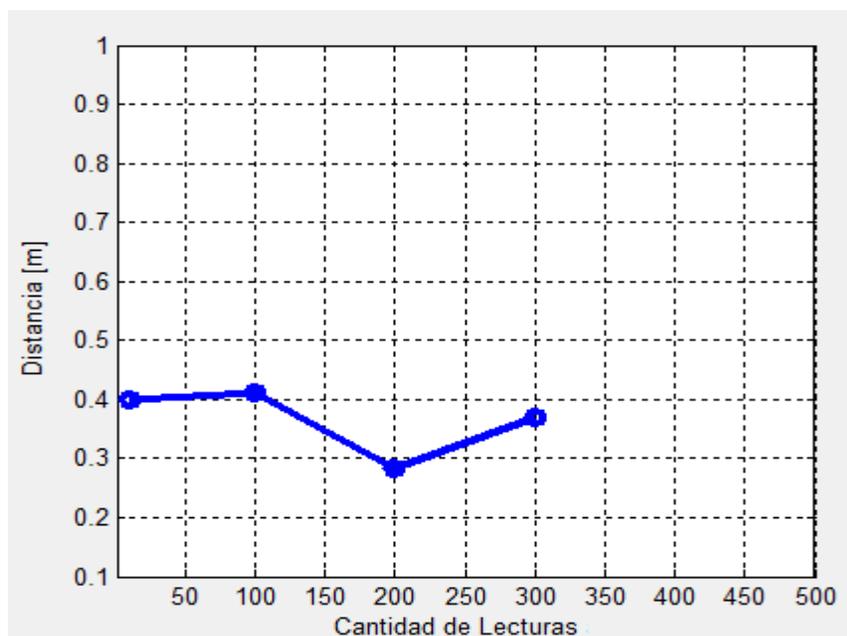
Botón (Estimar Posición): Se corre el sistema de *radiolocalización* y se realizan las siguientes funcionalidades.

- Se llama a la función de adquisición de datos y se crea un objeto del puerto serie COM4 (puede ser modificado por otro), especificando las unidades de señal por segundo (BaudRate), luego se procede a leer los datos del puerto que se muestran en el programa en formato entero. Se leen los paquetes hasta la cantidad de lecturas seleccionadas en el botón Cargar Datos. En el paquete leído se encuentra el ID de la mota que envía los paquetes (mota desconocida), el ID de la mota que los recibe y el valor del registro PHY\_ED\_LEVEL. Estos datos van a ser llevados al programa principal mediante esta función.
- Una vez que ya se tienen estos datos se procede a organizarlos de mejor manera para que puedan ser modificados y utilizados para el cálculo de las distancias. Con las distancias estimadas hacia cada mota fija y con

las coordenadas reales de las motas se procede a aplicar el algoritmo de trilateración para obtener la posición de la mota desconocida.

Botones (Trayectoria/ Historial del Error / Distancia vs No de Muestras): Como es posible que se trabaje muchas veces sobre un mismo escenario, se ha generado un modo de almacenamiento rápido de los datos obtenidos de las últimas configuraciones para ahorrar tiempo al usuario.

Cada vez que se presiona el botón estimar posición se puede formar una trayectoria en el plano con cada posición obtenida, además se puede calcular el error absoluto de las coordenadas (x,y) y calcular la distancia euclidiana entre la posición real de la mota desconocida de la posición estimada en función del número de lecturas para mostrar la precisión de la técnica de *radiolocalización*. En las Figuras 22, 23 y 24 se puede observar un ejemplo de los resultados obtenidos de prototipo de sistema de radiolocalización, en el cual se realizaron 4 mediciones con distintas cantidades de lecturas tomadas (10,100, 200 y 300).



**Figura 21 Ej. Distancia Euclidiana entre Posición Estimada**

y Real, en función de la Cantidad de Lecturas

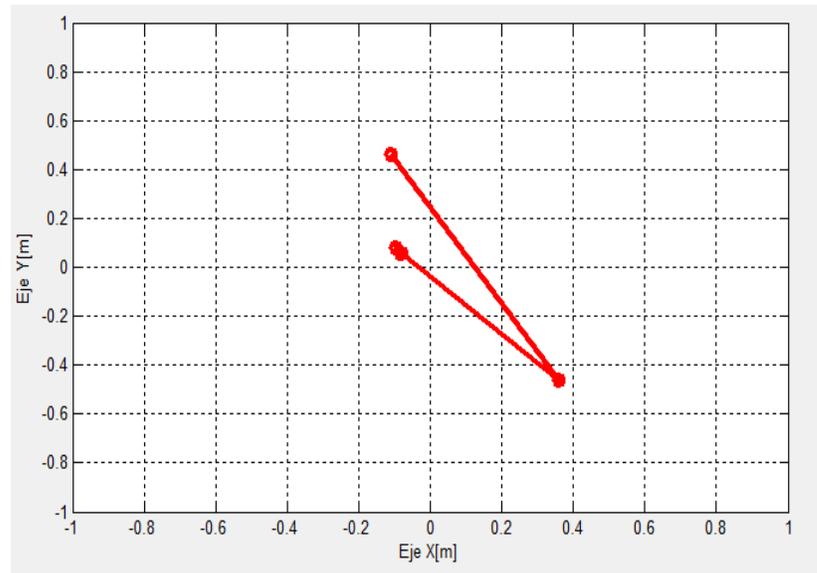


Figura 22 Ej. Trayectoria

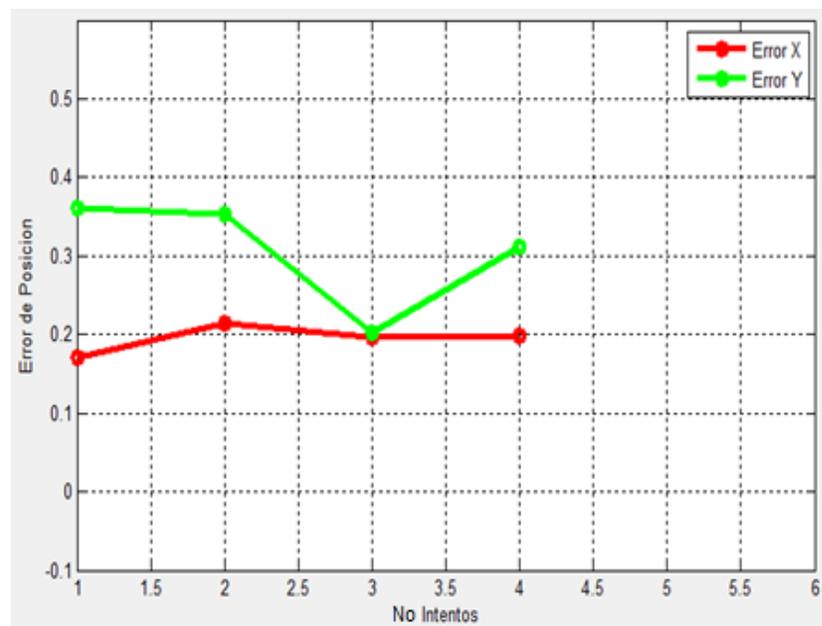


Figura 23 Ej. Error en X e Y

Botón (Borrar Datos): Se utiliza para borrar los datos de las casillas donde se muestran los datos obtenidos del sistema. Para una visualización más detallada de los cálculos realizados por el sistema, en la parte izquierda de la interfaz se han dispuesto campos informativos con los valores de la posición estimada, las distancias calculadas hacia cada mota fija y la distancia euclidiana entre la posición estimada y la posición real.

En la Tabla 4, se muestra un ejemplo de toma de muestras provenientes de la función de adquisición de datos en Matlab. Se observan los valores de PHY\_ED\_LEVEL de cada transmisión entre la mota desconocida (ID 6) hacia las motas fijas (IDs 2, 3, 4 y 5), posteriormente en el programa de procesamiento de datos se va a realizar la conversión de los valores a RSSI [dBm].

En una lectura se pueden tomar varias muestras hacia cada mota fija, como en el ejemplo se observa que en la lectura No 1 se realizan varios envíos a un mismo nodo, la cantidad de envíos puede variar según el número de lecturas. Para determinar el número de lecturas óptimo que permita que el sistema tenga un error aceptable, se planteó realizar 10, 100, 200, 300, 400 y 500 lecturas.

**Tabla 4**

**Toma de Muestras**

Numero de Lecturas	ID Mota Desconocida	ID Motas Fijas	PHY_ED_LEVEL
Lectura No1	6	5	44
Lectura No1	6	5	44
Lectura No1	6	3	30
Lectura No1	6	3	30
Lectura No1	6	4	26
Lectura No1	6	4	26
Lectura No1	6	2	37
Lectura No1	6	2	37

El tiempo de duración del sistema puede abarcar de 214 a 270 seg, distribuidos de acuerdo a la Tabla 5, todas las muestras medidas son consecutivas y se tarda en obtener cada muestra aproximadamente 50 ms, este tiempo se puede variar modificando el temporizador en la aplicación de NesC de envío de paquetes de la mota desconocida.

**Tabla 5**

**Tiempo de Espera del Sistema**

No Lecturas	Tiempo de Duración [seg]
10	2 a 3
100	15 a 20
200	34 a 39
300	46 a 60
400	50 a 73
500	67 a 75

**4.4. Calibración Escenarios**

Para evaluar el sistema de *radiolocalización* se planteó realizar pruebas donde se pueda modelar una propagación en espacio libre, por eso la implementación de prototipo de sistema se llevó a cabo en la parte posterior del Laboratorio de Electrónica. Como se muestra en la Figura 25 el primer escenario escogido esta elevado a 0.75 m de altura, situado sobre el césped para tratar de eliminar la atenuación debida a la reflexión provocada por el suelo y el segundo es un piso que no tiene ningún tipo de obstáculo, para minimizar las interferencias constructivas y destructivas debidas al multicamino (ver Figura 26).

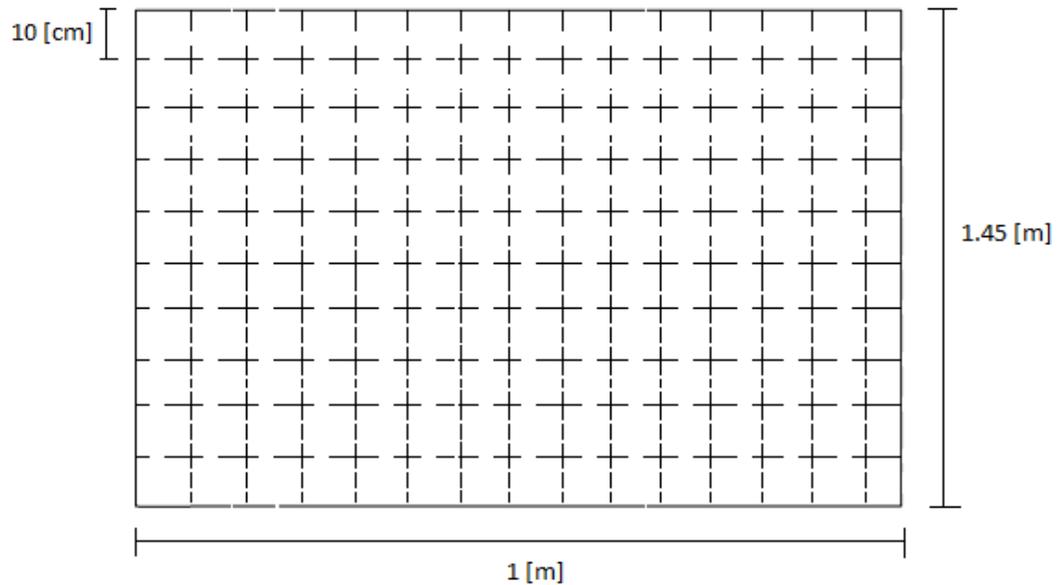


**Figura 24 Escenario 1 (sobre el césped, Elevado a 0.75 m de altura)**



**Figura 25 Escenario 2 (sobre el piso)**

Las motas iris tienen un alcance de 500 metros sin línea de vista y sin amplificación (Crossbow Technology), sin embargo para el sistema de *radiolocalización* se utilizó un eje de coordenadas de 1.45 m de ancho y 1 m de largo dividido cada 10 cm (ver Figura 27), para la resolución espacial y poder realizar las pruebas en los escenarios descritos anteriormente.

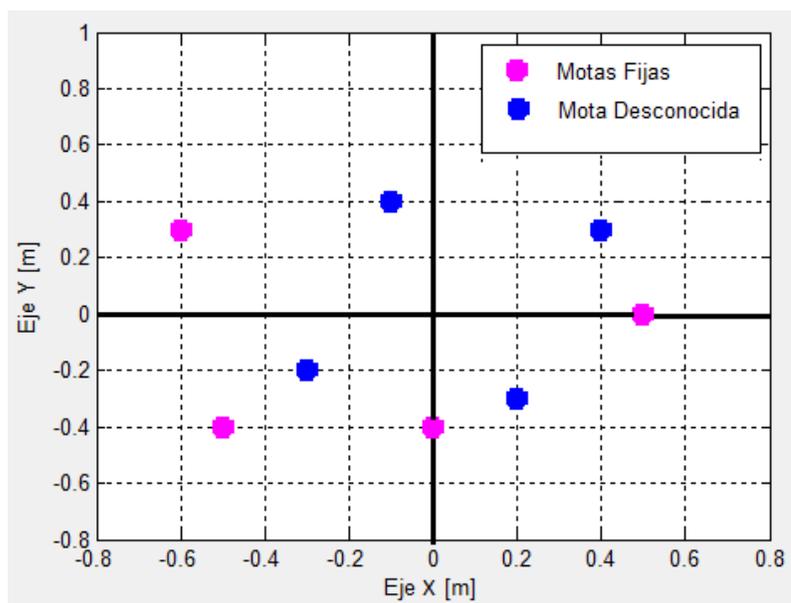


**Figura 26 Eje de Coordenadas**

Con las mediciones obtenidas se pretende mostrar el comportamiento del sistema frente a escenarios reales y determinar cuál es el mejor escenario de *radiolocalización* que proporcione menor error y robustez frente a los inconvenientes que puedan presentarse.

#### **4.5. Definición de Variables**

Antes de poder calcular la posición de la mota desconocida se deben establecer previamente las coordenadas de las motas fijas, las coordenadas reales de la mota desconocida y demás parámetros necesarios para la implementación. Esto dará una aproximación de la propagación de los entornos, la interferencia en el espectro de radio y los disturbios que puedan afectar los resultados medidos.



**Figura 27 Posición Motas**

Como se muestra en la Figura 28, para los dos escenarios se va a utilizar la siguiente configuración de Motas Fijas (ver Tabla 6). El algoritmo debe contener por lo menos 3 motas fijas para que pueda dar un resultado fiable, sin embargo también se van a realizar mediciones con otra mota, para conocer el comportamiento del sistema con 4 valores distintos de RSSI y la estimación de 4 distancias. La mota desconocida se va a posicionar en los cuatro cuadrantes para cada escenario de acuerdo a la Tabla 7.

**Tabla 6**

**Coordenadas Motas Fijas**

Nodos	Cordenada X [m]	Coordenada Y[m]
Mota Fija 1	0.5	0
Mota Fija 2	-0.5	-0.4
Mota fija 3	-0.6	0.3
Mota Fija 4	0	-0.4

**Tabla 7****Coordenadas Mota Desconocida**

Nodos	Cordenada X [m]	Coordenada Y[m]
Cuadrante 1	0.4	0.3
Cuadrante 2	-0.1	0.4
Cuadrante 3	-0.3	-0.2
Cuadrante 4	0.2	-0.3

Es necesario también obtener parámetros que modelen las pérdidas de propagación como son los valores de la constante de pathloss ( $n$ ) y RSSI medidos a un metro de distancia ( $X$ ). Con estos parámetros y el valor promedio de RSSI se podrán calcular las distancias de la mota desconocida hacia las motas fijas, necesarios para emplear el algoritmo seleccionado y obtener la posición (expresión 26).

**Tabla 8****X[m] (RSSI a un metro de Distancia)**

Escenarios	X[m] (RSSI a un metro)
Escenario 1	-62dBm /-56dBm
Escenario 2	-65 dBm

Como se comentó anteriormente se va a utilizar un valor promedio de  $X$ , esto se consigue realizando varios envíos a la misma mota, calculando la media entre los valores captados en esos envíos. Este parámetro debe obtenerse en los escenarios de trabajo para proporcionar una mejor representación de las pérdidas en función de la distancia. El valor promedio de  $X$  para los 2 escenarios se muestra en la Tabla 8 y gráficamente en la Figura 29.

El valor  $n$  depende del entorno de trabajo y es el que describe como la potencia de la señal decrece cuando la distancia aumenta (ver Tabla 3). Teniendo en cuenta que en la realidad no será un  $n$  igual a 2 como en espacio libre, se deberá encontrar un valor óptimo de  $n$  que mejor se adapte a los

ambientes de trabajo por lo que se realizarán mediciones de RSSI con diferentes valores de  $n$  (1, 1.5, 2 y 3), para saber cuál es el valor idóneo para el entorno con el que se va a trabajar.

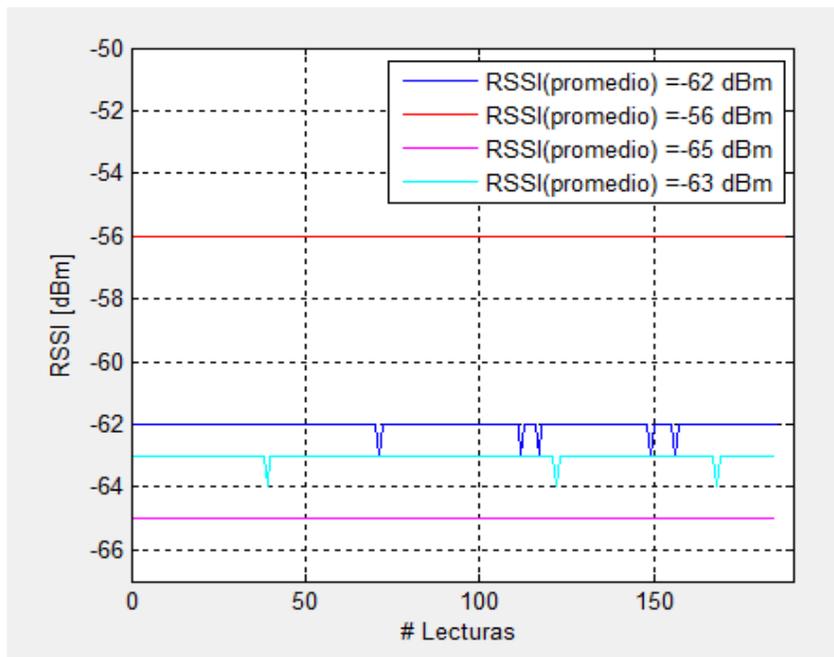


Figura 28 X[m] (RSSI a un metro de Distancia)

## CAPÍTULO 5

### ANÁLISIS DE RESULTADOS

En el presente capítulo se van a detallar las pruebas realizadas para validar el comportamiento del sistema de *radiolocalización* en diferentes entornos. Se va a realizar el análisis descriptivo de los resultados (media, varianza, distancias entre la mota desconocida con las motas fijas, error absoluto, distancia euclidiana entre la posición real y la estimada, RSSI en relación con la distancia, número de muestras óptimas, tiempos de proceso del sistema, etc.), por cuadrante efectuando una comparación de los dos escenarios descritos anteriormente, comentando las conclusiones de los resultados obtenidos.

Las distancias euclidianas entre la posición real y la posición estimada se van a realizar mediante la siguiente ecuación:

$$(x_d - x_{d_{est}})^2 + (y_d - y_{d_{est}})^2 = d \quad (28)$$

Donde:

$d$ : Distancia entre la posición estimada y la posición real de la mota desconocida [m].

$x_d$  y  $y_d$  = Posición real mota desconocida [m].

$x_{d_{est}}$  y  $y_{d_{est}}$  Posición estimada mota desconocida [m].

Entre los parámetros estadísticos que se van a analizar es la varianza que es la medida de dispersión que ostenta una variable aleatoria respecto a su esperanza, que se relaciona con la desviación típica o desviación estándar, la cual se denota como  $\sigma$ . La varianza se va a calcular mediante la siguiente fórmula:

$$\sigma = \frac{\sum_i (X_i - \bar{X})^2}{n} \quad (29)$$

Donde:

$X_i$ : Valores de RSSI.

$\bar{X}$  : Valor promedio de RSSI.

$n$ : Cantidad de muestras.

### 5.1. Análisis Primer Cuadrante

En las Tabla 9 y 10 se muestran las distancias euclidianas entre la posición estimada y la posición real, en función del número de motas fijas utilizadas, la cantidad de lecturas tomadas y los diferentes valores de  $n$  para identificar cuáles son los que mejor se adaptan al entorno de comunicaciones. En el escenario 1 el valor máximo de error es de 0.79 m y en el escenario 2 es de 0.53 m, ambos escenarios se realizaron con mediciones de 10 lecturas y con una constante  $n$  igual a 1. El valor mínimo de error en el escenario 1 es de 0.21m y del escenario 2 es de 0.04 m que corresponden a las mediciones de 100 y 200 lecturas respectivamente, para ambos escenarios la constante  $n$  es igual a 3.

**Tabla 9**

**Distancia Euclidiana entre Posición Real y Estimada [m]) (Escenario 1- 1er Cuadrante)**

No motas fijas	No Lecturas 10		No Lecturas 100		No Lecturas 200		No Lecturas 300		No Lecturas 400		No Lecturas 500	
	3	4	3	4	3	4	3	4	3	4	3	4
n=1	0.37	0.79	0.58	0.59	0.48	0.58	0.45	0.64	0.46	0.4	0.48	0.62
n=1.5	0.47	0.43	0.52	0.54	0.42	0.52	0.45	0.51	0.4	0.49	0.4	0.5
n=2	0.44	0.44	0.37	0.52	0.39	0.46	0.36	0.23	0.36	0.46	0.38	0.5
n=2.5	0.42	0.32	0.35	0.37	0.31	0.36	0.33	0.36	0.33	0.33	0.33	0.23
n=3	0.3	0.34	0.31	0.21	0.31	0.22	0.31	0.24	0.31	0.22	0.31	0.27

Tabla 10

**Distancia Euclidiana entre Posición Real y Estimada [m]) (Escenario 2-1er Cuadrante)**

No motas fijas	No Lecturas 10		No Lecturas 100		No Lecturas 200		No Lecturas 300		No Lecturas 400		No Lecturas 500	
	3	4	3	4	3	4	3	4	3	4	3	4
n=1	0.53	0.29	0.43	0.27	0.43	0.24	0.42	0.13	0.37	0.19	0.31	0.17
n=1.5	0.32	0.29	0.30	0.10	0.30	0.10	0.30	0.09	0.31	0.08	0.25	0.10
n=2	0.18	0.30	0.19	0.04	0.19	0.05	0.20	0.06	0.21	0.10	0.15	0.05
n=2.5	0.16	0.16	0.14	0.07	0.16	0.05	0.15	0.05	0.34	0.05	0.17	0.07
n=3	0.18	0.22	0.13	0.06	0.15	0.04	0.14	0.04	0.17	0.04	0.15	0.07

Se esperaba que  $n$  igual a 2 fuera la constante de pathloss que mejor condición de error tendría en el ambiente de propagación en espacio libre, sin embargo el valor óptimo de  $n$  fue de 3 ya que la señal sufrió dispersión de diferentes formas. Los cambios en las condiciones atmosféricas tales como la temperatura del aire también pueden afectar a la propagación de las ondas y por lo tanto a nivel de señal recibido. Desafortunadamente la frecuencia de 2.4 Ghz es la de resonancia del agua de forma que las personas, animales y plantas absorben las ondas de radio en esta banda de frecuencias donde trabaja el estándar ZigBee. Además pueden aparecer interferencias si otro dispositivo genera señales de radio en la misma frecuencia y con un valor de potencia comparable a la señal transmitida (Díaz & Tabernilla, 2010).

Se pretende analizar los parámetros obtenidos en la condición de mínimo y máximo error de posición en los diferentes escenarios. En la Tabla 11 se muestra la cantidad de lecturas, los parámetros  $X$  y  $n$  con las que se obtuvieron los valores obtenidos de cada condición, las posiciones estimadas de la mota desconocida y el error absoluto de las coordenadas  $X$  e  $Y$ .

Tabla 11

## Error de Posición Mínimo y Máximo (1er Cuadrante)

Escenario	Error Posición	No Lecturas	X [dBm]	n	Posición Estimada Mota Desconocida		Error Absoluto	
					X[m]	Y[m]	X[m]	Y[m]
Escenario 1	Máximo	10	-62	1	0.567	1.069	0.17	0.77
	Mínimo	100	-62	3	0.409	0.513	0.01	0.21
Escenario 2	Máximo	10	-65	1	0.908	0.158	0.51	0.14
	Mínimo	200	-65	3	0.438	0.285	0.04	0.02

En la Figura 30 en el escenario 1, el mayor error se produjo en el eje de coordenadas Y y en el segundo en el eje X, por lo que los valores de RSSI orientados a esos ejes en cada escenario fueron afectados por reflexiones, difracciones y dispersión. Se puede verificar que la orientación (vertical, horizontal o diagonal) entre las motas, tiene un efecto no despreciable sobre las mediciones de RSSI.

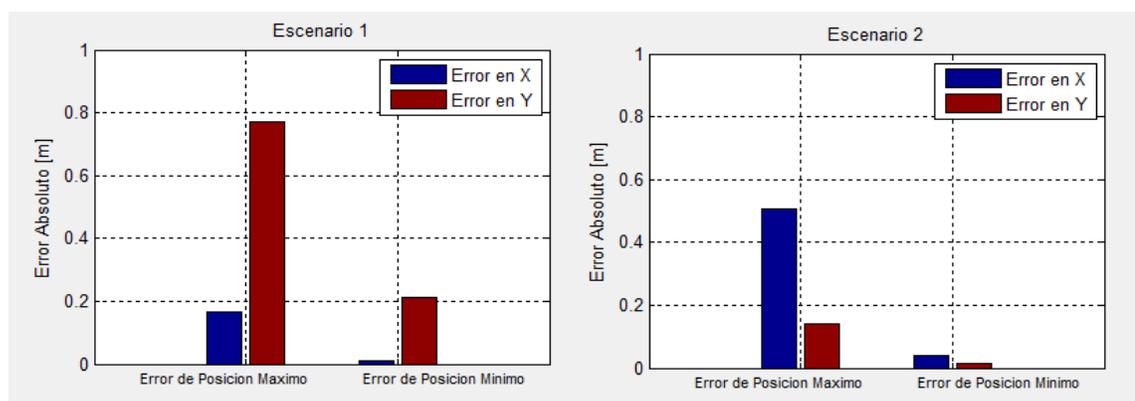


Figura 29 Error Eje X e Y (1er Cuadrante)

En las Tablas 12 y 13 se resume las distancias calculadas desde la mota desconocida hacia cada mota fija, los valores de RSSI promedios, las varianzas y el número de muestras de los datos tomados del error de posición máximo y mínimo. En el escenario 1 en el error de posición máximo se verifica que la distancia hacia la mota fija 2 supera el rango de eje de coordenadas (1.45m),

por lo que para disminuir el error del algoritmo, se configuró que si las distancias superan ese rango automáticamente se les asigne el valor de 1.45m. Así mismo en el escenario 2 se verifica que la distancia hacia la mota fija 2 y hacia la mota 3 supera el rango de eje de coordenadas, por lo que también se les asignó el valor de 1.45m para disminuir el error de localización.

No existe mucha diferencia entre los valores de RSSI de la condición con error máximo de la condición con error mínimo, sin embargo la constante n interfiere en el cálculo de las distancias por lo que al aumentar el n a 3 disminuye considerablemente el error de localización porque se ajustan los valores de las distancias estimadas a las reales.

**Tabla 12**

**Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 1-1er Cuadrante)**

Error de Posición	Distancia [m]		RSSI [dBm]		Varianza		Cantidad de Muestras	
	Max	Min	Max	Min	Max	Min	Max	Min
mota fija 1	0.32	0.39	-57	-49	0	0	19	199
mota fija 2	1.58	1.26	-64	-65	0	0	16	188
mota fija 3	0.79	0.91	-61	-61	0	0.2233	18	202
mota fija 4	1.25	0.81	-63	-59	0	0	18	190

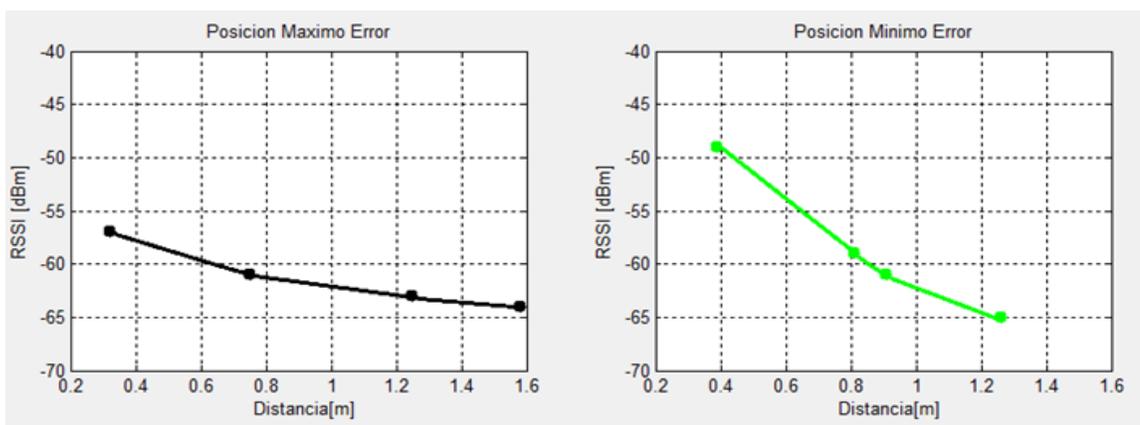
Tabla 13

**Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 2-1er Cuadrante)**

Error de Posición	Distancia [m]		RSSI [dBm]		Varianza		Cantidad de Muestras	
	Max	Min	Max	Min	Max	Min	Max	Min
mota fija 1	0.01	0.27	-41	-48	0	0.14	9	399
mota fija 2	1.58	1	-67	-67	0.21	0	10	452
mota fija 3	1.58	0.93	-67	-65	0	0	10	392
mota fija 4	-	0.72	-	-61	-	0	-	408

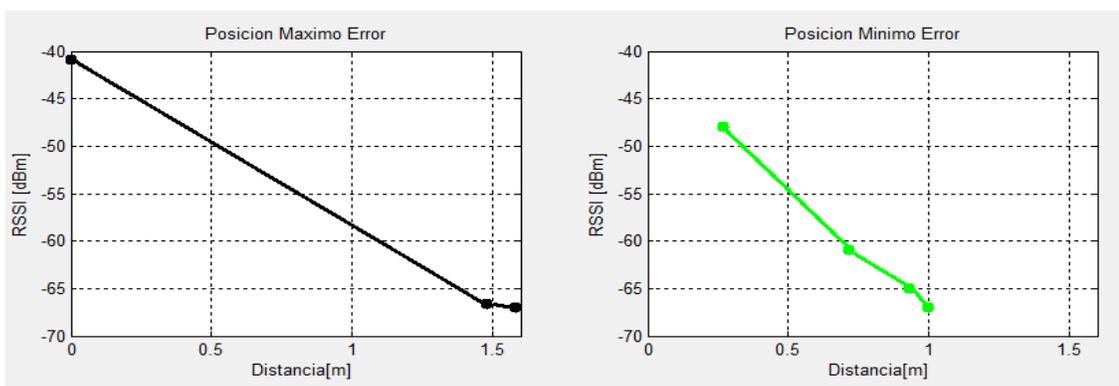
Las lecturas de RSSI tienden a variar en función del tiempo. Se muestra que existió una varianza en el escenario 1 igual a 0.2233 de 202 valores medidos en la mota fija 3. En el escenario 2 una varianza igual a 0.14 de 399 valores medidos en mota fija 1 y una varianza igual a 0.21 de 452 valores medidos en la mota fija 2.

En la Figura 31, se puede observar que la variabilidad del RSSI con la distancia es pequeña en los dos escenarios, esto es debe a que las motas fijas se encuentran estáticas en su sitio. Cuando la varianza es igual a 0 lo que significa es que los valores medidos en esas motas permanecieron constantes en función de la cantidad de muestras tomadas. Teóricamente la relación de la distancia con el RSSI es inversamente proporcional y en el caso práctico la distribución ideal si es aplicable para estos escenarios (distancias cortas) a pesar de existan interferencias.



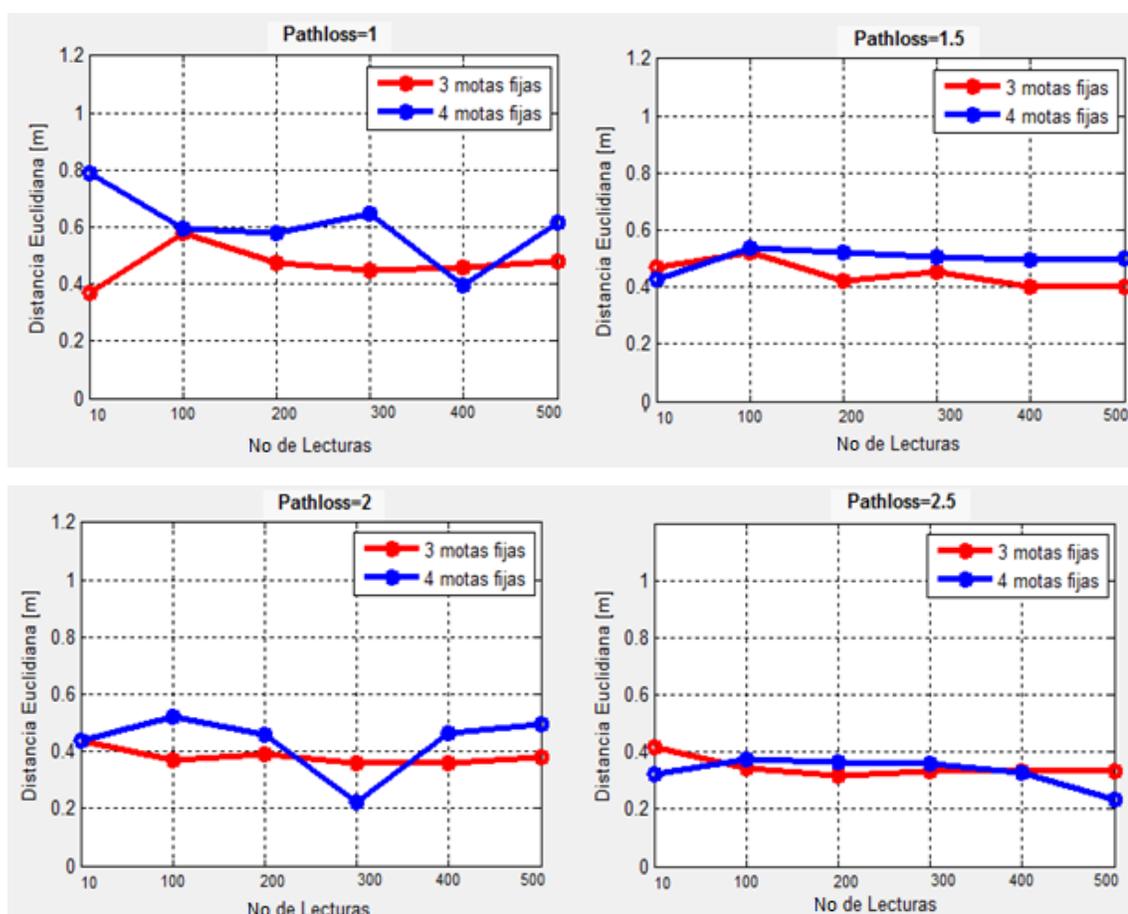
**Figura 30 Distancia[m]- RSSI [dBm] (Escenario 1-1er Cuadrante)**

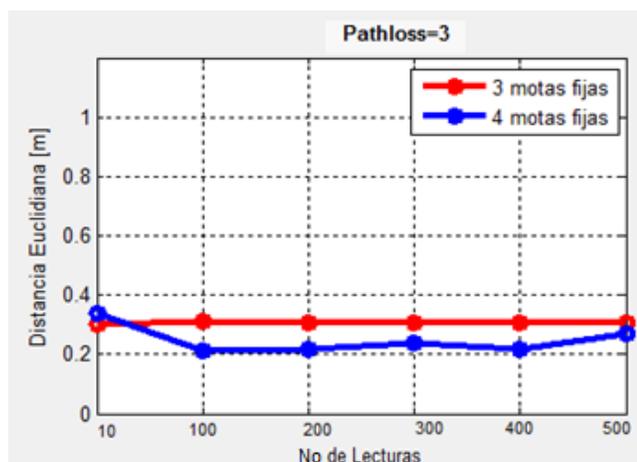
En el escenario 1 se puede observar que en la condición de máximo error la distancia alcanza casi 1.6m a -64 dBm, excediendo el eje de coordenadas implementado, pero el error se ajusta cuando se aplica la condición de mínimo error alcanzando una distancia de 1.26m a -65 dBm. Los valores de RSSI son similares en ambos escenarios, sin embargo como se comentó anteriormente en el cálculo de la distancia también influye los valores de  $n$  y de  $X$ . En el escenario 2 en la condición de máximo error se alcanza una distancia de 1.58m a -67 dBm (excediendo también el eje de coordenadas) y en la condición de mínimo error se alcanza una distancia de 0.93 m a -65 dBm.



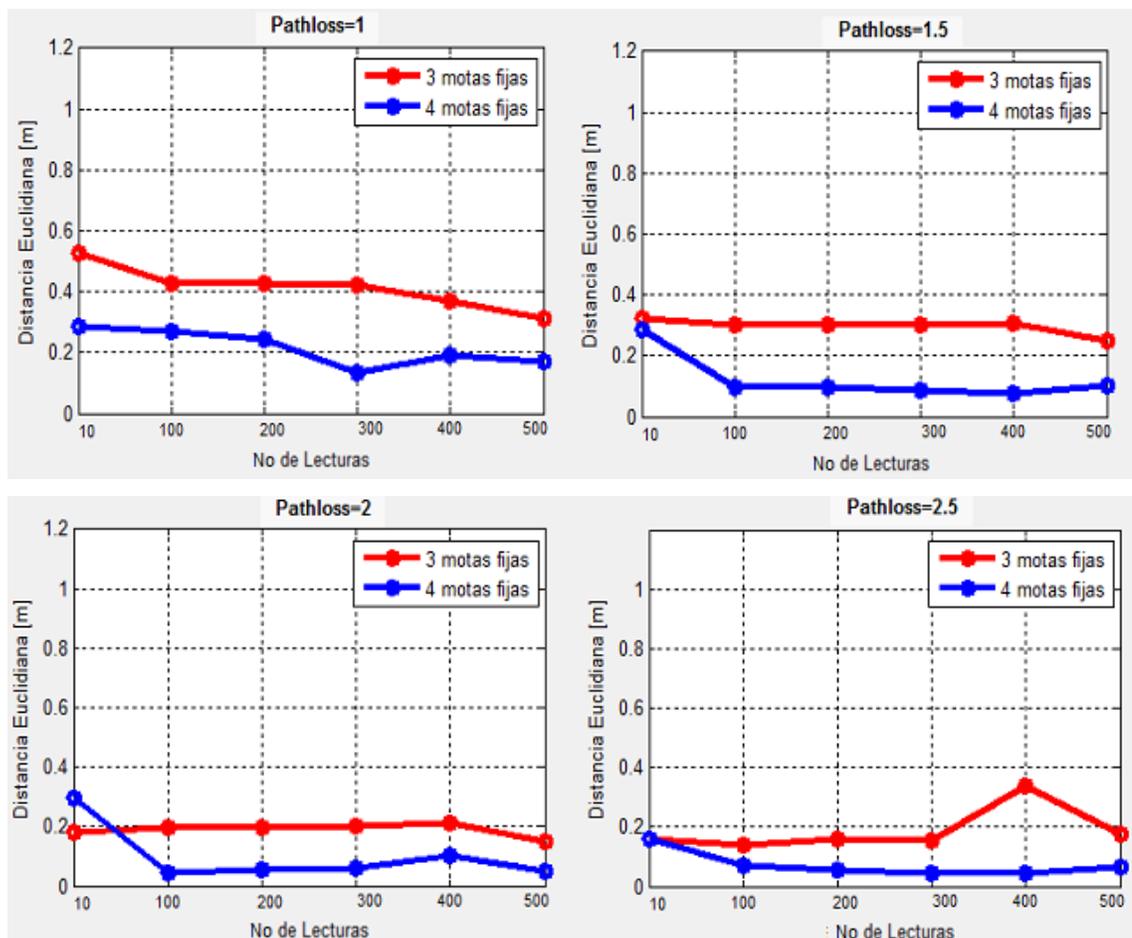
**Figura 31 Distancia[m]-RSSI [dBm] (Escenario 2-1er Cuadrante)**

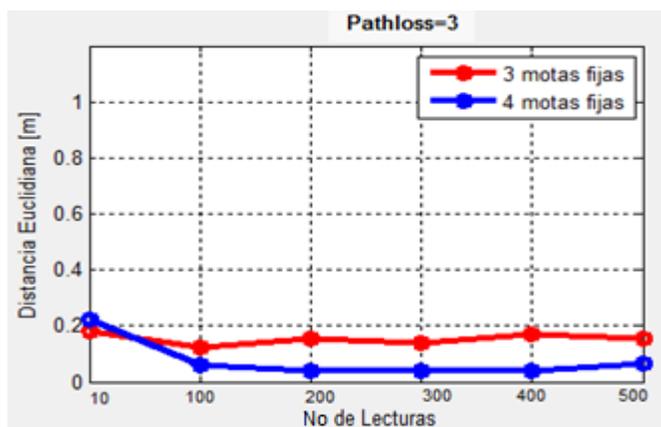
En la Figura 33, la curva de error presenta algunos picos pero tiende a ser constante en función de la cantidad de lecturas realizadas. Se puede observar que el error utilizando 4 motas fijas la curva se estabiliza a nivel de las 100 lecturas tomadas para  $n$  igual a 1.5, 2, 2.5 y 3, sin embargo los datos tomados con 3 motas el error permanece constante con algunos picos presentes. Se puede observar en la Figura 34, que en los diferentes valores de  $n$  las mediciones tomadas con 3 motas tienen más error que las mediciones tomadas con 4, esto se debe a que si se usan 3 puntos de referencia la influencia de cada mota es mayor y un valor de RSSI erróneo puede cambiar la posición estimada significativamente.





**Figura 32 Error de Posición al utilizar 3 y 4 motas (Escenario 1-1er Cuadrante)**





**Figura 33 Error de Posición al utilizar 3 y 4 motas (Escenario 2-1er Cuadrante)**

## 5.2. Análisis Segundo Cuadrante

Como se muestra en las tablas 14 y 15, en el escenario 1 el valor máximo de error es de 1.17 m y en el escenario 2 es de 0.64 m, ambos escenarios se realizaron con mediciones de 10 lecturas y con una constante  $n$  igual a 1. El valor mínimo de error en el escenario 1 es de 0.06 m que corresponde a la toma de 200 lecturas con un  $n$  de 2.5 y del escenario 2 es de 0.02 m que corresponden a las mediciones de 300 lecturas con un  $n$  igual a 3. Al igual que en el 1er cuadrante es evidente que el valor de  $n$  deberá ser mayor que 2 que sería el que se usara en una propagación ideal. Por esto se puede observar que la estimación de las distancias en ningún caso se ajusta a la realidad, con el parámetro de  $n$  igual a 3 la estimación parece bastante adecuada.

Tabla 14

**Distancia Euclidiana entre Posición Real y Estimada [m]) (Escenario 1-2do Cuadrante)**

No motas fijas	No Lecturas 10		No Lecturas 100		No Lecturas 200		No Lecturas 300		No Lecturas 400		No Lecturas 500	
	3	4	3	4	3	4	3	4	3	4	3	4
n=1	1.17	0.67	1.01	0.28	0.76	0.27	0.73	0.30	0.69	0.26	0.41	0.26
n=1.5	0.91	0.48	0.66	0.31	0.49	0.26	0.47	0.25	0.30	0.24	0.24	0.23
n=2	0.93	0.33	0.37	0.15	0.41	0.20	0.40	0.08	0.20	0.20	0.20	0.21
n=2.5	0.97	0.24	0.37	0.19	0.34	0.06	0.26	0.18	0.19	0.17	0.16	0.27
n=3	0.49	0.32	0.23	0.16	0.16	0.19	0.16	0.17	0.13	0.15	0.12	0.18

Tabla 15

**Distancia Euclidiana entre Posición Real y Estimada [m]) (Escenario 2-2do Cuadrante)**

No motas fijas	No Lecturas 10		No Lecturas 100		No Lecturas 200		No Lecturas 300		No Lecturas 400		No Lecturas 500	
	3	4	3	4	3	4	3	4	3	4	3	4
n=1	0.64	0.56	0.32	0.07	0.28	0.05	0.16	0.06	0.27	0.06	0.24	0.05
n=1.5	0.51	0.45	0.17	0.13	0.26	0.14	0.25	0.09	0.15	0.09	0.16	0.10
n=2	0.37	0.59	0.19	0.16	0.20	0.09	0.06	0.09	0.20	0.07	0.23	0.10
n=2.5	0.20	0.27	0.12	0.12	0.10	0.08	0.07	0.09	0.17	0.09	0.06	0.09
n=3	0.18	0.18	0.08	0.16	0.10	0.05	0.06	0.02	0.05	0.10	0.06	0.04

Como se puede observar en la Tabla 16, los valores de X difieren en el escenario 1, esto se debe a que las mediciones se realizaron en diferentes días y hubo cambios en las condiciones climáticas. Se muestra que la estimación de la posición con el valor mayor de X (-56 dBm), presentó más error de precisión en el eje Y, que los valores de X (-65 dBm).

Tabla 16

## Error de Posición Mínimo y Máximo (2do Cuadrante)

Escenario	Error Posición	No Lecturas	X [dBm]	n	Posición Estimada Mota Desconocida		Error Absoluto	
					X[m]	Y[m]	X[m]	Y[m]
Escenario 1	Maximo	100	-56	1	-0.436	1.355	0.336	0.955
	Mínimo	200	-65	2.5	-0.069	0.348	0.031	0.052
Escenario 2	Máximo	10	-65	1	0.324	0.878	0.424	0.478
	Mínimo	300	-65	3	-0.087	0.419	0.013	0.019

Se muestra en la Figura 35, que en los 2 escenarios el menor error se produjo en el eje X y el mayor se presentó en el eje Y. En el escenario 1, en la condición de error máximo, se aumentó el error en ambos ejes. El error en el eje Y alcanza aproximadamente 1 m, en este caso los niveles de la potencia de la señal no permanecieron estacionarios a lo largo del tiempo, los multitrayectos hicieron que los RSSI oscilen. En relación con el 1er cuadrante en la condición de error mínimo, tanto en el eje X como en el eje Y los errores permanecen casi iguales.

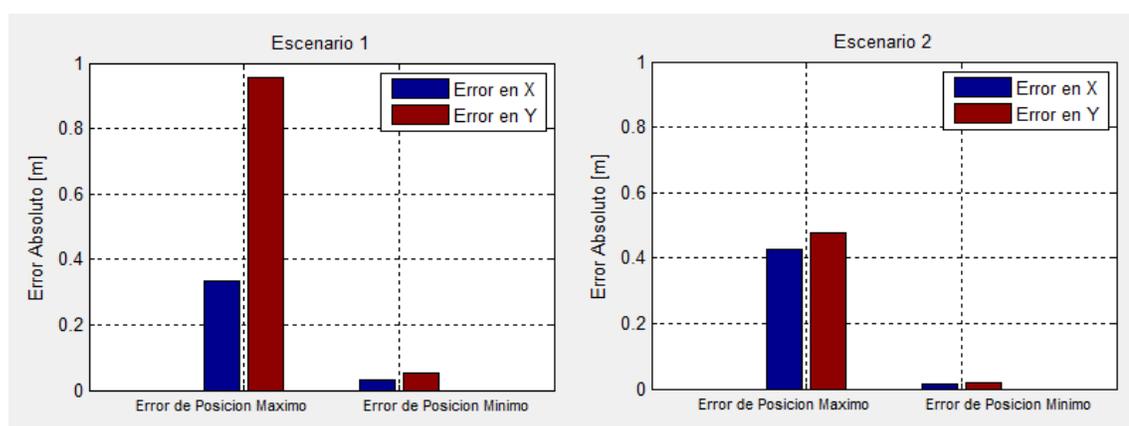


Figura 34 Error Eje X e Y (2do Cuadrante)

Como se puede observar en la Tabla 17, en el caso de la mota fija 3 la distancia aumento de manera considerable ya que superó el valor establecido de X a pesar de que los valores e RSSI no diferían mucho solo 5 dBm. En la Tabla 18 en la condición de máximo error, las distancias hacia la mota fija 2 y hacia la 3 superan el rango de eje de coordenadas, por lo que se les asignó el valor de 1.45m, a pesar de que los valores de RSSI son similares con los de la condición de mínimo error a excepción del valor tomado hacia la mota fija 2. En este caso el ruido del suelo produce un gran efecto en la potencia recibida, proporcionando un valor menor de RSSI al que se debería tener idealmente y por ello nos estima una distancia mayor.

**Tabla 17**

**Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 1-2do Cuadrante)**

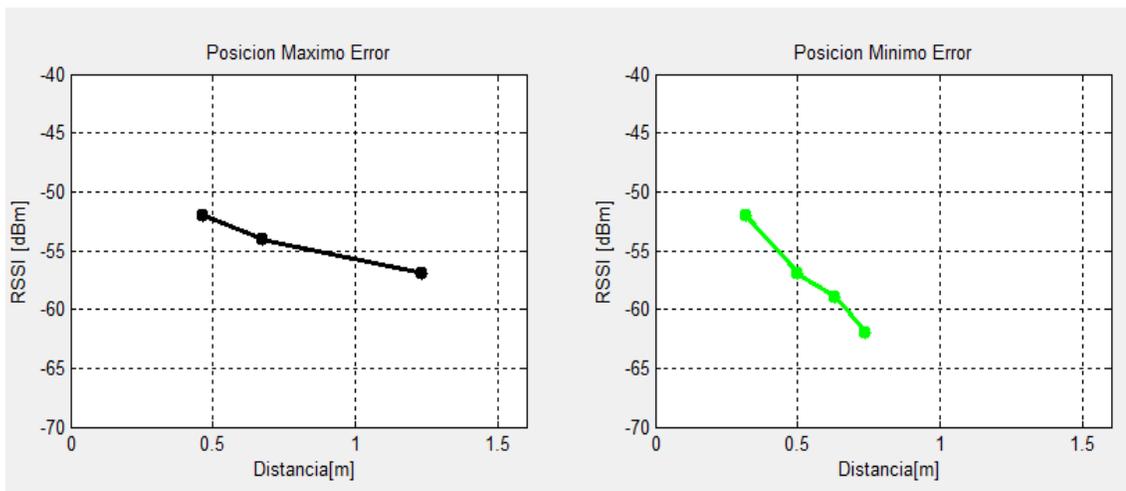
Error de Posición	Distancia [m]		RSSI [dBm]		Varianza		Cantidad de Muestras	
	Max	Min	Max	Min	Max	Min	Max	Min
<b>mota fija 1</b>	0.46	0.50	-53	-57	0.22	0	9	399
<b>mota fija 2</b>	0.67	0.74	-54	-62	0.198	0.015	11	390
<b>mota fija 3</b>	1.24	0.32	-57	-52	2.4097	0	12	402
<b>mota fija 4</b>	-	0.63	-	-59	-	0	-	390

Tabla 18

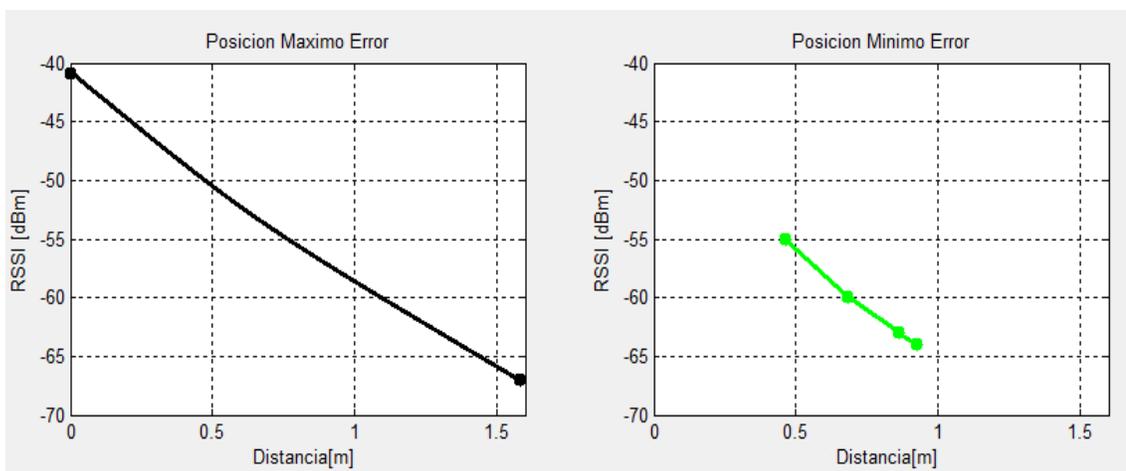
**Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 2-2do Cuadrante)**

Error de Posición	Distancia [m]		RSSI [dBm]		Varianza		Cantidad de Muestras	
	Max	Min	Max	Min	Max	Min	Max	Min
mota fija 1	0.004	0.69	-41	-60	0.2	0	9	799
mota fija 2	1.58	0.92	-67	-64	0.21	0.051	10	770
mota fija 3	1.58	0.46	-67	-55	0	0	10	796
mota fija 4	-	0.86	-	-63	-	0.002	-	808

Como se muestra en la Tabla 18, la posición de dos motas fijas coinciden en un mismo punto 1.584m a -67 dBm esto se debe a que los valores de RSSI de esas dos motas son iguales. Se puede observar que en la condición de mínimo error las distancias hacia la mota desconocida se ajustan mejor a las distancias reales. Como se muestra en la Figura 36, en la condición de mínimo error la curva decrece hacia valores más pequeños de RSSI, que en la curva de la condición de máximo error. En la Figura 37, se observa que en la condición de máximo error la distancia alcanza casi 1.6 m a -67 dBm, excediendo el eje de coordenadas implementado, pero el error se ajusta cuando se aplica la condición de mínimo error alcanzando una distancia de 0.92 m a -64 dBm hacia la mota fija 2 y de 0.4642 a -55dBm hacia la mota fija 3.



**Figura 35 Distancia[m]-RSSI [dBm] (Escenario 1-2do Cuadrante)**



**Figura 36 Distancia[m]-RSSI [dBm] (Escenario 2-2do Cuadrante)**

En la Figura 38, se muestra que la curva de error es decreciente hasta a que se estabiliza en algún punto. Para  $n$  igual a 1.5 el error se vuelve constante en 500 lecturas, para  $n$  igual a 2 y a 2.5 en 400 y para  $n$  igual a 3 a nivel de 100. Se verifica que se necesitan más de 500 lecturas, para que el error con  $n$  igual a 1 se estabilice. Al igual que en el escenario 1 la curva en el escenario 2 también tiene una tendencia decreciente (ver Figura 39). Para  $n$  igual a 1, 1.5, 2, 2.5 el error es constante a nivel de 100 lecturas tomadas y para

n igual a 3 aproximadamente a nivel de 200. Al igual que en el 1er cuadrante, los errores son mayores cuando se utilizan solo 3 motas fijas.

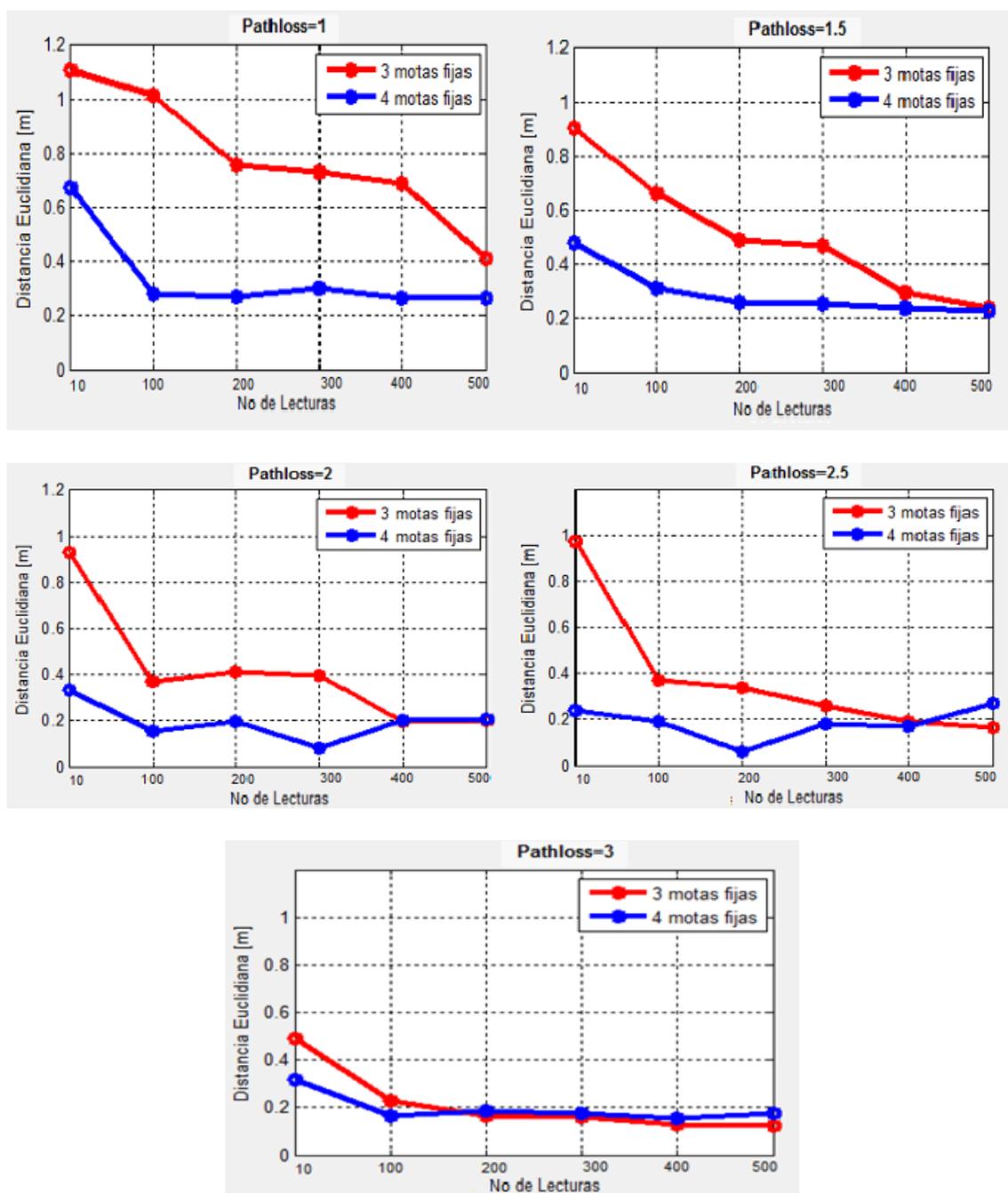


Figura 37 Error de Posición al utilizar 3 y 4 motas (Escenario 1-2do Cuadrante)

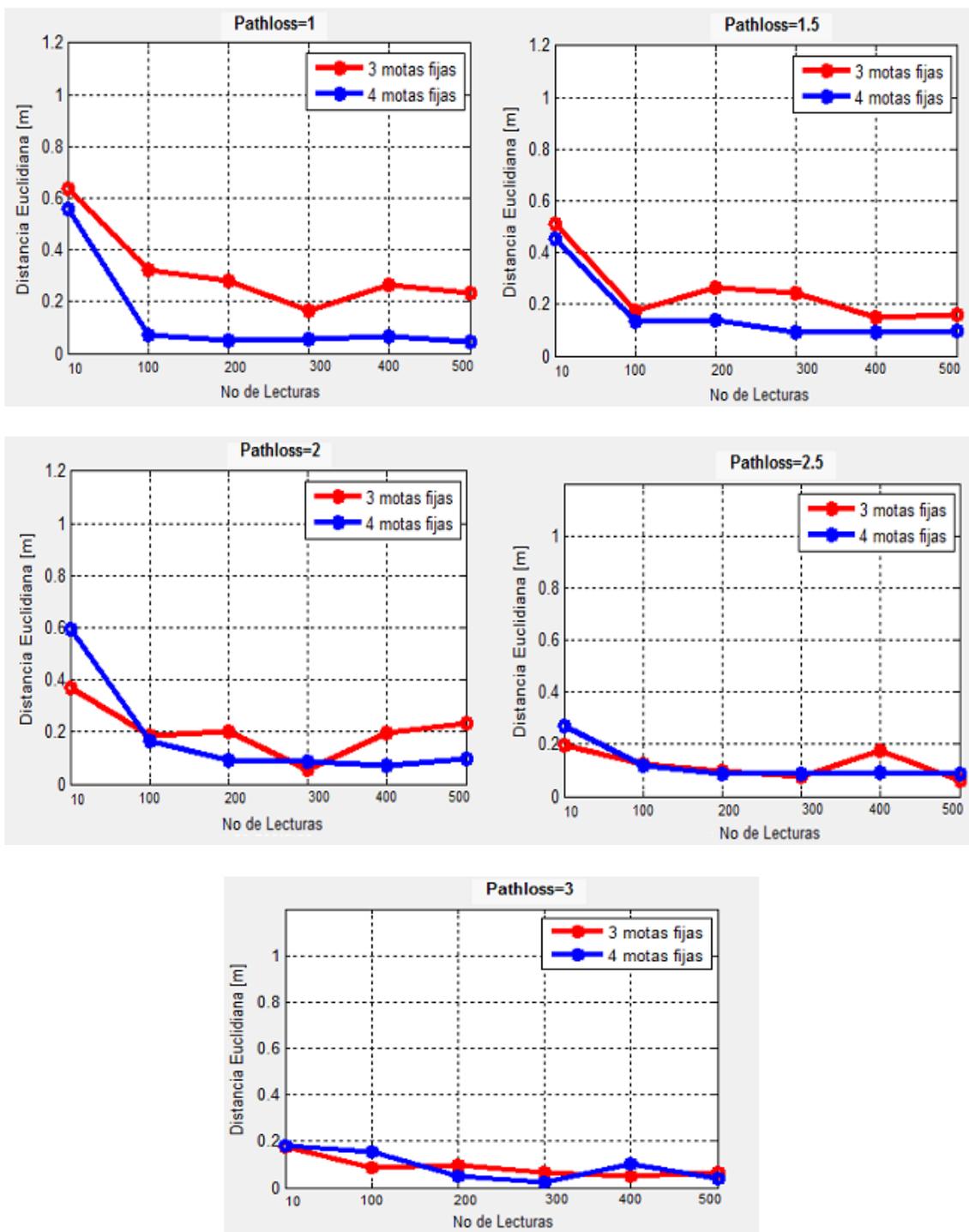


Figura 38 Error de Posición al utilizar 3 y 4 motas (Escenario 2-2do Cuadrante)

### 5.3. Análisis Tercer cuadrante

Como se muestra en las Tabla 19 y 20, en el escenario 1 el valor máximo de error es de 0.70 m y en el escenario 2 es de 0.39 m, ambos escenarios se realizaron con mediciones de 100 lecturas y con una constante n igual a 1. El valor mínimo de error en el escenario 1 es de 0.07 m que corresponde a la medición de 400 lecturas con un n igual a 3 y en el escenario 2 es de 0.03 m que corresponden a las mediciones de 500 lecturas con un n igual a 1.

**Tabla 19**

**Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 1- 3er Cuadrante)**

No motas fijas	No Lecturas 10		No Lecturas 100		No Lecturas 200		No Lecturas 300		No Lecturas 400		No Lecturas 500	
	3	4	3	4	3	4	3	4	3	4	3	4
n=1	0.67	0.14	0.70	0.15	0.65	0.16	0.61	0.34	0.66	0.14	0.48	0.14
n=1.5	0.67	0.16	0.46	0.14	0.51	0.14	0.43	0.14	0.47	0.13	0.45	0.14
n=2	0.23	0.13	0.50	0.14	0.44	0.13	0.41	0.13	0.32	0.12	0.41	0.09
n=2.5	0.19	0.12	0.42	0.14	0.36	0.15	0.28	0.12	0.40	0.14	0.39	0.09
n=3	0.17	0.11	0.32	0.13	0.32	0.12	0.27	0.11	0.32	0.07	0.32	0.12

Tabla 20

**Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 2- 3er Cuadrante)**

No motas fijas	No Lecturas 10		No Lecturas 100		No Lecturas 200		No Lecturas 300		No Lecturas 400		No Lecturas 500	
	3	4	3	4	3	4	3	4	3	4	3	4
n=1	0.25	0.26	0.25	0.39	0.22	0.29	0.29	0.18	0.23	0.13	0.23	0.03
n=1.5	0.20	0.16	0.20	0.09	0.27	0.12	0.18	0.12	0.27	0.12	0.12	0.20
n=2	0.17	0.06	0.15	0.05	0.15	0.07	0.15	0.08	0.15	0.18	0.15	0.08
n=2.5	0.14	0.16	0.13	0.07	0.14	0.09	0.13	0.09	0.13	0.09	0.13	0.10
n=3	0.11	0.07	0.11	0.07	0.11	0.06	0.12	0.06	0.11	0.06	0.12	0.06

Se puede verificar que en los escenarios con mayor error, corresponde a la medición de 10 y 100 lecturas, se comprueba que entre menos lecturas se realicen se obtiene una menor precisión de la posición, sin embargo al utilizar más muestras se incorpora una mayor congestión en la red y una mayor espera en la respuesta del sistema. Como se muestra en la Tabla 21, los valores de X difieren en el escenario 1, esto se debe a que las mediciones se realizaron en diferentes días y hubo cambios en el clima.

Tabla 21

**Error de Posición Mínimo y Máximo (3er Cuadrante)**

Escenario	Error Posición	No Lecturas	X [dBm]	n	Posición Estimada Mota Desconocida		Error Absoluto	
					X[m]	Y[m]	X[m]	Y[m]
Escenario 1	Máximo	100	-56	1	0.182	-0.712	0.482	0.512
	Mínimo	400	-62	3	-0.265	-0.136	0.035	0.064
Escenario 2	Máximo	10	-65	1	-0.714	-0.366	0.414	0.166
	Mínimo	500	-65	1	-0.298	-0.234	0.001	0.034

En la Figura 40, se muestra que en el escenario 1, en la condición de error máximo, el error en el eje Y aumentó, y en el eje X disminuyó con respecto al 2do cuadrante, sin embargo en el escenario 2 se disminuyó el error en ambos ejes. Como se muestra en la Tabla 22, en relación con el 1er y 2do cuadrante en la condición de error mínimo, tanto en el eje X como en el eje Y los errores no superan los 0.10 m, por eso la diferencia entre estos valores no es significativa.



**Figura 39 Error Eje X e Y (3er Cuadrante)**

En la práctica los RSSI permanecen casi constantes en función del tiempo, solo varían en uno o dos valores, por lo que la presencia de oscilaciones no dificulta a la hora de conseguir unas buenas estimaciones de distancias. Como se observa en la Tabla 22, en la condición de mínimo error, la mayor parte de mediciones hacia las motas fijas tuvieron varianza diferente de cero. En el escenario 2 la varianza se presentó hacia la mayor parte de las motas fijas en la condición de máximo error de posición (ver Tabla 23). A diferencia del 1er y 2do cuadrante no se estimaron distancias que superen el eje de coordenadas implementado en el proyecto, por lo que no se asignó a ninguna distancia el valor de 1.45 m.

Tabla 22

**Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 1-3er Cuadrante)**

Error de Posición	Distancia [m]		RSSI [dBm]		Varianza		Cantidad de Muestras	
	Max	Min	Max	Min	Max	Min	Max	Min
mota fija 1	0.33	0.93	-51	-61	0	0.112	199	399
mota fija 2	0.25	0.23	-50	-43	0	0.043	202	404
mota fija 3	1.06	0.47	-56	-53	0.165	0.480	200	405
mota fija 4	-	0.24	-	-46	-	0	-	392

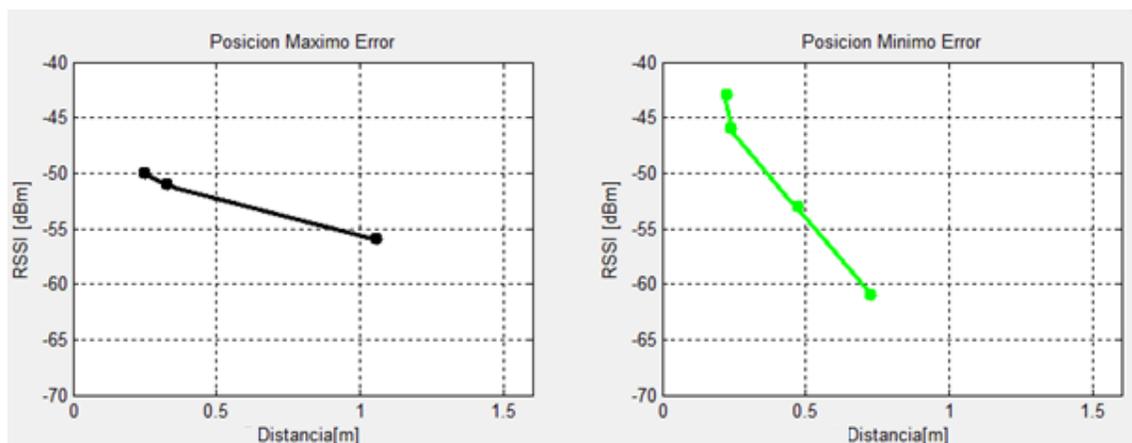
Tabla 23

**Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 2-3er Cuadrante)**

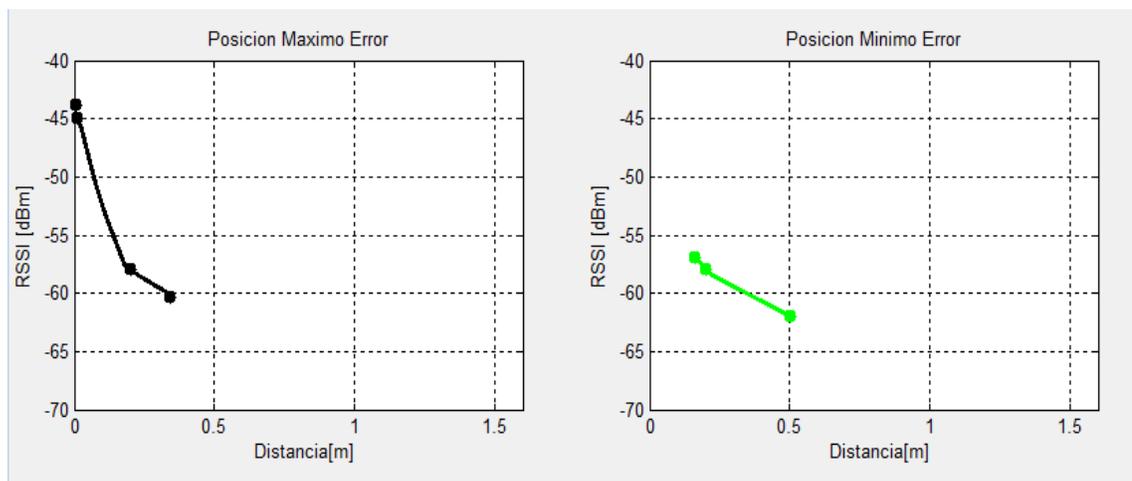
Error de Posición	Distancia [m]		RSSI [dBm]		Varianza		Cantidad de Muestras	
	Max	Min	Max	Min	Max	Min	Max	Min
mota fija 1	0.20	0.51	-58	-62	0	0	99	999
mota fija 2	0.34	0.16	-60	-57	0.252	0	94	986
mota fija 3	0.01	0.50	-45	-62	0.160	0	87	994
mota fija 4	0.007	0.20	-43	-58	0.601	0	94	1024

Como se muestra en las Figuras 41 y 42, al igual que en los otros cuadrantes se comprueba que a mayor distancia, menor RSSI. En el escenario 1 se puede observar que la curva está más inclinada en la condición de menor error, porque alcanza valores más grandes de RSSI. Sucede lo contrario en el escenario 2 donde los RSSI en la condición de mínimo error no alcanzan valores grandes. La precisión en la posición va a depender si los valores de

RSSI medidos se ajustan a los valores reales para poder estimar las distancias con mínimo error.



**Figura 40 Distancia[m]-RSSI[dBm] (Escenario 1-3er Cuadrante)**



**Figura 41 Distancia[m]-RSSI[dBm] (Escenario 2-3er Cuadrante)**

Al igual que en los otros cuadrantes, como se muestra en las Figuras 43 y 44, se puede observar que el aumento de 3 a 4 motas fijas tiene un efecto importante en la disminución del error de localización. Para los dos escenarios la curva de error es constante en función de la cantidad de lecturas con la presencia de algunos picos.

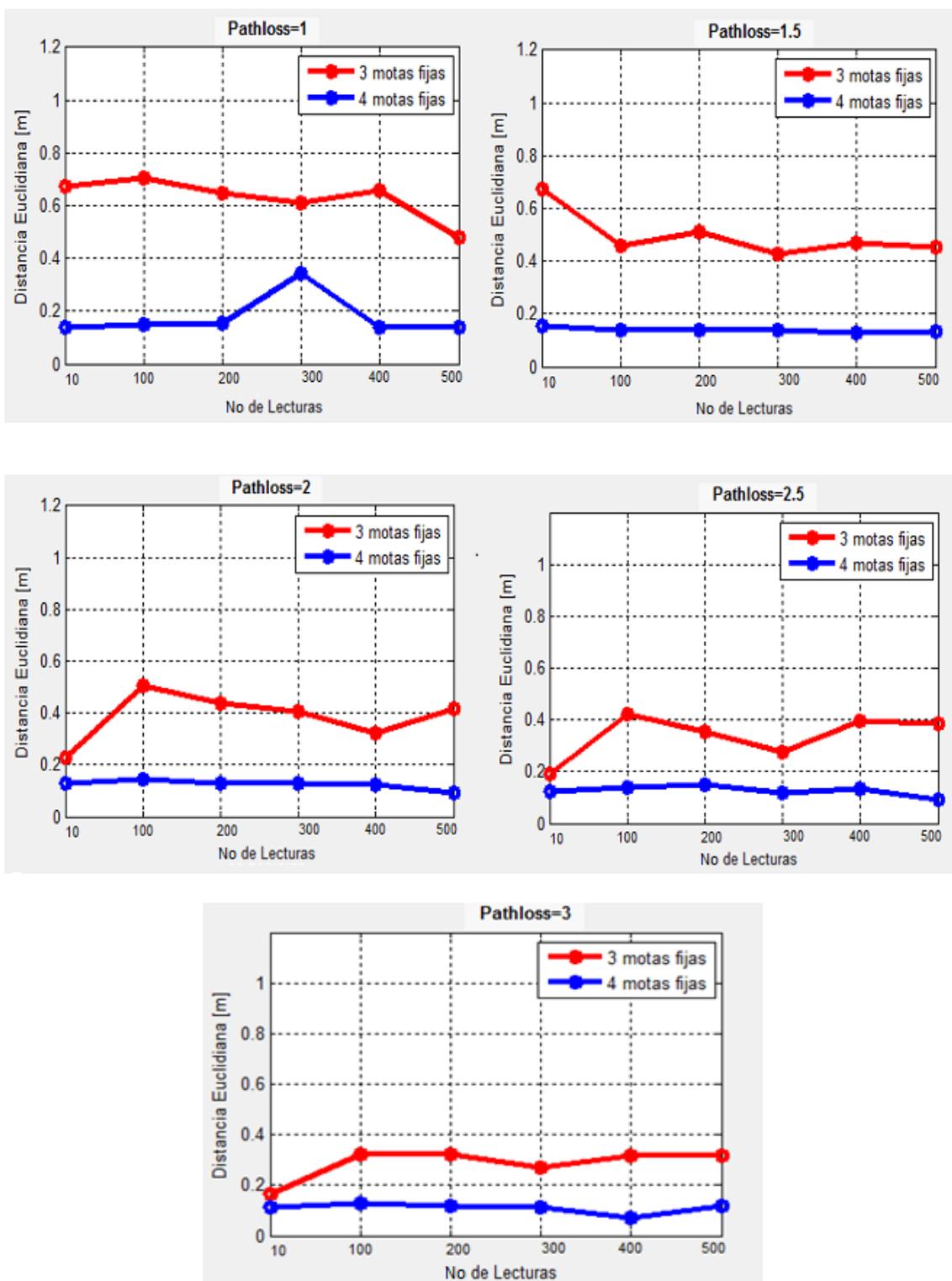


Figura 42 Error de Posición al utilizar 3 y 4 motas (Escenario 1-3er Cuadrante)

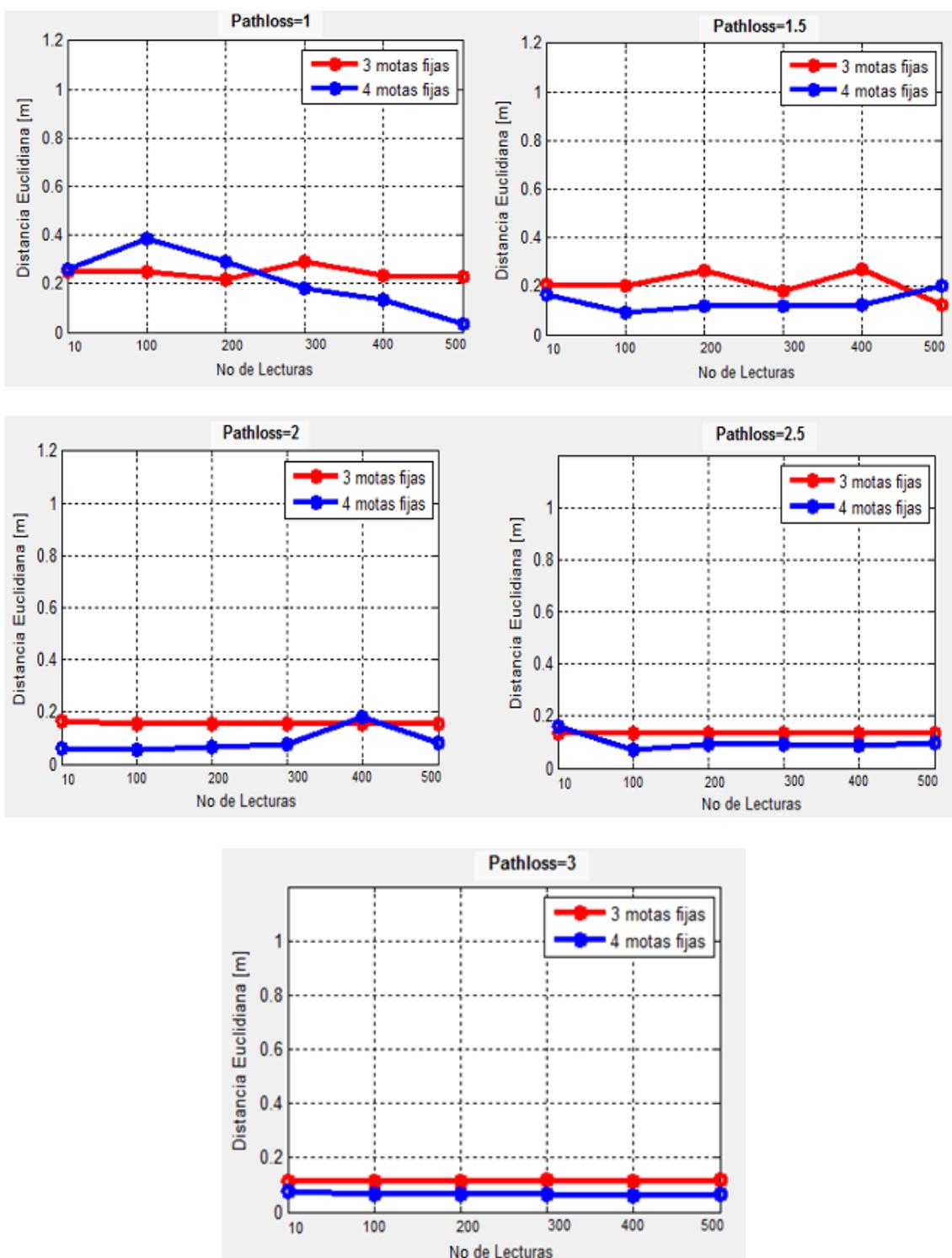


Figura 43 Error de Posición al utilizar 3 y 4 motas (Escenario 2-3er Cuadrante)

#### 5.4. Análisis Cuarto Cuadrante

Como se observa en las Tablas 24 y 25, en el escenario 1 el valor máximo de error es de 0.63 m, que corresponde a la medición de 10 lecturas y en el escenario 2 es de 0.40 m que corresponde a la medición de 200 lecturas, ambos escenarios se realizaron con una constante  $n$  igual a 1. El valor mínimo de error en el escenario 1 es de 0.05 m y del escenario 2 es de 0.01 m que corresponden a las mediciones de 500 y 300 lecturas respectivamente, para ambos escenarios la constante  $n$  es igual a 2.5. Al igual que en otros cuadrantes la condición de mínimo error se realizan menos lecturas que en la condición de máximo error.

**Tabla 24**

**Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 1- 4to Cuadrante)**

No motas fijas	No Lecturas 10		No Lecturas 100		No Lecturas 200		No Lecturas 300		No Lecturas 400		No Lecturas 500	
	3	4	3	4	3	4	3	4	3	4	3	4
n=1	0.39	0.63	0.62	0.53	0.48	0.34	0.37	0.34	0.42	0.42	0.44	0.36
n=1.5	0.57	0.48	0.28	0.18	0.21	0.21	0.24	0.23	0.24	0.27	0.27	0.11
n=2	0.41	0.11	0.24	0.14	0.18	0.11	0.22	0.14	0.21	0.17	0.26	0.14
n=2.5	0.18	0.47	0.12	0.13	0.13	0.14	0.12	0.11	0.10	0.11	0.26	0.05
n=3	0.30	0.34	0.15	0.12	0.09	0.10	0.09	0.10	0.11	0.09	0.23	0.09

Tabla 25

**Distancia Euclidiana entre Posición Real y Estimada [m] (Escenario 2- 4to Cuadrante)**

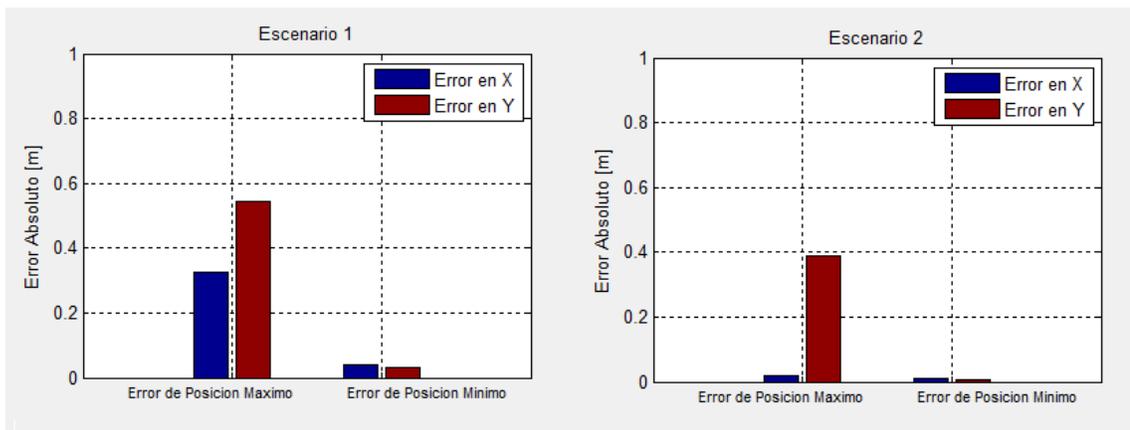
No motas fijas	No Lecturas 10		No Lecturas 100		No Lecturas 200		No Lecturas 300		No Lecturas 400		No Lecturas 500	
	3	4	3	4	3	4	3	4	3	4	3	4
n=1	0.31	0.23	0.31	0.07	0.40	0.04	0.34	0.34	0.34	0.06	0.30	0.04
n=1.5	0.30	0.07	0.25	0.08	0.23	0.05	0.22	0.04	0.29	0.05	0.26	0.14
n=2	0.27	0.11	0.37	0.05	0.23	0.05	0.26	0.04	0.25	0.03	0.34	0.09
n=2.5	0.21	0.24	0.24	0.02	0.25	0.02	0.28	0.01	0.27	0.02	0.34	0.13
n=3	0.17	0.17	0.15	0.05	0.21	0.04	0.21	0.04	0.22	0.12	0.28	0.04

En la Figura 45, se muestra que en el escenario1, el error en X y en Y no varía mucho con respecto al 3er cuadrante en la condición de máximo error, sin embargo en el escenario 2, el error de X disminuyó considerablemente pasando de 0.4138 m a 0.0183 m. En la condición de mínimo error los errores permanecen menores a 0.10 m. Como se puede observar en la Tabla 26, los valores de X difieren en los dos escenarios, esto se debe a que las mediciones se realizaron en diferentes días y hubo cambios en las condiciones climáticas.

Tabla 26

**Error de Posición Mínimo y Máximo (4to Cuadrante)**

Escenario	Error Posición	No Lecturas	X [dBm]	n	Posición Estimada Mota Desconocida		Error Absoluto	
					X[m]	Y[m]	X[m]	Y[m]
Escenario 1	Máximo	10	-62	1	0.524	-0.843	0.324	0.543
	Mínimo	500	-63	2.5	0.240	-0.333	0.040	0.033
Escenario 2	Máximo	200	-62	1	0.218	0.087	0.018	0.387
	Mínimo	300	-65	2.5	0.212	-0.293	0.012	0.006



**Figura 44 Error Eje X e Y (4to Cuadrante)**

Como se observa en la Tablas 27 y 28, en la mayor parte de las mediciones hacia las motas fijas tuvieron varianza igual a cero, esto quiere decir que los valores de RSSI no presentaron oscilaciones en función del tiempo. Además en el escenario 1 en la condición de máximo error se verifica que la distancia hacia la mota fija 3, supera el rango de eje de coordenadas, por lo que también se les asignó el valor de 1.45m para disminuir el error en la posición. En ambos escenarios se muestra que los valores de RSSI no difieren mucho de la condición de máximo y mínimo error por lo que la exactitud en la posición al igual que en otros cuadrantes va a depender de la correcta estimación del parámetro  $n$ .

Tabla 27

**Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 1-4to Cuadrante)**

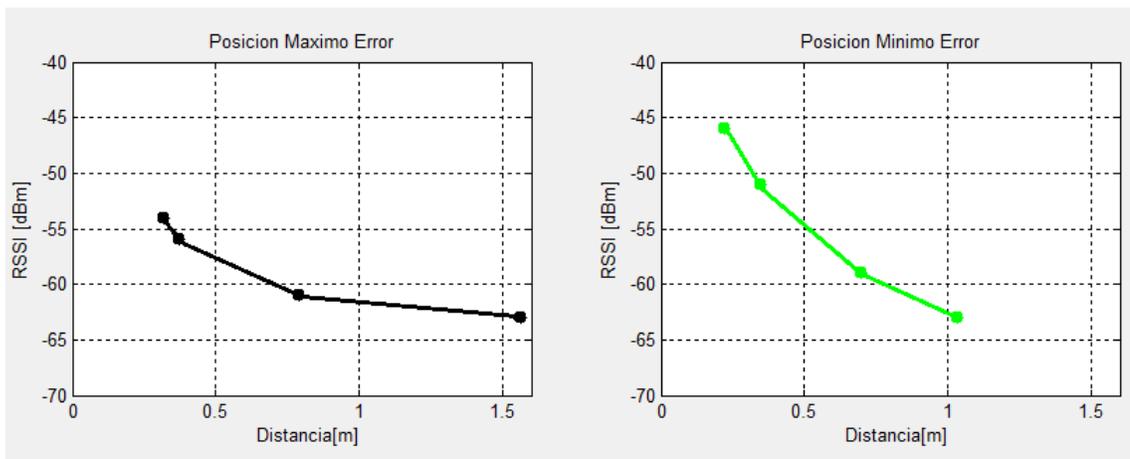
Error de Posición	Distancia [m]		RSSI [dBm]		Varianza		Cantidad de Muestras	
	Max	Min	Max	Min	Max	Min	Max	Min
mota fija 1	0.25	0.35	-56	-51	0	0	19	999
mota fija 2	0.79	0.67	-61	-59	0	0	20	968
mota fija 3	1.58	1.04	-63	-63	0	0	22	1000
mota fija 4	0.16	0.22	-54	-46	0	0	22	1118

Tabla 28

**Distancia, RSSI, Varianza y Cantidad de Muestras (Escenario 2 -4to Cuadrante)**

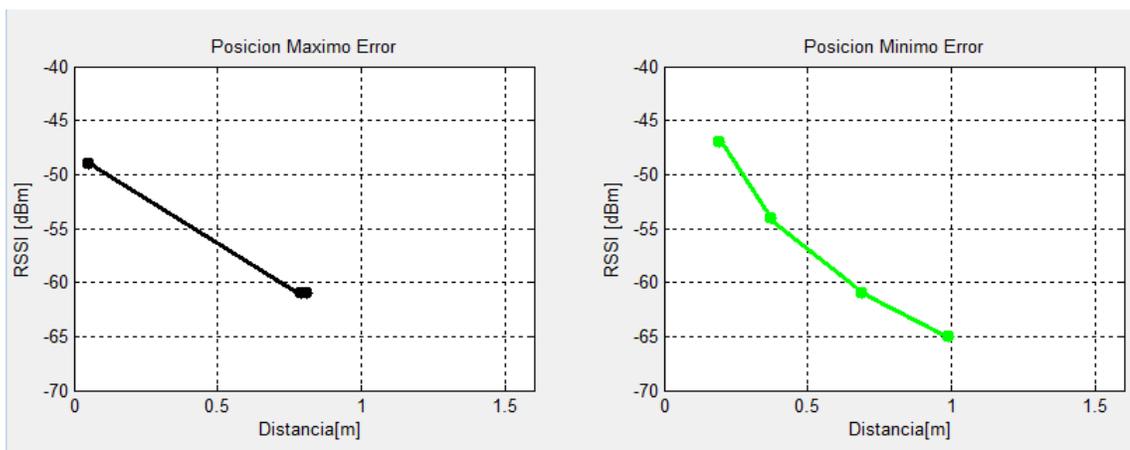
Error de Posición	Distancia [m]		RSSI [dBm]		Varianza		Cantidad de Muestras	
	Max	Min	Max	Min	Max	Min	Max	Min
mota fija 1	0.05	0.37	-49	-54	0	0	399	599
mota fija 2	0.81	0.69	-61	-61	0.019	0	408	574
mota fija 3	0.79	0.99	-61	-65	0	0	406	590
mota fija 4		0.19		-47		0		610

En la Figura 46, se puede observar que en la condición de máximo error la distancia alcanza casi 1.6 m a -63 dBm, excediendo el eje de coordenadas implementado, pero el error se ajusta cuando se aplica la condición de mínimo error alcanzando una distancia de 1.0356 m a -63 dBm hacia la mota fija 3. Al igual que en los otros cuadrantes se comprueba que la distancia es inversamente proporcional al RSSI.



**Figura 45 Distancia[m]-RSSI [dBm] (Escenario 1-4to Cuadrante)**

Como se observa en la Figura 47, en el escenario 1 se puede observar que la curva está más inclinada en la condición de mínimo error porque alcanza valores más grandes de RSSI. La precisión en la posición va a depender si los valores de RSSI medidos, se ajustan a los valores reales para poder estimar las distancias con menor error.



**Figura 46 Distancia[m]-RSSI [dBm] (Escenario 2-4to Cuadrante)**

En la Figura 48, se muestra que no existe mucha diferencia entre los valores de error utilizando 3 y 4 motas fijas como en otros cuadrantes. Para  $n$  igual a 1 la curva se estabiliza en 200 lecturas, para  $n$  igual a 1.5, 2, 2.5 y 3 a

nivel de 100 lecturas. Se muestra que con 3 motas fijas el error tiende a aumentar en 500 lecturas. En el escenario 2, la curva de error es constante en función de la cantidad de lecturas tomadas con la presencia de algunos picos.

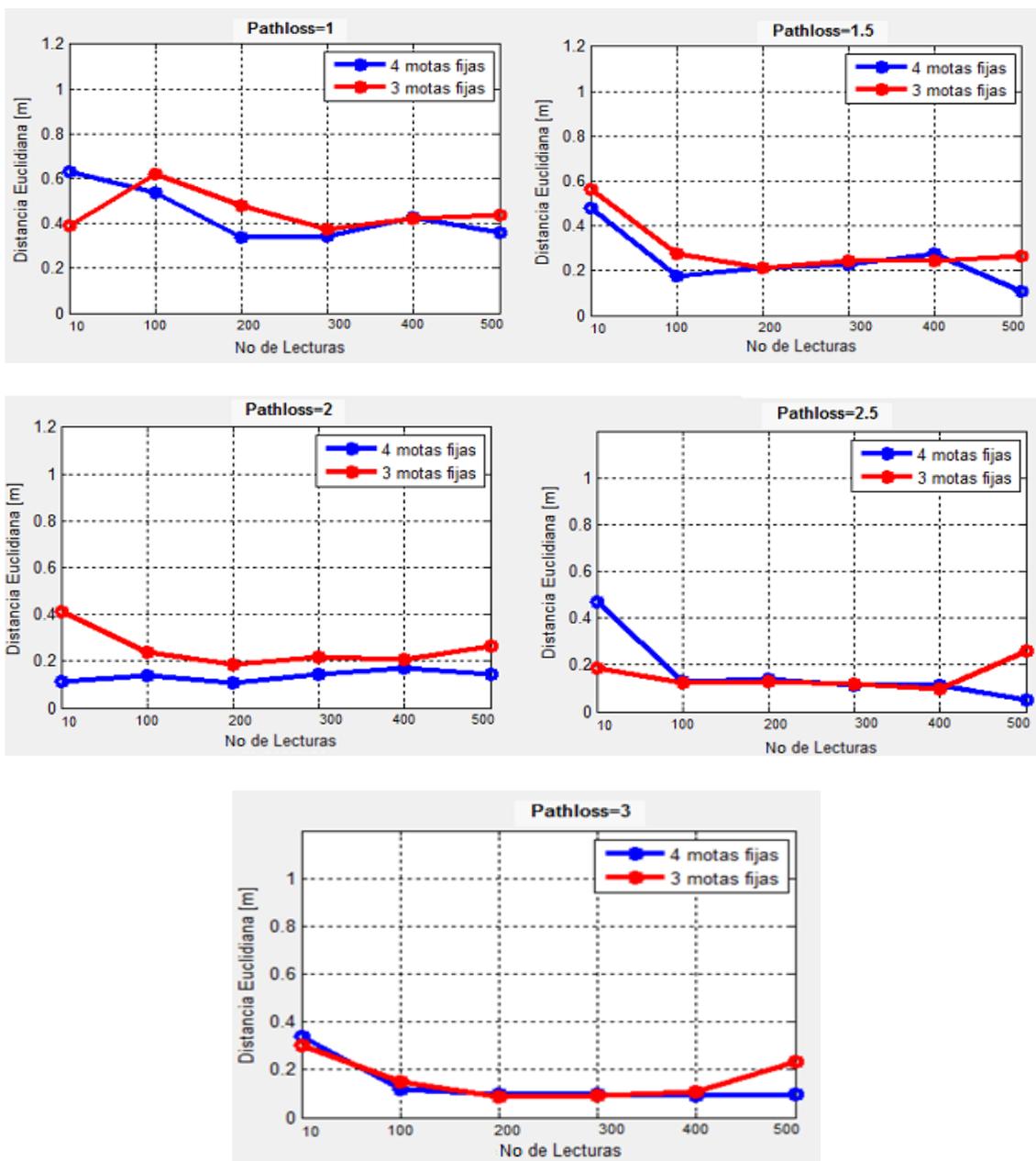


Figura 47 Error de Posición al utilizar 3 y 4 motas (Escenario 1-4to Cuadrante)

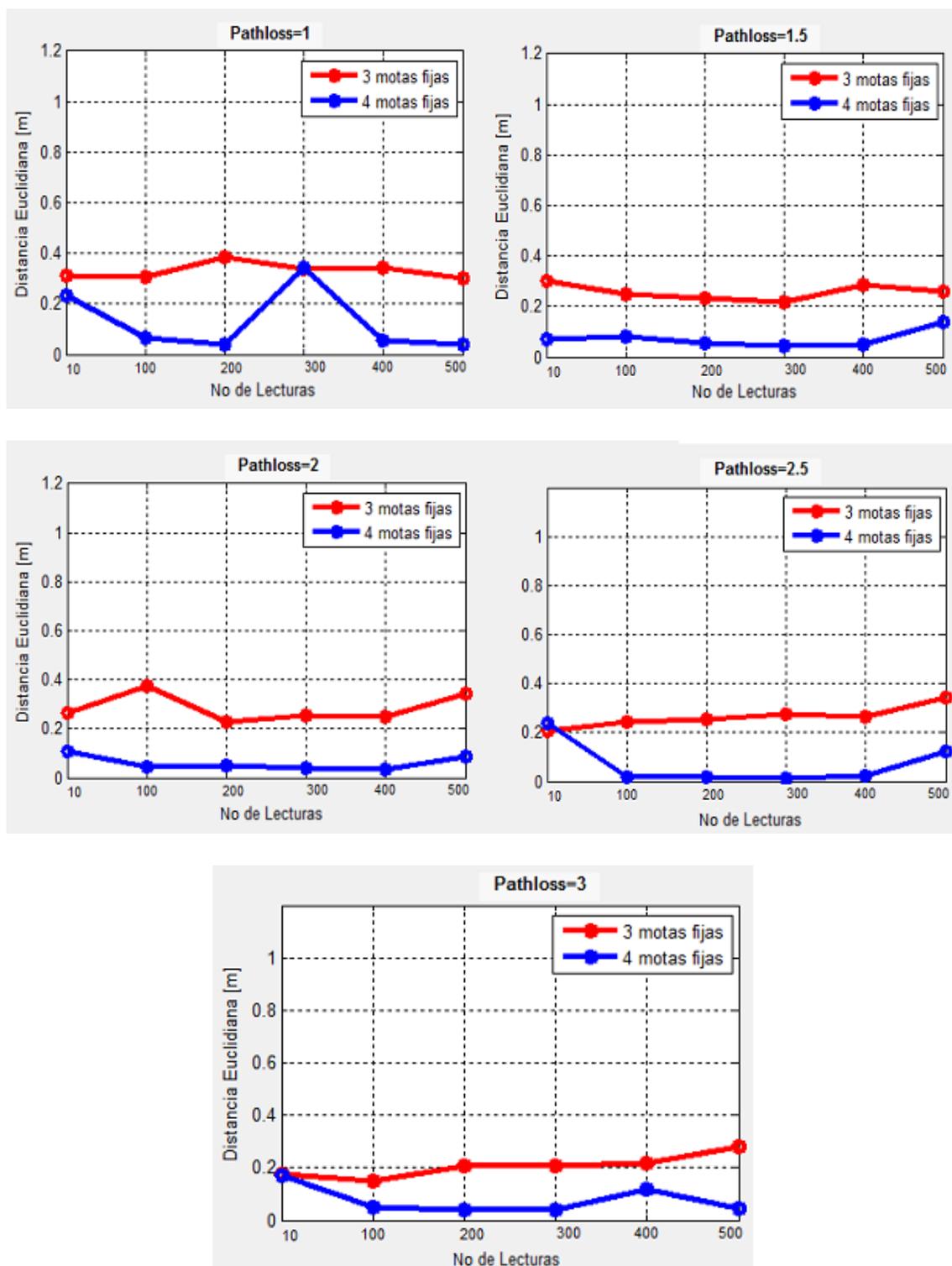


Figura 48 Error de Posición al utilizar 3 y 4 motas (Escenario 2 -4to Cuadrante)

Como se puede comprobar tras las diversas pruebas realizadas (ver Tabla 29-32), el prototipo de sistema de *radiolocalización* es capaz de calcular la posición con cierta precisión en sus cálculos. Aunque los resultados obtenidos están lejos de sistemas como los que emplean ultrasonidos, si se acercan a las prestaciones dadas por los principales sistemas de *radiolocalización* como por ejemplo el GPS el cual utiliza triangulación como algoritmo de localización.

La realización de experimentos con diferentes valores de n, cantidad de lecturas y pruebas utilizando más motas fijas de las requeridas, ha sido la clave para determinar cuál es la mejor condición de error frente al entorno de trabajo. Si no hay condiciones que alteren el comportamiento de la señal del sistema el error cometido en el escenario 1 varía entre 0.05 a 1.17m y en el escenario de 0.01 a 0.64 m.

**Tabla 29**  
**Condición máximo error (Escenario 1)**

Escenario 1	Error [m]	No lecturas	n	Cantidad de Motas Fijas
Primer cuadrante	0.79	10	1	4
Segundo cuadrante	1.17	10	1	3
Tercer cuadrante	0.70	100	1	3
Cuarto cuadrante	0.63	200	1	4

**Tabla 30**  
**Condición máximo error (Escenario 2)**

Escenario 2	Error [m]	No lecturas	n	Cantidad de Motas Fijas
Primer cuadrante	0.53	10	1	3
Segundo cuadrante	0.64	10	1	3
Tercer cuadrante	0.39	100	1	4
Cuarto cuadrante	0.4	10	1	3

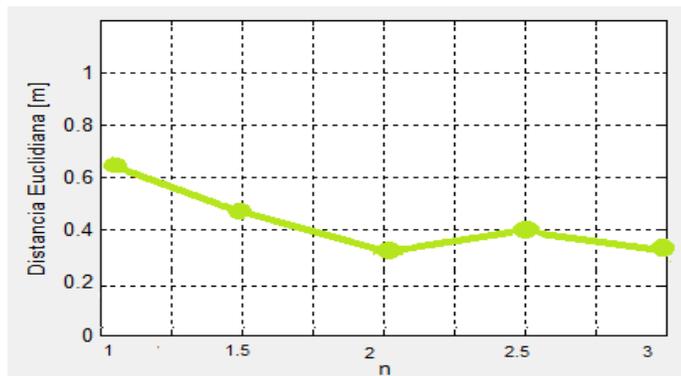
**Tabla 31**  
**Condición mínimo error (Escenario 1)**

Escenario 1	Error [m]	No lecturas	N	Cantidad de Motas Fijas
Primer cuadrante	0.21	100	3	4
Segundo cuadrante	0.06	200	2.5	4
Tercer cuadrante	0.07	400	3	4
Cuarto cuadrante	0.05	500	2.5	4

**Tabla 32**  
**Condición mínimo error (Escenario 2)**

Escenario 2	Error [m]	No lecturas	N	Cantidad de Motas Fijas
Primer cuadrante	0.04	200	3	4
Segundo cuadrante	0.02	300	3	4
Tercer cuadrante	0.03	500	1	4
Cuarto cuadrante	0.01	300	2.5	4

En los errores altos de localización, se utilizó para la estimación de las distancias un n igual a 1 y se tomaron entre 10 y 100 lecturas y para los errores mínimos se utilizó un n entre 1-3 y se tomaron entre 100 y 500 lecturas. Para evitar un elevado porcentaje de error en los cálculos es necesario utilizar el valor de n que mejor se adapte al entorno de comunicación, en este caso deberá ser mayor que 2 porque la fuerza de la señal recibida presentan valores muy dispares debido a la degradación de la señal, estos errores vienen marcados por los objetos que puedan actuar de obstáculos en la trayectoria de los paquetes, los rebotes de las señales durante el transcurso del envío hasta el receptor, o incluso la propia congestión del canal puede alterar los cálculos debido a alteraciones sobre la señal original. En los escenarios las paredes se sitúan lejos, por lo que los multitrayectos se deberán mayormente a los agentes externos como las personas y cualquier otro obstáculo que se encuentre alrededor.

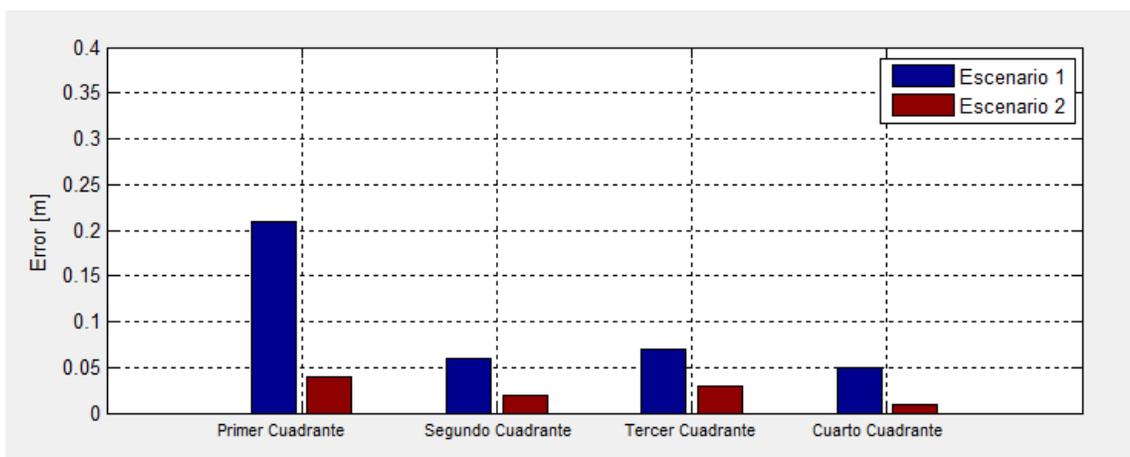


**Figura 49 Error de Posición en función de n**

En la Figura 50 se muestra un ejemplo del error producido al variar los valores de n. Se puede observar que la curva tiene una tendencia decreciente, al aumentar el valor de n disminuye el error de localización. Al analizar los datos obtenidos es evidente que al aumentar la cantidad de lecturas realizadas desde 100 a 500 aumenta la precisión de la posición, sin embargo la curva de error en los diferentes escenarios tuvo una tendencia decreciente al aumentar el número de muestras, pero a nivel de algún punto se vuelve constante. La desventaja que se presenta al tomar varias muestras es que existe una mayor congestión en la red inalámbrica y aumenta la espera en la respuesta del sistema, por lo que se debería buscar un punto intermedio en donde exista un error aceptable, utilizando el menor número de muestras posible.

En el algoritmo de trilateración utilizado, la cantidad de motas fijas mínimas requeridas para realizar los cálculos es de 3, sin embargo en los resultados obtenidos las mejores condiciones de error se presentaron utilizando 4 motas fijas. Teniendo en cuenta que las estimaciones de distancias pueden estar sujetas a imprecisión debido al ruido, interferencias o multicamino, la precisión del posicionamiento puede ser mejorada incorporando un gran número de motas fijas, en vez del mínimo requerido para una estimación de la posición.

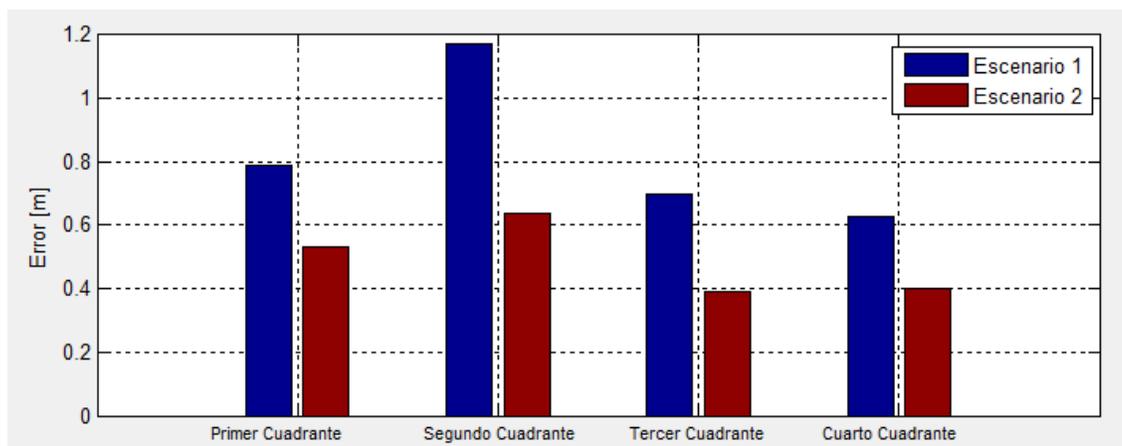
En las Figuras 51 y 52 se muestra la comparación de los 2 escenarios de *radiolocalización* implementados (el escenario 1 a 0.75 m de altura y el escenario 2 en el piso), para esto se tomaron los datos de la mejor y peor condición de error (mínimo y máximo error obtenido) de cada escenario en los diferentes cuadrantes. En la mejor condición de error en el primer cuadrante el error varía entre 0.04 hasta 0.21 m, en el segundo cuadrante de 0.02 a 0.06m, en el tercer cuadrante de 0.03 a 0.07m y en el cuarto cuadrante de 0.01 a 0.05. En la Figura 51 se observa que el menor error obtenido se presentó en el cuarto cuadrante.



**Figura 50 Condición Mínimo Error de Posición**

Como se observa en la Figura 52, en la peor condición de error, en el primer cuadrante el error varía entre 0.53 hasta 0.79 m, en el segundo cuadrante de 0.64 a 1.17m, en el tercer cuadrante de 0.39 a 0.70 m y en el cuarto cuadrante de 0.40 a 0.63. Se puede observar que los mejores resultados se presentan en el escenario 2 en todos los cuadrantes. En la práctica, se presentaban errores que alcanzaban 1.17 m cuando la red de comunicaciones se situaba a 0.75 m de altura, esto puede deberse a que el plano en el que se encontraban las motas fijas y la desconocida estaba muy por encima del plano de la mota base provocando que los planos de máxima radiación de las motas no coincidiesen, esta situación provocó que en la mayoría de los casos la señal

se viera afectada por el multitrayecto. Al repetir las pruebas situando la red a nivel del suelo la exactitud y la precisión en la ubicación de la mota desconocida se incrementó.



**Figura 51 Condición Máximo Error de Posición**

## CAPÍTULO 6

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1. Conclusiones

- A partir del estudio del Marco Teórico y conceptual de las WSN se determinó que las características de las motas utilizadas en el proyecto como son su bajo consumo energético, tamaño pequeño, escalabilidad, versatilidad en su funcionamiento, facilidad de instalación en diferentes puntos de un escenario cualquiera, tanto interior como exterior, entre otros hacen que las WSN sean adecuados para aplicaciones de *radiolocalización*.
- En base a la necesidad presentada de almacenar, procesar y visualizar los datos obtenidos del prototipo de sistema de *radiolocalización* en tiempo real, se optó por utilizar GUI de MATLAB al ser una herramienta de software matemática que permite el fácil desarrollo de conceptos de *radiolocalización* y que permite al usuario mediante la interfaz gráfica predecir de manera eficiente el comportamiento de las motas en entornos reales.
- Además en este proyecto se realizó el estudio del lenguaje de programación NesC para poder configurar las motas en función de las necesidades de localización. El diseño en NesC permite la incorporación de nuevas aplicaciones rápidamente y hace que funcione bajo las restricciones de memoria que se dan en este tipo de redes.
- Se desarrolló un algoritmo de *radiolocalización* basado en la técnica de trilateración utilizando el método LS (Least Squares), esto se logró a

través del análisis experimental y teórico con las WSN, se concluye que cada vez que una mota pueda obtener el RSSI del payload del paquete y transmitir su ID se pueden estimar las distancias y obtener la ubicación de una mota que se desconoce su posición. Se deben utilizar mínimo tres motas que servirán como eje de referencia (motas fijas) para que se pueda implementar la técnica de trilateración, sin embargo se comprobó que entre más motas de referencia se utilicen mejor precisión de la ubicación se obtendrá.

- El algoritmo de *radiolocalización* seleccionado utiliza el indicador de la fuerza de la señal recibida (RSSI) para calcular la posición, el problema surge de la relación distancia/RSSI, la presencia de reflexiones, difracciones y multitrayecto, alteran las estimaciones de las distancias, por lo que para disminuir el error de localización se trabajó con valores promedio de RSSI. También la precisión va a depender de la correcta estimación de los parámetros que modelan el entorno de comunicaciones que son la constante de pathloss ( $n$ ) y el RSSI medido a un metro de distancia.
- Se comparó los resultados obtenidos en los 2 escenarios de trabajo y se determinó que estos son adecuados para la implementación del algoritmo de *radiolocalización*, sin embargo el escenario 1 (que se encuentra a nivel de piso) presentó mejores resultados que el escenario 2 (que se encuentra a 0.75 m de altura), esto es debido a que en el escenario 2 las motas fijas y la desconocida no estuvieron a la misma altura que la mota base lo que ocasionó que se presenten inconsistencias en los datos obtenidos.

## 6.2. Recomendaciones

- Para el correcto funcionamiento del sistema se deben evitar escenarios que presenten algún tipo de obstáculo o dispositivos que trabajen a la misma frecuencia que las motas (2.4 GHz), para evitar reflexiones, multitrayectos y otros tipos de interferencia que puedan afectar a la señal.
- Se recomienda que las motas fijas se distribuyan en el área en forma de mallas o en los 4 cuadrantes para optimizar la recepción de los RSSI en diferentes entornos de trabajo.
- Cuando el sistema se encuentre en funcionamiento se recomienda que las pilas se encuentren cargadas para evitar que la señal se degrade y afecte al cálculo de posicionamiento.
- Se debe tener cuidado al montar las motas en el escenario, teniendo precaución en no dañar el conector de alimentación ni la antena ya que son muy frágiles.
- Los valores de RSSI varían en función de la orientación de las antenas de las motas, por lo que se recomienda que en la transmisión y recepción de los paquetes éstas se encuentren rectas (polarizadas vertical u horizontalmente) en todo momento.
- Se recomienda evitar el uso de máquinas virtuales para la instalación del software del proyecto, ya que podría interferir con el funcionamiento de la PC y que ocurran retardos en la adquisición y procesamiento de la información proveniente de la red de sensores inalámbricos.

### 6.3. Descripción de Utilidad del Proyecto para Trabajos Futuros

- Se podría mejorar a la técnica de *radiolocalización* empleada en este proyecto, para que el prototipo de sistema sea más preciso y robusto cuando se presenten obstáculos que degraden a la señal recibida. Además se podría evaluar el comportamiento del prototipo de sistema en distintos escenarios (interiores o exteriores), aumentando la cantidad de motas fijas y expandiendo el eje de coordenadas utilizado en este proyecto.
- Se podría utilizar otro tipo de software para la aplicación de la interfaz de usuario y para la adquisición-procesamiento de datos proveniente de la red inalámbrica, que permita al igual que Matlab manejar al sistema de *radiolocalización* en un entorno amigable e ilustrativo como por ejemplo Java, C#, labview, entre otros.
- Se podría probar y adecuar al prototipo de sistema desarrollado, utilizando otro tipo de arquitectura de sensores inalámbricos como por ejemplo TelosB, Micaz, Imote2, entre otros, para verificar y analizar las ventajas-desventajas que estos sensores presentan en distintos entornos.
- El prototipo de sistema desarrollado podría servir como base para la implementación de otro algoritmo de *radiolocalización* o se podría complementar con la técnica utilizada en este proyecto (mediante RSSI) que permita igualar o incrementar la precisión de la ubicación de la mota desconocida en distintos escenarios.
- Este proyecto se ha centrado principalmente en el proceso de *radiolocalización* y monitorización en entornos básicos, en un futuro será

posible añadir diversas funcionalidades a la aplicación como por ejemplo la gestión de inventarios, detección de intrusos, monitoreo de tráfico, atención de la salud, vigilancia de incendios de matorrales, monitoreo de la calidad del agua, agricultura de precisión y reconocimiento.

- Se podría utilizar otro tipo de dispositivos de WSN para la implementación del proyecto, que trabajen en la banda de 868 MHz o 915 MHz, para lograr mayor alcance y evaluar el comportamiento del prototipo de sistema a grandes distancias.

## Trabajos Citados

(s.f.).

Adams, J. T. (2006). *An Introduction to IEEE STD 802.15.4*. Obtenido de [https://www.sonoma.edu/users/f/farahman/sonoma/courses/cet543/resources/802\\_intro\\_01655947.pdf](https://www.sonoma.edu/users/f/farahman/sonoma/courses/cet543/resources/802_intro_01655947.pdf)

AKAR, F. y. (2010). *Recent Advances in Circuits, Systems, Signal Processing and Communications*. Bogazici University, Department of Electrical Engineering.

Alberca, Y., & Berrazueta, F. (2007). *Zigbee como una aplicación a la Domótica, y el Estándar IEEE 802.15.4*. Obtenido de [http://clusterfie.epn.edu.ec/ibernal/html/CURSOS/Oct05Marzo06/Inalamblicas/Trabajo1/Paper/Proyecto1\\_g2.pdf](http://clusterfie.epn.edu.ec/ibernal/html/CURSOS/Oct05Marzo06/Inalamblicas/Trabajo1/Paper/Proyecto1_g2.pdf)

ATMEL. (2009). *AT86RF230 Data sheet*. Obtenido de [www.atmel.com/literature](http://www.atmel.com/literature)

Autores, V. (Diciembre de 2004). ZigBee Specifications. *ZigBee Alliance*.

Crossbow Technology. (s.f.). *IRIS\_Datasheet*. Obtenido de IRIS\_Datasheet: [http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS\\_Datasheet.pdf](http://www.memsic.com/userfiles/files/Datasheets/WSN/IRIS_Datasheet.pdf)

Daboín, A., & Verde, G. (2012). *Medición RSSI, LQI y pruebas de cobertura para diferentes escenarios de propagación en una red inalámbrica de sensores*. Obtenido de Revista Digital de Investigación y Postgrado de la Universidad Nacional Experimental Politécnica “Antonio José de Sucre”, Vicerrectorado Barquisimeto. Venezuela: <http://redip.bqto.unexpo.edu.ve>.

Dargie, W., & Poellabauer, C. (2010). *Fundamentals of Wireless Sensor Networks: Theory and Practice*. John Wiley & Sons Ltd.

Díaz, L., & Tabernilla, A. (2010). *Localización en interiores con redes de sensores mediante técnicas de fingerprinting*. Madrid, España: Universidad Politécnica de Madrid.

Fernandez, N. (Marzo de 2006). *Modelo de cobertura en redes inalámbricas*. Obtenido de <http://di002.edv.uniovi.es/~cueva/investigacion/tesis/Nestor.pdf>

- Fidan, B., & Mao, G. (2006). *Location Algorithms and Strategies for Wireless Sensor Networks. 1ra Ed.* New York-USA: Information Science Reference.
- Figuera, C. (2009). Algoritmos de aprendizaje estadístico aplicados a la radiolocalización en interiores. *Tesis Dr. Ciencias.* Fuenlabrada, España.
- Freescale. (2007). IEEE 802.15.4 y Zigbee. *SILICA.*
- García, A. (2009). Localización de nodos dentro de una red inalámbrica de sensores. Barcelona, España.
- García, J. (2008). Desarrollo y Estudio de Capacidades de Redes 802.15.4 Usando Dispositivos Micaz . Cartagena, España: Universidad Politécnica de Cartagena.
- Gay, D. (2005). *nesC 1.2 Language Reference Manual - TinyOS.*
- GMBH. (s.f.). White Paper Real Time Locations Systems (RTLS). *Nanotron, Technologies.*
- González, A. G. (2009). *Comunicaciones Industriales Avanzadas.* Obtenido de Universidad Pontificia Comillas - ICAI:  
[http://www.dea.icai.upco.es/sadot/Comunicaciones/avanzadas/Alberto\\_Gasc%C3%B3n\\_Zigbee%20y%20el%20Est%C3%A1ndar%20IEEE%20802.15.4.pdf](http://www.dea.icai.upco.es/sadot/Comunicaciones/avanzadas/Alberto_Gasc%C3%B3n_Zigbee%20y%20el%20Est%C3%A1ndar%20IEEE%20802.15.4.pdf)
- Grilo, A. (2009). *Wireless Sensor Networks: Localization and Positioning.* Portugal: Carnegie Mellon.
- Guidas, L. (2005). *Wireless Sensor Networks. 1ra Ed.* San Francisco-USA: ELSEVIER.
- Jiménez, J. (2007). Localización en interiores con redes de sensores mediante técnicas de fingerprinting. Murcia, España: Universidad de Murcia Facultad de Informática.
- Laaraiedh, M. (2009). *Enhancing Positioning Accuracy Through RSS Based.* HAL Archives.
- Levis, P. (2012). *message\_t.* Documentary, Core Working Group.
- Maroto, S. (2010). Desarrollo de Aplicaciones Basadas en WSN. Valencia, España: Universidad Politécnica de Valencia.

- MEMSIC. (2006). *MPR-MIB Users Manual*.
- Moreno, C. (2011). *Localización en Redes Inalámbricas de Sensores*. Obtenido de [http://mcyti.izt.uam.mx/archivos/Tesis/Generacion2008/ICR\\_CarlosMorenoEscobar.pdf](http://mcyti.izt.uam.mx/archivos/Tesis/Generacion2008/ICR_CarlosMorenoEscobar.pdf)
- Oguejiofor O.S, A. A. (2013). *Trilateration based locatitation algorithm*. International Journal of Science and Modern Engineering (IJISME).
- Poellabauer, W. D. (2010). *Fundamentals of Wireless Sensor Networks: Theory and Practice*. John Wiley & Sons Ltd.
- Polly Huang, & Seng-Yong Lau. (2009). *Wireless Sensor Network and Laboratories*. Obtenido de Department of Electrical Engineering National Taiwan University : <http://cc.ee.ntu.edu.tw/~phuang>  
phuang@cc.ee.ntu.edu.tw
- Polo, E. M. (2013). *Técnicas de Localización en Redes Inalámbricas de Sensores*. Instituto de Investigación en Informática de Albacete, Departamento de Sistemas Informáticos, Córdoba.
- Pozuelo, C. F. (2009). *Algoritmos de aprendizaje estadístico aplicados*. UNIVERSIDAD REY JUAN CARLOS, TEORIA DE LA SEÑAL Y COMUNICACIONES.
- Rugeles, J. (2010). *Técnicas de localización de nodos inalámbricos mediante redes de sensores*. Bogotá, Colombia: Universidad Militar Nueva Granada.
- Santana, J. (2010). *Localización en interiores con redes de sensores mediante técnicas de fingerprinting*. Murcia, España: Universidad de Murcia.
- Suárez, J. U. (2009). *ESTUDIO COMPARATIVO DE SISTEMAS OPERATIVOS DE TIEMPO REAL APLICADO A REDES DE SENSORES INALAMBRICAS*. Sociedad Agrícola Ojos Buenos Ltda.
- Tinyos-2.x. (2006). *This lesson has introduced radio communications TinyOS 2.x*. Obtenido de <http://www.tinyos.net/dist-2.0.0/tinyos-2.0.0beta1/doc/html/tutorial/lesson5.html>
- Tinyos-2.x. (2008). *RSSI DEMO y BLINK TO RADIO*. Obtenido de <http://www.tinyos.net/tinyos-2.x/apps/tutorials/>

Tseng, Y. (2007). *Wireless Ad hoc Networking. 1ra Ed.* New York-USA: Auerbach Publications.

Vidri, I. (2012). *ZigBee y sus aplicaciones - Universidad Pontificia Comillas.*

Obtenido de

<http://www.dea.icaei.upco.es/sadot/Comunicaciones/avanzadas/Zigbee%20y%20sus%20aplicaciones.pdf>