



---

---

---

## **UNIVERSIDAD NACIONAL DE LOJA.**

**ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS  
RECURSOS NATURALES NO RENOVABLES.**

### **CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES.**

#### **TÍTULO:**

**“DESARROLLO DE UN PROTOTIPO DE RED  
INALÁMBRICA DE SENSORES BAJO EL ESTANDAR  
IEEE 802.15.4 (ZIGBEE) PARA EL MONITOREO DE  
VARIABLES AMBIENTALES EN EL JARDÍN  
BOTÁNICO REINALDO ESPINOSA DE LA  
UNIVERSIDAD NACIONAL DE LOJA.”**

TESIS DE GRADO PREVIO A LA  
OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y  
TELECOMUNICACIONES.

**DIRECTOR:** Ing. Juan Gabriel Ochoa Aldeán, Mg. Sc.

**AUTORA:** María Elena Ochoa Azuero.

**LOJA-ECUADOR**

**2016**

## **CERTIFICACIÓN.**

Señor Ingeniero.

Ing. Juan Gabriel Ochoa Aldeán, Mg. Sc.

**DIRECTOR DEL TRABAJO DE TESIS**

### **CERTIFICA:**

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación, cuyo tema versa; **“DISEÑO DE UN PROTOTIPO DE RED INALÁMBRICA DE SENSORES BAJO EL ESTANDAR IEEE 802.15.4 (ZIGBEE) PARA EL MONITOREO DE VARIABLES AMBIENTALES EN EL JARDIN BOTÁNICO REINALDO ESPINOSA DE LA UNIVERSIDAD NACIONAL DE LOJA.”** previo a la obtención del título de **Ingeniero en Electrónica y Telecomunicaciones**, realizado por la señorita egresada: Maria Elena Ochoa Azuero, la misma que cumple con el reglamento y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

**Loja, marzo de 2016**



Ing. Juan Gabriel Ochoa Aldeán, Mg. Sc.

**DIRECTOR DEL TRABAJO DE TESIS**

## **AUTORÍA.**

Yo **MARÍA ELENA OCHOA AZUERO**, declaro ser autora del presente trabajo de tesis y eximo expresamente a la Universidad de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi trabajo de tesis en el Repositorio Institucional- Biblioteca Virtual.

**Firma:**

A handwritten signature in blue ink, appearing to read 'María Elena Ochoa Azuero', with a long horizontal flourish extending to the right.

**Cedula:** 1105681769

**Fecha:** 16/03/2016

**CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LA AUTORA, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.**

**Yo MARIA ELENA OCHOA AZUERO**, declaro ser autora de la tesis titulada **“DISEÑO DE UN PROTOTIPO DE RED INALÁMBRICA DE SENSORES BAJO EL ESTANDAR IEEE 802.15.4 (ZIGBEE) PARA EL MONITOREO DE VARIABLES AMBIENTALES EN EL JARDIN BOTÁNICO REINALDO ESPINOSA DE LA UNIVERSIDAD NACIONAL DE LOJA”**, como requisito para optar al grado de: **ING. EN ELECTRÓNICA Y TELECOMUNICACIONES**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, dieciséis días del mes de marzo del dos mil dieciséis

**Firma:**



**Autor:** Maria Elena Ochoa Azuero.

**Cédula:** 1105681769

**Dirección:** Ciudadela Ciudad Victoria

**Correo Electrónico:** mary.elena93@hotmail.es

**Teléfono:** 232-62-57 **Celular:** 0968838511

**DATOS COMPLEMENTARIOS**

**Director de Tesis:** Ing. Juan Gabriel Ochoa Aldeán, Mg. Sc.

**Tribunal de Grado:** Ing. Andy Fabricio Vega León, Mg. Sc

Ing. Paulo Alberto Samaniego Rojas, Mg. Sc.

Ing. John Jossimar Tucker Yépez, Mg. Sc.

## **DEDICATORIA.**

Dedico el presente trabajo a Dios por la vida, la inteligencia y por guiarme en el transcurso de la carrera.

A mis padres por la ayuda y fortaleza incondicional que supieron brindarme para ser mejor cada día, y para poder culminar con éxito mi carrera universitaria.

## **AGRADECIMIENTO.**

Yo, Maria Elena Ochoa autor del presente trabajo de tesis dejo constancia de mi agradecimiento a la Universidad Nacional de Loja, institución que ha brindado un gran aporte a mi formación profesional a través del Área de la Energía, las Industrias y los Recursos Naturales no Renovables y la carrera de Ingeniería Electrónica y Telecomunicaciones.

Mis más sinceros agradecimientos al Ing. Mg. Sc. Juan Gabriel Ochoa Aldeán, director de la presente tesis, y a toda la planta docente de la carrera que con sus conocimientos supieron enseñarme e incentivar me, para lograr con éxito el desarrollo del presente trabajo de titulación.

## TABLA DE CONTENIDO.

CERTIFICACIÓN.....	I
AUTORÍA.....	III
CARTA DE AUTORIZACIÓN.....	IV
DEDICATORIA.....	V
AGRADECIMIENTO.....	VI
TABLA DE CONTENIDO.....	VII
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS.....	XVI
1. TÍTULO.....	1
2. RESUMEN.....	2
ABSTRACT.....	3
3. INTRODUCCIÓN.....	4
4. REVISIÓN DE LITERATURA.....	6
4.1. INTRODUCCIÓN A LAS REDES DE SENSORES INALÁMBRICAS.....	6
4.1.1. Redes de Computadoras.....	6
4.1.1.1. Definición.....	6
4.1.2. Redes Inalámbricas.....	6
4.1.2.1. Redes Inalámbricas de área Extendida (WWAN).....	7
4.1.2.2. Redes Inalámbricas de área Metropolitana (WMAN).....	8
4.1.2.3. Redes Inalámbricas de área Local (WLAN).....	8
4.1.2.4. Redes Inalámbricas de área personal (WPAN).....	9
4.2. REDES INALÁMBRICAS DE SENSORES (WSN).....	9
4.2.1. Definición.....	10
4.2.2. Elementos.....	10

4.2.3.	Características.....	12
4.2.4.	Aplicación de las WSN en la agricultura.....	12
4.2.5.	Estándares para WSN. ....	14
4.2.5.1.	Estándar ZigBee. ....	14
4.2.5.2.	Estándar Wifi.....	14
4.2.5.3.	Estándar Bluetooth. ....	15
4.3.	ESTÁNDAR IEEE 802.15.4 (ZIGBEE).....	16
4.3.1.	Definición. ....	16
4.3.2.	Estándar 802.15.4. ....	18
4.3.3.	Características de ZigBee. ....	18
4.3.4.	Tipos de dispositivos ZigBee. ....	19
4.3.5.	Topologías ZigBee. ....	21
4.3.6.	Arquitectura ZigBee. ....	23
4.3.6.1.	Capa Física (PHY).....	24
4.3.6.2.	Capa de Acceso al Medio. ....	25
4.3.6.3.	Capa de Red.....	27
4.3.6.4.	Capa de Soporte a la Aplicación. ....	27
4.3.6.5.	Capa de Aplicación.....	27
4.4.	MÓDULOS X-BEE- ZIGBEE.....	27
4.4.1.	Definición. ....	27
4.4.2.	Características.....	28
4.4.3.	Series. ....	29
4.4.4.	Tipo de Antenas de los módulos XBEE. ....	31
4.4.5.	Diagrama de pines de un módulo XBee. ....	32
4.4.6.	Modos de operación. ....	32
4.4.6.1.	Modo recepción/transmisión. ....	32



4.4.6.2.	Modo de bajo consumo. ....	33
4.4.6.3.	Modo Comando. ....	33
4.4.6.4.	Modo Transparente. ....	33
4.4.6.5.	Modo API. ....	34
5.	MATERIALES Y MÉTODOS. ....	35
5.1.	SIMULACIÓN DE LA RED INALÁMBRICA DE SENSORES. ....	35
5.1.1.	Elección del Simulador. ....	35
5.1.2.	Simulador NS2. ....	39
5.1.2.1.	Generalidades. ....	39
5.1.2.2.	Arquitectura de NS2. ....	40
5.1.2.3.	Instalación NS2. ....	42
5.1.2.4.	Herramientas Complementarias de NS2. ....	48
5.1.3.	Fichero de Simulación. ....	51
5.1.3.1.	Definición de variables Globales. ....	51
5.1.3.2.	Definición de los ficheros de trazas (NS-2 y NAM). ....	53
5.1.3.3.	Definición de las características de la red Inalámbrica. ....	53
5.1.3.4.	Configuración de los Nodos. ....	58
5.1.3.5.	Creación de los nodos. ....	60
5.1.3.6.	Tráfico entre los nodos. ....	60
5.1.3.7.	Finalización del simulador. ....	61
5.1.4.	Archivo de trazas. ....	61
5.1.5.	Simulación de las topologías de Red. ....	64
5.1.5.1.	Variables de Análisis. ....	64
5.1.5.2.	Métricas de evaluación para el análisis de los datos. ....	65
5.2.	DESARROLLO DEL PROTOTIPO DE WSN. ....	67
5.2.1.	Consideraciones para Diseño e Implementación del prototipo de WSN. ....	68

5.2.2.	Identificación de las variables ambientales a monitorear.....	71
5.2.3.	Componentes y/o Dispositivos.....	72
5.2.3.1.	Sensores.....	72
5.2.3.2.	Sensor de Temperatura LM35DZ.....	73
5.2.3.3.	Sensor de Humedad, Sparkfun Soil Moisture Sensor.....	74
5.2.3.4.	Arduino Uno.....	76
5.2.3.5.	Módulo XBee Pro S2B.....	79
5.2.3.6.	Shield XBee Pro.....	82
5.2.3.7.	Baterías.....	83
5.2.3.8.	XBee Explorer USB.....	83
5.2.4.	NODO SENSOR.....	84
5.2.4.1.	Diseño del nodo Sensor.....	84
5.2.4.2.	Implementación del Nodo Sensor.....	87
5.2.5.	NODO COORDINADOR.....	93
5.2.5.1.	Diseño del Nodo Coordinador.....	93
5.2.5.2.	Implementación del Nodo Coordinador.....	94
5.2.6.	Trama de datos de un XBee en Modo API.....	98
5.2.7.	INTERFAZ DE MONITOREO: Aplicación en VS Express 2013 .....	100
5.2.7.1.	Desarrollo de la Interfaz de Monitoreo.....	101
5.2.7.2.	Descripción de la Interfaz de monitoreo.....	104
5.3.	CÁLCULOS.....	107
5.3.1.	Cálculo del RSSI (Nivel de intensidad de la señal Recibida).....	107
5.3.2.	Cálculo de la Potencia de recepción.....	108
5.3.3.	Cálculo de las pérdidas en el enlace inalámbrico.....	108
5.3.4.	Cálculo del nivel de potencia de la señal recibida a 50m.....	110
5.3.5.	Cálculo del nivel de potencia de la señal recibida a 100m.....	110

5.3.6.	Cálculo del nivel de potencia de la señal recibida a 200m.....	111
5.3.7.	Cálculo del nivel de potencia de la señal recibida a 300m.....	112
6.	RESULTADOS.....	113
6.1.	Resultados de la simulación de la Red de sensores realizada en NS2. ....	113
6.1.1.	Análisis de los resultados de la simulación de la WSN.....	117
6.2.	Pruebas y Resultados del prototipo de WSN. ....	118
6.2.1.	Prueba de Recepción de Señal.....	118
6.2.1.1.	Interpretación de resultados de la prueba de recepción de señal. ....	122
6.2.2.	Prueba de envío y recepción de datos entre el nodo sensor y el nodo coordinador.....	123
6.2.2.1.	Interpretación de resultados de la prueba de Tx y Rx de datos. ....	124
6.2.3.	Prueba General del funcionamiento del prototipo de WSN. ....	124
6.2.4.	Costo del prototipo implementado .....	130
7.	DISCUSIÓN.....	132
8.	CONCLUSIONES. ....	136
9.	RECOMENDACIONES.....	138
10.	BIBLIOGRAFÍA.....	140
11.	ANEXOS.....	144

## ÍNDICE DE FIGURAS.

<b>Figura. 1:</b> Clasificación de las redes inalámbricas según su alcance. ....	7
<b>Figura. 2:</b> Red Inalámbrica de Sensores. ....	10
<b>Figura. 3:</b> Elementos de una WSN. ....	11
<b>Figura. 4:</b> Aplicación de las WSN en la Agricultura.....	13
<b>Figura. 5:</b> Tipos de dispositivos ZigBee .....	20
<b>Figura. 6:</b> Tipos de topologías ZigBee.....	22
<b>Figura. 7:</b> Arquitectura ZigBee .....	24
<b>Figura. 8:</b> Tipos de XBee en la banda de 2.4GHz.....	30
<b>Figura. 9:</b> Tipos de antenas XBee .....	31
<b>Figura. 10:</b> Diagrama de pines del módulo XBee. (Vista Superior) .....	32
<b>Figura. 11:</b> Arquitectura de Ns2.....	41
<b>Figura. 12:</b> Página oficial del simulador NS2. ....	42
<b>Figura. 13:</b> Línea Archivo ls.h que debe modificarse .....	43
<b>Figura. 14:</b> Línea Archivo ls.h modificada.....	44
<b>Figura. 15:</b> Ejecución de NS2 sin argumentos .....	46
<b>Figura. 16:</b> Ejecución de NS2 con argumentos .....	46
<b>Figura. 17:</b> Archivo de configuración para la topología en árbol (.tcl).....	47
<b>Figura. 18:</b> Archivo de configuración para la topología en estrella (.tcl). ....	47
<b>Figura. 19:</b> Herramienta NAM. ....	49
<b>Figura. 20:</b> Interfaz gráfica de la herramienta NAM.....	50
<b>Figura. 22:</b> Escenario de simulación de la topología en estrella. ....	57
<b>Figura. 21:</b> Escenario de simulación de la topología en árbol. ....	58
<b>Figura. 23:</b> Archivo de trazas del simulador NS2 .....	61
<b>Figura. 24:</b> Campos del archivo de trazas del simulador NS2. ....	63

<b>Figura. 25:</b> Diseño de la WSN en estrella. ....	71
<b>Figura. 26:</b> Sensor LM35DZ .....	73
<b>Figura. 27:</b> Sensor de Humedad .....	75
<b>Figura. 28:</b> Esquemático de Arduino UNO .....	76
<b>Figura. 29:</b> Conexión para un módulo XBee.....	79
<b>Figura. 30:</b> Descripción de los Pines del Xbee S2B.....	80
<b>Figura. 31:</b> XBee Shield. ....	82
<b>Figura. 32:</b> XBee Explorer USB .....	83
<b>Figura. 33:</b> Diagrama de bloques de un Nodo WSN Sensor. ....	84
<b>Figura 34:</b> Arduino Uno alimentado mediante una fuente de alimentación externa.....	85
<b>Figura 35:</b> Modulo XBee Pro S2B .....	86
<b>Figura 36:</b> Comunicación XCTU-XBee.....	88
<b>Figura 37:</b> Configuración del nodo Sensor en XCTU.....	90
<b>Figura 38:</b> Direcciones SH y SL de un XBee. ....	91
<b>Figura 39:</b> Diagrama de flujo del Nodo Sensor. ....	91
<b>Figura 40:</b> Diagrama de conexión de los elementos de un Nodo Sensor. ....	92
<b>Figura 41:</b> Diagrama de bloques de un Nodo WSN Coordinador.....	93
<b>Figura 42:</b> Conexión Pc-XBee Explorer. ....	94
<b>Figura 43:</b> Configuración del nodo Coordinador en XCTU .....	96
<b>Figura 44:</b> Diagrama de conexión de los elementos de un Nodo Coordinador.....	97
<b>Figura 45:</b> Tipos de tramas API de un módulo XBee. ....	98
<b>Figura 46:</b> Tramas API del nodo Coordinador.....	99
<b>Figura 47:</b> Bytes de una trama API (Hexadecimal). ....	100
<b>Figura 48:</b> Control Serial Port de VS Express 2013.....	101
<b>Figura 49:</b> Apertura del control Serial Port de VS Express 2013. ....	102

<b>Figura 50:</b> Buffer de lectura de los datos. ....	102
<b>Figura 51:</b> Función de recepción de los datos. ....	102
<b>Figura 52:</b> Definición de direcciones para cada sensor en el software. ....	103
<b>Figura 53:</b> Función para decodificar la trama de datos recibida. ....	103
<b>Figura 54:</b> Función para actualizar la información del Sensor 1 en la aplicación. ....	103
<b>Figura 55:</b> Función para actualizar la información del Sensor 2 en la aplicación. ....	104
<b>Figura 56:</b> Vista general de la Interfaz de Monitoreo. ....	104
<b>Figura 57:</b> Monitoreo del canal virtual USB a través de la aplicación.....	105
<b>Figura 58:</b> Pantalla de monitoreo de variables ambientales sector Orquideario. ....	106
<b>Figura 59:</b> Pantalla de monitoreo de variables ambientales sector Invernadero. ....	106
<b>Figura 60:</b> Pantalla de registro y almacenamiento de los datos de monitoreo. ....	107
<b>Figura 61:</b> Resultados del archivo .nam (Topología estrella) .....	113
<b>Figura 62:</b> Resultados del archivo .tr (Topología estrella).....	114
<b>Figura 63:</b> Resultado general de la simulación en NS2. (Topología estrella).....	114
<b>Figura 64:</b> Resultados del archivo .nam (Topología Arbol).....	115
<b>Figura 65:</b> Resultados del archivo .tr (Topología Arbol).....	116
<b>Figura 66:</b> Resultado general de la simulación en NS2. (Topología Árbol).....	116
<b>Figura 67:</b> Diagrama para pruebas de recepción de señal en exteriores .....	118
<b>Figura 68:</b> Prueba 50 metros exteriores.....	119
<b>Figura 69:</b> Prueba 100 metros exteriores.....	120
<b>Figura 70:</b> Prueba 200 metros exteriores.....	120
<b>Figura 71:</b> Prueba 300 metros exteriores.....	121
<b>Figura 72:</b> Resultados de RSSI en exteriores. ....	122
<b>Figura 73:</b> Resultados del nivel recepción de paquetes en exteriores. ....	122
<b>Figura 85:</b> Tramas API que llegan al coordinador enviadas por el nodo sensor.....	124

<b>Figura 86:</b> Nodo sensor. ....	125
<b>Figura 87:</b> Nodo Coordinador. ....	125
<b>Figura 88:</b> Interfaz de monitoreo (Selección del puerto COM) .....	126
<b>Figura 89:</b> Monitoreo de variables ambientales a través de la Interfaz gráfica. (Sector Invernadero).....	127
<b>Figura 90:</b> Monitoreo de variables ambientales a través de la Interfaz gráfica. (Sector Orquideario).....	128
<b>Figura 91:</b> Registro de mediciones a través de la Interfaz gráfica. ....	129
<b>Figura 92:</b> Pestaña para guardar el registro de mediciones .....	129

## ÍNDICE DE TABLAS.

<b>Tabla 1:</b> Comparativa entre los estándares WSN: Wifi, Bluetooth, y ZigBee .....	17
<b>Tabla 2:</b> Funciones de los dispositivos ZigBee. ....	21
<b>Tabla 3:</b> Topologías de WSN. ....	23
<b>Tabla 4:</b> Comparación entre los simuladores de WSN.....	38
<b>Tabla 5:</b> Características de la simulación. ....	56
<b>Tabla 5:</b> Variables ambientales a monitorear. ....	72
<b>Tabla 6:</b> Características del sensor de temperatura LM35DZ. ....	74
<b>Tabla 7:</b> Características y especificaciones del sensor de humedad.....	75
<b>Tabla 8:</b> Características de la placa Arduino Uno. ....	78
<b>Tabla 9:</b> Descripción de los pines del XBee Pro S2B. ....	80
<b>Tabla 10:</b> Especificaciones Técnicas del módulo XBee Pro S2B .....	81
<b>Tabla 11:</b> Características de la unidad de energía. ....	83
<b>Tabla 12:</b> Parámetros configurados en el XBee del Nodo Sensor.....	89
<b>Tabla 13:</b> Elementos utilizados en un nodo Sensor.....	92
<b>Tabla 14:</b> Parámetros configurados en el XBee del Nodo Coordinador. ....	96
<b>Tabla 15:</b> Elementos utilizados en un nodo Sensor.....	98
<b>Tabla 16:</b> Resultado de la comparación entre las métricas usadas en la simulación...	117
<b>Tabla 17.</b> Presupuesto del prototipo de WSN.....	130



**1. TÍTULO.**

**“DESARROLLO DE UN PROTOTIPO DE RED INALÁMBRICA DE  
SENSORES BAJO EL ESTANDAR IEEE 802.15.4 (ZIGBEE) PARA  
EL MONITOREO DE VARIABLES AMBIENTALES EN EL  
JARDIN BOTÁNICO REINALDO ESPINOSA DE LA  
UNIVERSIDAD NACIONAL DE LOJA.”**

## **2. RESUMEN.**

En este trabajo primeramente se realizó un estudio comparativo entre las topologías físicas que soportan las Redes Inalámbricas de Sensores WSN, (de sus siglas en inglés, wireless sensor network), con la finalidad de determinar cuál es la topología más eficiente para ser aplicada a una red de sensores ambientales; el estudio se realizó mediante el simulador NS2 (Network Simulator V2), que permite simular el funcionamiento de la red, los resultados obtenidos de las pruebas de envío de paquetes y consumo de energía, nos permitieron analizar las ventajas e inconvenientes de cada topología y finalmente determinar la más óptima, se concluyó que la topología en estrella es la indicada para la transmisión en tiempo real de los datos recogidos por los sensores.

A través de los resultados que se obtuvieron de la simulación se procedió a desarrollar un prototipo de Red Inalámbrica de sensores para el monitoreo de variables ambientales en el Jardín Botánico Reinaldo Espinosa, empleando la topología estipulada, para esto se diseñó el hardware y el software de dos nodos sensores y un nodo coordinador. Los nodos sensores son dispositivos terminales que se encargan de medir las variables de temperatura y humedad en determinada zona de cultivo, y posteriormente enviar estas mediciones hacia el dispositivo coordinador, quien a su vez procesa los datos de entrada y permite el reconocimiento y comunicación con los demás nodos presentes en la red.

Para el desarrollo del prototipo de WSN se utilizó principalmente módulos de comunicación inalámbrica XBee Pro S2B, módulos de procesamiento Arduino UNO, sensores y unidades de energía; finalmente para mostrar los datos al usuario se diseñó mediante el software VS Express 2013, una aplicación que nos permite monitorear en tiempo real los datos proporcionados por los nodos sensores que conforman la Red WSN.

## **ABSTRACT.**

In this work a comparative study was first made between the physical topologies which support the Wireless Networks of Sensors WSN, (because its initials in English, Wireless Sensor Network), with the purpose of determining which is the most efficient topology to be applied to a network of environmental sensors; The study was made by means of the simulator NS2 (Network Simulator V2), that allows to simulate the functioning of the network, the results obtained of the tests of sending of packages and energy consumption allowed us to analyze the advantages and disadvantages of every topology and finally to determine the most ideal, It was concluded that the topology in star is the indicated one for the real time transmission of the information gathered by the sensors.

Through the obtained results of the simulation we proceeded to develop a prototype of Wireless Network of sensors for the monitoring of environmental variables in the Botanical Garden Reinaldo Espinosa, using the stipulated topology, for this reason the hardware and the software of two nodes sensors and a coordinating node were designed. The nodes sensors are terminal devices that take charge measuring the variables of temperature and humidity in certain zone of crop, and then it sends these measurements towards the coordinating device, which in turn processes the information of entry and allows the recognition and communication with other present nodes in the network.

For the development of WSN's prototype it was used principally modules of wireless communication XBee Pro S2B, modules of processing Arduino ONE, sensors and units of energy; finally to show the information to the user we designed by means of the software VS Express 2013, one application that allows us to monitor real time the information provided by the nodes sensors that form part of the Network WSN.

### **3. INTRODUCCIÓN.**

La evolución constante en los últimos años de las tecnologías inalámbricas, ha permitido facilitar las comunicaciones y obtener datos de variables muy importantes, en el año 2003 el MIT (Instituto tecnológico de Massachusetts), identificó las 10 tecnologías emergentes que cambiarán el mundo entre las cuales las WSN (Wireless Sensor Networks) aparecen como unas de las primeras, a partir de este acontecimiento el desarrollo a lo largo del mundo de esta tecnología ha incrementado y están implementándose millones de redes de este tipo; inicialmente su campo solamente estaba relacionado con aplicaciones militares, ahora se ha extendido a muchas otras áreas industriales y de interés social, como a controles del proceso de producción, monitorización de la salud, automatización de hogares o control de tráfico.

Hoy en día, las Redes de Sensores Inalámbricas se encuentran en auge debido a que son una tecnología emergente muy prometedora para una amplia variedad de aplicaciones en ingeniería, están actualmente en el punto de mira de muchos investigadores y empresas tecnológicas. A nivel mundial, son varios los países que consideran a las WSN como un indicador tecnológico, Ecuador no es la excepción y empieza a sumergirse en este campo, gracias al proyecto realizado en conjunto por investigadores de las universidades Politécnica del Ejército, Universidad de Harvard y la Universidad de Carolina del Norte que ha conducido recientemente a la implementación de una WSN para monitorear erupciones en el Volcán Tungurahua, un volcán activo en el centro de Ecuador.

Al igual las redes de sensores inalámbricas son un tema muy activo de investigación en varias universidades, en la Universidad Nacional de Loja no existe un estudio preliminar sobre lo que sería el uso de las tecnologías WSN al monitoreo de variables ambientales y es lo que se pretende con la realización de la presente tesis cuya área de aplicación es el jardín botánico "Reinaldo Espinosa" mismo que es considerado el laboratorio natural más antiguo del país, fue fundado en el año 1949 por el Botánico Reinaldo Espinosa Aguilar, con el fin de impulsar la investigación y conservación ambiental; en sus 5 hectáreas de extensión se puede observar la más variada muestra de especies vegetales entre nativas, exóticas y endémicas, siendo un lugar para la exhibición de la biodiversidad

de la provincia de Loja; se encuentra bajo la dirección de un profesor de Botánica, quien decide el tipo de riego que se debe usar para cada sección del jardín bien sea este por aspersión o por goteo, de la misma forma tiene a su cargo establecer cada que tiempo se realiza este riego, cantidad de abono que se aplica, resuelve cuando es necesario tener un ambiente húmedo y de sombra para que las especies de flora puedan conservarse de la mejor manera, la toma de estas decisiones, por parte del administrador son en base a sus conocimientos en la materia tanto teóricos como prácticos, esto conlleva a que exista una adecuada conservación de la diversidad vegetal dentro del Jardín Botánico; sin embargo, no se tiene la precisión de que cantidad de riego, luz, abono etc., se está aplicando a determinado genero vegetal; puede suceder que se riegue determinada área aun cuando en ésta exista un 100% de humedad, lo que provoca un consumo innecesario del recurso del agua.

La mayor parte de ambientes agrícolas en nuestro país no poseen un sistema de comunicación en tiempo real para la obtención de los datos y por ende para análisis de estos siempre se dependerá de un factor humano que traslade la información hacia los centros de estudio, es evidente que las oportunidades que hoy en día brindan las TICs, para la monitorización y prevención temprana de fenómenos medio ambientales en tiempo real mediante el uso de WSN, resultaría beneficioso para la sociedad y las autoridades ecuatorianas.

Por ello este proyecto se apoya en el desarrollo de un prototipo de red inalámbrica de sensores para el monitoreo de variables ambientales en tiempo real; mediante el protocolo de comunicaciones Zigbee, con la finalidad de apoyar a los expertos a diagnosticar o prevenir un problema o urgencia ambiental, mediante el uso de nodos sensores conectados inalámbricamente, logrando con esto establecer técnicas de agricultura de precisión que permitan una reducción en el consumo de agua y pesticidas, contribuyendo a la preservación del entorno.

## **4. REVISIÓN DE LITERATURA.**

### **4.1. INTRODUCCIÓN A LAS REDES DE SENSORES INALÁMBRICAS.**

#### **4.1.1. Redes de Computadoras.**

La rápida evolución tecnológica de las computadoras en los últimos años nos conllevan a mejorar sus capacidades, que en principio estaban limitadas al procesamiento individual y aislado de datos; al poco tiempo de su aparición, ya se sintió la necesidad de interconectar estos dispositivos para que se puedan comunicar entre sí, de tal manera que creen nuevas formas de intercambio, tratamiento y manipulación de información, generando como resultado la aparición de lo que hoy en día conocemos como redes de computadoras. Las redes evolucionan a una velocidad significativa; constantemente aparecen nuevos protocolos, aplicaciones y dispositivos que mejoran las comunicaciones en diferentes niveles por ello en la actualidad no sólo las computadoras hacen uso de las redes, podemos encontrar sensores, dispositivos de control, celulares, PDAs, etc. que se conectan directamente a la red. [1] [7]

##### **4.1.1.1. Definición.**

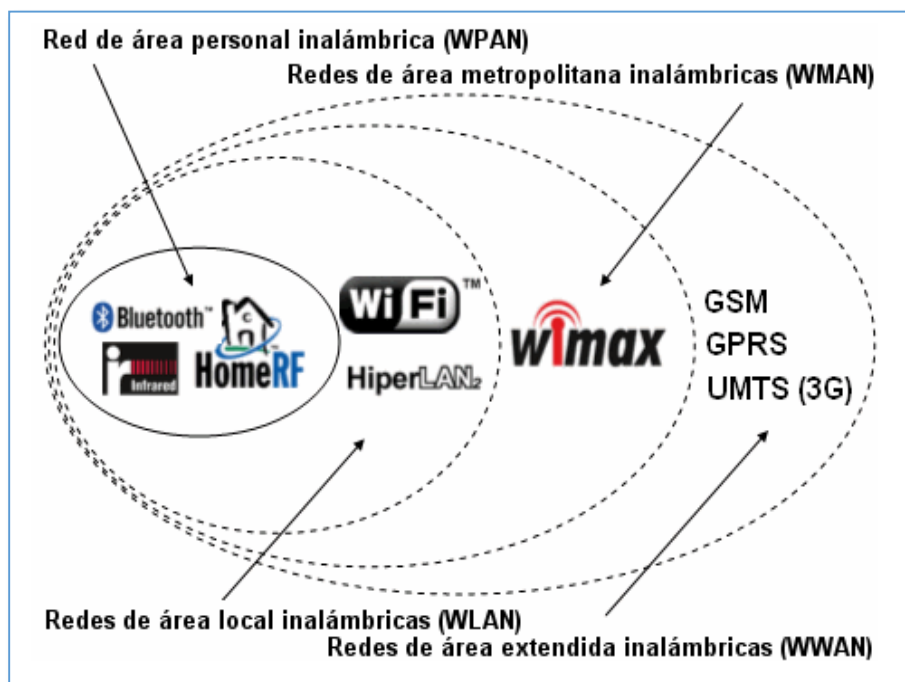
De manera general definimos a una red como un conjunto de elementos unidos, entre sí mediante algún medio. Dentro del ámbito de la informática se define a una Red como el conjunto o sistema de computadoras y otros dispositivos periféricos conectados entre sí y que establecen líneas de comunicación, ya sea de forma alámbrica o inalámbrica, para facilitar el intercambio de información y recursos de manera local o remota. El objetivo principal de una red de computadoras es compartir recursos e información a distancia, asegurando su confiabilidad y disponibilidad. [1]

#### **4.1.2. Redes Inalámbricas.**

Una red Inalámbrica se define como la interconexión de distintos dispositivos con la capacidad de compartir información entre ellos, pero sin un medio físico de transmisión.

Las redes inalámbricas constituyen actualmente una tecnología prometedora; sin embargo a pesar de que han presentado una gran acogida en el mundo de las redes no se espera que estas lleguen a reemplazar a las redes cableadas puesto que estas últimas presentan velocidades de transmisión mayores; el reto está en combinar las funcionalidades de las redes cableadas con las de las redes inalámbricas, para generar una "Red Híbrida" capaz de resolver problemas que de manera aislada estos dos tipos de redes no lograrían solucionar. [1] [8]

Por lo general, las redes inalámbricas se clasifican en varias categorías; en la Figura 1 se muestra la clasificación de las Redes Inalámbricas según su alcance o cobertura.



**Figura. 1:** Clasificación de las redes inalámbricas según su alcance.

Fuente: <http://es.ccm.net/contents/818-redes-inalambricas>.

#### 4.1.2.1. Redes Inalámbricas de área Extendida (WWAN).

Una W-WAN es una Red Inalámbrica de área amplia que abarca un área geográfica extensa, mediante este tipo de red podemos conectar múltiples localidades tales como

países, ciudades, universidades, instituciones, e incluso continentes. Está constituida por varias Redes de área local o Redes de Área Metropolitana.

Las Redes Inalámbricas de área amplia utilizan varias tecnologías de conexión entre las que podemos mencionar como las más usadas, las conexiones satelitales y las realizadas por antenas de radio microondas mismas que son mucho más flexibles, económicas y fáciles de instalar; la forma más común de implementación de una red W-WAN es por medio de Satélites con velocidades de transmisión del orden de los GHz; estos enlazan una o más estaciones bases, para la emisión y recepción, conocidas como estaciones terrestres; utilizan una banda de frecuencias para recibir la información, luego amplifican y repiten la señal para enviarla en otra frecuencia. [2] [9]

Entre los estándares W-WAN más usadas tenemos: GSM (Sistema Global de Comunicaciones Móviles), GPRS (Servicio General de paquetes de radio), UMTS (3G) (Sistema Universal de Telecomunicaciones Móviles).

#### **4.1.2.2. Redes Inalámbricas de área Metropolitana (WMAN).**

Las redes inalámbricas de área metropolitana también se conocen como bucle local inalámbrico (WLL, Wireless Local Loop). Las WMAN conectan diversas W-LAN cercanas geográficamente se basan en el estándar IEEE 802.16. Ofrecen una velocidad total efectiva de 1 a 10 Mbps, con un alcance de 4 a 10 kilómetros, algo muy útil para compañías de telecomunicaciones. La mejor red inalámbrica de área metropolitana es WiMax, que puede alcanzar una velocidad aproximada de 70 Mbps en un radio de varios kilómetros. [2]

#### **4.1.2.3. Redes Inalámbricas de área Local (WLAN).**

Una W-LAN (Wireless Local Area Network) es una red local inalámbrica que permite interconectar varios dispositivos periféricos conectados a ella, pero que la extensión o área que pueden abarcar se encuentra limitada físicamente a un entorno de unos pocos kilómetros que comprenden, por ejemplo, unos cuantos edificios o fábricas. Las Wireless LANs ofrecen muchas ventajas sobre las LANs Ethernet convencionales, tales son,



movilidad, flexibilidad, escalabilidad, velocidad, simplicidad, y costos reducidos de instalación. En la actualidad se han convertido en una solución para edificios que por su arquitectura, o su valor histórico, no pueden ser modificados para instalar cableado estructurado. El estándar más usado en redes inalámbricas de área local es el IEEE 802.11 (Wi-Fi). [9]

#### **4.1.2.4. Redes Inalámbricas de área personal (WPAN).**

Las redes de tipo Wireless PAN (Personal Área Network) son redes inalámbricas de área personal que nos permiten interconectar dispositivos que se encuentran cercanos al punto de acceso de la red, son redes de corto alcance que abarcan áreas de decenas de metros. Una Red de éste tipo está compuesta por una mezcla de dispositivos que se encuentran interactuando entre sí, por medio de una conexión inalámbrica como Bluetooth, Wi-Fi, ZigBee, etc. La principal limitación de este tipo de redes inalámbricas radica en su alcance pues está limitado, aunque en teoría podrían tener hasta 100 metros de cobertura bajo condiciones ideales. [9]

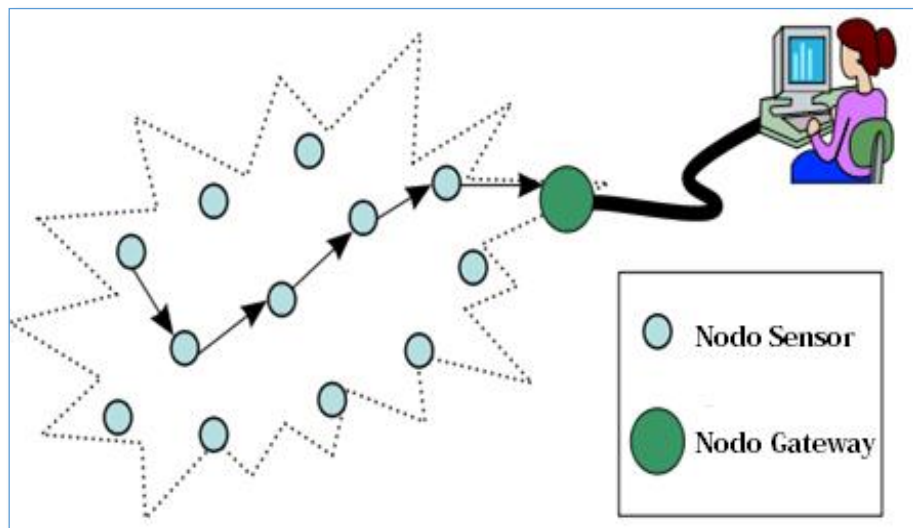
#### **4.2. REDES INALÁMBRICAS DE SENSORES (WSN).**

Las WSN forman parte de las Redes Inalámbricas de Área Personal (WPAN), están formadas por numerosos sensores los cuales están distribuidos espacialmente en un área geográfica determinada y recogen información de distintos fenómenos físicos y ambientales, con la finalidad de que estos sean procesados y se puedan obtener estadísticas y un control en tiempo real de los mismos. Este tipo de redes están diseñadas para aplicaciones en las cuales no se cuente con una infraestructura cableada, o para lugares donde sea inaccesible el suministro energético como por ejemplo un invernadero en donde llegar con cable tanto de red como eléctrico resultaría un verdadero inconveniente. Por esta razón es necesario que los nodos funcionen de manera inalámbrica y con la mínima cantidad de energía proporcionada por fuentes independientes tales como pilas, baterías o panel solar. [4]

Las redes inalámbricas de sensores se basan en dispositivos de bajo coste que son capaces de obtener información de su entorno, procesarla localmente y comunicarla a través de enlaces inalámbricos hasta un nodo coordinador central.

#### 4.2.1. Definición.

Una WSN se define como una red inalámbrica constituida por nodos o dispositivos autónomos, que a su vez están conformados por sensores que se encuentran distribuidos en un área geográfica determinada, para cumplir con una tarea en común. En la Figura 2 se muestra una red inalámbrica de sensores. [4]



**Figura. 2.** Red Inalámbrica de Sensores.

**Fuente:** <http://image.slidesharecdn.com/presentacinpfc-130304133037-phpapp02/95/improvement-on-energy-consumption-in-wsn-applications-3-638.jpg?cb=1362404251>

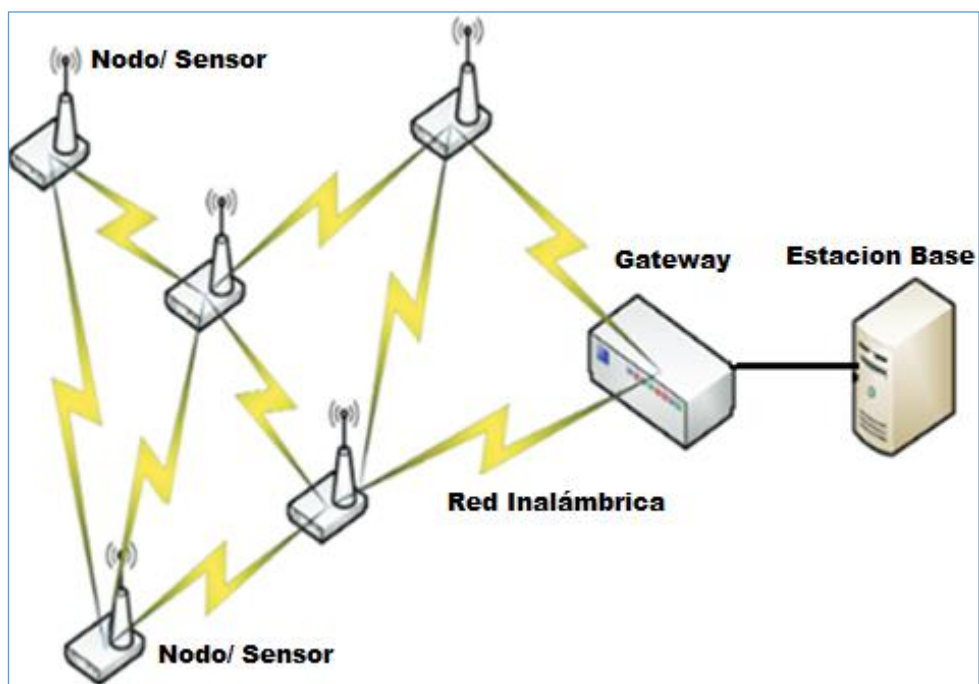
#### 4.2.2. Elementos.

Una red inalámbrica de sensores está compuesta por los siguientes elementos:

- ✓ **SENSORES:** De distinta naturaleza y tecnología toman del medio la información y la convierten en señales eléctricas.

- ✓ **NODOS SENSOR:** Dispositivos que toman los datos del sensor a través de sus puertas de datos, y envían la información a la estación base.
- ✓ **GATEWAY:** Elementos destinados para la interconexión entre la red de sensores y la estación base. Es un nodo especial sin elemento sensor, cuyo objetivo es actuar como puente entre dos redes de diferente tipo.
- ✓ **ESTACIÓN BASE:** Recolector de datos basado en un ordenador común o sistema embebido.
- ✓ **RED INALÁMBRICA:** Típicamente basada en el estándar IEEE 802.15.4 ZigBee. [4]

En la Figura.3 podemos observar los elementos que conforman una WSN.



**Figura. 3:** Elementos de una WSN.

Fuente: <http://www.elprofesionaldelainformacion.com/contenidos/2015/ene/08.pdf>

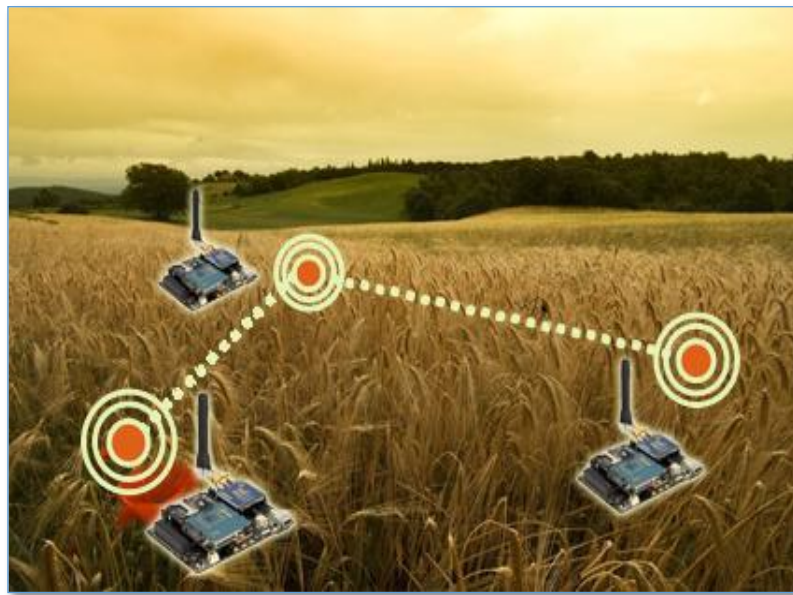
### **4.2.3. Características.**

- ✓ Una WSN se adapta a cualquier cambio en la red debido a que los nodos son auto-configurables, con tolerancia a fallos y presentan una elevada fiabilidad.
- ✓ No se utiliza una infraestructura de red predeterminada es decir una red de sensores no tiene necesidad alguna de infraestructura para poder operar, ya que sus nodos pueden actuar de emisores, receptores o enrutadores.
- ✓ Tolerancia a errores es decir un nodo dentro de la red debe ser capaz de mantener su funcionamiento a pesar de errores en el sistema.
- ✓ Bajo consumo energético debido a que dependen de pilas o baterías y tienen una larga autonomía de funcionamiento. Pueden operar sin mantenimiento durante varios meses o años.
- ✓ Limitaciones de Hardware: Con el fin de ahorrar energía, el hardware debe ser lo más sencillo posible. Esto nos limita a tener una capacidad de procesamiento limitada.
- ✓ Soporta múltiples opciones de conectividad: Las redes WSN no limitan al uso de la tecnología a utilizar, esto depende del uso y de las condiciones que presente el lugar en donde se las va a aplicar, la tecnología más utilizada es Zigbee sin embargo se puede utilizar Wi-Fi, Wimax, Bluetooth, etc.
- ✓ Costo de producción: normalmente bajo en comparación a otras redes. [4] [5]

### **4.2.4. Aplicación de las WSN en la agricultura.**

Las aplicaciones que ofrecen las WSN al ser una tecnología inalámbrica de bajo consumo de energía son numerosas, pues son utilizadas en todas o en la mayoría de áreas en las que se requiera monitorizar, estudiar y almacenar datos recogidos desde un sensor. Debido a que la presente tesis se enfoca en el monitoreo de variables ambientales de un

Jardín botánico se hará referencia al uso de las WSN dentro del sector agrícola. La agricultura es una de las áreas en donde las redes de sensores han tenido mayor acogida ya que se necesita la recolección de datos que quizá para el hombre sean imperceptibles como por ejemplo la cantidad de CO<sub>2</sub> en el medio o la cantidad de radiación que esté presente en determinada área de sembrío, estas variables, ahora son posibles monitorear gracias al uso de redes de sensores. [4]



**Figura. 4:** Aplicación de las WSN en la Agricultura.

**Fuente:** <http://www.probesrl.net/eng/wp-content/uploads/2010/06/campo.jpg>

Otros factores importantes que se han podido monitorear y controlar gracias a las WSN son:

- ✓ Control de la cantidad de agua, fertilizante o pesticida que las plantas necesitan.
- ✓ Medida de la humedad del suelo.
- ✓ Decisión del momento óptimo para realizar la cosecha.
- ✓ Optimización de la producción y la calidad de una cosecha. [4]

- ✓ Gestión de alarmas por intrusión de animales o daños provocados por heladas.
- ✓ Detección de incendios forestales.

#### **4.2.5. Estándares para WSN.**

Los Estándares Bluetooth, Wi-Fi (IEEE 802.11) y Zigbee son las opciones más populares y comercialmente disponibles cuya utilización en redes inalámbricas de sensores ha sido evaluada.

##### **4.2.5.1. Estándar ZigBee.**

Es probablemente el estándar que mejor se adapta a las Redes de Área Personal (PANs) y, por tanto, a las redes inalámbricas de sensores, actualmente, es objeto de gran interés a nivel comercial y de investigación. Se basa sobre el estándar IEEE 802.14.5 que garantiza un radio de comunicación teórico de 70-100m; la frecuencia es 2.4Ghz a 250Kbps. Su punto fuerte es el bajo coste de comunicación relacionado con las prestaciones proporcionadas: una red inalámbrica de sensores, más que ancho de banda, requiere poco gasto energético. [10]

##### **4.2.5.2. Estándar Wifi.**

La familia de estándares IEEE 802.11 (802.11a, 802.11b y 802.11g), más conocida como WiFi, tiene asignadas las bandas ISM (Industrial, Scientific and Medical) 902-928 MHz, 2.400-2.4835 GHz, 5.725-5.850 GHz para uso en las redes inalámbricas. WiFi comparte la mayoría de su funcionamiento interno con Ethernet, sin embargo este difiere en la especificación de la capa física (PHY) utilizando señales de radio en lugar cable. Wifi presenta un gran ancho de banda (entre 1 y 11 Mbps para 802.11b y hasta 54Mbps para 802.11a/g), como una de las mejores opciones para la transmisión de datos y redes de telefonía empleando VoIP (voz sobre IP). [10]

En términos generales, el estándar de protocolos IEEE 802.11 (WI-FI) tiene los siguientes inconvenientes al ser aplicado a Redes WSN.

- ✓ Si bien se proveen algunas funcionalidades de ahorro de energía, en general está orientado a altas tasas de transmisión, y los transceptores disponibles requieren una cantidad de energía que son órdenes de magnitud mayores que lo aceptable en aplicaciones de redes de sensores.
- ✓ Tiene un número limitado de canales no interferentes, 3 en 2.4 GHz y 8 en 5.8 GHz. [10]

#### **4.2.5.3. Estándar Bluetooth.**

Bluetooth es un estándar diseñado para redes inalámbricas de área personal, posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. su principal aplicación es la conexión de dispositivos a una computadora personal.

Basado sobre el estándar IEEE 802.15.1, proporciona, respecto a ZigBee, una mayor velocidad de transmisión a costa de un mayor gasto energético. En principio este aspecto puede resultar determinante en sentido negativo para la utilización en ámbito de redes de sensores; de todos modos, hay unos temas de interés, como las aplicaciones multimediales sobre redes de sensores inalámbricas, en el cual Bluetooth puede ser un válido antagonista de ZigBee e, incluso, proporcionar mejores resultados que este. En la actualidad se han implementado prototipos de redes de sensores basadas en Bluetooth, los nodos son organizados en pico-redes (red formada por dos o más dispositivos Bluetooth) con un nodo maestro y un máximo de siete nodos esclavos activos. El maestro elige la secuencia de saltos que deben seguir los esclavos. [10]

Entre los inconvenientes de la aplicación de Bluetooth a redes inalámbricas de sensores tenemos:

- ✓ La necesidad de tener un nodo maestro constantemente, con el costo de interrogar sus esclavos.
- ✓ La cantidad limitada de esclavos por picored que soporta.

- ✓ Para el caso de redes de sensores densas, se necesitaría un número enorme de nodos maestros.
- ✓ Un esclavo activo debe permanecer siempre encendido, ya que no puede predecir cuándo será interrogado por el maestro.
- ✓ Un esclavo pasivo debe postularse con el maestro para cambiar a activo, y si ya hay siete nodos activos, será rechazado.
- ✓ Se requiere que cada nodo pueda asumir el rol de maestro o esclavo, agregando una complejidad considerable. [10]

En la Tabla 1 se presenta una comparación entre las características principales de estos tres estándares. [11]

### **4.3. ESTÁNDAR IEEE 802.15.4 (ZIGBEE).**

#### **4.3.1. Definición.**

ZigBee es una tecnología propia de Redes Inalámbricas de Área Personal (W-Pan) diseñada por la ZigBee Alliance; constituye una solución de comunicación de corto alcance y de bajo consumo energético, basado principalmente en el estándar IEEE 802.15.4, su objetivo principal es utilizar un conjunto de dispositivos llamados sensores, que funcionan a bajos niveles de energía y bajas transferencias de datos en comparación con otras tecnologías inalámbricas de corto alcance. [12]

Es actualmente la tecnología más utilizada en el diseño e implementación de WSN (Redes Inalámbricas de Sensores.) para entornos industriales, médicos, de monitorización y sobre todo domóticos.



**Tabla 1:** Comparativa entre los estándares WSN: Wifi, Bluetooth, y ZigBee

<b>ESTÁNDAR</b>	<b>ZIGBEE</b>	<b>BLUETOOTH</b>	<b>WI-FI</b>
<b>Especificación IEEE</b>	802.15.4	802.15.1	802.11.b
<b>Velocidad de Transmision</b>	250Kbps	1Mbps	11 Mbps
<b>Tamaño de la red</b>	65.000 nodos	8 nodos	255 nodos
<b>Latencia</b>	30ms	10s	3s
<b>Tipo de datos</b>	Pequeños paquetes de datos	Audio-Gráficos	Video Audio Gráficos
<b>Topologías</b>	Estrella- árbol- malla	Estrella	Estrella
<b>Cobertura (metros)</b>	70-1600	10-100	100
<b>Duración de baterías</b>	100-1000 días	1 semana	0.5 a 5 días
<b>Precio</b>	Accesible	Bajo	Costoso
<b>Aplicación principal</b>	Control y monitorización	W-PAN	W-LAN
<b>Memoria necesaria</b>	4Kb-32Kb	250Kb	1MB
<b>Expansión</b>	si	no	Roaming
<b>Parámetros importantes</b>	Fiabilidad- Bajo coste Bajo consumo	Costes y perfiles de aplicación	Velocidad Flexibilidad

Fuente: <https://riunet.upv.es/bitstream/handle/10251/8592/PFC%20-%20DESARROLLO%20DE%20APLICACIONES%20BASADAS%20EN%20WSN.pdf?sequence=1>

#### **4.3.2. Estándar 802.15.4.**

Las redes W-Pan están divididas para su estudio en cuatro grandes grupos de trabajo, de los cuales el grupo 4 (Task Group 4) es el encargado de del desarrollo del estándar IEEE 802.15.4 enmarcado en el estudio de adelantos de investigaciones y soluciones que requieren baja transmisión de datos, bajo consumo de energía y baja complejidad.

El estándar IEEE 802.15.4 está encargado de definir las dos capas más bajas de la pila de protocolos ZigBee que son: la capa Física y la capa de enlace de datos. La revisión más actual de este estándar se aprobó en el 2009. [12]

#### **4.3.3. Características de ZigBee.**

- ✓ Opera en las bandas libres ISM (Industrial, Scientific & Medical) de 2.4 GHz, 868 MHz (Europa) y 915 MHz (Estados Unidos).
- ✓ Velocidad de transmisión entre 25-250 kbps lo que lo hace útil para aplicaciones que no requieran alta transmisión de datos.
- ✓ Rango de cobertura de 10 a 75 metros.
- ✓ 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, entonces, mayor número de rutas alternas existirán para garantizar que un paquete llegue a su destino.
- ✓ Soporta diferentes tipos de topologías como estrella, punto a punto, malla, árbol.
- ✓ Acceso de canal mediante CSMA/CA (acceso múltiple por detección de portadora con evasión de colisiones).
- ✓ ZigBee tiene un consumo de 30mA transmitiendo y de 3 $\mu$ A en reposo.

- ✓ Teóricamente una red ZigBee puede estar constituida por hasta 65 000 nodos, distribuidos en subredes de 255 nodos; estos límites se pueden ver afectados por algunas restricciones físicas como memoria disponible, ancho de banda, etc.
- ✓ Se basa en un estándar de comunicación 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, de esta manera, se incrementa el área de cobertura de la red.
- ✓ Los dispositivos de estas redes pueden funcionar en un modo de bajo consumo, lo que supone años de duración de sus baterías.
- ✓ A pesar de coexistir en la misma frecuencia con otro tipo de redes como WiFi o Bluetooth su desempeño no se ve afectado, esto debido a su baja tasa de transmisión y, a características propias del estándar IEEE 802.15.4. Se puede decir que ZigBee ocupa el vacío que hay por debajo de Bluetooth, para comunicaciones de datos que no requieren altas velocidades. [12]

#### **4.3.4. Tipos de dispositivos ZigBee.**

Una vez comentadas las características de esta tecnología, podemos pasar a hablar sobre los dispositivos que ésta utiliza.

- ✓ **Coordinador ZigBee (ZC).**

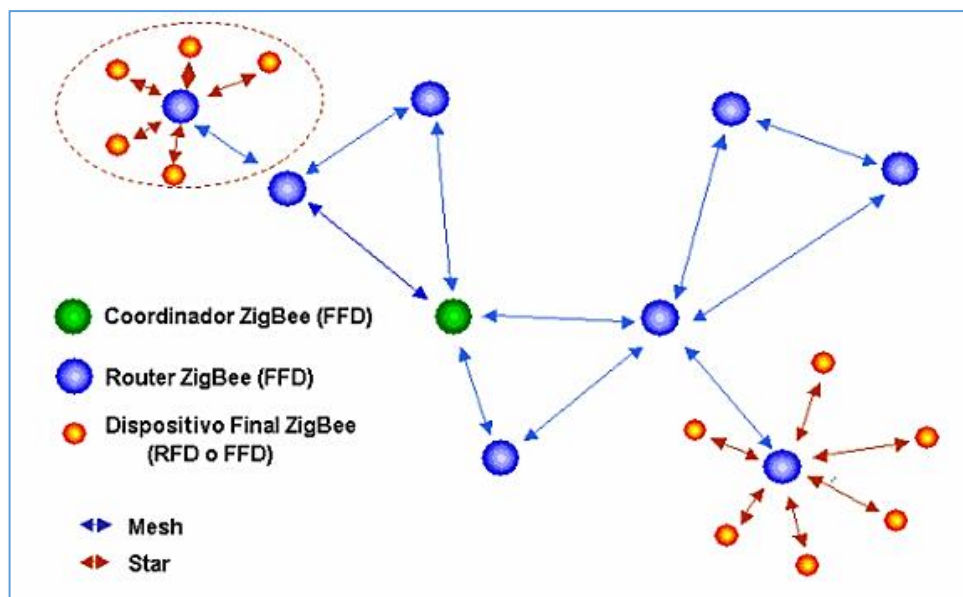
Se trata del dispositivo más completo e importante de la WSN. Su función principal es almacenar la información que fluye a través de la red; puede actuar como director de una red en árbol así como servir de enlace con otras redes. Debe existir uno por red.

- ✓ **Router ZigBee (ZR).**

Como su nombre lo indica actúa como un enrutador interconectando dispositivos separados en la red y limitados debido a su rango.

✓ **Dispositivo final (ZED).**

Este dispositivo puede comunicarse con su nodo padre bien sea este 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 es por tanto significativamente más barato, que el resto de dispositivos. Los dispositivos descritos se representan en la Figura 5.



**Figura. 5:** Tipos de dispositivos ZigBee

Fuente: [http://www.gta.ufrj.br/grad/10\\_1/zigbee/dispositivos.html](http://www.gta.ufrj.br/grad/10_1/zigbee/dispositivos.html)

Basándose en su funcionalidad, puede plantearse una segunda clasificación:

✓ **Dispositivo de funcionalidad completa (FFD).**

También conocidos como nodo activo. Es 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. [13]

✓ **Dispositivo de funcionalidad reducida (RFD).**

También conocido como nodo pasivo. Tiene capacidad y funcionalidad limitadas (especificada en el estándar) con el objetivo de conseguir un bajo coste y una gran simplicidad. Básicamente, son los sensores/actuadores de la red. [13]. Las funciones que estos dispositivos deben cumplir dentro de una red ZigBee, se detallan en la Tabla 2.

**Tabla 2:** Funciones de los dispositivos ZigBee.

Función	Dispositivo Coordinador	Dispositivo Router	Dispositivo Final
Establecer una nueva Red ZigBee	*		
Direccionamiento lógico de la red.	*	*	
Permitir que dispositivos entren o salgan de la red ZigBee.	*	*	
Mantener lista de vecinos y rutas.	*	*	
Enrutar los paquetes en la capa de red	*	*	
Transferencia de paquetes a la capa de red	*	*	*

Fuente: [http://www.gta.ufrj.br/grad/10\\_1/zigbee/dispositivos.html](http://www.gta.ufrj.br/grad/10_1/zigbee/dispositivos.html)

**4.3.5. Topologías ZigBee.**

✓ **Topología Estrella.**

En una Red basada en IEEE802.15.4 y configurada en topología estrella el dispositivo coordinador de la red se sitúa en el centro de la misma y el resto de dispositivos

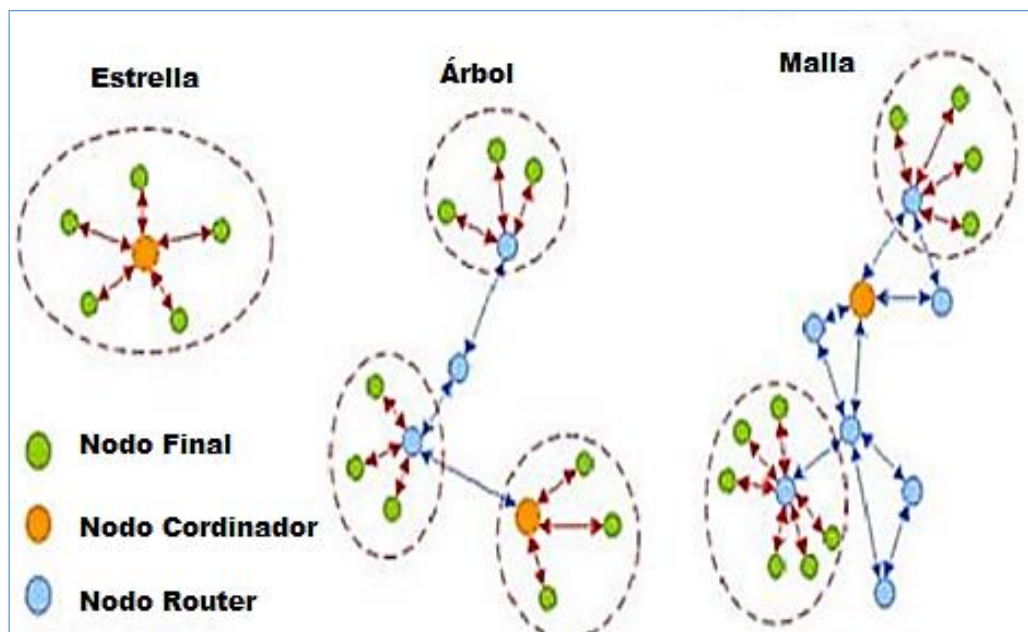
pertenecientes a la red, sean FFD (Dispositivos de funcionalidad completa) o RFD (Dispositivos de funcionalidad reducida) se comunican directamente con él.

✓ **Topología en Árbol.**

En topología en árbol el coordinador de la red se comunica con dispositivos FFDs, los cuales a su vez se comunican de forma directa con los nodos de su cluster.

✓ **Topología en Malla.**

En topología en malla los nodos tendrán más de una conexión, con lo cual existen muchas rutas para hacer llegar mensajes al coordinador de red en caso de que una ruta falle.



**Figura. 6:** Tipos de topologías ZigBee.

Fuente:<http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/229/A6.pdf?sequence=6>

En la tabla 3 se presentan las principales ventajas y desventajas de estas topologías dentro de una red ZigBee, considerando una red aplicada a la agricultura donde los nodos van a estar ubicados de forma fija. [14]

**Tabla 3:** Topologías de WSN.

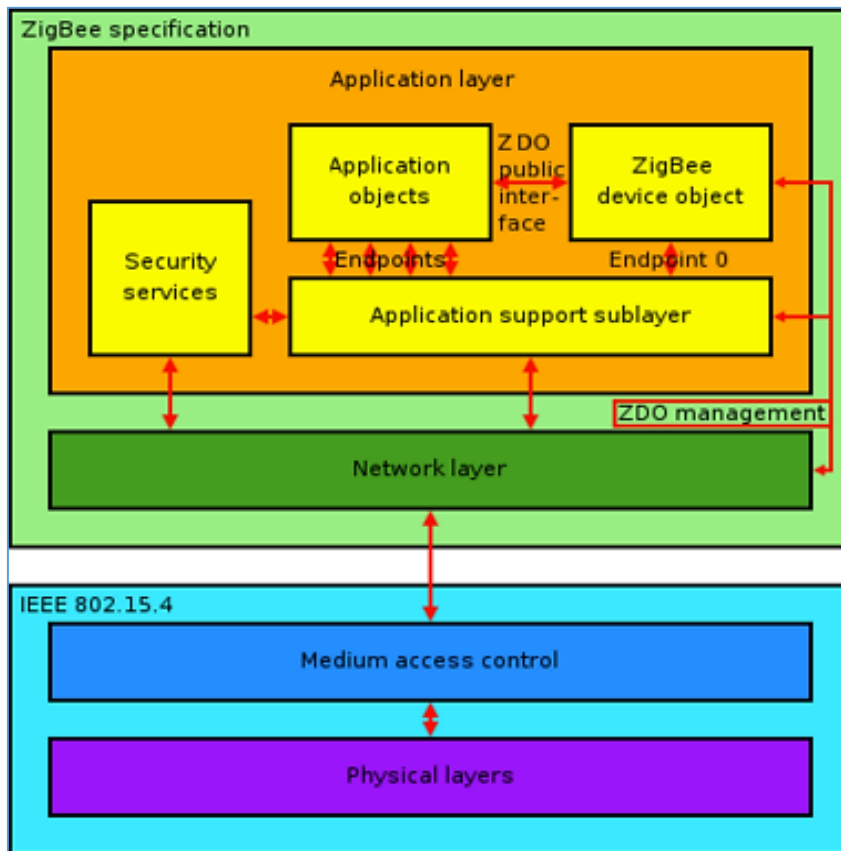
Topología	Ventajas	Inconvenientes
<b>Estrella</b>	<p>Baja latencia</p> <p>Topología muy robusta.</p> <p>Fiabilidad muy alta.</p> <p>Sencillez y rapidez en el desarrollo.</p> <p>Bajo consumo energético.</p>	<p>No siempre es posible desplegar una topología de este tipo.</p> <p>Posibles problemas de colisiones cuando aumenta el número de ZEDs.</p> <p>Si falla el nodo central cae toda la red.</p>
<b>Árbol</b>	<p>Mantiene las ventajas de la topología Estrella.</p> <p>Alta escalabilidad.</p> <p>Menor porcentaje de colisiones.</p>	<p>La inclusión de routers puede encarecer significativamente la solución final.</p> <p>Puede caer una parte significativa de la red al caer un router.</p>
<b>Malla</b>	<p>Pueden caer uno o varios nodos que la información seguirá obteniéndose, al existir rutas alternativas.</p>	<p>Complejidad del sistema alta.</p> <p>Alta cantidad de colisiones.</p> <p>Empeora la latencia de la red.</p> <p>Despliegue muy costosos.</p>

Fuente: <http://www.javierlongares.com/arte-en-8-bits/introduccion-a-zigbee-y-las-redes-de-sensores-inalambricas/>

#### 4.3.6. Arquitectura ZigBee.

La arquitectura de la pila de protocolos ZigBee está hecha de una serie de bloques llamados capas, cada capa realiza una serie de servicios específicos para la capa posterior. Esta arquitectura está basada en el modelo OSI, el estándar IEEE 802.15.4 define las dos

capas inferiores de esta pila de protocolos que son: la capa física (PHY) y la subcapa de control de accesos al medio (MAC), por su parte la Alliance ZigBee define la capa de red (NWK) y la capa de aplicación; esto se muestra en la Figura 7. [15]



**Figura. 7:** Arquitectura ZigBee

Fuente: <http://ricveal.com/contenido/uploads/2014/07/arquitectura-zig-bee.png>

#### 4.3.6.1. Capa Física (PHY).

Es la encargada de la transmisión y recepción de datos utilizando un canal de datos determinado. El estándar 802.15.4 puede utilizar tres bandas de operación: 868MHz, 915MHz y 2.4 GHz. En la primera banda entre 868MHz y 868,6MHz tenemos un canal, en la segunda banda de frecuencia entre 902 y 928 MHz tenemos 10 canales y en la tercera banda entre 2.4 y 2.48 GHz tenemos 16 canales de operación, Las velocidades de transferencia son de 20Kbps en 868MHz, 40Kbps a 915MHz y 250Kbps a 2.4 GHz. Las



frecuencias más bajas son las más adecuadas para la transmisión debido a las bajas pérdidas de propagación, mejor sensibilidad y mayor cobertura. Al momento de elegir nuestro rango de frecuencia debemos tener en cuenta un factor muy importante:

- ✓ 868MHz solo opera en Europa,
- ✓ 915MHz solo opera en Estados Unidos y Australia.
- ✓ 2.4 GHz es universal, opera en la mayoría de los países.

#### ⊗ **Funciones de la Capa Física.**

- ✓ Activar y desactivar el nodo sensor: el nodo sensor tiene tres tipos de funcionamiento: Recepción, Transmisión y modo Sleep.
- ✓ Indicar la calidad del enlace: mide la potencia en dBm de la señal que ha transmitido el último paquete.
- ✓ Detección de Energía de cada canal: función que luego es utilizada por la capa de red para determinar que potencia tiene determinado canal.
- ✓ Evaluación del canal libre: usado por el protocolo CSMA-CA. [15]

#### **4.3.6.2. Capa de Acceso al Medio.**

La capa MAC o de acceso al medio proporciona una interfaz de comunicación entre la capa física y protocolos de capas superiores, en esta capa el estándar 802.15.4 posee un protocolo llamado CSMA-CA, que es el encargado de reducir la probabilidad de colisiones, ya que las colisiones son más frecuentes en redes de baja transferencia de datos como los son las redes ZigBee.

### ⊗ **Características de la capa MAC.**

- ✓ Generar balizas<sup>1</sup> en el caso que el dispositivo sea un coordinador.
- ✓ Sincronizar las balizas de la red.
- ✓ Aplicar el protocolo CSMA-CA para la elección del canal.

### ⊗ **Funciones de la capa MAC.**

Al momento de la inicialización de la red ZigBee el dispositivo coordinador es el encargado de seleccionar el modo de trabajo de esta capa, mismo que puede ser de dos modos:

#### ✓ **Modo Baliza:**

El coordinador es el encargado de generar balizas en forma periódica para la sincronización de los dispositivos, dicha sincronización sirve de ayuda para colocar los dispositivos en modo de operación, ya que estos se encuentran la mayor parte del tiempo en modo Sleep, colocándolos a transmitir datos almacenados que contengan cada nodo cuando la red así lo necesite, una vez que termine la acción los configura de nuevo a estado original en modo reposo, para que exista ahorro de energía.

#### ✓ **Modo No Baliza:**

En este modo los únicos dispositivos que están en modo de reposo son los dispositivos finales ya que todos los dispositivos de la red no se encuentran sincronizados unos con otros, los Routers y el Coordinador se encuentran en modo de trabajo constante, debido a que no saben cuándo entren en acción los dispositivos finales. [15]

---

<sup>1</sup> Beacon o baliza es una señal enviada por el cliente inalámbrico o router para indicar a la red que se encuentra en estado activo.

#### **4.3.6.3. Capa de Red.**

Esta capa actúa como interfaz entre la capa MAC y la capa aplicación. Esta capa es la responsable de iniciar la red, unirse a la red, enrutar paquetes dirigidos a otros nodos en la red, proporcionar los medios para garantizar la entrega del paquete al destinatario final, filtrar paquetes recibidos, cifrarlos y autentificarlos. Por lo tanto, esta capa implementa seguridad y encamina tramas a sus respectivos destinos. Cabe destacar que la capa de red del controlador de red es la responsable de crear una nueva red y asignar direcciones a los dispositivos de la misma.

Por último debemos comentar que esta es la capa donde están implementadas las diferentes topologías de red que soporta Zigbee.

#### **4.3.6.4. Capa de Soporte a la Aplicación.**

Se encarga de filtrar paquetes a nivel de aplicación, mantener la relación de grupos y dispositivos con los que la aplicación interactúa y simplificar el envío de datos a los diferentes nodos de la red. Las capas de red y de soporte a la aplicación son definidas por la ZigBee Alliance.

#### **4.3.6.5. Capa de Aplicación.**

Es la capa de la que se encargan los fabricantes, donde se encuentran los ZDO (ZigBee Device Objects) aquí se define el papel del dispositivo en la red. Podemos decir que esta capa es la aplicación misma. [16]

### **4.4. MÓDULOS X-BEE- ZIGBEE.**

#### **4.4.1. Definición.**

De acuerdo a Digi la empresa líder en la fabricación de estos dispositivos; se define a los módulos XBee como soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Estos módulos basan su funcionamiento

en el protocolo de comunicaciones inalámbricas para WPAN IEEE 802.15.4, y permiten la creación tanto de redes punto a multipunto; como de redes punto a punto. [17]

Los módulos XBee son dispositivos de radiofrecuencia que integran un transmisor-receptor de ZigBee y un procesador en un mismo módulo, lo que permite a los usuarios desarrollar sus aplicaciones programando directamente en el módulo, eliminando la necesidad de configuración de un procesador independiente. [18] [19]

#### **4.4.2. Características.**

Entre las características principales de los Modulos X-Bee tenemos:

- ✓ Alcance: hasta 300ft (100 mts) para los módulos Xbee y hasta 1 milla (1.6 Km) para los módulos Xbee Pro.
- ✓ Bajo consumo menor a 50mA cuando están en funcionamiento y menor 10uA cuando están en modo sleep.
- ✓ Escalabilidad: se puede tener 65,000 direcciones (Nodos) para cada uno de los 16 canales disponibles.
- ✓ Son económicos y cuentan con Interfaz serial que nos permite la configuración de estos dispositivos.
- ✓ Interoperabilidad, son fáciles de integrar y son compatibles con otros dispositivos Zigbee.
- ✓ Permiten la configuración de diversas topologías de Red tales como Estrella, Árbol y topología en malla.
- ✓ Poseen versiones de Firmware programables para cada dispositivo X-Bee, lo que nos permite personalizar nuestro módulo de acuerdo al objetivo de nuestra aplicación. [19]

#### 4.4.3. Series.

Existen 2 series de módulos XBee. La serie 1 y la serie 2 o también conocida como 2.5.

- ✓ **XBEE Series 1:** También conocidos como XBee 802.15.4 son la serie más fácil para trabajar, no necesitan ser configurados, pero incluso así se pueden obtener beneficios según la aplicación a la que sean destinados; son útiles en comunicaciones Punto-a-Punto. Esta serie está basada en el chipset Freescale y se pueden configurar en modo AT como en modo API. [17] [20]
- ✓ **XBEE Serie 2:** Los módulos de la serie 2 necesitan ser configurados previamente para su correcto funcionamiento están basados en el chipset del fabricante Ember y están diseñados para ser utilizados en comunicaciones punto a punto, punto a multipunto y en aplicaciones que requieren repetidores o para red mesh. Al igual que los de la serie 1 pueden ser utilizados en los modos AT y API. Dentro de la serie 2 se encuentran los módulos XBEE Serie 2B son nuevos módulos que poseen mejoras en el hardware respecto de los de la Serie 2, como en el uso de potencia y rango de cobertura. [17] [20]
- ✓ **Regular vs Pro:** Tanto la serie 1 como la serie 2 de módulos Xbee cuentan con su versión PRO entre las diferencias a destacar entre un XBee regular y un XBee PRO en cuanto a hardware su dimensión el XBee PRO es un poco largo y con respecto a comunicación, la versión Pro tiene un mayor alcance (1,6 Km línea vista), pero para ello tiene un mayor consumo de potencia. [17]

El criterio para tomar la decisión de cual usar es la distancia a la que se requiera comunicar dos XBee. Los dos modelos tanto regular como pro son interoperables y se pueden usar dentro de la misma red, siempre cuando pertenezcan a la misma serie.

En la Figura 8 podemos observar una comparación entre las características principales de los diferentes módulos XBee que actualmente se comercializan en la banda de 2.4GHz.

XBEE	Velocidad	Frecuencia	Potencia Tx	Antena	ID Digital	Entradas ADC	Rango
 XBee 1mW PCB Antenna	250kbps	2.4 GHz	1mW output (+0dBm)	Built-in	8	(6) 10-bit	300ft (100m)
 XBee 1mW Wire Antenna	250kbps	2.4 GHz	1mW output (+0dBm)	Built-in	8	(6) 10-bit	300ft (100m)
 XBee 2mW Chip Antenna - Series 2	250kbps	2.4 GHz	2mW output (+3dBm)	Built-in	8	(6) 10-bit	400ft (120m)
 XBee 2mW RPSMA - Series 2	250kbps	2.4 GHz	2mW output (+3dBm)	RPSMA	8	(6) 10-bit	400ft (120m)
 XBee 2mW Wire Antenna - Series 2 (Mesh)	250kbps	2.4 GHz	2mW output (+3dBm)	Built-in	8	(6) 10-bit	400ft (120m)
 XBee Pro 50mW RPSMA - Series 2 (Mesh)	250kbps	2.4 GHz	50mW output (+17dBm).	RPSMA	8	(6) 10-bit	1 mile (1600m)

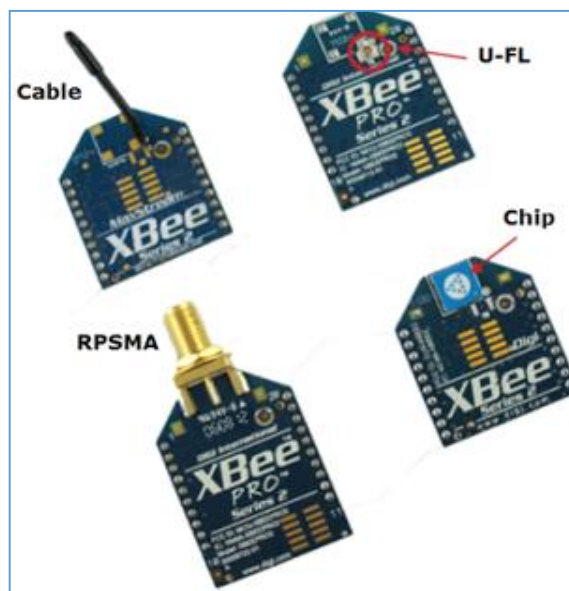
**Figura. 8:** Tipos de XBee en la banda de 2.4GHz.

Fuente: <http://xbee.cl/comparativa/>

#### 4.4.4. Tipo de Antenas de los módulos XBEE.

Las antenas que componen un módulo XBEE pueden ser de distintos tipos y las podemos visualizar en la Figura. 9, entre las cuales se tiene:

- ✓ **Chip Antena:** Básicamente es un pequeño chip que actúa como antena, rápido, sencillo y barato.
- ✓ **Wire Antena (Whip Antena):** Es un pequeño cable que sobresale del módulo.
- ✓ **u.FL Antena:** Es un conector pequeño para adaptar una antena externa; es perfecto si se tiene el equipo en una caja y se desea la antena fuera de ésta.
- ✓ **RPSMA Antena:** consiste de un conector grande que nos permite conectar una antena externa. Al igual esto resulta perfecto si se tiene nuestro dispositivo XBee en una caja y se desea la antena fuera de ésta.

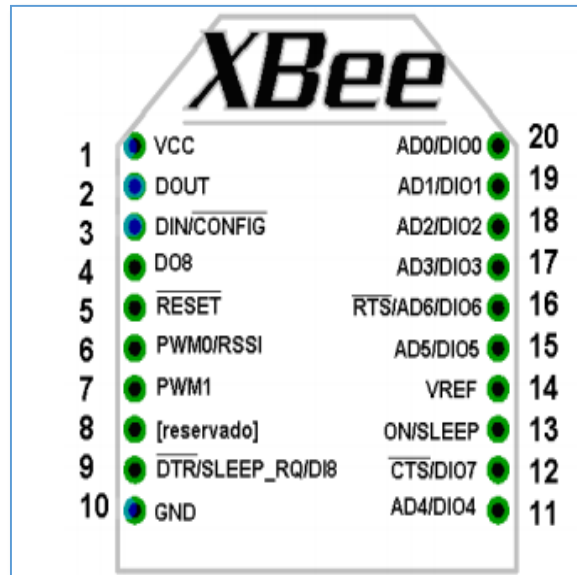


**Figura. 9:** Tipos de antenas XBee

Fuente:[http://1.bp.blogspot.com/-xi5iEYpaFGc/T521PZmYumI/AAAAAAAAApA/Zhxb\\_JBIHtc/s320/antenas.png](http://1.bp.blogspot.com/-xi5iEYpaFGc/T521PZmYumI/AAAAAAAAApA/Zhxb_JBIHtc/s320/antenas.png)

#### 4.4.5. Diagrama de pines de un módulo XBee.

El esquemático de los pines se muestra en la siguiente figura 10:



**Figura. 10:** Diagrama de pines del módulo XBee. (Vista Superior)

Fuente: [www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia\\_Usuario.pdf](http://www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia_Usuario.pdf)

#### 4.4.6. Modos de operación.

Los módulos XBee tienen 5 modos diferentes de operar:

##### 4.4.6.1. Modo recepción/transmisión.

El módulo XBee se encuentra en estos modos cuando el módulo recibe algún paquete RF a través de la antena de recepción (modo Receive) o cuando se envía información serial al buffer del pin 3 (UART Data in) que luego será transmitida (modo Transmit). [18]

La información transmitida puede ser Directa o Indirecta. En modo directo la información se envía inmediatamente a la dirección de destino. En el modo Indirecto la información es retenida por el módulo durante un periodo de tiempo y es enviada solo cuando la dirección de destino la solicita.



#### **4.4.6.2. Modo de bajo consumo.**

El modo de Bajo Consumo o modo Sleep, se da cuando el modulo entra en un estado de bajo consumo de energía. Esto depende de la configuración en la que se encuentra. Se debe indicar que el ahorro de energía depende enormemente del voltaje de alimentación. La configuración de los ciclos de sueño se realiza mediante comandos. Por defecto, los ciclos de sueño están deshabilitados; Mediante el pin de hibernación Sleep\_RQ (pin 9) de los dispositivos XBee en estado alto, el módulo termina cualquier transmisión, recepción o procedimiento de asociación y entra en modo reposo y luego en modo sueño. En este estado no responde a comandos entrantes de ningún tipo. Sólo cuando se baje el estado lógico de Sleep\_RQ el módulo saldrá del estado de sueño y podrá volver a enviar y recibir datos. [18] [20]

#### **4.4.6.3. Modo Comando.**

Este modo permite ingresar comandos AT al módulo XBee, para configurar, ajustar o modificar distintos parámetros. Permite ajustar valores como la dirección propia o la de destino, así como su modo de operación entre otras cosas. Para poder ingresar los comandos AT es necesario utilizar el Hyperterminal de Windows, el programa X-CTU o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria o los adquiera de alguna otra forma.

#### **4.4.6.4. Modo Transparente.**

En modo transparente todo lo que ingresa por el pin 3 (DIN) se guarda en el buffer de entrada y se transmite y aquello que ingresa como paquete RF se guarda en el buffer de salida y se envía por el pin 2 (DOUT) inmediatamente o tras el paso de un tiempo. Este modo se destina a comunicación punto a punto, donde no es necesario ningún tipo de control. El modo Transparente viene por defecto en los módulos XBee. [18]

#### **4.4.6.5. Modo API.**

Cuando se encuentra en este modo, el módulo empaqueta en tramas toda la información que entra y sale. Entre las opciones que permite la API, se definen:

- ✓ Transmitir información a múltiples destinatarios, sin entrar al modo de comandos.
- ✓ Recibir estado de éxito/fallo de cada paquete RF transmitido.
- ✓ Identificar la dirección de origen de cada paquete recibido. [20]

El modo API, es especialmente recomendado para el módulo que debe cumplir la función de coordinador.

## 5. MATERIALES Y MÉTODOS.

### 5.1. SIMULACIÓN DE LA RED INALÁMBRICA DE SENSORES.

#### 5.1.1. Elección del Simulador.

Existen cantidad de plataformas para simulación de redes en general y de WSNs en particular. En la elección del simulador primeramente se tomó en cuenta que el objetivo del presente trabajo es utilizar para la creación de la WSN el estándar IEEE 802.15.4, la última revisión de este estándar fue aprobada en el año 2009;<sup>2</sup> por ello los proveedores de software que lo incluyen como estándar dentro sus herramientas de simulación, son aun escasos; en la mayoría de simuladores de WSN los módulos que implementan IEEE 802.15.4/ ZigBee, son fruto de contribuciones externas y casi siempre están aún en fase de desarrollo, por lo que no se garantiza el correcto funcionamiento del estándar dentro de estos, de la misma forma la documentación y el soporte técnico de dichos módulos es escaso; como por ejemplo en el simulador NS3, cuyo módulo IEEE 802.15.4 para redes ZigBee aún se encuentra en desarrollo y para su uso es necesario la contribución de librerías y módulos desde NS2 que constituye una versión anterior de este simulador.<sup>3</sup>

En la Tabla 4; se presenta una comparación entre las herramientas de software que se consideraron más sobresalientes para la simulación de la Red Inalámbrica de Sensores bajo el estándar IEEE 802.15.4.

Entre las características consideradas para la elección del simulador tenemos:

- ✓ **Uso investigativo:** tanto OPNET como OMNET++ y NS2 son usados ampliamente en áreas académicas por lo que su uso en la investigación es Alto lo que indica que las tres herramientas han sido utilizadas en un gran número de referencias y proyectos científicos.

---

<sup>2</sup> [https://ceseand.net/es/download\\_file.cfm?file=18&area=49](https://ceseand.net/es/download_file.cfm?file=18&area=49)

<sup>3</sup> U. of California-Berkeley, "The ns-3 network simulator." <http://www.nsnam.org/>,2006.

✓ **Tipo de licencia:** Los simuladores OMNET++ y NS2 cuentan con licencia libre; al hablar de software libre no necesariamente nos referimos a una herramienta gratuita, sino a la posibilidad que ofrece a los usuarios para editarla, copiarla, ejecutarla, distribuirla, estudiarla y mejorarla. Desde la perspectiva de la Free Software Foundation y la GNU Operation System, los usuarios de programas libres tienen cuatro libertades esenciales:

- Ejecutar el programa para cualquier propósito,
- Estudiar cómo funciona el programa y adecuarlo a la forma que se desee,
- Distribuir el código de un usuario a otro y
- Distribuir el código modificado a otros.

El software libre debe contar con estas cuatro libertades; el hecho de que se cobre algún valor por un programa no inhibe que el programa pueda ser libre o utilizado según las mismas pautas. Por su parte la herramienta OPNET posee licencia comercial por ello se presentan ciertas restricciones para el usuario, su comercialización, costo, duración de uso, edición y libertad de permisos, son controlados y definidos por los propietarios de la herramienta; por ello este tipo de licencia también es conocida como licencia propietaria.

✓ **Curva de aprendizaje:** tanto OPNET como OMNET++ y NS2 poseen una curva de aprendizaje Alta, es decir para usar correctamente estas tres herramientas es muy importante tener conocimientos en redes y programación, puesto que el manejo de estas exige programación de los dispositivos por modificación de su código fuente.

✓ **Plataformas de soporte:** Las herramientas OPNET Y OMNET++ son compatibles y pueden correr sin ningún problema en sistemas operativos como

Windows y Unix. Por su parte NS2 es compatible en sistemas operativos tales como: Windows, Linux, Mac OS y Unix.

- ✓ **Interfaz gráfica:** OPNET posee una interfaz gráfica que le permite una Alta interacción con el usuario dado esto requiere un nivel de programación mínimo, ya que la herramienta tiene la disposición de trabajar desde todas sus perspectivas con una interfaz gráfica. El simulador OMNET ++ interactúa a nivel Medio con el usuario pues implementa una interfaz gráfica que facilita su uso, pero lo hace de forma limitada; algunas de sus implementaciones deben definirse mediante programación. Finalmente la interacción entre el usuario y NS2 es Baja, pues la herramienta no cuenta con interfaz gráfica, lo cual implica la programación de cada elemento dentro de una simulación para su ejecución final.
  
- ✓ **Tecnologías y protocolos de nivel 2 y 3 que soporta:** tanto OPNET como OMNET++ y NS2 soportan diversas tecnologías y protocolos de nivel 2 y 3 del Modelo OSI. Cabe mencionar que las tres herramientas son útiles para simular el estándar IEEE 802.15.4 que era nuestro requerimiento principal, poseen módulos propios con la arquitectura necesaria para que este estándar sea soportado y desplegado de manera correcta, con el fin de acercarse a implementaciones reales.
  
- ✓ **Documentación:** todos los simuladores cuentan con la documentación necesaria para trabajar correctamente con ellos sin embargo las herramientas OPNET y NS2 en sus páginas oficiales presentan foros y tutoriales que ayudan al usuario a tener una mejor experiencia con el simulador, de igual manera en la web existe Alto nivel de información sobre estas herramientas.
  
- ✓ **Grado de complejidad:** el simulador OMNET++ por ser un software de aplicación en áreas comerciales y para efectos de investigación y desarrollo, tiene un Alto grado de complejidad en su manejo. Por su parte tanto NS2 como OPNET son simuladores que por ser basados en el lenguaje C++ tienen un grado de complejidad Medio. [21] [22]

**Tabla 4:** Comparación entre los simuladores de WSN.

	<u>OPNET</u>	<u>OMNET++</u>	<u>NS2</u>
<b>Uso investigativo</b>	Alto	Alto	Alto
<b>Tipo de licencia</b>	Comercial	Libre	Libre
<b>Plataformas de soporte</b>	Windows Unix	Windows Unix	Linux Windows Mac, Unix
<b>Interfaz Gráfica</b>	Alta	Medio	Baja
<b>Curva de aprendizaje</b>	Alta	Alta	Alta
<b>Soporte de tecnologías de nivel dos y tres.</b>	Alto	Alto	Alto
<b>Tráfico que permite modelar</b>	Alto	Alto	Alto
<b>Documentación disponible</b>	Alta	Media	Alta
<b>Grado de complejidad</b>	Medio	Alto	Medio

Fuente: Elaboración propia.

Por lo analizado se toma la decisión que el simulador que será utilizado en el presente proyecto es Network Simulator versión 2 (NS2), principalmente porque soporta el estándar IEEE 802.15.4, es un software libre lo que nos proporciona mejores prestaciones como usuarios, y además es compatible con el Sistema Operativo Linux con el cual ya se ha venido trabajando en el transcurso de la materia de Sistemas operativos, considerando también que NS2 es el simulador más utilizado actualmente en ámbitos académicos y de investigación orientados WSN.

## **5.1.2. Simulador NS2.**

### **5.1.2.1. Generalidades.**

NS es un simulador de eventos discretos destinado a la creación de redes de investigación. NS presta un apoyo sustancial para la simulación de protocolos de enrutamiento y protocolos de multidifusión a través de redes cableadas e inalámbricas. Desarrollado inicialmente en los laboratorios de investigación de la universidad californiana de Berkeley (LBNL, Lawrence Berkeley National Laboratory) está concebido para el estudio del comportamiento de una red cualquiera de telecomunicaciones. [23]

NS empieza como una variante del REAL network simulator en 1989 y ha evolucionado substancialmente durante los últimos años. Actualmente el desarrollo de NS lo lleva a cabo DARPA junto a SAMAN y otros desarrolladores de software. NS ha sido desarrollado principalmente para dos diversas categorías de usuarios: aquellos interesados sencillamente en las simulaciones y a los usuarios interesados en el desarrollo de nuevos códigos y objetos. De aquí que trabaja con dos distinguidos lenguajes de programación: el OTcl y el C++. [24]

NS2 constituye una herramienta de simulación que nos permite definir y observar:

- ✓ Redes terrestres, inalámbricas y por satélite con varios algoritmos de enrutado (DV, LS, PIM-DM, PIM-SM, AODV, DSR).
- ✓ Multitud de protocolos de la capa de aplicación (http, ftp, cbr, etc.),

- ✓ Multitud de protocolos de la capa transporte (TCP, UDP, RTP, SRM),
- ✓ Mecanismos referentes a la capa de Enlace de Datos en Redes WAN, LAN Y PAN, tales como protocolos MAC (Control de Acceso al Medio) del tipo CSMA/CD, CSMA/CA etc.
- ✓ Distintos Algoritmos de tratamiento de colas FQ (Encolamiento Justo), SFQ (Encolamiento Estocástico Justo), DRR (Deficit Round Robin), FIFO (Primero en Entrar es Primero en Salir), etc. [25]

Y a su vez mediante NS2 podemos visualizar:

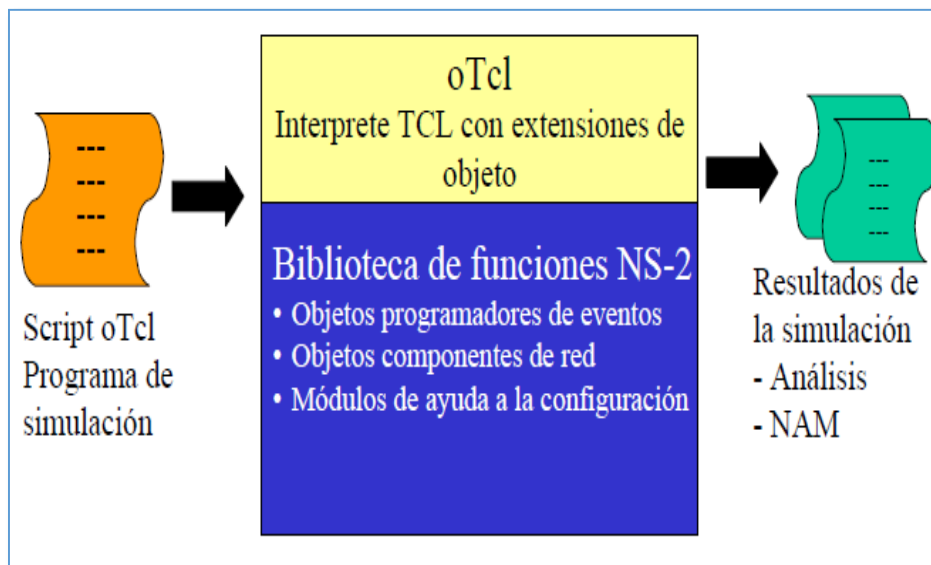
- ✓ Flujo del paquete, su encolado y su posible descarte.
- ✓ Fallos en el rendimiento de la red, fallos en la conexión, etc.
- ✓ Comportamientos del protocolo: comienzo lento de TCP, control de congestión, retransmisión rápida y recuperación.
- ✓ Movimiento de nodos en redes inalámbricas.
- ✓ Notas de los sucesos más importantes. [25]

Para esto cuenta con herramientas complementarias como NAM y XGRAPH, que nos permiten tanto la manipulación como el realizar gráficas de los datos obtenidos en la simulación para su posterior análisis.

#### **5.1.2.2. Arquitectura de NS2.**

Como se mencionó en la sección anterior el simulador NS2 está basado en los lenguajes de programación C++ y OTcl.





**Figura. 11:** Arquitectura de Ns2

Fuente: <http://downloads.tuxpuc.pucp.edu.pe/linuxweek2007/martes/LW2007-NS2.pdf>

Como se puede observar en la figura.11 el funcionamiento del simulador Ns2 comienza con un script en OTcl que viene a hacer lo que el usuario codifica para simular, es el único INPUT que da el usuario al programa.

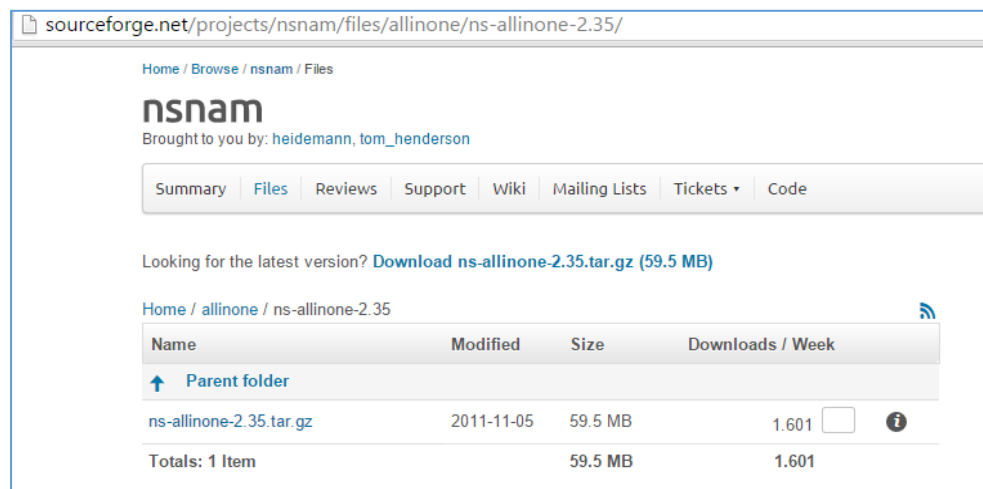
El resto es el procesamiento interno de NS. La simulación arroja como resultado un archivo de trazas que puede ser bastante incomodo de leer o analizar para el usuario, sin embargo, usando una aplicación especial o cualquier herramienta complementaria del simulador como NAM, GPLOT, XGRAPH, TRACERGRAPH etc. se puede mostrar mediante una interfaz gráfica.

El script programado es un archivo escrito en Tcl orientado a objetos, es decir, OTcl, que tiene diversos componentes internos que se muestran en la figura 11. En estos componentes se configura la topología de la red, se carga las funciones necesarias para la simulación, se planifica cuando iniciar o terminar el tráfico de un determinado paquete, entre otros aspectos de la red, y módulos de ayuda de la configuración. [24] [26]

### 5.1.2.3. Instalación NS2.

Debido a que el simulador de redes NS2 es compatible con cualquier sistema operativo linux, se decidió realizar la instalación del mismo sobre Ubuntu-14.04 LTS en su versión de 64 bits, el procedimiento para esto se describe a continuación:

1. Se descarga el archivo fuente de NS2 llamado “ns-allinone-2.35.tar.gz” desde la página oficial <http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/> como lo muestra la figura 12.



**Figura. 12:** Página oficial del simulador NS2.

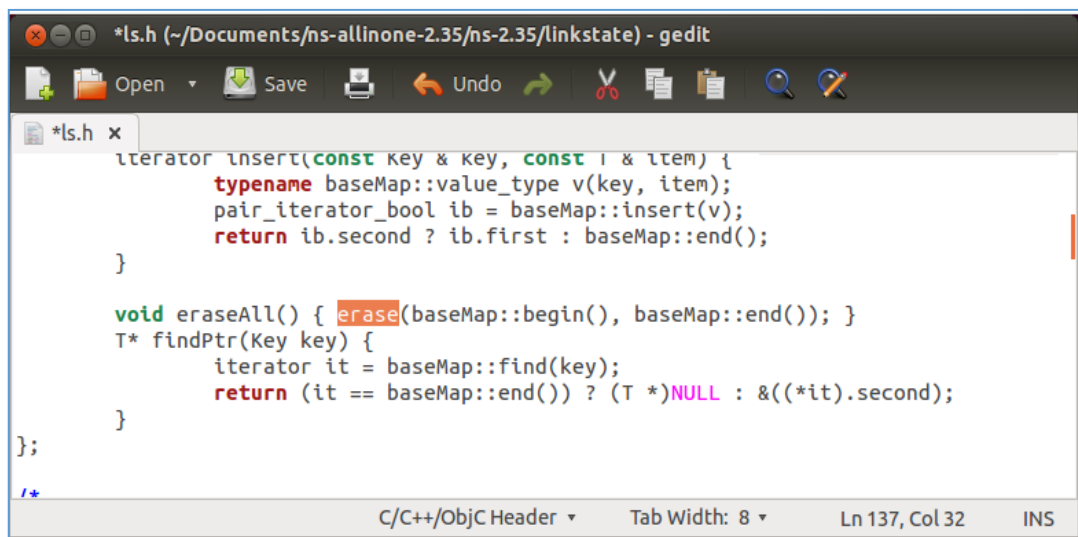
Fuente: Elaboración propia

2. Actualizar Ubuntu a través de la consola, para ello se abre un terminal y se ejecutan los siguientes comandos:
  - sudo apt-get update
  - sudo apt-get dist-upgrade
  - sudo apt-get update
3. Instalar los paquetes esenciales requeridos por el simulador NS2, para ello se ejecuta los siguientes comandos:
  - sudo apt-get install automake build-essential autoconf
  - sudo apt-get install tcl8.5-dev tk8.5-dev
  - sudo apt-get install perl xgraph libXt-dev libx11-dev libxmu-dev

4. Extraer el paquete NS descargado ya sea mediante el menú contextual del botón derecho "Extraer aquí" o el uso de los siguientes comandos en un terminal:
  - `cd / home / marys/ Documentos`
  - `tar -xvzf ns-allinone-2.35.tar.gz`
5. Después de extraer la carpeta "ns-allinone-2.35", se debe abrir el archivo "/ns-allinone-2.35/ns-2.35/linkstate/ls.h" en un editor de texto, esto se puede hacer desde el terminal o desde el explorador de archivos (Nautilus); se debe realizar algunos cambios en el archivo ls.h de lo contrario se mostrará un error durante la instalación del simulador.

Una vez abierto el archivo en la línea 137 del código se debe sustituir **erase** como lo señala la Figura 13 con **this-> erase** tal como lo muestra la Figura 14 y guardamos los cambios.

6. Se finaliza la instalación de NS2 a través de los siguientes comandos:
  - Ir a `cd /home/marys/Documentos/ns-allinone-2.35`
  - `sudo ./install`




```
*ls.h (~/Documents/ns-allinone-2.35/ns-2.35/linkstate) - gedit
Open Save Undo
*ls.h x
iterator insert(const key & key, const i & item) {
    typename baseMap::value_type v(key, item);
    pair_iterator_bool ib = baseMap::insert(v);
    return ib.second ? ib.first : baseMap::end();
}

void eraseAll() { erase(baseMap::begin(), baseMap::end()); }
T* findPtr(Key key) {
    iterator it = baseMap::find(key);
    return (it == baseMap::end()) ? (T *)NULL : &((*it).second);
}
};
/*
C/C++/ObjC Header Tab Width: 8 Ln 137, Col 32 INS
```

**Figura. 13:** Línea Archivo ls.h que debe modificarse

**Fuente:** Elaboración propia



```
*ls.h x
typedef pair<iterator, bool> pair_iterator_bool;
iterator insert(const Key & key, const T & item) {
    typename baseMap::value_type v(key, item);
    pair_iterator_bool ib = baseMap::insert(v);
    return ib.second ? ib.first : baseMap::end();
}

void eraseAll() { this->erase(baseMap::begin(), baseMap::end()); }
T* findPtr(Key key) {
    iterator it = baseMap::find(key);
    return (it == baseMap::end()) ? (T *)NULL : &((*it).second);
}
};

C/C++/ObjC Header Tab Width: 8 Ln 137, Col 38 INS
```

**Figura. 14:** Línea Archivo ls.h modificada.

Fuente: Elaboración propia

7. Una vez instalado NS2 es necesario agregar algunas variables de entorno a su perfil. Esto se puede hacer mediante la edición del archivo .bashrc. para ello:
  - Desde un nuevo terminal abrir el archivo mediante el comando `sudo gedit .bashrc`
  - Agregar las siguientes líneas al final del archivo:

```
# LD_LIBRARY_PATH
OTCL_LIB = / home/Documentos/marys /ns-allinone-2.35/otcl-1.14/
NS2_LIB = / home/Documentos/marys /ns-allinone-2.35/lib/
USR_Local_LIB = / usr / local / lib /
export LD_LIBRARY_PATH = $ LD_LIBRARY_PATH: $ OTCL_LIB: $
NS2_LIB: $ USR_Local_LIB
# TCL_LIBRARY
TCL_LIB = / home/Documentos/marys /ns-allinone-2.35/tcl8.5.10/library/
USR_LIB = / usr / lib /
export TCL_LIBRARY = $ TCL_LIBRARY: $ TCL_LIB: $ USR_LIB
# CAMINO
```

```
XGRAPH= / home/Documentos/marys /ns-allinone-2.35/xgraph-12.2/: /
home/Documentos/marys /ns-allinone-2.35/bin/:/ home/Documentos/marys
/ns-allinone-2.35/tcl8.5.10/unix/: / home/Documentos/marys /ns-allinone-
2.35/tk8.5.10/unix/
NS = / home/Documentos/marys /ns-allinone-2.35/ns-2.35/
NAM = / home/Documentos/marys /ns-allinone-2.35/nam-1.15/
export PATH = $ PATH: $ xgraph: $ NS: $ NAM
```

- Guardamos el archivo y reiniciamos el sistema ejecutando el comando:  
source ~/ .bashrc

8. Finalmente antes de ejecutar el simulador es necesario validar la instalación de NS2 para ello:

- Abrir un terminal y situarnos en directorio "/ home /Documentos/ns-allinone-2.35/ns-2.35/
- Ejecutar el comando ./validate

9. Ejecutar NS2.

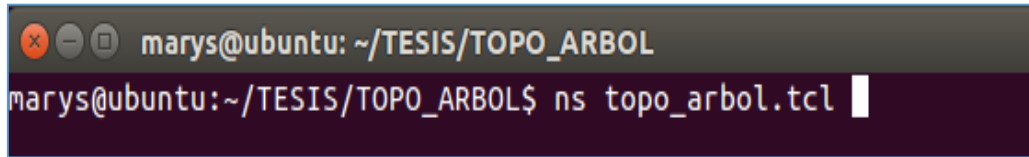
Existen dos maneras distintas de ejecutar el simulador NS2, la primera en modo Argumento que consiste en ejecutar el comando ns seguido del nombre del archivo Otcl a ejecutar y la segunda ejecutando sólo el comando ns y entrando en el modo interactivo a continuación especificamos ambos modos:

- ✓ **Con argumentos:** se entra a este modo mediante el comando:

```
# marys@ubuntu:~/TESIS/TOPO_ARBOL$ ns topo_arbol.tcl
```

Mediante este comando NS-2 abre y ejecuta el archivo que se pasa como argumento como se muestra en la figura 15, al acabar la simulación vuelve a la línea de comandos.

El archivo pasado como argumento tiene que tener extensión .tcl y en él deben estar todas las instrucciones necesarias para realizar la simulación.



```
marys@ubuntu: ~/TESIS/TOPO_ARBOL
marys@ubuntu:~/TESIS/TOPO_ARBOL$ ns topo_arbol.tcl
```

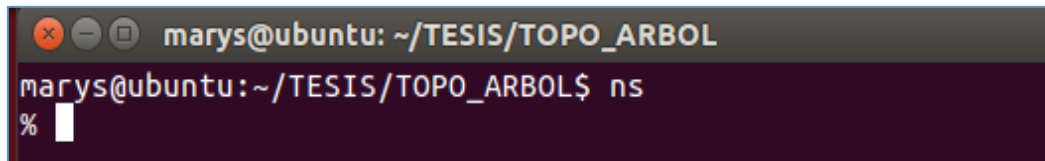
**Figura. 15:** Ejecución de NS2 sin argumentos

**Fuente:** Elaboración propia

- ✓ **Sin argumentos:** se entra a este modo mediante el comando:

```
# marys@ubuntu:~/TESIS/TOPO_ARBOL$ ns
```

En modo interactivo o sin argumentos el simulador NS-2 se comporta como cualquier intérprete de comandos, es decir, analiza y ejecuta cada instrucción una a una, al entrar en este modo parece el símbolo % en la consola de comandos tal como lo muestra la Figura 16, para salir de este modo ejecutamos exit.



```
marys@ubuntu: ~/TESIS/TOPO_ARBOL
marys@ubuntu:~/TESIS/TOPO_ARBOL$ ns
%
```

**Figura. 16:** Ejecución de NS2 con argumentos

**Fuente:** Elaboración propia

Para las simulaciones de la tesis se utilizó NS2 bajo el modo con argumentos creándose para cada topología de red un archivo con extensión .tcl, tal como se muestra en la figuras 17 y 18.

```

*topo_arbol.tcl (~:/TESIS/TOPO_ARBOL) - gedit
# =====
#                                UNIVERSIDAD NACIONAL DE LOJA
#                                A.E.I.R.N.N.R
#                                ING. EN ELECTRONICA Y TELECOMUNICACIONES
#                                RED DE SENSORES INALAMBRICA
# Simulacion Topologia Arbol
# =====
set val(chan)      Channel/WirelessChannel      ;# Tipo de Canal
set val(prop)      Propagation/TwoRayGround     ;# Modelo de Radiopropagacion
set val(netif)     Phy/WirelessPhy/802_15_4
set val(mac)       Mac/802_15_4
set val(ifq)       Queue/DropTail/PriQueue     ;# Tipo de cola
set val(ll)        LL                           ;# Tipo de capa de enlace
set val(ant)       Antenna/OmniAntenna         ;# Modelo de antena
set val(ifqlen)    50                           ;# tamaño del paquete en cola
set val(nn)        11                           ;# numero de nodos
set val(rp)        AODV                         ;# protocolo de enrutamiento
set val(x)         500
set val(y)         500
set val(nam)       arbol.nam
set val(traffic)   cbr                          ;# cbr

```

**Figura. 17:** Archivo de configuración para la topología en árbol (.tcl).

Fuente: Elaboración propia

```

*estrella.tcl (~:/TESIS/TOPO_ESTRELLA) - gedit
# =====
#                                UNIVERSIDAD NACIONAL DE LOJA
#                                A.E.I.R.N.N.R
#                                ING. EN ELECTRONICA Y TELECOMUNICACIONES
#                                RED DE SENSORES INLAMBRICA
# TOPOLOGIA ESTRELLA
# =====
set val(chan)      Channel/WirelessChannel      ;# TIPO DE CANAL
set val(prop)      Propagation/TwoRayGround     ;# Modelo de Radiopropagacion
set val(netif)     Phy/WirelessPhy/802_15_4
set val(mac)       Mac/802_15_4
set val(ifq)       Queue/DropTail/PriQueue     ;# Tipo de Cola
set val(ll)        LL                           ;# Tipo de Capa de enlace
set val(ant)       Antenna/OmniAntenna         ;# Modelo de Antena
set val(ifqlen)    150                          ;# Tamaño del paquete en Cola
set val(nn)        7                            ;# Numero de Nodos
set val(rp)        AODV                         ;# Protocolo capa 3
set val(x)         500
set val(y)         500
set val(nam)       estrella.nam                 ;# Archivo nam
set val(traffic)   cbr                          ;# Tipo de trafico

```

**Figura. 18:** Archivo de configuración para la topología en estrella (.tcl).

Fuente: Elaboración propia

#### 5.1.2.4. Herramientas Complementarias de NS2.

Como se mencionó en la sección 5.1.2.2 los resultados que se obtienen del simulador se almacenan en archivos de trazas con extensión .tr, estos son complejos de entender, debido a que se presentan como archivos de gran tamaño y multitud de datos, por ello se precisa de herramientas de software complementarias que nos permiten entender e interpretar los resultados obtenidos, algunos son de representación gráfica, como NAM (network animator), TraceGraph, Xgraph, etc y otros son programas basados en códigos como AWK. En este proyecto se han usado tanto NAM para visualizar el comportamiento de la red como filtros AWK para entender y analizar los resultados.

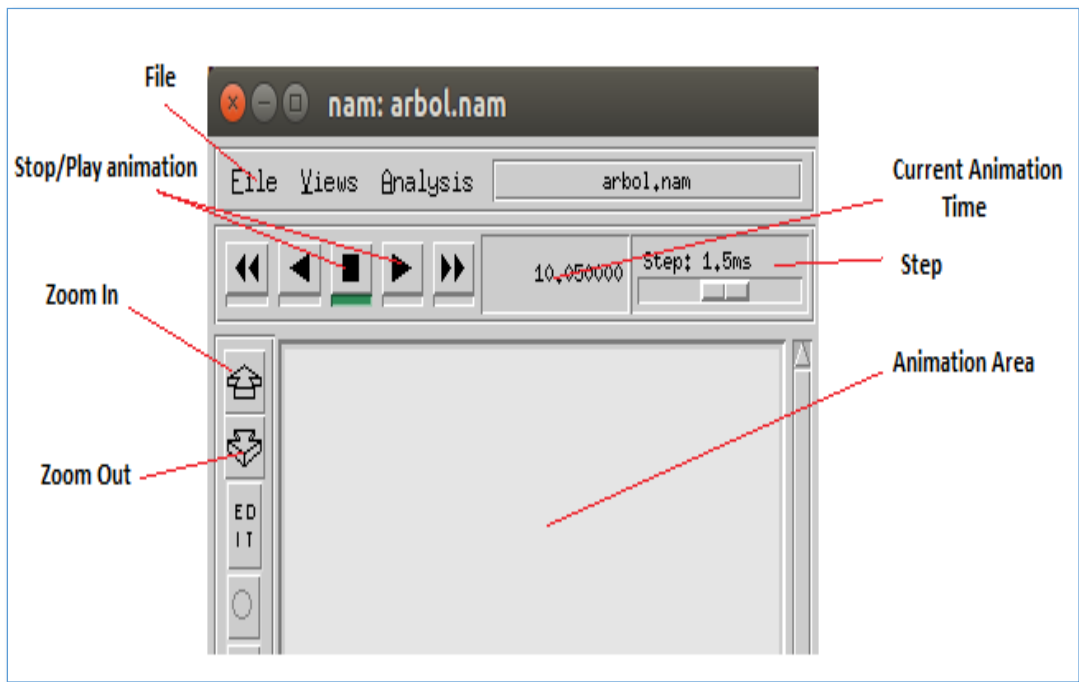
#### ✓ NAM.

NAM (Network Animador) es una herramienta de software externa al simulador NS-2, pero que viene incluida en el paquete de instalación del simulador. Esta aplicación está basada en el lenguaje OTcl y nos permite visualizar de forma gráfica los escenarios de simulación de las distintas topologías de red. [26]

La figura 19 muestra la pantalla de visualización de la herramienta NAM algunos de sus componentes son:

- **Animation Area:** aquí se puede visualizar la topología de la red especificada en el simulador.
- **Zoom In /Zoom Out:** permite aumentar o disminuir el área de simulación.
- **Stop/Play Animation:** inicia la simulación o la detiene.
- **Current Animation Time:** muestra el tiempo transcurrido desde el inicio de la simulación. El tiempo máximo se halla determinado al inicio de la simulación.
- **Step:** indica la velocidad con la que transcurre la simulación. Está marcado en milisegundos.
- **File:** permite grabar la simulación, imprimir el área de simulación, etc. [26]





**Figura. 19:** Herramienta NAM.

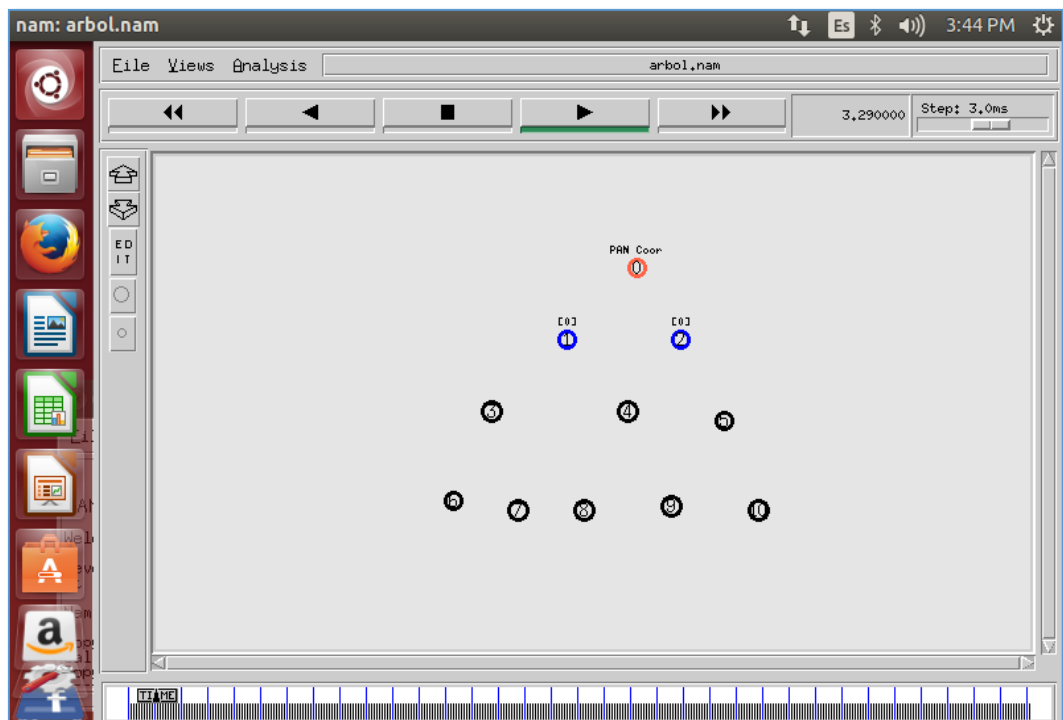
Fuente: Elaboración propia

Todos los datos que se pueden visualizar mediante la herramienta NAM se procesan y almacenan en un fichero con extensión \*.nam al momento en que se corre la simulación; para abrir el archivo que se genera al finalizar la simulación se utiliza el siguiente comando que se ejecuta desde la consola:

```
# marys@ubuntu:~/TESIS/TOPO_ARBOL$ nam arbol.nam
```

Es importante detallar que la herramienta NAM no es imprescindible dentro de una simulación debido a que no aporta ningún tipo de datos o información con los cuales se pueda extraer directamente conclusiones sobre el funcionamiento de una red; pero es importante el uso de este programa, sobre todo al principio, para entender que es lo que está sucediendo en la simulación y posteriormente tratar los resultados obtenidos.

Al ejecutar el comando se obtiene una pantalla de visualización similar a la que muestra la figura 20.



**Figura. 20:** Interfaz gráfica de la herramienta NAM

**Fuente:** Elaboración propia

Con la herramienta NAM es posible visualizar los siguientes elementos que intervienen en la simulación:

- Nodos con su respectiva numeración
- Enlaces entre los nodos y rangos de cobertura
- Flujo de información que cruza entre cada nodo de la red
- Transmisión- Recepción y pérdida de paquetes.

✓ **Filtros AWK.**

Los filtros AWK son un programa de GNU constituyen una herramienta muy útil para el tratamiento de archivos de gran tamaño, buscar y transformar datos y, en general, realizar

cualquier tipo de tratamiento masivo de ficheros. Con un programa AWK es posible contar el número de líneas de un archivo, seleccionar columnas, realizar cruces, borrar campos de cada línea para analizar específicamente solo los que son de nuestro interés, hacer sumalizaciones, comprobar duplicados, muestreos, etc. Dentro de la investigación se usa la herramienta AWK para analizar el archivo de trazas (.tr) que se genera como resultado de la simulación en NS2, mediante AWK podemos operar sobre este archivo para mostrar o filtrar de la traza de datos solo la información deseada para una mejor comprensión. [27]

### **5.1.3. Fichero de Simulación.**

Para realizar la simulación de una Red Inalámbrica con NS-2 es necesario crear un archivo .tcl en el que se definen todas las características y parámetros propios de la red; los archivos que se crearon para la simulación de este trabajo presentan la siguiente estructura:

- Definición de las variables globales.
- Definición de los ficheros de trazas (NS-2 y NAM).
- Definición de las características de la red Inalámbrica.
- Configuración de los nodos.
- Creación de los nodos.
- Finalización del simulador. [28]

#### **5.1.3.1. Definición de variables Globales.**

Para empezar lo primero que se realiza es crear una nueva instancia de simulación, para esto se define una variable mediante el comando set seguido del nombre que se le dará a

la variable y el valor inicial. Si se trata de definir un objeto de una clase hay que usar el método new. Entonces para crear una nueva simulación se debe introducir, por ejemplo:

```
set ns_ [new Simulator]
```

Una vez creada la nueva instancia de simulación, cualquier procedimiento referido a esta se realizará empezando con \$ns\_. Posteriormente se crea el espacio en el que se desarrolla la simulación, esto se consigue adquiriendo una instancia a la clase Topography mediante el siguiente comando:

```
set topo [new Topography]
```

Como en el caso anterior, una vez definida la topología cualquier procedimiento que se refiera a ella debe empezar con \$topo.

Asignamos a esta topología un plano x-y en donde se ejecutaran los eventos; mediante esta línea definimos el tamaño del escenario donde se desarrolla la simulación, en el caso particular de la presente investigación en esta línea se coloca las 5 Ha que corresponden a la extensión del Jardín Botánico Reinaldo Espinosa esto se realiza mediante el siguiente comando.

```
$topo load_flatgrid $val(x) $val(y)
```

Se crea el objeto God encargado de gestionar los enlaces inalámbricos entre nodos, en él se indica el número de nodos a trabajar dentro de la red inalámbrica.

```
set god_ [create-god $val(nn)]
```

Un paso importante que se debe realizar es activar el nuevo formato de trazas disponible en el simulador NS2 para ello usamos la siguiente instrucción:

```
$ns_ set WirelessNewTrace_ ON
```

Es indispensable activar esta configuración y poner el archivo de trazas en un formato adecuado para poder aplicar los filtros .awk que permiten la recolección de las muestras.

### 5.1.3.2. Definición de los ficheros de trazas (NS-2 y NAM).

Una vez creadas las variables globales, el siguiente paso consiste en definir los archivos donde se almacenan los resultados de la simulación, tales como los archivos de trazas de NS-2 (.tr) y los de NAM (.nam). Para definirlos se usan las siguientes instrucciones:

```
set tracefd [open ./topo_estrella.tr w]
$ns_ trace-all $tracefd
$ns_ use-newtrace
set namtrace [open estrella.nam w]
```

Estas instrucciones que se muestran como ejemplo generan dos archivos al terminar la simulación: topo\_estrella.tr y estrella.nam, uno para analizarlo mediante filtros AWK y el otro para que lo reproduzca la herramienta grafica NAM.

Es importante destacar que se ha incluido la instrucción necesaria para usar el nuevo formato de trazas de NS-2, si no se introduce esta instrucción se usaría el formato de trazas antiguo, que es el que viene por defecto en el simulador; el nuevo formato contiene mayor información sobre la red y por lo tanto es preferible usarlo, y es el que se ha utilizado en la presente investigación. [26] [28]

### 5.1.3.3. Definición de las características de la red Inalámbrica.

Son las variables o parámetros de configuración que se crean para caracterizar la simulación de acuerdo a nuestros requerimientos; así se tiene:

```
# Definición de los Parámetros de Simulación.
set val(chan) Channel/WirelessChannel ;# Tipo de canal
set val(netif) Phy/WirelessPhy/802_15_4 ;# Interfaz Wireless de comunicación
set val(mac) Mac/802_15_4 ;#Definición del protocolo de capa MAC
set val(ll) LL ;# Tipo de capa de enlace
set val(nn) 11 ;# Numero de nodos
```

```
set val(x) 225 ;#Área de cobertura
set val(y) 225
set val(ant) Antenna/OmniAntenna ;# Modelo de antena
set val(prop) Propagation/TwoRayGround ;# Modelo de radio Propagación
set val(rp) AODV ;# Protocolo de enrutamiento
set val(traffic) cbr ;# Trafico UDP/CBR
```

✓ **Canal de comunicación.**

La primera instrucción nos determina el tipo de canal de comunicación; en el trabajo de investigación se realiza la simulación de una Red inalámbrica de Sensores (WSN) por tal motivo se define un canal inalámbrico de comunicación.

✓ **Protocolo de comunicación.**

Se define de igual forma el protocolo WPAN IEEE 802.15.4 que es el que se está usando para el desarrollo de la Red de sensores y se realizó una simulación con 11 nodos comunicados inalámbricamente mediante este protocolo.

✓ **Tipo de tráfico.**

La simulación se realizó mediante tráfico CBR (Constant Byte Rate), que nos permite una transmisión de una sola vía y el tamaño del paquete e intervalo entre paquetes se encuentra definido, este parámetro es fundamental en el desarrollo de la simulación pues se debe simular un tráfico lo más cercano a la realidad en la que se transmite de esta forma se obtiene una correcta perspectiva del funcionamiento de la red real, por ello se usa CBR que funciona sobre UDP.

CBR envía bits a una velocidad constante sin variaciones de ningún tipo, llegando a explotar la conexión y que se asemeja al tipo de tráfico generado por los datos que se van a recoger en el prototipo de WSN a implementar.

✓ **Tiempo de simulación.**

Este parámetro permite ver la evolución de la red desde el momento inicial en el que los nodos buscan establecer conexión con el dispositivo coordinador, hasta su comportamiento en el envío aleatorio de datos que se producen durante la simulación. Se simuló durante 100 segundos, debido a que en la escala de tiempo del simulador 1s en la realidad, representa solamente 2ms dentro de la simulación, por lo que los 100s fueron suficientes para recoger los datos que nos permitieron hacer un análisis del rendimiento de la red. [28] [29]

✓ **Modelo de Antena.**

Al simular una red WSN, para alcanzar nodos finales y mantener la conectividad, la cobertura es indispensable, por ello se define para la cobertura las antenas de los módulos XBEE PRO S2B que son los módulos XBee que se utilizaran en el diseño de la Red Inalámbrica de Sensores; estas son antenas omnidireccionales que nos permiten transmitir a todas las regiones con la misma intensidad y presentan un rango de cobertura teórico por nodo de 1600 metros.

✓ **Modelo de radio propagación.**

Ns2 permite la simulación de dos tipos de modelos de radio propagación el modelo a dos rayos y el modelo de espacio libre; en el presente trabajo se elige el modelo de dos rayos (Two-Ray Ground), esta es una versión mejorada del modelo de espacio libre (Free Space), que considera no solo el camino directo de las ondas de antena a antena, sino también el efecto de la reflexión de esas ondas en el suelo. [28] [29]

En este tipo de modelo para el cálculo de la potencia de la señal recibida a una distancia  $d$ , se tiene la ecuación 1.

$$P_r = \frac{P_t \times G_t \times G_r \times h_r^2 \times h_t^2}{d^4} \quad \text{Ecuación 1}$$

**Tabla 5:** Características de la simulación.

Característica.	Definición	Valor
Distancia máxima a cubrir en el jardín botánico Reinaldo Espinoza que es el escenario de monitorización.	d	225m
Ganancia de la antena de recepción (ganancia en dBi de la antena utilizada en el diseño del nodo sensor)	$G_t$	5dBi
Ganancia de la antena de transmisión. (ganancia en dBi de la antena utilizada en el diseño del nodo coordinador)	$G_r$	5dBi
Altura de la antena de transmisión	hr	1m
Altura de la antena de recepción	ht	1m
Potencia de transmisión correspondiente al dispositivo de comunicación inalámbrica XBee Pro S2B usado en diseño de la WSN.	$P_t$	63mW 18dBm

Fuente: Elaboración propia.

Al remplazar los valores de la tabla 5 en la ecuación 1 se obtuvo el siguiente valor de potencia de recepción bajo el modelo de propagación Two Ray Ground el cual se define en el código del simulador para cada nodo:

$$P_r = \frac{P_t \times G_t \times G_r \times hr^2 \times ht^2}{d^4}$$

$$P_r = \frac{63\text{mW} \times 5\text{dBi} \times 5\text{dBi} \times (1\text{m})^2 \times (1\text{m})^2}{(225\text{m})^4}$$

$$P_r = 6.145 \times 10^{-10} \text{ W} = -63.71 \text{ dBm.}$$

Podemos observar que la potencia de recepción de los módulos de comunicación inalámbrica XBEE PRO S2B a 225m es adecuada a la sensibilidad de recepción de estos dispositivos. (-102 dBm).



✓ **Protocolo de Enrutamiento.**

Para el protocolo de enrutamiento dentro del simulador NS2 se dispone de diversos protocolos tales como DSDV, DSR, TORA y AODV que para el caso de estudio el que mejor se ajusta de acuerdo a la red de sensores inalámbricos es el protocolo AODV el cual adiciona las características de descubrimiento y mantenimiento de rutas del protocolo DSR y el enrutado salto a salto, los números de secuencia y los beacons de DSDV.

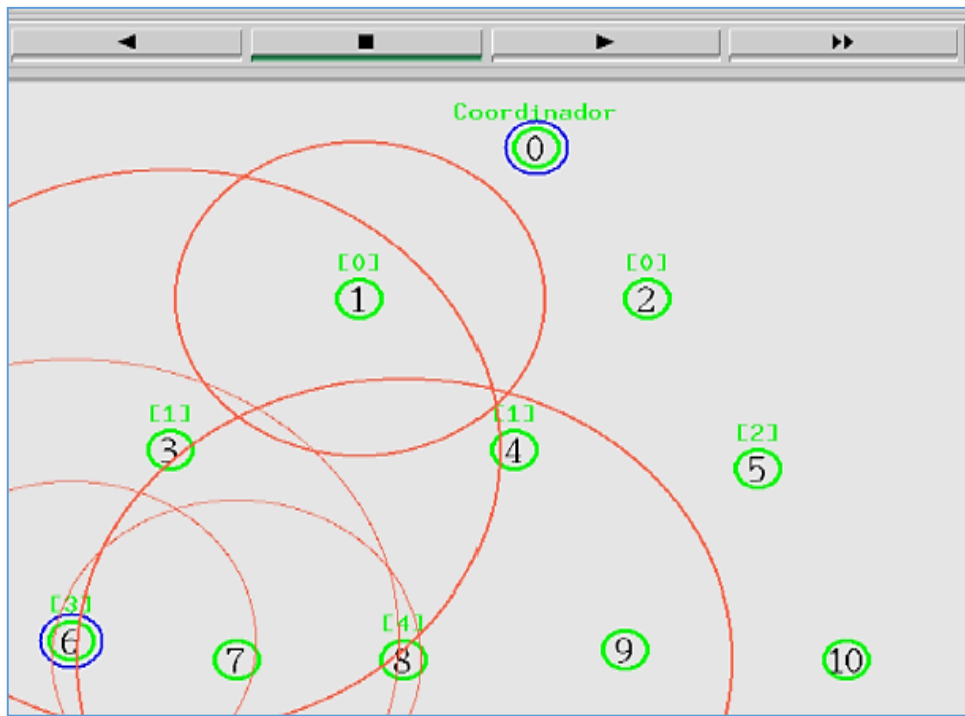
✓ **Tamaño del escenario.**

El tamaño del escenario de simulación se relaciona directamente con la cantidad de nodos que se posee en la red como se puede ver en las figuras 21 y 22; tanto el nodo configurado como coordinador como los diez nodos restantes se organizaron de distinta manera según sea la topología en árbol o en estrella; para el desarrollo de ambas se seleccionó un escenario de 225m por 225m que corresponde a la extensión del jardín botánico Reinaldo Espinosa que constituye nuestro ambiente de monitorización.



**Figura. 21:** Escenario de simulación de la topología en estrella.

Fuente: Elaboración propia



**Figura. 22:** Escenario de simulación de la topología en árbol.

Fuente: Elaboración propia

#### 5.1.3.4. Configuración de los Nodos.

Se procede a definir y configurar las características inalámbricas de los nodos creados llamando los parámetros ya anteriormente configurados mediante el siguiente esquema:

```
# Configuración del nodo inalámbrico
$ns_ node-config -adhocRouting $val(rp) \
llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-topoInstance $topo \
```

Dentro de la configuración de los nodos inalámbricos es importante especificar los eventos que generaran las trazas, de ruteo, de acceso al medio y de agente, así como el movimiento de los nodos; al tratarse de una simulación de WSN para monitoreo de

variables ambientales no se necesita activar el movimiento en los nodos por ello en la configuración se definen nodos estáticos tal como lo muestran las siguientes líneas:

```
-agentTrace ON \  
-routerTrace ON \  
-macTrace ON \  
-movementTrace OFF \
```

Un parámetro importante a considerar es la energía de los nodos para ello se inicia un modelo del escenario con los parámetros que se desean medir.

```
-energyModel "EnergyModel" \  
-initialEnergy 56160\  
-rxPower $potenciaRx \  
-txPower $potenciaTx \  
-sleepPower \  
-channel $chan_1_
```

El parámetro \$potenciaRx corresponde a la potencia de recepción calculada mediante la ecuación 1. El parámetro \$potenciaTx es la potencia de transmisión de los módulos de comunicación inalámbrica XBEE PRO S2B (dada por el fabricante).

Para realizar el cálculo de la energía inicial del nodo WSN sabemos que la unidad para medir la energía acumulada en una batería es el Julio; sin embargo, para fines prácticos, y dado que el voltaje de la batería del nodo es fijo, se utiliza como unidad de carga el [Ah], haciendo referencia al tiempo de carga y descarga de la batería.

La energía máxima que se acumula en una batería viene dada por la cantidad de [Ah] de la batería, multiplicada por 3600 y por el voltaje según la ecuación 2. En este trabajo se simuló con las características de una batería de 1.3 Ah de capacidad de carga, que proporciona un voltaje de 12 V, eso quiere decir que puede acumular 56160 Julios según la ecuación 2.

$$Ah \times 3600 \times V = \text{Julios} \quad \text{Ecuación 2}$$

$$1.3 \text{ Ah} \times 3600 \times 12V = 56160 \text{ J}$$

La configuración de la posición inicial de los nodos se realiza manualmente; situándolos alrededor del escenario de simulación esto mediante las coordenadas cartesianas (x, y, z) que va a ocupar cada nodo. Para ello se usa las siguientes líneas de código:

```
$node_(0) set X_ 20
$node_(0) set Y_ 20
$node_(0) set Z_ 0
$node_(1) set X_ 20
$node_(1) set Y_ 15
$node_(1) set Z_ 0
```

#### 5.1.3.5. Creación de los nodos.

Una vez que se han definido y configurado las características de los once nodos inalámbricos que componen la simulación de la red, éstos pueden ser creados, automáticamente utilizando un bucle “for”, ya que todos los nodos tendrán las mismas características definidas anteriormente, esto lo realizamos mediante las siguientes líneas:

```
For {set I 0} {$i < $val(nn) } {incr i} {
set node_($i) [$ns_ node]
}
```

#### 5.1.3.6. Tráfico entre los nodos.

Los nodos por si solos no generan ni reciben tráfico, para hacerlo hay que crear agentes de tráfico y una vez creados se asocian a los distintos nodos, de esta forma se establece el flujo de tráfico que existirá entre los once nodos según la topología a simular, esto dentro de NS2 se realiza a través de las siguientes líneas:

```
$val(traffic)traffic 9 0 0.6 $appTime9
$val(traffic)traffic 10 0 0.6 $appTime10
$ns_ at $appTime9 “$ns_ trace-annotate \”(at $appTime1) $val(traffic) tráfico
desde nodo 9 al nodo 0\””
$ns_ at $appTime10 “$ns_ trace-annotate \”(at $appTime2) $val(traffic) tráfico
desde nodo 10 al nodo 0\””
```

### 5.1.3.7. Finalización del simulador.

Se detiene la simulación llamando al proceso:

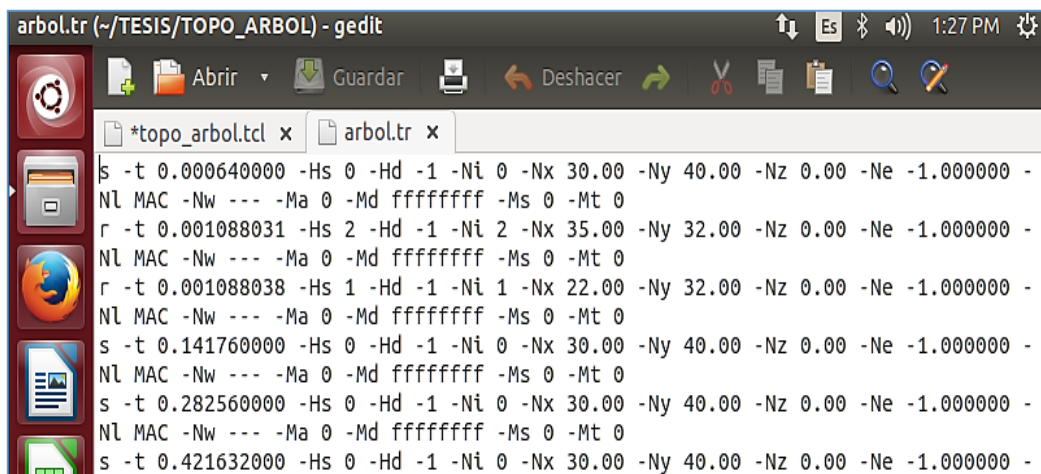
```
$ns_ at $stopTime "stop"  
$ns_ at $stopTime "puts \"NS Saliendo...\\n\""
```

En el proceso de finalización se cierran los archivos y se termina las trazas de datos, como se muestra a continuación:

```
proc stop {} {  
    global ns_ tracefd appTime1 val env  
    $ns_ flush-trace  
    close $tracefd  
}
```

### 5.1.4. Archivo de trazas.

Como se había mencionado anteriormente el resultado de la simulación es un archivo de trazas (.tr) que contiene gran cantidad de información como se observa en la figura 23.



**Figura. 23:** Archivo de trazas del simulador NS2

Fuente: Elaboración propia

Gracias a este archivo se puede examinar la eficiencia en la Red, adaptabilidad, tiempos de transmisión, número de ruta, tipo de tráfico, energía consumida, etc. Este tipo de archivos son difíciles de analizar a simple vista por ello es necesario conocer la forma en que se estructuran. Cada fila del archivo de trazas del simulador NS2 contiene un evento diferente, que puede ser un paquete enviado, un paquete recibido, un paquete perdido, etc. Dentro de cada fila hay una gran cantidad de campos separados por espacios en blanco, el número de campos puede ser distinto en función del tipo de evento. [29]

En general, los primeros campos de cada evento son los mismos y son los últimos los que más difieren. Algo muy importante como se mencionó anteriormente es que NS-2 ofrece distintos formatos de trazas. Por defecto dentro de la configuración del simulador viene un formato de traza antiguo con menos cantidad de información de la Red que se está simulando, pero para las simulaciones del presente trabajo se ha usado el nuevo formato de traza que se muestra en la figura 24 así como la explicación de cada campo de la trama.

1	2	3	4	5	6	7	8	9					
10	11	12	13	14	15	16	17	18	19	20	21	22	23
24	25	26											
s -t 10.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 370.37 -Ny 325.73 -Nz 0.00 -Ne -1.000000 -NI AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.0 -Id 1.0 -It cbr -Il 1500 -If 0 -Ii 0 -Iv 32 -Pn cbr -Pi 0 -Pf 0 -Po 2													

Campo	Marca	Descripción
1	-	Indica el tipo de evento. Puede tener cinco valores distintos: <ul style="list-style-type: none"> <li>• s → send, indica que se trata de un paquete enviado</li> <li>• r → receive, es un paquete recibido</li> <li>• f → forward, es un paquete reenviado por un nodo intermedio</li> <li>• d → drop, se trata de un paquete perdido</li> </ul>
2	-t	Tiempo en qué ocurre el evento.
-	-H...	Los campos que empiecen con esta marca se refieren a información acerca del siguiente salto.
3	-Hs	Número de nodo dónde acontece el evento.
4	-Hd	Número del siguiente nodo hacia el destino.
-	-N...	Los campos que empiecen con esta marca se refieren a información acerca de las propiedades del nodo.
5	-Ni	Identificador del nodo.
6	-Nx	Coordenada x en la que se encuentra el nodo.
7	-Ny	Coordenada y en la que se encuentra el nodo.
8	-Nz	Coordenada z en la que se encuentra el nodo.

9	-Ne	Nivel de energía del nodo.
10	-Ni	El valor de este campo depende de qué capa estamos teniendo en cuenta: <ul style="list-style-type: none"> <li>• MAC → paquete correspondiente a la capa MAC</li> <li>• AGT → capa de transporte (tcp ó udp)</li> <li>• RTR → paquete enrutado</li> <li>• IFQ → referente a la cola salida del nodo</li> </ul> PHY → capa física
11	-Nw	Motivo por el cual se descarta el paquete: <ul style="list-style-type: none"> <li>• --- → el paquete no se descarta</li> <li>• END → final de la simulación</li> <li>• COL → colisión a nivel MAC</li> </ul> Existen muchas más posibilidades, no hace falta enumerarlas todas.
-	-M...	Los campos que empiecen con esta marca se refieren a información acerca de los paquetes a nivel de capa MAC.
12	-Ma	Tiempo esperado en segundos para enviar el paquete a través del canal <i>wireless</i> .
13	-Md	Dirección MAC del nodo destino.
14	-Ms	Dirección MAC del nodo fuente.
15	-Mt	Tipo de capa MAC.
-	-I...	Los campos que empiecen con esta marca se refieren a información acerca de los paquetes a nivel de IP.
16	-Is	Dirección IP del nodo fuente seguido de "." Seguido de el número de puerto del nodo fuente.
17	-Id	Dirección IP del nodo destino seguido de "." Seguido del número de puerto del nodo destino.
18	-It	Tipo de paquete. Puede ser: cbr, DSR, ack, udp, tcp, ARP...
19	-Il	Tamaño del paquete en <i>bytes</i> .
20	-If	Identificador flowstate.
21	-Ii	Identificador de secuencia global del paquete. Identifica el paquete en el fichero de trazas.
22	-Iv	Valor del TTL (Time To Live).
-	-P...	Los campos que empiecen con esta marca se refieren a información acerca de los paquetes a nivel de "aplicación".
23	-P	En este campo se indica el tipo de "aplicación" que puede ser: <ul style="list-style-type: none"> <li>• arp → Address Resolution Protocol</li> <li>• dsr → Dynamic Source Routing</li> <li>• cbr → Constant Bit Rate</li> <li>• tcp → Transport Control Protocol</li> </ul> En función de este campo, el número y tipo de campos siguientes serán distintos. Este ejemplo está hecho a partir de un paquete cbr así que seguiremos con los campos de este tipo de paquete.
24	-Pi	Número de secuencia cbr.
25	-Pf	Número de veces que el paquete ha sido reenviado.
26	-Po	Número óptimo de reenvíos necesarios.

**Figura. 24:** Campos del archivo de trazas del simulador NS2.

Fuente: [http://upcommons.upc.edu/bitstream/handle/2099.1/8374/PFC\\_Jordi\\_Chalmeta.pdf?sequence=1](http://upcommons.upc.edu/bitstream/handle/2099.1/8374/PFC_Jordi_Chalmeta.pdf?sequence=1)

## 5.1.5. Simulación de las topologías de Red.

### 5.1.5.1. Variables de Análisis.

Para la simulación de las topologías físicas de la Red inalámbrica de sensores se tomó en cuenta las variables que se consideran más relevantes en el análisis de eficiencia y consumo de energía de una red WSN, para esto se analizó los siguientes puntos:

- ✓ **Tráfico de paquetes:** cantidad de paquetes enviados, recibidos, desechados y reenviados en la red.
  - **Paquetes enviados:** Son los paquetes generados por el nodo emisor (nodo final) y que se envían al nodo coordinador mediante un enlace de comunicación. Como se mencionó en la sección anterior estos paquetes son de tipo CBR sobre una conexión UDP.
  - **Paquetes recibidos:** son aquellos que lograron transmitirse exitosamente entre los nodos finales y el nodo coordinador dependiendo de cada topología, en la topología estrella se transmiten directo al nodo coordinador, por su parte en la topología árbol el paquete tendrá que pasar por nodos enrutadores hasta llegar al nodo coordinador. El número de paquetes recibidos por el nodo coordinador es proporcional al número de paquetes enviados por los nodos finales.
  - **Paquetes desechados:** son aquellos que se pierden en la red por distintas razones como cambios en la topología física, destino inalcanzable u otros y que no lograron llegar al nodo coordinador y en algún punto de la red fueron desechados.
  - **Paquetes reenviados:** son aquellos que para alcanzar su destino realizan distintos saltos a través de la red entre diferentes tipo de nodos enrutadores que integran una WSN, de esta manera los nodos reenvían paquetes que en



este caso se contabilizan dentro de la simulación como reenviados tienen la misma estructura que un paquete emitido por un nodo emisor. De este número de paquetes se puede concluir el número de saltos que utiliza cada conexión para enviar la información dado que entre más paquetes se reenvían significa que el número de saltos es mayor.

- ✓ Retardo en la transmisión de los paquetes que son enviados desde los nodos finales al nodo coordinador.
- ✓ Energía total Consumida: Al hablar de energía se precisa que cada nodo en su inicio cuenta con una capacidad inicial, para la simulación se asignó un valor inicial de energía expresado en Julios de 56.160 para cada nodo, de este valor se procede a restar la energía que se va consumiendo en el transcurso de la simulación, de esta manera se puede obtener un valor de la energía consumida en cada topología expresado en Julios.

#### 5.1.5.2. Métricas de evaluación para el análisis de los datos.

Para extraer la información de los archivos de simulación y obtener los datos a analizar se utilizaron filtros en lenguaje AWK, y se analizaron métricas típicas en el análisis de Redes inalámbricas de sensores como son:

- ✓ **Packet Delivery Ratio (Relación de distribución de paquetes):** muestra la relación entre el número de paquetes de datos entregados al nodo destino y el número de paquetes de datos enviados hacia el mismo. Esta relación nos permite cuantificar el nivel de datos entregados al destino mediante l

$$PDR = \frac{\sum \text{Paquetes recibidos}}{\sum \text{Paquetes enviados}} \text{ Ecuación 3}$$

- ✓ **Throughput de la Red:** esta métrica permite cuantificar el volumen de tráfico o de información que fluye a través de la red, en un determinado tiempo, en este caso el tiempo de simulación mediante la ecuación 4.

$$\text{Throughput} = \frac{\text{\# de Bits}}{\text{Tiempo de simulación}} \text{ Ecuación 4}$$

- ✓ **End-to-End Delay (Retardo de extremo a extremo):** métrica que permite conocer el tiempo medio que se demora un paquete de datos en llegar a su destino, mediante la ecuación 5:

$$\begin{aligned} & \text{End – to – End Delay} \\ = & \frac{\sum(\text{Tiempo en que se recibe el paquete} - \text{tiempo de envío})}{\text{número de conexiones}} \text{ Ecuación 5} \end{aligned}$$

- ✓ **Consumo de Energía:** Se realiza con una simple resta del total de energía inicial en cada nodo y el consumo de energía en cada suceso de la red.

## **5.2. DESARROLLO DEL PROTOTIPO DE WSN.**

En la presente sección se describe el desarrollo del prototipo de red inalámbrica de sensores ambientales; bajo la topología física en estrella esto en base a los resultados obtenidos mediante la simulación realizada en la sección anterior.

El contenido de la sección se encuentra estructurado de la siguiente manera:

En primera instancia se identifica las características y requerimientos propios del diseño de una Red inalámbrica de Sensores, y se procede con la identificación y análisis de las variables ambientales a monitorear según la importancia de estas para el Jardín Botánico Reinaldo Espinosa.

Posteriormente se detalla el diseño de la red inalámbrica de sensores y se seleccionan los componentes que conforman el Hardware de los nodos de nuestro prototipo (Nodos Sensores y Nodo Coordinador); describiendo las principales características de dispositivos tales como: los módulos de comunicación Xbee, el módulo de procesamiento "Arduino UNO"; la interfaz "Xbee shield" que interconecta nuestro módulo de procesamiento con los módulos de comunicación, el Xbee Explorer USB que actúa como Gateway entre nuestro modulo Coordinador de la red y la interfaz gráfica, la unidad de energía y los sensores de adquisición de las variables ambientales.

Se describe de igual forma el proceso de implementación de la Red, la forma cómo fueron programados cada uno de los módulos de comunicación Xbee para operar bajo la topología estrella, y la programación que se realizó en la placa base del nodo sensor para que procese y transmita los datos adquiridos de las variables ambientales en un formato adecuado hacia la estación coordinadora, finalmente se realiza el diseño del sistema HMI que permitirá el monitoreo de los datos que proporcionan los sensores que conforman el prototipo de la WSN.

### **5.2.1. Consideraciones para el Diseño e Implementación del prototipo de WSN.**

Para el diseño de la Red (hardware y software) se tuvieron en cuenta una serie de consideraciones que el sistema debía disponer para asegurar su correcto funcionamiento, esto de acuerdo a las especificaciones y características propias de las WSN que se estudiaron anteriormente, entre las que mencionamos:

#### **✓ Clasificación de los nodos.**

Los nodos de la red se clasificaron según el papel que desempeñan dentro del prototipo, considerando una topología física en estrella y siguiendo una estructura similar a la clasificación establecida por el protocolo de comunicaciones inalámbricas IEEE 802.15.4 ZigBee se diseñaron dos tipos de nodos: nodo coordinador y nodo sensor.

#### **✓ Energía.**

Un punto importante a considerar en las WSN es el consumo de energía, el objetivo es que los nodos consuman la menor cantidad de recursos energéticos; sobre todo cuando estos tienen que funcionar en entornos donde no se tiene disponible un suministro continuo de energía. Si bien se mencionó el protocolo ZigBee se diferencia del resto de protocolos WPAN porque incorpora características para reducir el consumo de las radios, sin embargo es fundamental la correcta elección del resto de los componentes electrónicos que conforman el prototipo, por lo que se considera principalmente para su elección el factor de consumo energético.

#### **✓ Comunicación Inalámbrica.**

En el diseño de la red de sensores es indispensable garantizar el correcto funcionamiento de la comunicación inalámbrica; se sabe que la tecnología ZigBee cuenta con la característica de coexistencia y a pesar de coexistir en la misma frecuencia con otro tipo de redes como WiFi o Bluetooth su desempeño no se ve afectado por interferencias, esto debido a su baja tasa de transmisión y, a características propias del estándar IEEE 802.15.4; se puede decir que ZigBee ocupa el vacío que hay por debajo de Bluetooth,

para comunicaciones de datos que no requieren altas velocidades, por tal motivo se decide usar el módulo de comunicación inalámbrica XBEE PRO S2B, dispositivo que está basado en el estándar IEEE 802.15.4, con ello se garantiza que la red de sensores ambientales no se vea afectada por aspectos como interferencias. Así mismo gracias a los resultados obtenidos de la simulación de la red y que se muestran en la sección 6.1 del presente trabajo, se decide implementar una topología física en estrella para optimizar el funcionamiento de las comunicaciones, reduciendo el número de saltos entre nodos que la información debe realizar hasta llegar a su destino.

✓ **Error de tolerancia.**

Debido a que algunos nodos sensores pueden fallar o bloquearse por la falta de potencia, daño físico, etc. el diseño de la red se realizó bajo la condición de que el mal funcionamiento de un nodo sensor no afecte la tarea global de la red de sensores. Es decir la WSN tiene la habilidad de sostener las funciones de la red sin ninguna interrupción debido a los fracasos que se puedan ocasionar en un nodo sensor.

✓ **Trama de Comunicación.**

La estructura y tamaño de la trama de comunicación de la Red es otra consideración importante que se tomó en cuenta en el diseño de la WSN, de ella depende el minimizar de la manera más óptima el uso de la red y el correcto funcionamiento de la misma. Es imprescindible que exista una unidad de procesamiento de las señales captadas por los sensores de esta forma solo los datos que son estrictamente necesarios se envían a través de la red, reduciendo de esta manera el tráfico y el consumo energético de la misma. Por ello se decide utilizar para el diseño del prototipo de WSN; como unidad de procesamiento una placa Arduino UNO, y el módulo XBee Coordinador se configura en modo de funcionamiento API asegurando de esta forma que las comunicaciones se establezcan de forma fiable. La estructura general de la trama de comunicación en modo API debe incluir por los menos los siguientes campos para identificar correctamente una medición: ID del nodo emisor, ID del nodo receptor y valor de la medición.

✓ **Software.**

A la hora de desarrollar el software para el prototipo de WSN se considera principalmente:

- **Software del nodo sensor:** se considera que el programa a utilizar sea de código abierto y de fácil acceso; en este caso la unidad de procesamiento Arduino cuenta con su propio lenguaje e IDE de programación es considerado como una plataforma de hardware y software libre, muy fácil de conseguir, aprender e implementar.
- **Software del módulo XBee:** en este caso el fabricante del dispositivo ofrece la aplicación X-CTU que es el software propio de estos dispositivos que permite configurar los diferentes parámetros de los módulos: su rol en la red, las conexiones con otros dispositivos, parámetros energéticos, y descargar actualizaciones del firmware de los módulos.
- **Software para la interfaz gráfica:** se elige de acuerdo a las funciones que tiene que cumplir dicha interfaz; tales como almacenamiento y procesamiento de los datos que provienen de la red, además de ser diseñada de tal manera que cuenta con la propiedad de escalabilidad aceptando la integración de nuevos nodos a la red.

Una vez conocidas las consideraciones generales que se tomaron en cuenta en el diseño de la Red Inalámbrica de sensores en los siguientes apartados se abordará la selección de las variables ambientales, selección de hardware, diseño e implementación de los nodos (coordinador y sensor), sus diagramas de operación, diagramas de flujo de la programación del software y pruebas del funcionamiento del prototipo que estará compuesto por dos nodos sensores y un coordinador tal como se observa en la figura 25.



**Figura. 25:** Diseño de la WSN en estrella.

Fuente: [http://www.ni.com/cms/images/devzone/tut/figure1\\_20100521153802.jpg](http://www.ni.com/cms/images/devzone/tut/figure1_20100521153802.jpg)

### 5.2.2. Identificación de las variables ambientales a monitorear.

Para poder identificar las variables físicas a monitorear se realizó una entrevista al encargado de apoyar y sostener las estructuras del jardín botánico Reinaldo Espinosa, estableciendo la importancia de mantener un monitoreo y control continuo de las variables de temperatura y Humedad relativa de los cultivos y especies vegetales que alberga el jardín, ya que cada de estas variables debe mantenerse dentro de los rangos adecuados para que exista un crecimiento favorable de dichas especies.

En la Tabla 5 se presenta el análisis que se realizó a través de la entrevista de estas variables, estableciendo el rango de valores óptimos de estas y las consecuencias que genera el no mantener su control adecuadamente.

**Tabla 6:** Variables ambientales a monitorear.

<b>Variable</b>	<b>Rango de valores óptimos</b>	<b>Daño ocasionado por nivel alto.</b>	<b>Daño ocasionado por nivel bajo</b>
<b>Temperatura</b>	15-23 °C	Alteración de permeabilidad	Estrés Térmico
<b>Humedad del suelo</b>	10-20 cB <sup>4</sup>	Destrucción de Pelos Radiculares	Marchitamiento de las plantas.

Fuente: Elaboración propia.

### 5.2.3. Componentes y/o Dispositivos.

#### 5.2.3.1. Sensores.

En la elección de los sensores se toma en cuenta principalmente las siguientes características:

- ✓ **Campo de medida:** Rango de valores de la magnitud de entrada comprendido entre el máximo y el mínimo detectados por un sensor, con una tolerancia de error aceptable.
- ✓ **Resolución:** Mínima diferencia entre dos valores próximos que el sensor es capaz de distinguir.
- ✓ **Sensibilidad:** Variación de la salida producida por una variación de entrada. Pendiente de la curva de calibración. Cuanto mayor, mejor.
- ✓ **Histéresis:** Diferencia entre valores de salida correspondientes a la misma entrada, según la trayectoria seguida por el sensor.

---

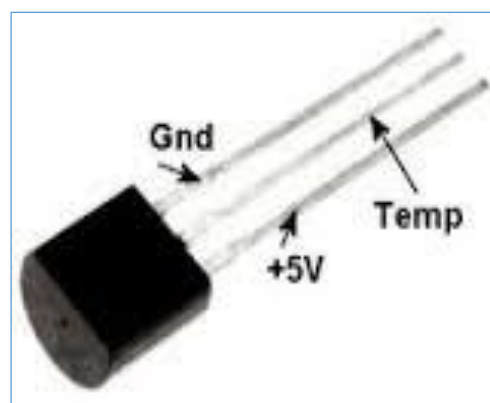
<sup>4</sup> cB: Centibares, unidad métrica de presión utilizada tradicionalmente en la agricultura como unidad de tensión del agua del suelo (la presión del agua en las raíces de las plantas).



- ✓ **Exactitud:** Diferencia entre la salida real y el valor verdadero. Se suele dar en valor absoluto o relativo.
- ✓ **Precisión:** Capacidad de obtener la misma salida cuando se realizan varias lecturas de la misma entrada y en las mismas condiciones.
- ✓ **Velocidad de respuesta:** Capacidad para que la señal de salida siga sin retraso las variaciones de la señal de entrada.
- ✓ **Estabilidad:** Desviación de salida del sensor, al variar ciertos parámetros exteriores distintos del que se pretende medir. [30]

### 5.2.3.2. Sensor de Temperatura LM35DZ.

Para la medición de temperatura se usó el sensor LM35DZ (Figura 26). Es un sensor de temperatura integrado de precisión, cuya tensión de salida es linealmente proporcional a la temperatura en grados centígrados (°C), para su funcionamiento no requiere ninguna calibración externa o ajuste puesto que el dispositivo se ajusta y calibra durante el proceso de producción; proporciona una precisión típica de  $\pm 1.4$  °C a temperatura ambiente y  $\pm 3.4$  °C a lo largo de su rango de temperatura (de -55 a 150 °C). [31] [32]



**Figura. 26:** Sensor LM35DZ

**Fuente:** <http://electronilab.co/wp-content/uploads/2014/05/LM35.jpg>

El sensor LM35DZ tiene baja impedancia de salida, su salida es lineal y posee una calibración inherente precisa, lo que lo hace útil en la creación de circuitos de lectura o control [31]. En la tabla 6 se muestra las características principales del sensor y en el Anexo VII de este trabajo se encuentra la hoja de especificaciones técnicas dadas por el fabricante.

**Tabla 7:** Características del sensor de temperatura LM35DZ.

Características	Valor
Calibración	Grados centígrados
Precisión	0,5 °C a +25 °C
Rango de trabajo	De -55 °C a +150 °C
Aplicación	Apropiado para aplicaciones remotas
Costo	Bajo
Alimentación	Entre 4V y 30V
Consumo.	Menos de 60 µA
Impedancia de salida	0,1Ω

Fuente: <http://www.ti.com/lit/ds/symlink/lm35.pdf>

Para convertir el voltaje a temperatura, se utiliza la fórmula básica del sensor

$$\text{Temperatura en } ^\circ\text{C} = [(\text{Vout en mV}) - 500] / 10.$$

### 5.2.3.3. Sensor de Humedad, Sparkfun Soil Moisture Sensor.

El sensor de humedad utilizado en el proyecto se muestra en la Figura 27, mediante este sensor podemos detectar la humedad del suelo o la presencia de agua alrededor del sensor. Para obtener las mediciones de humedad basta con insertarlo en el suelo y se podrá saber si la planta necesita agua mediante la observación de los resultados de la salida del sensor. En la tabla 7 se indican las especificaciones del mismo.



**Figura. 27:** Sensor de Humedad

Fuente: [http://www.adhiratech.com/index.php?route=product/product&product\\_id=2215](http://www.adhiratech.com/index.php?route=product/product&product_id=2215)

**Tabla 8:** Características y especificaciones del sensor de humedad  
Sparkfun Soil Moisture Sensor

Característica y/o especificación	Valor
Fuente de alimentación:	3,3 V o 5 V
Señal de salida de voltaje:	0 ~ 4.2v
Corriente:	35mA
Pines de conexión:	1-Salida analógica 2-GND. 3-Power.
Tamaño:	60x20x5mm
Rango de valores:	0 ~ 300: suelo seco. 300 ~ 700: suelo húmedo. 700 ~ 950: suelo en agua.

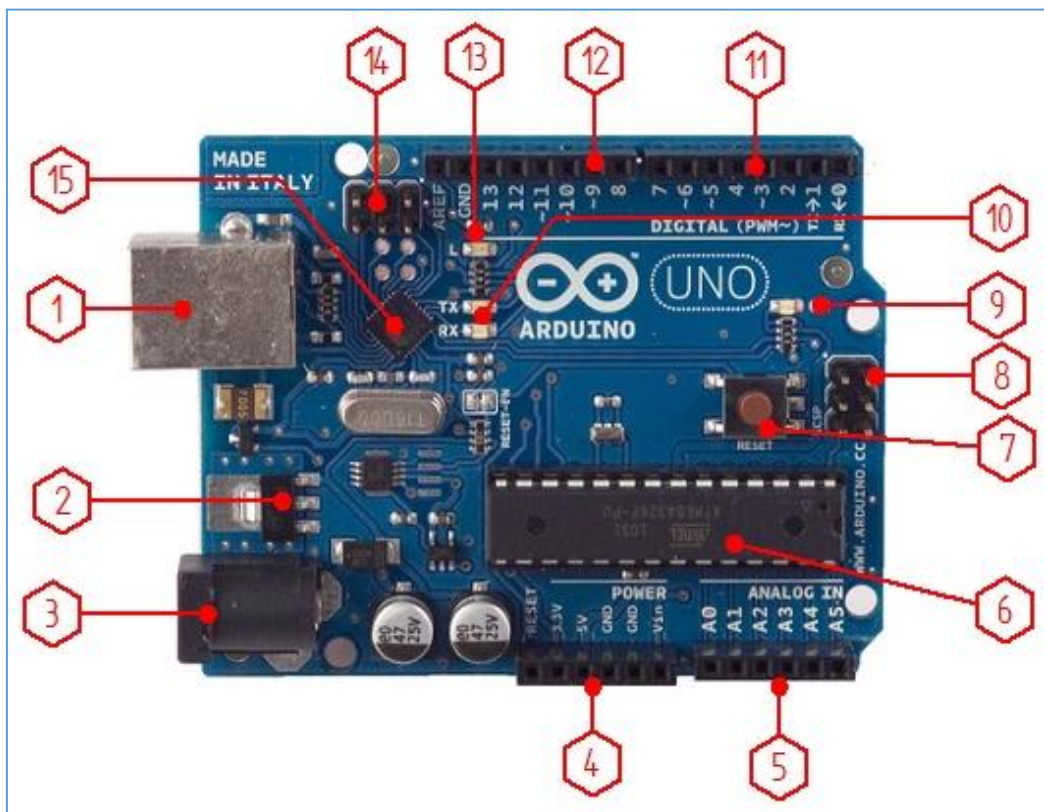
Fuente: [http://www.dfrobot.com/wiki/index.php/Moisture\\_Sensor\\_\(SKU:SEN0114\)](http://www.dfrobot.com/wiki/index.php/Moisture_Sensor_(SKU:SEN0114))

#### 5.2.3.4. Arduino Uno.

Arduino es una plataforma computacional física open-source basada en una simple tarjeta de I/O; (entradas y salidas, analógicas y digitales) y un entorno de desarrollo que implementa el lenguaje Processing/Wiring.

Arduino Uno R3 es un dispositivo que conecta el mundo físico con el mundo virtual, o el mundo analógico con el digital y que puede ser utilizado para desarrollar objetos interactivos o puede ser conectado a software de tu computadora (por ejemplo, Flash, Processing). El IDE open-source puede ser descargado gratuitamente actualmente para sistemas operativos como Mac OS X, Windows y Linux. [33]

En la Figura 28 vemos el esquemático del Hardware de la placa Arduino Uno.



**Figura. 28:** Esquemático de Arduino UNO

Fuente: [https://sites.google.com/site/temasdedisenoymanufactura/\\_/rsrc/1333349488409/arduino/arduino%20uno%20partes.jpg](https://sites.google.com/site/temasdedisenoymanufactura/_/rsrc/1333349488409/arduino/arduino%20uno%20partes.jpg)

## ✓ Descripción.

- 1- Conector USB, que puede ser tipo B o mini, este provee la comunicación para la programación y la toma de datos, también provee una fuente de 5VDC para alimentar al arduino, pero de baja corriente por lo que no sirve para alimentar motores grandes por ejemplo.
- 2- Regulador de voltaje de 5V, se encarga de convertir el voltaje ingresado por el plug 3, en un voltaje de 5V regulado. Necesario para el funcionamiento de la placa y para alimentar circuitos externos.
- 3- Plug de conexión para fuente de alimentación externa, el voltaje que se suministra por aquí debe ser directo y estar entre 6V y 18V, incluso 20V, generalmente se usa un adaptador.
- 4- Puerto de conexiones; constituido por 6 pines de conexión con las siguientes funciones: RESET, permite resetear el microcontrolador al enviarle un cero lógico. Pin 3.3V, este pin provee una fuente de 3.3VDC. Pin 5V, provee una fuente de 5VDC. Dos pines GND, que proveen la salida de cero voltios para dispositivos externos. Pin Vin, se usa para conectar la alimentación de la placa con una fuente externa de entre 6 y 12VDC.
- 5- Puerto de entradas análogas, aquí se conectan las salidas de los sensores analógicos. Estos pines funcionan como entradas recibiendo voltajes entre cero y cinco voltios directos.
- 6- Microcontrolador Atmega 328, es el microcontrolador implementado en los Arduino uno y sobre el cual vamos a programar.
- 7- Botón de RESET, este botón así como el pin mencionado anteriormente permiten resetear el microcontrolador haciendo que reinicie el programa.
- 8- Pines de programación ICSP, son usados para programar microcontroladores en protoboard o sobre circuitos impresos sin tener que retirarlos de su sitio.
- 9- LED ON, enciende cuando el Arduino está encendido.
- 10- LEDs de recepción y transmisión, estos se encienden cuando la tarjeta se comunica con el PC. El Tx indica transmisión de datos y el Rx recepción.
- 11- Puerto de conexiones, está constituido por los pines de entradas o salidas digitales desde la cero hasta la 7.

La configuración de estos pines como entradas o salidas debe ser incluida en el programa que se realice dentro del microcontrolador de la placa. Cuando se usa la terminal serial es conveniente no utilizar los pines cero (Rx) y uno (Tx). Los pines 3, 5 y 6 están precedidos por el símbolo ~ lo que indica que permiten su uso como salidas controladas por ancho de pulso PWM.

12- Puerto de conexiones, incluye 5 entradas o salidas adicionales (de la 8 a la 12), las salidas 9, 10 y 11 permiten control por ancho de pulso; la salida 13 es un poco diferente pues tiene conectada una resistencia en serie, lo que permite conectar un led directamente entre ella y tierra. Finalmente hay una salida a tierra GND y un pin AREF que permite ser empleado como referencia para las entradas análogas.

13- Led indica el estado del pin 13.

14- Chip de comunicación que permite la conversión de serial a USB. [34]

Las características principales de la placa se resumen en la Tabla 8.

**Tabla 9:** Características de la placa Arduino Uno.

Parámetro	Dato
Voltaje operativo	5 V
Voltaje de entrada recomendado	7-12 V
Voltaje de entrada límite	6-20 V
Pines de entrada y salida digital	14 (6 proporcionan PWM)
Pines de entrada analógica	6
Intensidad de corriente	40 mA
Memoria Flash	32KB (2KB reservados para el bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

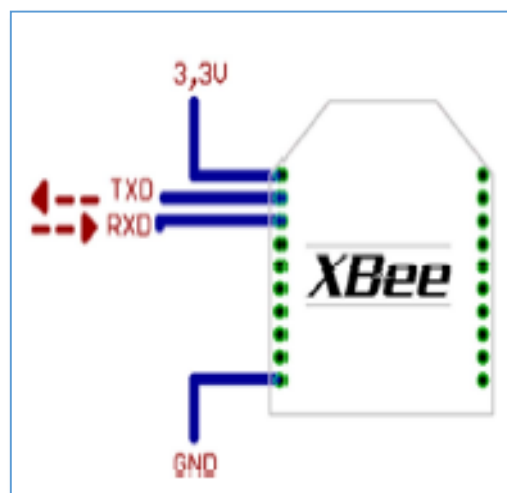
Fuente: <https://www.arduino.cc/en/Main/ArduinoBoardUno>

### 5.2.3.5. Módulo XBee Pro S2B.

El XBee Pro S2B es un módulo de comunicación inalámbrica innovador principalmente porque ofrece mayor estabilidad a la red a través de sus características de auto-curación y auto-descubrimiento, con esto asegura que el fallo de un nodo sensor no afecte al resto de la red y este pueda reintegrarse fácilmente cuando esté en condiciones de funcionamiento adecuadas, de igual forma son fáciles de usar, y son completamente interoperables con otros productos o series XBee.

#### ✓ Conexión básica.

Para funcionar en condiciones óptimas el modulo requiere una alimentación de 2.8 a 3.4V, y una conexión a tierra; para comunicarse con la unidad de procesamiento usa las líneas de transmisión de datos por medio del UART (TXD Y RXD) tal como lo muestra la Figura 29.



**Figura. 29:** Conexión para un módulo XBee

Fuente: <http://alvarounal.blogspot.com/2011/10/modulos-xbee-parte1.html>

#### ✓ Descripción de pines.

Es importante conocer para que sirve cada uno de los 20 pines del módulo XBee Pro S2B, para facilitar la ubicación y descripción de los pines se muestra el diagrama de la Figura 30.



**Figura. 30:** Descripción de los Pines del Xbee S2B

Fuente: [http://3.bp.blogspot.com/-EsBAEOpE9PE/TofmY\\_K1huI/AAAAAAAAAE8/G7S1yjA\\_Ji0/s1600/Pines.jpg](http://3.bp.blogspot.com/-EsBAEOpE9PE/TofmY_K1huI/AAAAAAAAAE8/G7S1yjA_Ji0/s1600/Pines.jpg)

En la Tabla 9 se describe en detalle cada uno de los pines del módulo Xbee Pro S2B y en la Tabla 10 sus especificaciones técnicas.

**Tabla 10:** Descripción de los pines del XBee Pro S2B.

PIN	NOMBRE	DIRECCIÓN	DESCRIPCIÓN
1	VCC	-----	Alimentación
2	DOUT	Salida	Salida UART
3	DIN/	Entrada	Entrada UART
4	DIO12	E/S	Digital E/S 12
5	RST	Entrada	Módulo de reinicio
6	PWM0/RSSI	E/S	PWM Salida 0 Indicador de la intensidad de señal recibida.
7	DIO11	E/S	Digital E/S 11
8	Reservado	-----	No conectar
9	SLEEP_RQ/DIO8	E/S	Pin Sleep, Control en línea Digital E/S 8
10	GND	-----	Tierra
11	DIO4	E/S	Digital E/S 4
12	DIO7	E/S	Control de flujo Clear-to-Send Digital E/S 7. CTS.
13	ON	Salida	Indicador del estado del módulo Digital E/S 9



14	VREF	Entrada	No se utiliza en este módulo, por compatibilidad con otros módulos XBee, si se desea muestreo analógico se recomienda conectar este pin a un voltaje de referencia, de lo contrario, conectarse a GND.
15	Asociado/DIO5	E/S	Indicador asociado, Digital E/S 5
16	DIO6	E/S	Control de flujo Request-to-Sent, Digital E/S 6.
17	AD3/DIO3	E/S	Entrada analógica 3 o Digital E/S 3
18	AD2/DIO2	E/S	Entrada analógica 2 o Digital E/S 2
19	AD1/DIO1	E/S	Entrada analógica 1 o Digital E/S 1
20	AD0/DIO0/	E/S	Entrada analógica 0, Digital E/S 0

Fuente: <http://www.adafruit.com/datasheets/XBee%20ZB%20User%20Manual.pdf>

✓ **Características Técnicas del módulo XBee Pro S2B.**

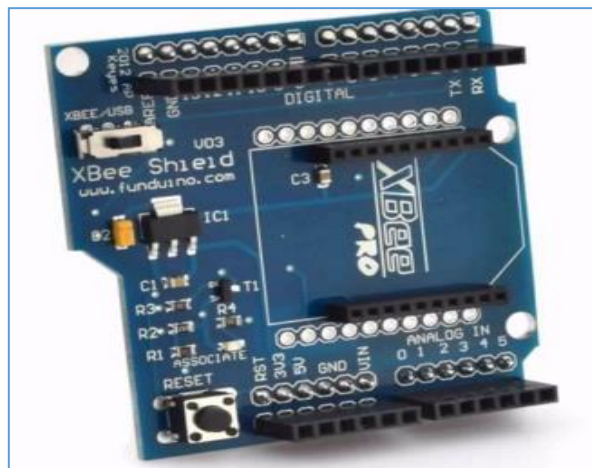
**Tabla 11:** Especificaciones Técnicas del módulo XBee Pro S2B

<b>ESPECIFICACIONES TÉCNICAS DEL XBEE PRO S2B</b>		
<b>Rendimiento</b>	Cobertura en interiores	Hasta 300 pies (90 m)
	Cobertura en exteriores	Hasta 1 milla (1600 m)
	Potencia de salida	63 mW (+18 dBm)
	RF, Velocidad de datos	250 Kbps
	Sensibilidad del receptor	-102 dBm
<b>Requerimientos de potencia</b>	Suministro de voltaje	2.7 V – 3.6 V DC
	Transmisión de corriente	205 mA @ 3.3 V
	Recepción de corriente	47 mA @ 3.3 V
	Intensidad de reposo	3.5 µA @ 25 °C
<b>Información general</b>	Banda de frecuencia	2.4 GHz ISM
	Métodos de configuración	AT y API, local o por aire
	Tipo de antena	Conector RP-SMA
<b>Trabajo en red</b>	Topologías de red	Punto a punto, punto a multipunto, y mesh
	Capacidad de canales	15 Canales de secuencia directa (11 a 25)
	Direccionamiento	PAN ID, 64-bit IEEE MAC

Fuente: <http://www.adafruit.com/datasheets/XBee%20ZB%20User%20Manual.pdf>

### 5.2.3.6. Shield XBee Pro.

La Shield XBee Pro (Figura 31) es una placa, creada por la empresa de Libelium que permite la conexión de los módulos Xbee con el microcontrolador de Arduino; dispone de pines digitales y analógicos y de dos jumpers que determinan esta conexión; existen dos posiciones posibles, la posición XBee y la posición USB.



**Figura. 31:** XBee Shield.

**Fuente:** Elaboración propia.

La posición XBee conecta el pin DOUT del módulo XBee al pin Rx del microcontrolador de la unidad de proceso y el pin DIN está conectado al pin de Tx. En esta posición el microcontrolador solo será capaz de recibir datos desde el módulo XBee y no desde el USB de un ordenador; esta posición es usada cuando se ubica los módulos remotamente, como sucede en la aplicación de este trabajo.

La posición USB conecta el pin DOUT del módulo XBee al pin Rx del chip FTDI y el pin DIN al pin de TX del chip FTDI, en esta posición podemos conectar directamente el módulo Xbee al ordenador, bajo esta posición se configuraron los diferentes firmware de los nodos que estructuran la WSN algo a tomar en cuenta es que se debe quitar el microcontrolador de la placa Arduino cuando se usara la Shield XBee en esta posición.

### 5.2.3.7. Baterías.

Como suministros de energía se usaron baterías con las siguientes características.

**Tabla 12:** Características de la unidad de energía.

Característica.	Valor.
Batería tipo gel	Recargable
Capacidad	1.3 A/h
Voltaje	12V
Marca	Press
Largo	97mm
Ancho	43mm
Altura	52mm
Tipo de conector	FO-01

Fuente: <http://www.pronext.com.ar/admin/archivos/PR1213.pdf>

### 5.2.3.8. XBee Explorer USB.

Se usa en los módulos de Series 2/2.5/ZB para conectarlos a un PC e instalar el firmware; son útiles para cambiar la configuración de los dispositivos, y nos permiten enviar y recibir información a un PC actuando como Gateway entre esta y el módulo XBee. Viene tanto en una versión micro USB como en una para conectar directamente al puerto USB como se ve en la Figura 32.



**Figura. 32:** XBee Explorer USB

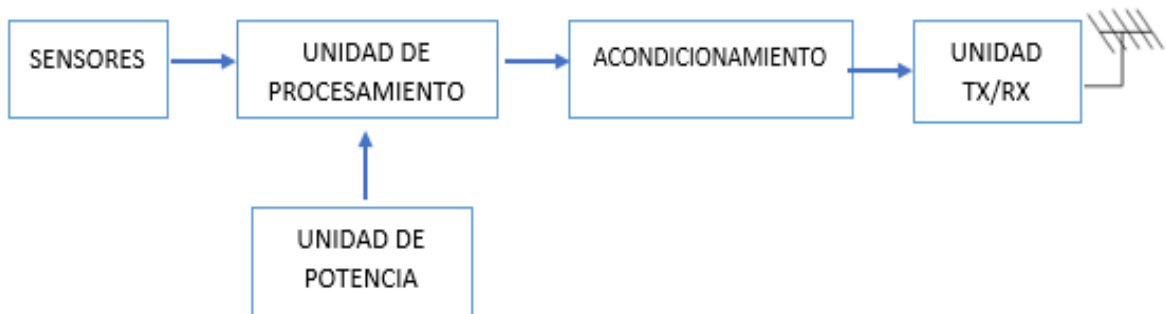
Fuente: <http://webdelcire.com/wordpress/wp-content/uploads/2012/02/XBeeexplorerusb.jpg>

## 5.2.4. NODO SENSOR.

### 5.2.4.1. Diseño del nodo Sensor.

El diseño del dispositivo final o nodo sensor se basó en cinco componentes:

- Sensores,
- Unidad de procesamiento,
- Unidad de transmisión-recepción inalámbrica,
- Unidad de acondicionamiento y
- Unidad de potencia, tal como lo muestra el diagrama de bloques de la Figura 33.



**Figura. 33:** Diagrama de bloques de un Nodo WSN Sensor.

Fuente: Elaboración Propia

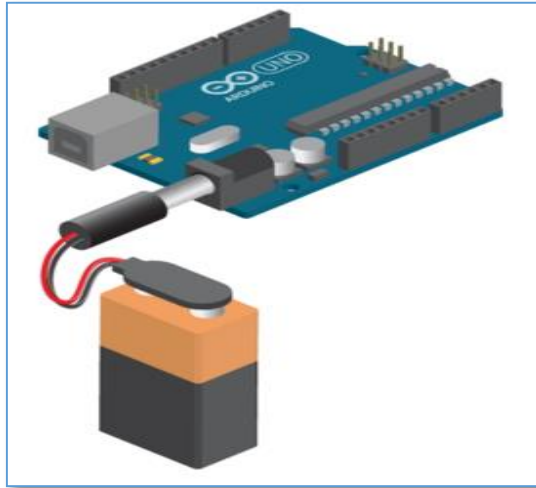
#### ✓ **Sensores.**

Los sensores tienen como objetivo obtener una medición de las variables físicas y convertirlas a una señal eléctrica que será entregada a la unidad de procesamiento. Cada nodo final cuenta con dos sensores; que permiten la medición de las variables de temperatura y humedad relativa del lugar donde se encuentran ubicados.

#### ✓ **Unidad de procesamiento.**

Los Nodos cuentan con un Arduino Uno V1.8.1 para procesar e interpretar los datos que envían los sensores. La unidad de proceso estará conectada directamente a la unidad de

potencia mediante el plug de conexión de la placa para una fuente de alimentación externa; en la Figura 34 se observa esta conexión.



**Figura 34:** Arduino Uno alimentado mediante una fuente de alimentación externa.

Fuente: <http://zgz.verkstad.cc/digital-water-pavilion-es/>

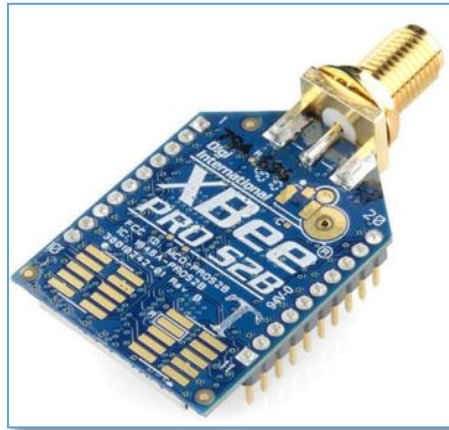
La unidad de procesamiento tiene la función de interpretar y procesar los datos provenientes de los sensores y a su vez tendrá que proveer de energía a los sensores y a la unidad de comunicación Xbee.

#### ✓ **Unidad de transmisión/recepción inalámbrica.**

La unidad de transmisión/recepción inalámbrica es muy importante dentro del diseño de las redes WSN pues es la encargada de establecer los enlaces de comunicación inalámbrica entre los distintos nodos y transmitir la información de los sensores de manera eficiente hacia el nodo coordinador. Para la elección del módulo XBee correcto se tomó a consideración primeramente el tipo de antena (chip, alambre o conector RP-SMA) y la potencia de transmisión (2mW para 300 pies o 60mW para 1.6 km).

Se utilizó en el diseño, el módulo Xbee Pro S2B con conector RP-SMA (figura 35), cuyas características se mencionan en la sección anterior, debido a que cumple con los requerimientos que se necesitan para el desarrollo del prototipo de WSN; soporta la configuración de la topología estrella, que es la que se desea configurar y permite según

datos del fabricante, un alcance de 1.6 Km en exteriores y de 300m en interiores, cubriendo satisfactoriamente los 225m de extensión del jardín botánico Reinaldo Espinosa que es el escenario en donde se desplegara la red.



**Figura 35:** Modulo XBee Pro S2B

**Fuente:** <https://www.sparkfun.com/products/10419>

✓ **Unidad de acondicionamiento.**

Con la finalidad de acoplar los voltajes proporcionados por Arduino UNO a nuestro módulo de comunicación inalámbrica XBee Pro S2B se decidió usar como unidad de acondicionamiento una Shield XBee Pro.

✓ **Unidad de Potencia.**

La unidad de potencia o baterías son elementos fundamentales en el diseño de nuestra WSN porque deben estar en la capacidad de otorgar autonomía a los nodos remotos; los principales parámetros que se tomaron en cuenta a la hora de la elección de la unidad de energía fueron:

- Tensión que aporta: Los requerimientos de diseño imponen que nuestra batería proporcione una alimentación entre 6 V y 20 V, que es el rango de tensiones de alimentación que necesita nuestra unidad de procesamiento Arduino UNO para funcionar adecuadamente.

- Capacidad de la batería, medida en mAh; es un factor de vital importancia ya que de esta manera podemos cuantificar el tiempo vida de la Red de sensores. Por ejemplo, si la batería tiene una capacidad de 250 mAh, y provee una corriente media de 2 mA a la carga, en teoría la batería podría otorgar una vida de 125 horas a nuestro equipo.
- El diseño en la placa base: la batería no debe ocupar demasiado espacio y al mismo tiempo debe integrarse adecuadamente con el resto del hardware de nuestro prototipo.

Una vez analizados los criterios de diseño se decidió utilizar como suministros de energía baterías de 12V con una capacidad de 1.3 A/h.

#### **5.2.4.2. Implementación del Nodo Sensor.**

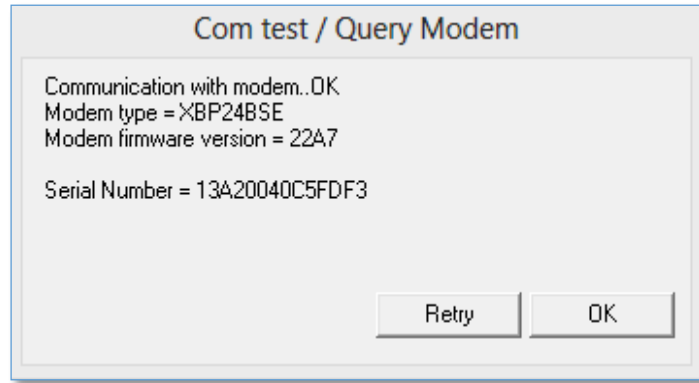
Una vez que se ha diseñado el nodo sensor, se procede a su implementación, a continuación se describe paso a paso la configuración del módulo XBee Pro S2B como nodo sensor mediante el software XCT-U; el diagrama de flujo del funcionamiento y el diagrama de conexión de los diferentes componentes y dispositivos.

##### **✓ Configuración y diagramas de flujo.**

Para la configuración [35] del módulo XBee Pro S2B como dispositivo final se deben seguir los siguientes pasos:

- Se inserta el módulo XBee en la tarjeta XBee Explorer USB.
- Se abre una ventana de la terminal X-CTU.
- Se selecciona el puerto USB donde está conectado el XBee mediante el explorador USB.
- Se configura los valores de Baud Rate: 9600, Flow Control: none, Data Bits: 8, Parity: none; Stop Bits: 1

- Se comprueba la comunicación con el XBee a través del botón Test / Query del software XCTU; obtenemos una pantalla similar a la de la Figura 36.



**Figura 36:** Comunicación XCTU-XBee

Fuente: Elaboración propia

- Se dirige a la pestaña Modem Configuration del Software XCTU y se da click en el botón Read con esto se puede leer los parámetros que por defecto tiene configurados el XBee. Dependiendo de la versión de hardware es decir serie 1, 2, pro etc. será el firmware que se desplegará.
- En la opción “Function Set”, se configura el XBee con el firmware de Router AT.
- En la carpeta “Networking”, se configura el Pan ID para el dispositivo tomando en cuenta que este debe ser igual al que se configura en el módulo Coordinador de esta forma queda direccionado al mismo.
- El dispositivo coordinador por defecto asignara al dispositivo final el canal de operación que escogió.
- En la carpeta “Addressing”, se configura la dirección del Coordinador en el parámetro Destination Address donde se enviaran los paquetes de resultados de la variable sensada.



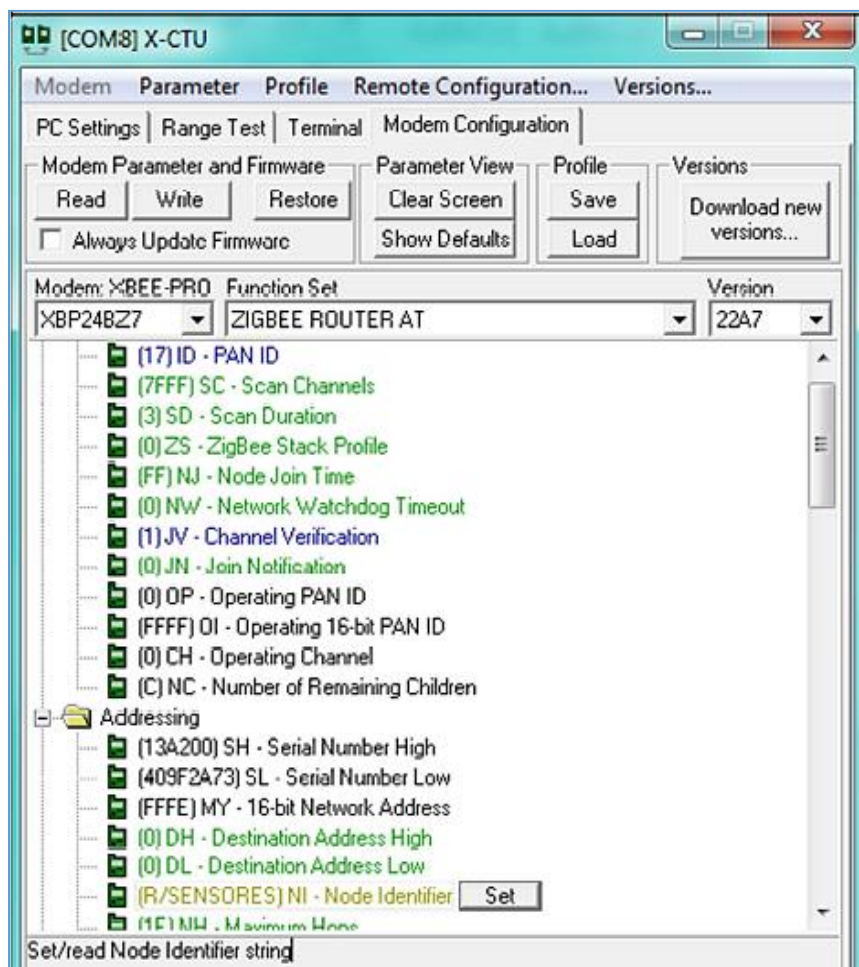
- Se configura el parámetro llamado Nodo Identifier donde colocaremos el nombre del nodo para diferenciarlo de los demás.
- De igual forma se configura el tiempo que permanecerá en estado de reposo el dispositivo y cada que tiempo envía los datos.
- Terminada la configuración se pulsa el botón “Write” de XCTU y se espera a que se grabe la configuración del firmware en nuestro XBee.
- Finalmente se cierra el X-CTU, desconectamos el XBee Explorer y el XBee que se acaba de configurar.

En la Tabla 12 se observa todos los parámetros configurados para este dispositivo y en la Figura 37 la configuración realizada en el software XCT-U.

**Tabla 13:** Parámetros configurados en el XBee del Nodo Sensor.

<b>CONFIGURACIÓN AT DEL NODO SENSOR</b>		
<b>INDICADOR</b>	<b>NOMBRE</b>	<b>VALOR</b>
<b>NI</b>	Identificador del nodo	R/sensor
<b>ID</b>	PAN ID	17
<b>MY</b>	Dirección de Red	10
<b>SD</b>	Duración del escaneo	3
<b>CH</b>	Canal de operación	11
<b>SC</b>	Examina el canal elegido	40
<b>DH</b>	Dirección destino en Bajo	13A200 (Dirección en Bajo del Coordinador)
<b>DL</b>	Dirección destino en Bajo	409F2A79 (Dirección en Alto del Coordinador)
<b>SH</b>	Número de Serie en Alto	13A200
<b>SL</b>	Número de Serie en Bajo	4092C8EA
<b>D1/D3</b>	Configuración AD1/AD3	2
<b>SP</b>	Periodo de Reposo para 10 min	3E8
<b>SN</b>	Nº de periodos de reposo	3C

Fuente: Elaboración propia



**Figura 37:** Configuración del nodo Sensor en XCTU

Fuente: Elaboración propia

- ✓ El parámetro PAN ID es el identificador de la red, en este caso se ha indicado que use el identificador 17.
- ✓ Los parámetros de Destination Address High y Destination Address Low están configurados con los valores Serial Number High y Serial Number Low del XBee configurado como coordinador, de esta forma los dispositivos finales únicamente enviarán información al módulo coordinador. (Característica propia de la Topología estrella)
- ✓ Channel 11 es el canal que se ha asignado al módulo Xbee del dispositivo final para realizar la comunicación, al indicarle al módulo que trabaje en el Channel 11 se está trabajando en el rango de frecuencia de: 2,4075 a 2,4125GHz.

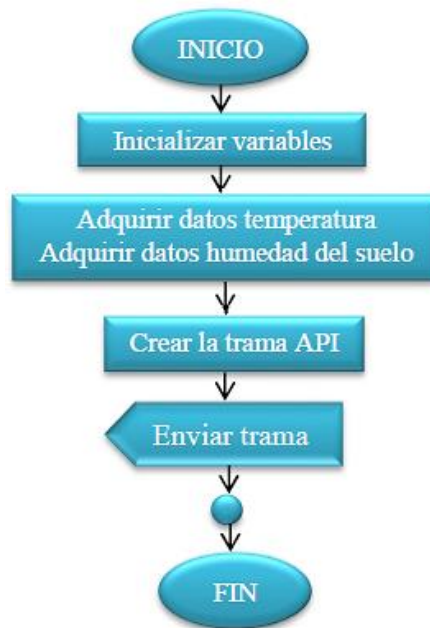
- ✓ Los parámetros Serial Number High (SH) y Serial Number Low (SL) son los datos identificativos del módulo XBee de cada dispositivo final (No son modificables por el usuario y se los puede encontrar impresos en la parte inferior de cada módulo), como lo muestra la Figura 38.



**Figura 38:** Direcciones SH y SL de un XBee.

**Fuente:** <https://mechatronicauasp.wordpress.com/2013/07/04/tutorial-xbee-parte-2-configuracion-xbee-serie-2/>

El la figura 39 se muestra el diagrama de flujo de la programación que se realiza en la unidad de procesamiento Arduino UNO para la adquisición de las variables ambientales y su posterior transmisión al nodo coordinador.

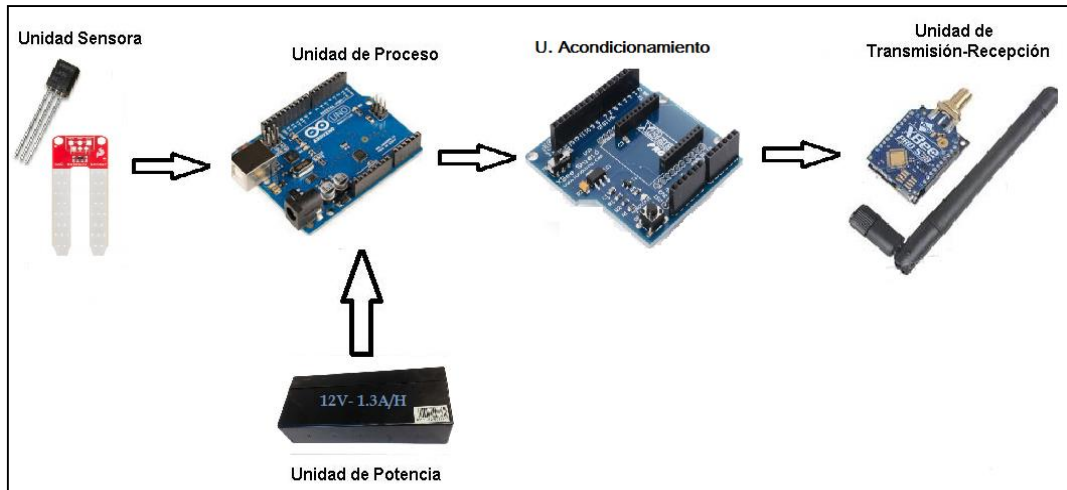


**Figura 39:** Diagrama de flujo del Nodo Sensor.

**Fuente:** Elaboración propia

✓ **Conexión.**

En la Figura 40 se observa el diagrama de conexión final de todos los dispositivos que conforman un nodo sensor; y en la Tabla 13 se muestran los elementos utilizados para la implementación de este nodo.



**Figura 40:** Diagrama de conexión de los elementos de un Nodo Sensor.

Fuente: Elaboración propia

**Tabla 14:** Elementos utilizados en un nodo Sensor.

Elemento	Valor/Tipo
1 XBee Pro	S2B
1 Batería	12V/1.3 A/h
1 Placa Arduino	Uno V1.8.1
1 XBee Shield	Pro S2B
1 Sensor de temperatura	LM35DZ
1 Sensor de Humedad	Sparkfun SEN92355P
1 Antena XBee	5dBi
1 Conector	RP-SMA

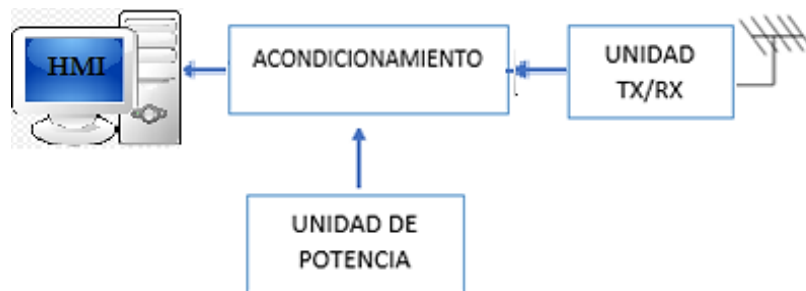
Fuente: Elaboración propia

## 5.2.5. NODO COORDINADOR.

### 5.2.5.1. Diseño del Nodo Coordinador.

El diseño del dispositivo coordinador se basó en la elección de tres componentes.

- Unidad de transmisión-recepción inalámbrica,
- Unidad de acondicionamiento y
- Unidad de potencia, tal como lo muestra el diagrama de bloques de la Figura 41.



**Figura 41:** Diagrama de bloques de un Nodo WSN Coordinador.

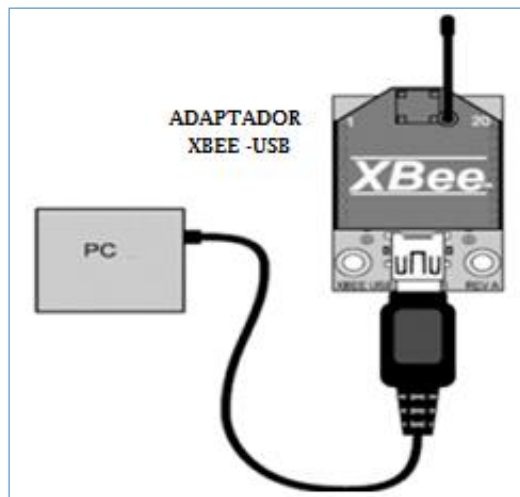
Fuente: Elaboración propia

#### ✓ **Unidad de transmisión/recepción inalámbrica.**

Al igual que en el diseño del nodo sensor se utilizó un módulo de comunicación inalámbrica XBEE PRO S2B para establecer el enlace inalámbrico entre los nodos sensores y el nodo Coordinador, esta unidad es la encargada de recibir los datos provenientes de los nodos finales y enviarlos hacia la interfaz HMI que se encarga de procesarlos para su posterior interpretación por el usuario.

#### ✓ **Unidad de acondicionamiento.**

Con la finalidad de acoplar los niveles de voltaje del ordenador en donde está corriendo la aplicación de monitoreo a los de la unidad de tx/rx y poder recibir los datos en la aplicación se hizo uso de un Xbee USB Explorer como unidad de acondicionamiento esta conexión se muestra en la Figura 42.



**Figura 42:** Conexión Pc-XBee Explorer.

Fuente: [http://www.geekytheory.com/wp-content/uploads/2013/04/tumblr\\_lzay5levCL1qf00w4.png](http://www.geekytheory.com/wp-content/uploads/2013/04/tumblr_lzay5levCL1qf00w4.png)

#### ✓ **Unidad de Potencia.**

El nodo Coordinador es el modulo principal de la red de sensores de su funcionamiento depende el correcto desempeño del resto de dispositivos que conforman la red por ello este nodo estará continuamente alimentado por lo voltajes proporcionados por una computadora por medio del puerto USB.

#### **5.2.5.2. Implementación del Nodo Coordinador.**

Una vez diseñado el nodo Coordinador, se procede a su implementación, a continuación se describe paso a paso la configuración del módulo XBee Pro S2B como Coordinador mediante el software XCT-U; y se muestra el diagrama de conexión de los diferentes componentes y dispositivos.

#### ✓ **Configuración.**

Para la configuración del XBee Pro S2B [35] como dispositivo Coordinador se deben seguir los siguientes pasos:

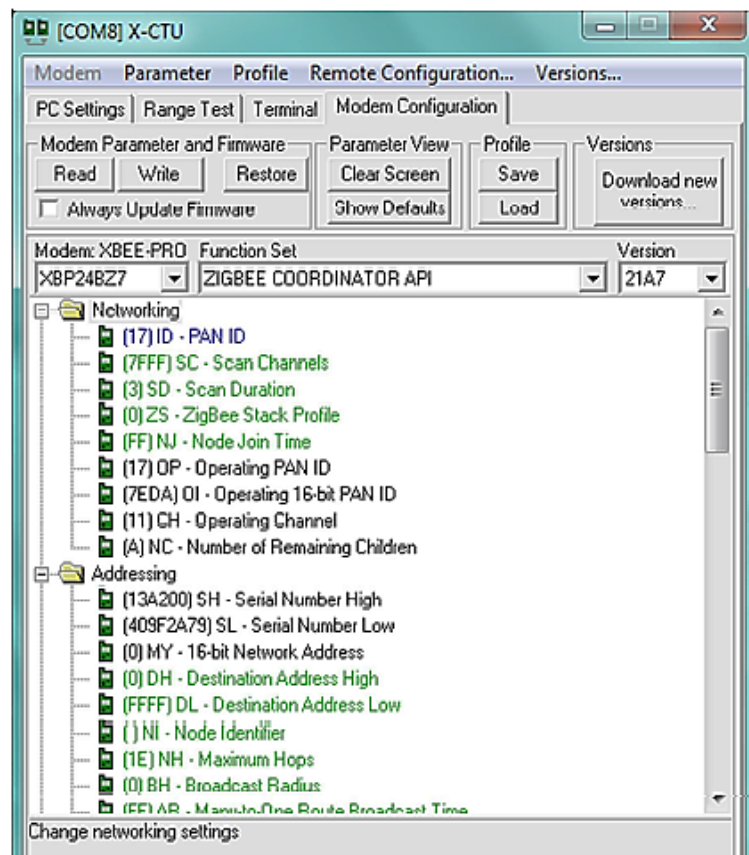
- Se inserta el módulo XBee en la tarjeta XBee Explorer USB.
- Se abre una ventana de la terminal X-CTU.y se selecciona el puerto USB en donde se tiene conectado el XBee mediante el explorador USB.
- Se configura los valores de Baud Rate: 9600, Flow Control: none, Data Bits: 8, Parity: none; Stop Bits: 1 y se comprueba la comunicación con el XBee a través del botón Test / Query del software XCTU.
- Se dirige a la pestaña Modem Configuration del Software XCTU, en la opción “Function Set”, se configura el XBee con el firmware del dispositivo a Coordinador API.
- En la carpeta “Networking”, se configura el Pan ID para el dispositivo Coordinador o en su defecto el coordinador la escoge aleatoriamente dentro de los rangos establecidos.
- El Coordinador escogerá aleatoriamente la MY que es la dirección corta de direccionamiento de 16 Bits a través de la cual se tendrá comunicación con los dispositivos esclavos que estén direccionados a este dispositivo.
- Al configurarlo por defecto elegirá el canal de operación.
- Se configura el parámetro Nodo Identifier donde colocaremos el nombre del nodo Coordinador para diferenciarlo de los nodos sensores.

En la Tabla 14 se detallan todos los parámetros que se configuran para este dispositivo y en la Figura 43 se muestra la configuración que se realiza en el software XCT-U.

**Tabla 15:** Parámetros configurados en el XBee del Nodo Coordinador.

CONFIGURACIÓN API DEL NODO CORDINADOR		
INDICADOR	NOMBRE	VALOR
NI	Identificador del nodo	Coordinador
ID	PAN ID	17
MY	Dirección de Red	10
SD	Duración del escaneo	3
CH	Canal de operación	11
SC	Examina el canal elegido	40
DH	Dirección destino en Bajo	0
DL	Dirección destino en Bajo	FFFF
SH	Número de Serie en Alto	13A200
SL	Número de Serie en Bajo	409F2A79
D0/D5	Configuración AD1/AD3	1
AP	Modo API	1
P0	Configuración DIO10/PWM0	1

Fuente: Elaboración propia



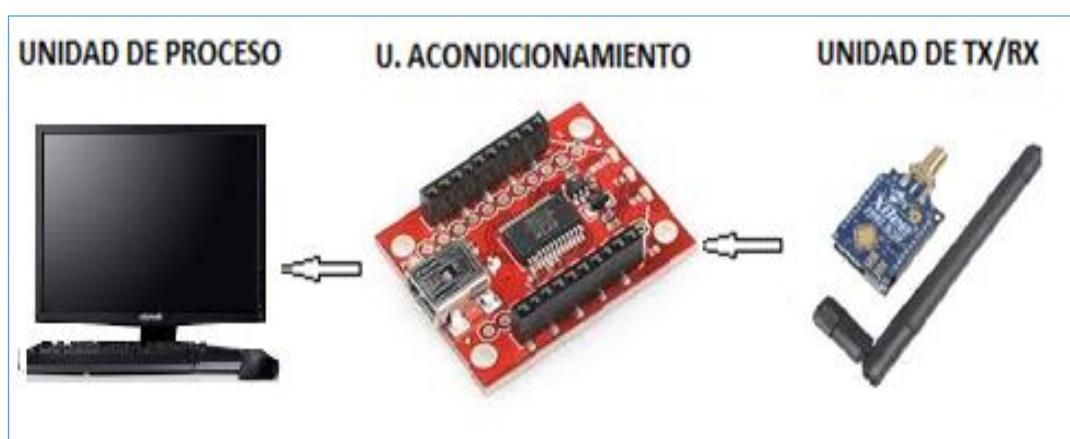
**Figura 43:** Configuración del nodo Coordinador en XCTU

Fuente: Elaboración propia



- ✓ El parámetro PAN ID es el identificador de la red, debe ser la misma que se configura en el dispositivo del nodo sensor en este caso se ha indicado que use el identificador 17.
- ✓ Los parámetros de Destination Address High y Destination Address Low están configurados con los valores de 0000 y FFFF, respectivamente de esta forma le indicamos al módulo que debe establecer comunicación con todos los nodos de la red.
- ✓ Channel 11 es el canal que se ha asignado al módulo Xbee del dispositivo final para realizar la comunicación, al indicarle al módulo que trabaje en el Channel 11 se está trabajando en el rango de frecuencia de: 2,4075 a 2,4125GHz.
- ✓ Los parámetros Serial Number High (SH) y Serial Number Low (SL) son los datos identificativos del módulo XBee que se está configurando como coordinador.
- ✓ **Conexión.**

En la Figura 44 se observa el diagrama de conexión final de los dispositivos que conforman un nodo coordinador; y en la Tabla 15 se muestran los elementos utilizados para la implementación de este nodo.



**Figura 44:** Diagrama de conexión de los elementos de un Nodo Coordinador.

Fuente: Elaboración propia

**Tabla 16:** Elementos utilizados en un nodo Sensor.

Elemento	Valor/Tipo
1 XBee Pro	S2B
1 XBee Explorer	USB
1 Antena XBee	5dBi
1 Conector	RP-SMA

Fuente: Elaboración propia

### 5.2.6. Trama de datos de un XBee en Modo API.

Los módulos XBEE como ya se ha mencionado en la revisión de literatura, tienen dos formas de trabajo: en modo transparente y en modo API. En el modo transparente el dispositivo funciona como un puente Serial-Zigbee, donde toda información que se recibe por el XBEE es transmitida por el aire. Esta configuración es la más sencilla debido a que no requiere ningún tipo de ajuste, pero no es la más adecuada para la aplicación que describe este proyecto. Por ello se decide trabajar en el modo de comunicación API, en este modo de trabajo, la información desde y hacia otros módulos se hace a través de tramas API, las cuales nos brindan información variada sobre el host que emite la trama y el destino de la misma. [35]

Existen varios tipos de trama API, en la Figura 45 se enumeran algunas de estas:

Frame type	Description	Frame type	Description
0x08	AT command (immediate)	0x10	TX request
0x09	AT command (queued)	0x8B	TX response
0x17	Remote Command Request	0x90	RX received
0x88	AT command response	0x92	RX I/O data received
0x8A	Modem Status	0x95	Node Identification Indicator
		0x97	Remote Command Response

**Figura 45:** Tipos de tramas API de un módulo XBee.

Fuente: Building Wireless Sensor Networks by Robert Faludi

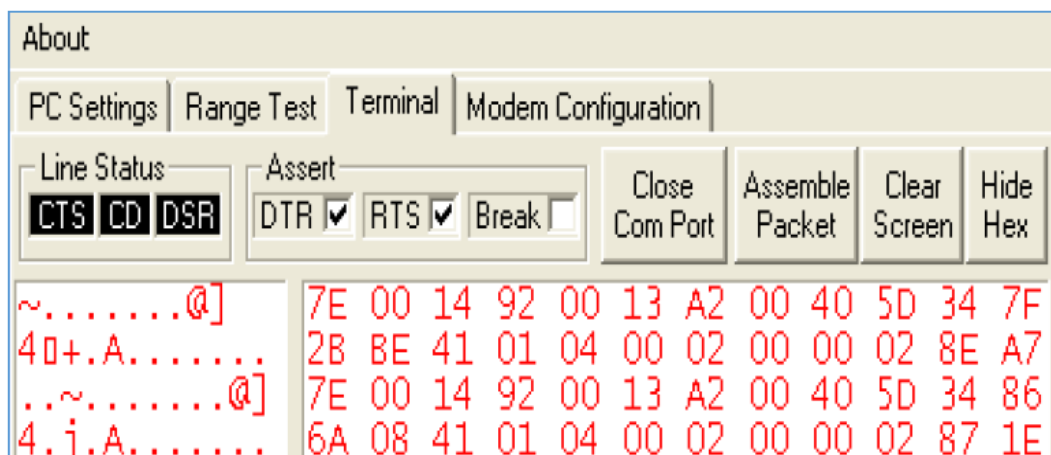
El tipo de trama API que interesa en este proyecto es la 0x92 RX I/O data received.

En ella es posible identificar el origen del paquete y visualizar las configuraciones de entradas analógicas o digitales con sus respectivos valores.

Una trama API inicia con el byte de comienzo de trama 0x7E seguido de 2 bytes que nos indican la longitud de la trama, luego una serie de bytes que nos dan la información del paquete de datos que lo podríamos llamar un campo de información y por último 1 byte llamado checksum. El campo de información, a su vez, lo podemos dividir en 2 partes, un byte que nos indica el tipo de trama y una serie de bytes conteniendo datos dependientes del tipo de trama.

Nuestros módulos XBee Pro S2B se configuraron como coordinador API y como Router AT, en este caso en nuestro dispositivo coordinador recibimos una trama API tipo 0x92 que nos proporciona dentro de su campo de información las mediciones analógicas o digitales tomadas por nuestros nodos remotos a través de los sensores de temperatura y humedad.

En la Figura 46 podemos visualizar a través de XCTU las tramas que llegan al Coordinador emitidas por los dispositivos finales que conforman la red de sensores. A continuación se explica cada Byte que conforma la trama.



**Figura 46:** Tramas API del nodo Coordinador.

Fuente: Elaboración propia.

```
7E 00 14 92 00 13 A2 00 40 5D 34 7F
2B BE 41 01 04 00 02 00 00 02 8E A7
```

**Figura 47:** Bytes de una trama API (Hexadecimal).

Fuente: Elaboración propia.

A continuación se describen los campos que conforman la trama de comunicación API que llegan al dispositivo Coordinador.

✓ Cabecera Trama API:	7E
✓ Longitud de trama:	00 14
✓ Tipo de trama:	92
✓ Dirección larga origen:	00 13 A2 00 40 5D 34 7F
✓ Dirección corta. (MY):	2B BE
✓ Opciones de recepción:	41
✓ Numero de muestras:	01
✓ Mapa de canales digitales habilitados:	04 00
✓ Mapa de entradas análogas habilitadas:	02 00
✓ Estados digitales habilitados:	00 00
✓ Estado análogos habilitados:	02 8e
✓ Checksum:	A7

Al conocer cada dato de la trama API, se puede deducir que los bytes 22 y 23 son los estados de los sensores colocados en las entradas análogas del módulo XBee-Pro S2B que emitió la trama.

### 5.2.7. INTERFAZ DE MONITOREO: Aplicación en VS Express 2013

Como se mencionó el prototipo de Red Inalámbrica de sensores está compuesto por un coordinador, y 2 lectores de variables ambientales que van a ir instalados en las tarjetas del nodo sensor desarrolladas.

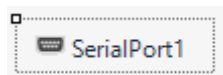
Cada módulo sensor se configuro para que automáticamente tome el valor de las variables ambientales y la transmita por la red hacia el coordinador, el cual retransmitirá las tramas

API (figura 47) de todos los sensores hacia la aplicación en el computador para visualizada en pantalla.

Para la elaboración de la aplicación de monitoreo se elige como lenguaje de programación la herramienta de Microsoft Corporation VS Express 2013 que ofrece una licencia académica de fácil acceso por medio del sitio web: <http://www.microsoft.com/express/Downloads/>. La función de la interfaz de monitoreo que se diseñó con esta herramienta es la de visualizar en tiempo real la información que fluye a través de la red. En este caso se tiene 2 nodos sensores enviando información cada 10 minutos, y por medio de la interfaz se conoce el estado de las variables (temperatura y humedad) del sitio donde se encuentra el modulo sensor. Esta información se puede visualizar en la aplicación mediante una gráfica y un archivo del histórico de las mediciones y a través de la visualización numérica de la última medición adquirida por cada sensor independientemente.

#### **5.2.7.1. Desarrollo de la Interfaz de Monitoreo.**

Una vez que las tramas Api llegan al Coordinador el módulo de comunicación inalámbrica XBee de este nodo es el encargado de enviarlas hacia el PC que se encarga de procesarlas; para lograr esto lo primero es establecer una comunicación física entre la tarjeta XBee Explorer USB y el software VS Express 2013, para esto, se implementó el control Serial Port (figura 48), incluido en la versión de VS Express 2013; este control permite abrir el puerto serial de un computador y leer o escribir datos en él.



**Figura 48:** Control Serial Port de VS Express 2013

**Fuente:** Elaboración propia.

Para realizar la apertura del puerto de transmisión y recepción de datos dentro del software VS Express 2013 debemos invocar las funciones que se muestran en la figura 49.

```
SerialPort.PortName = comboCOM.Text 'asignamos al puerto serie el COM seleccionado en el combobox
SerialPort.Open() 'Se abre el puerto seleccionado'
```

**Figura 49:** Apertura del control Serial Port de VS Express 2013.

**Fuente:** Elaboración propia.

Una vez abierto el puerto serie de comunicaciones el computador empieza a recibir los datos conforme van llegando hasta que se reúna la cantidad específica en un “buffer” (figura 50), el cual ha sido configurado previamente en 24 que es la longitud fija de las tramas que envían los nodos sensores de nuestro prototipo.

```
Public Class WSN
    'variables para la lectura del puerto serie
    Dim buffer(0 To 24) As Byte 'buffer de lectura, 24 bytes en mi caso
```

**Figura 50:** Buffer de lectura de los datos.

**Fuente:** Elaboración propia.

Cuando llegan los 24 bytes que corresponden a una trama API, el computador ejecuta una función de recepción, la cual inicialmente valida la trama y procede a decodificarla para abstraer la información contenida. (figura 51)

```
nbytes = SerialPort.BytesToRead 'Comprobamos la cantidad de bytes a leer
If nbytes = 24 Then
    SerialPort.Read(buffer, 0, nbytes) 'Si son 24, los volcamos en el vector "buffer"
    mydata = buffer(0)
    If mydata = 126 Then 'Comprobamos que el primer caracter recibido corresponde al delimitador de trama 7E'
        mydata = buffer(3)
        If mydata = 146 Then
            Me.Invoke(New EventHandler(AddressOf decodifica))
        End If
    End If
End If
```

**Figura 51:** Función de recepción de los datos.

**Fuente:** Elaboración propia.

Para conocer la procedencia de la trama que se va a decodificar, el en software se deben especificar las direcciones largas (Serial Number) de cada sensor que se desee monitorear, estas se establecen dentro de la programación de la aplicación como constantes tal como lo muestra la figura 52.

```

Const Sensor1 = "0013A20040C5FDE2"
Const Sensor2 = "0013A20040C5FDF3"

```

**Figura 52:** Definición de direcciones para cada sensor en el software.

**Fuente:** Elaboración propia.

El numero serie de cada nodo sensor y el valor analógico leído por el mismo, se abstraen de la trama a través de la función “decodifica” (figura 53), la cual adicionalmente calcula el valor de temperatura y humedad en base a la medición en mV.

```

Private Sub decodifica()

    myMy = Hex(CByte(buffer(4))).PadLeft(2, "0") +
           Hex(CByte(buffer(5))).PadLeft(2, "0") +
           Hex(CByte(buffer(6))).PadLeft(2, "0") +
           Hex(CByte(buffer(7))).PadLeft(2, "0") +
           Hex(CByte(buffer(8))).PadLeft(2, "0") +
           Hex(CByte(buffer(9))).PadLeft(2, "0") +
           Hex(CByte(buffer(10))).PadLeft(2, "0") +
           Hex(CByte(buffer(11))).PadLeft(2, "0")

    temperatura = (buffer(19) * 256) + buffer(20) 'unimos los 10 bits separados al nacer :)
    RH = (buffer(21) * 256) + buffer(22)

```

**Figura 53:** Función para decodificar la trama de datos recibida.

**Fuente:** Elaboración propia.

Los números de serie se comparan mediante la función que actualiza la ventana de la aplicación, para luego graficar, visualizar y registrar el valor del respectivo sensor que envió la información, como lo muestran las figuras 54 y 55.

```

Public Sub SetText(ByVal text As String, ByVal ref As String)
    Select Case ref
        Case Sensor1
            If Me.Label17.InvokeRequired Then
                Dim d As New SetTextCallback(AddressOf SetText)
                Me.Invoke(d, New Object() {text, ref})
            Else
                Me.Label17.Text = FormatNumber(text, 1)
                Me.Label15.Text = FormatNumber(RH, 1)
                Charttemporq.Series("°C").Points.AddXY(DateTime.Now.ToString, temperaturaC.ToString)
                Charthumorq.Series("RH (%)").Points.AddXY(DateTime.Now.ToString, RH.ToString)
            End If
    End Select

```

**Figura 54:** Función para actualizar la información del Sensor 1 en la aplicación.

**Fuente:** Elaboración propia.

```

Case Sensor2
  If Me.Label8.InvokeRequired Then
    Dim d As New SetTextCallback(AddressOf SetText)
    Me.Invoke(d, New Object() {text, ref})
  Else
    Me.Label8.Text = FormatNumber(text, 1)
    Me.Label9.Text = FormatNumber(RH, 1)
    Charttempinv.Series("°C").Points.AddXY(DateTime.Now.ToString, temperaturaC.ToString)
    Charthuminv.Series("RH (%)").Points.AddXY(DateTime.Now.ToString, RH.ToString)
  End If

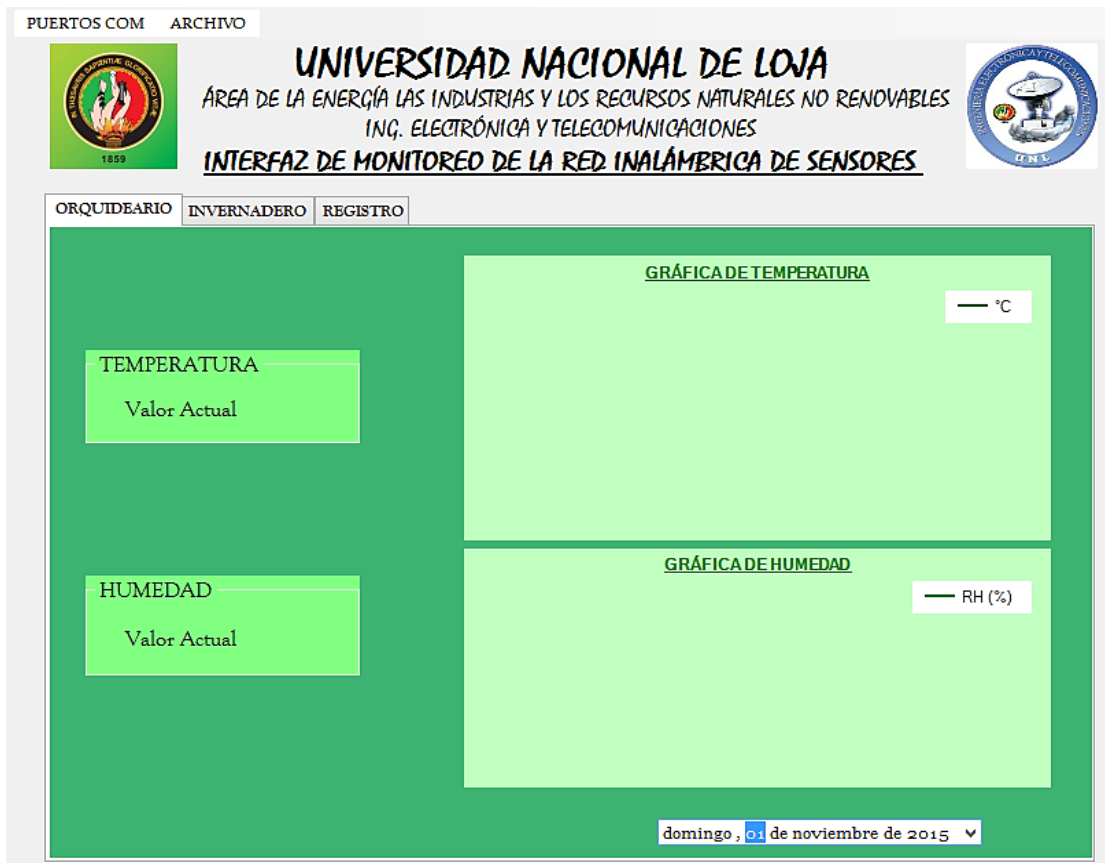
```

**Figura 55:** Función para actualizar la información del Sensor 2 en la aplicación.

Fuente: Elaboración propia.

### 5.2.7.2. Descripción de la Interfaz de monitoreo.

La figura 56 muestra una pantalla general de la interfaz de monitoreo desarrollada en Visual Studio Express 2013.



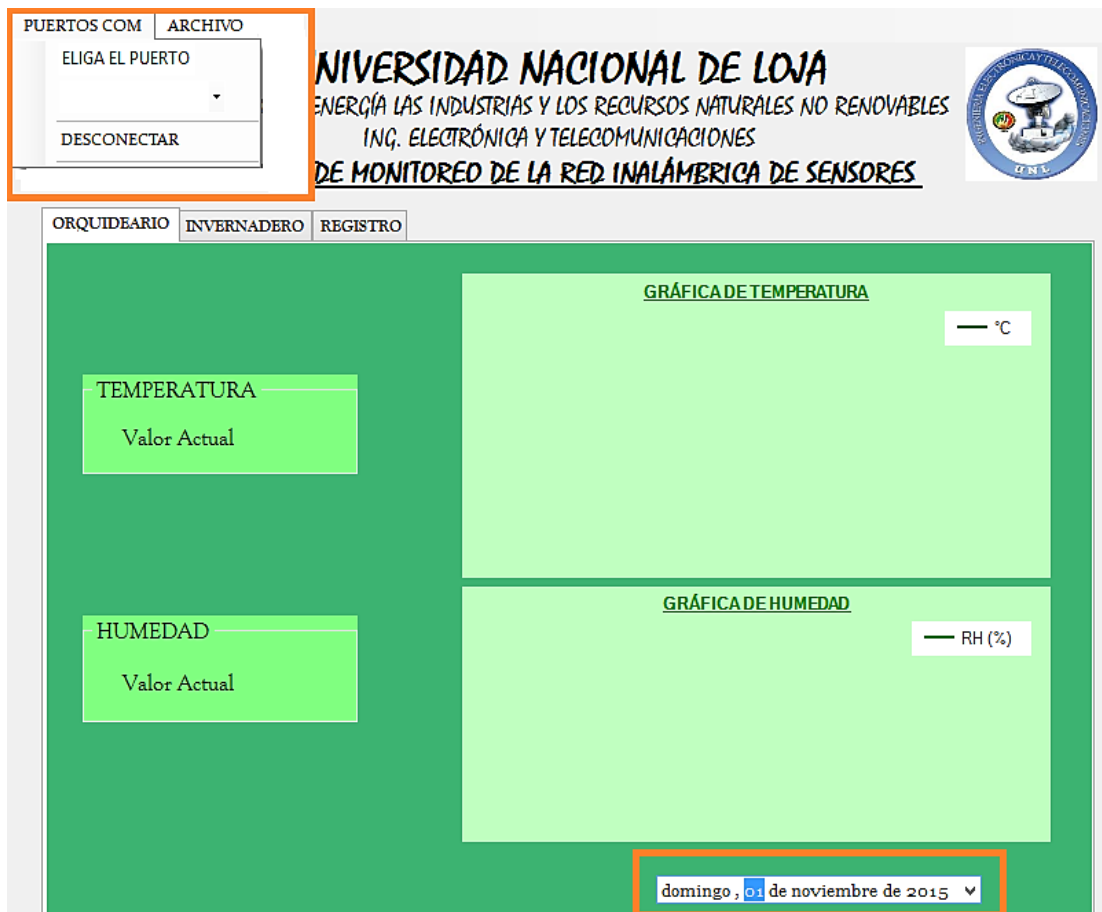
**Figura 56:** Vista general de la Interfaz de Monitoreo.

Fuente: Elaboración propia.



Las secciones que se definen en la aplicación son:

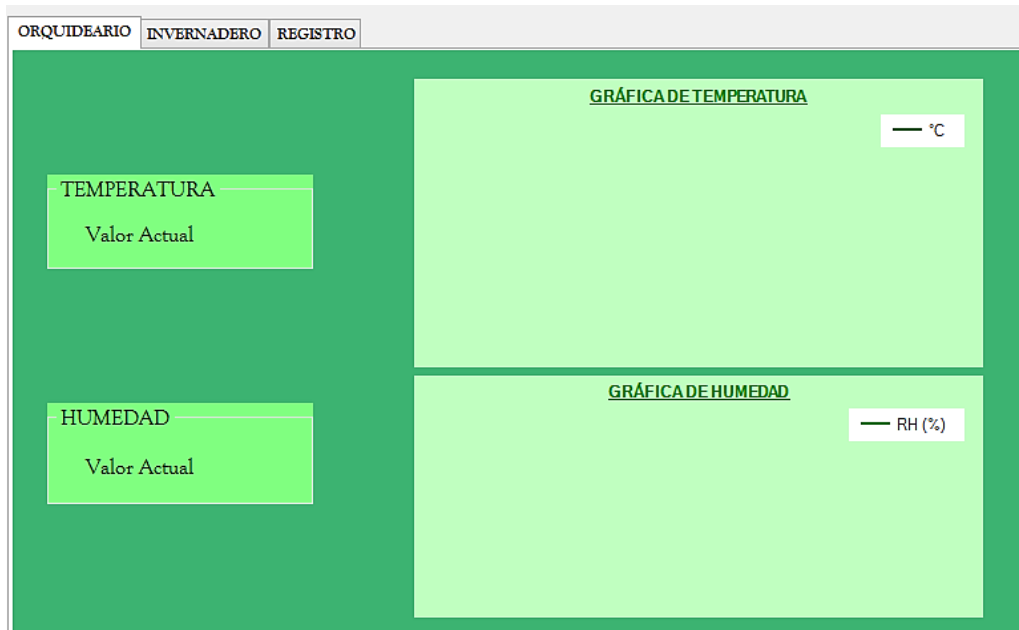
- ✓ Monitoreo del canal virtual USB. (Figura 57)
  - Estado (Conectado o Desconectado).
  - Elección del Puerto de comunicación.
  - Fecha y hora.



**Figura 57:** Monitoreo del canal virtual USB a través de la aplicación.

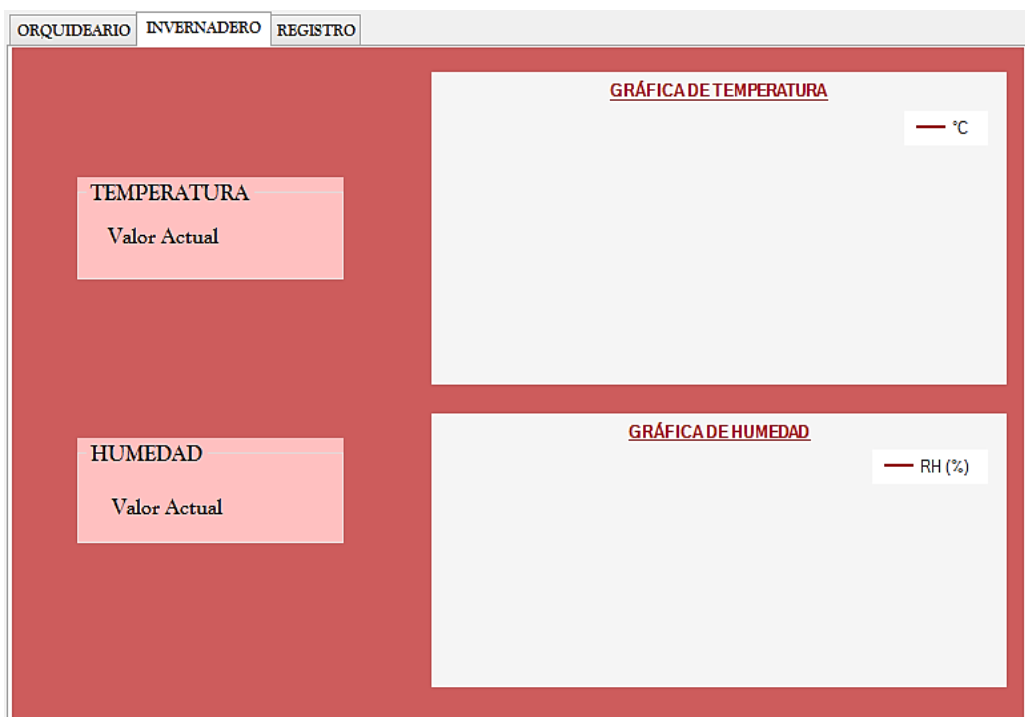
**Fuente:** Elaboración propia.

- ✓ Monitoreo de las variables físicas (temperatura y humedad), de las distintas dependencias (Orquideario-Invernadero) en donde están ubicados los nodos sensores a través de indicadores y gráficos. (Figuras 58-59).



**Figura 58:** Pantalla de monitoreo de variables ambientales sector Orquideario.

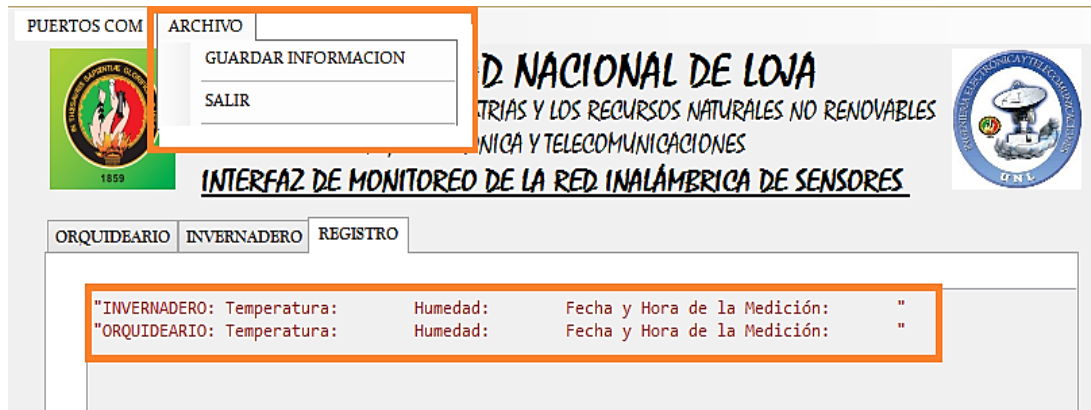
Fuente: Elaboración propia.



**Figura 59:** Pantalla de monitoreo de variables ambientales sector Invernadero.

Fuente: Elaboración propia.

- ✓ Registro y almacenamiento de los datos. (Figura 60)



**Figura 60:** Pantalla de registro y almacenamiento de los datos de monitoreo.

**Fuente:** Elaboración propia.

### 5.3. CÁLCULOS. [3]

#### 5.3.1. Cálculo del RSSI (Nivel de intensidad de la señal Recibida).

La factibilidad de un enlace de comunicación inalámbrica depende principalmente del nivel de potencia de la señal obtenido en la recepción y de la sensibilidad del receptor.

Teóricamente un enlace radioeléctrico es factible siempre que se alcance al menos un valor de recepción igual al valor de la sensibilidad del dispositivo de comunicación inalámbrica, sin embargo de manera práctica se recomienda un margen de seguridad mínimo de unos 10 dB sobre este valor, por ello para garantizar la eficiencia de la red es necesario calcular el nivel de intensidad de la señal recibida (RSSI), por el receptor inalámbrico utilizado en el diseño de la WSN.

El nivel de intensidad de la señal recibida, es el nivel de potencia de recepción de una señal en relación a un nivel de referencia de 1mW y se calcula mediante la ecuación 6.

$$\text{RSSI(dBm)} = -10 \log \left( \frac{P_{\text{Rx}}}{1\text{mW}} \right) \text{ Ecuacion 6}$$

Donde:

- ✓  $P_{RX}$  = Potencia de recepción.

Por tanto el RSSI es la medida de la potencia de la señal recibida por el receptor expresada en dBm.

### 5.3.2. Cálculo de la Potencia de recepción.

Para calcular la potencia de recepción se realiza un balance de potencias en base a las características técnicas de los módulos de comunicación inalámbrica XBee Pro S2B y mediante la ecuación 7

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} - L_t \text{ Ecuación 7}$$

Donde:

- ✓  $P_{TX}$  = Potencia de transmisión del módulo XBee Pro S2B = 63mW= 18dBm;
- ✓  $G_{TX}$  = Ganancia de la antena de transmisión del módulo XBee Pro S2B = 5dBi;
- ✓  $G_{RX}$  = Ganancia de la antena de recepción del módulo XBee Pro S2B = 5dBi;
- ✓  $L_t$  = Pérdidas en el enlace inalámbrico

### 5.3.3. Cálculo de las pérdidas en el enlace inalámbrico.

Las pérdidas totales de propagación  $L_t$ , para un enlace radioeléctrico a través de un medio cualquiera se obtienen de la sumatoria de la pérdida básica de propagación en condiciones de espacio libre  $L_o(dB)$ , más la atenuación de campo  $L_v(dB)$ , en este caso la atenuación ocasionada por la presencia de vegetación en el escenario de monitorización (Jardín Botánico Reinaldo Espinosa), tal como lo determina la ecuación 8.

$$L_t = L_o(dB) + L_{v-ITU}(dB) \text{ Ecuación 8}$$

La pérdida básica de propagación se calcula mediante la ecuación 9.

$$L_o(\text{dB}) = 32.4 + 20 \log(f) + 20 \log(d) \text{ Ecuación 9}$$

Donde:

- ✓  $f$  = Frecuencia de operación del enlace radioeléctrico en MHz.
- ✓  $d$  = Distancia a cubrir en el enlace en km.

El cálculo de las pérdidas ocasionadas por la presencia de vegetación  $L_v(\text{dB})$ , en el área de monitorización se realiza según la norma UIT-R P833<sup>5</sup>, mediante la ecuación 10, que fue desarrollada por la unión internacional de telecomunicaciones, para calcular la atenuación por vegetación en enlaces que operan en la banda de UHF. (Ultra High Frequency 300MHz – 3GHz), y fue planteada para casos en los que la distancia entre el transmisor y el receptor de un enlace inalámbrico es menor a 400m.

$$L_{v-ITU}(\text{dB}) = 0.2f^{0.3}d^{0.2} \text{ Ecuación 10.}$$

Donde:

- ✓  $f$  = Frecuencia de operación del enlace radioeléctrico en MHz.
- ✓  $d$  = Distancia a cubrir en el enlace en m.

Con el fin de evaluar el rendimiento del enlace en el escenario de monitorización (225m x 225m), se realizó el cálculo del nivel de potencia de la señal recibida entre un nodo coordinador y un nodo sensor a diferentes distancias (50m 100m 200m 300m), asegurando la eficiencia del enlace a 225m de distancia, esto a través de las siguientes características.

- ✓  $P_{Tx}$  = Potencia de transmisión = 63mW= 18dBm;
- ✓  $G_{Tx}$  = Ganancia de la antena de transmisión = 5dBi;

---

<sup>5</sup> [https://www.itu.int/dms\\_pubrec/itu-r/rec/p/R-REC-P.833-7-201202-S!!PDF-E](https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.833-7-201202-S!!PDF-E)  
Revista Tecnológica ESPOL – RTE, Vol. 28, N. 3, 115-130, (Noviembre 2015)  
<http://www.rte.espol.edu.ec/index.php/tecnologica/article/download/381/265.pdf>

- ✓  $G_{RX}$  = Ganancia de la antena de recepción = 5dBi;
- ✓ Frecuencia de operación 2.4 GHz.

#### 5.3.4. Cálculo del nivel de potencia de la señal recibida a 50m.

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} - L_t$$

$$L_t = L_o(dB) + L_v(dB)$$

$$L_{v-ITU}(dB) = 0.2f^{0.3}d^{0.2}$$

$$L_{v-ITU}(dB) = 0.2(2400)^{0.3}(50)^{0.2}$$

$$L_{v-ITU}(dB) = 4.517 \text{ dB}$$

$$L_o(dB) = 32.4 + 20 \log(f) + 20 \log(d)$$

$$L_o(dB) = 32.4 + 20 \log(2400) + 20 \log(0.05)$$

$$L_o(dB) = 73.98 \text{ dB}$$

$$L_t = L_o(dB) + L_v(dB)$$

$$L_t = 73.98 \text{ dB} + 4.5 \text{ dB}$$

$$L_t = 78.48 \text{ dB}$$

$$P_{RX}(dBm) = P_{TX}(dBm) + G_{TX}(dBi) + G_{RX}(dBi) - L_t(dB)$$

$$P_{RX}(dBm) = 18 + 5 + 5 - 78.45$$

$$P_{RX}(dBm) = -50.48 \text{ dBm} //$$

$$RSSI(50m) = P_{RX} \text{ dBm} (50m) = -50.48 \text{ dBm} //$$

#### 5.3.5. Cálculo del nivel de potencia de la señal recibida a 100m.

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} - L_t$$

$$L_t = L_o(dB) + L_v(dB)$$

$$L_{v-ITU}(dB) = 0.2f^{0.3}d^{0.2}$$

$$L_{v-ITU}(dB) = 0.2(2400)^{0.3}(100)^{0.2}$$

$$L_{v-ITU}(dB) = 5.18 \text{ dB}$$

$$L_o(dB) = 32.4 + 20 \log(f) + 20 \log(d)$$

$$L_o(dB) = 32.4 + 20 \log(2400) + 20 \log(0.1)$$

$$L_o(dB) = 80 \text{ dB}$$

$$L_t = L_o(dB) + L_v(dB)$$

$$L_t = 80 \text{ dB} + 5.18 \text{ dB}$$

$$L_t = 85.18 \text{ dB}$$

$$P_{RX}(dBm) = P_{TX}(dBm) + G_{TX}(dBi) + G_{RX}(dBi) - L_t(dB)$$

$$P_{RX}(dBm) = 18 + 5 + 5 - 85.18$$

$$P_{RX}(dBm) = -57.18 \text{ dBm//}$$

$$RSSI(100m) = P_{RX} \text{ dBm}(100m) = -57.18 \text{ dBm//}$$

### 5.3.6. Cálculo del nivel de potencia de la señal recibida a 200m.

$$P_{RX} = P_{TX} + G_{TX} + G_{RX} - L_t$$

$$L_t = L_o(dB) + L_v(dB)$$

$$L_{v-ITU}(dB) = 0.2f^{0.3}d^{0.2}$$

$$L_{v-ITU}(dB) = 0.2(2400)^{0.3}(200)^{0.2}$$

$$L_{v-ITU}(dB) = 5.96 \text{ dB}$$

$$L_o(dB) = 32.4 + 20 \log(f) + 20 \log(d)$$

$$L_o(dB) = 32.4 + 20 \log(2400) + 20 \log(0.2)$$

$$L_o(dB) = 86.02 \text{ dB}$$

$$L_t = L_o(dB) + L_v(dB)$$

$$L_t = 86.02 \text{ dB} + 5.96 \text{ dB}$$

$$L_t = 91.98 \text{ dB}$$

$$P_{R_x}(dBm) = P_{T_x}(dBm) + G_{T_x}(dBi) + G_{R_x}(dBi) - L_t(dB)$$

$$P_{R_x}(dBm) = 18 + 5 + 5 - 91.98$$

$$P_{R_x}(dBm) = -62.98 \text{ dBm//}$$

$$RSSI(100m) = P_{R_x} \text{ dBm} (200m) = -62.98 \text{ dBm//}$$

### 5.3.7. Cálculo del nivel de potencia de la señal recibida a 300m.

$$P_{R_x} = P_{T_x} + G_{T_x} + G_{R_x} - L_t$$

$$L_t = L_o(dB) + L_v(dB)$$

$$L_{v-ITU}(dB) = 0.2f^{0.3}d^{0.2}$$

$$L_{v-ITU}(dB) = 0.2(2400)^{0.3}(300)^{0.2}$$

$$L_{v-ITU}(dB) = 6.78 \text{ dB}$$

$$L_o(dB) = 32.4 + 20 \log(f) + 20 \log(d)$$

$$L_o(dB) = 32.4 + 20 \log(2400) + 20 \log(0.1)$$

$$L_o(dB) = 91.45 \text{ dB}$$

$$L_t = L_o(dB) + L_v(dB)$$

$$L_t = 91.45 \text{ dB} + 6.78 \text{ dB}$$

$$L_t = 98.23 \text{ dB}$$

$$P_{R_x}(dBm) = P_{T_x}(dBm) + G_{T_x}(dBi) + G_{R_x}(dBi) - L_t(dB)$$

$$P_{R_x}(dBm) = 18 + 5 + 5 - 98.23$$

$$P_{R_x}(dBm) = -70.23 \text{ dBm//}$$

$$RSSI(100m) = P_{R_x} \text{ dBm} (300m) = -70.23 \text{ dBm//}$$



## 6. RESULTADOS.

En la primera parte de esta sección se describen los resultados obtenidos de la simulación de la Red Inalámbrica de sensores realizada en NS2, mediante la cual se determinó la mejor topología física a utilizar para garantizar eficiencia en la Red.

Así mismo se describen las pruebas que se realizaron para garantizar el funcionamiento del prototipo de WSN, y se analizan los resultados.

### 6.1. Resultados de la simulación de la Red de sensores realizada en NS2.

#### ✓ Topología en Estrella.

En la Figura 61 se muestra el resultado del archivo .nam para la topología estrella.



**Figura 61:** Resultados del archivo .nam (Topología estrella)

Fuente: Elaboración propia.

La Figura 61 nos permite observar la estructura de la Red inalámbrica de sensores WSN bajo una topología en estrella, la misma que está compuesta por once nodos, diez de ellos configurados como nodos finales y uno como nodo coordinador, de igual forma se

observan los enlaces inalámbricos entre los distintos nodos que forman la red; por tratarse de una topología física en estrella todos los datos provenientes de los nodos finales tienen como destino el nodo coordinador.

La simulación de la red en estrella arrojó como resultado un archivo de trazas .tr como el de la Figura 62.

```

estrella.tr (-/TESIS/TOPO_ESTRELLA) - gedit
-----
s -t 0.002560000 -Hs 0 -Hd -1 -Ni 0 -Nx 25.00 -Ny 25.00 -Nz 0.00 -Ne -1.000000 -
NL MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
r -t 0.003008033 -Hs 3 -Hd -1 -Ni 3 -Nx 20.00 -Ny 33.66 -Nz 0.00 -Ne -1.000000 -
NL MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
r -t 0.003008033 -Hs 1 -Hd -1 -Ni 1 -Nx 20.00 -Ny 16.34 -Nz 0.00 -Ne -1.000000 -
NL MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
r -t 0.003008033 -Hs 4 -Hd -1 -Ni 4 -Nx 30.00 -Ny 33.66 -Nz 0.00 -Ne -1.000000 -
NL MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
r -t 0.003008033 -Hs 6 -Hd -1 -Ni 6 -Nx 30.00 -Ny 16.34 -Nz 0.00 -Ne -1.000000 -
NL MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
r -t 0.003008033 -Hs 2 -Hd -1 -Ni 2 -Nx 15.00 -Ny 25.00 -Nz 0.00 -Ne -1.000000 -
NL MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
r -t 0.003008033 -Hs 5 -Hd -1 -Ni 5 -Nx 35.00 -Ny 25.00 -Nz 0.00 -Ne -1.000000 -
NL MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
  
```

**Figura 62:** Resultados del archivo .tr (Topología estrella)

Fuente: Elaboración propia.

En la Figura 63 se muestra el resultado que se obtuvo del simulador NS2 al aplicar al archivo de trazas estrella.tr (Figura 62) el filtro AWK respectivo.

```

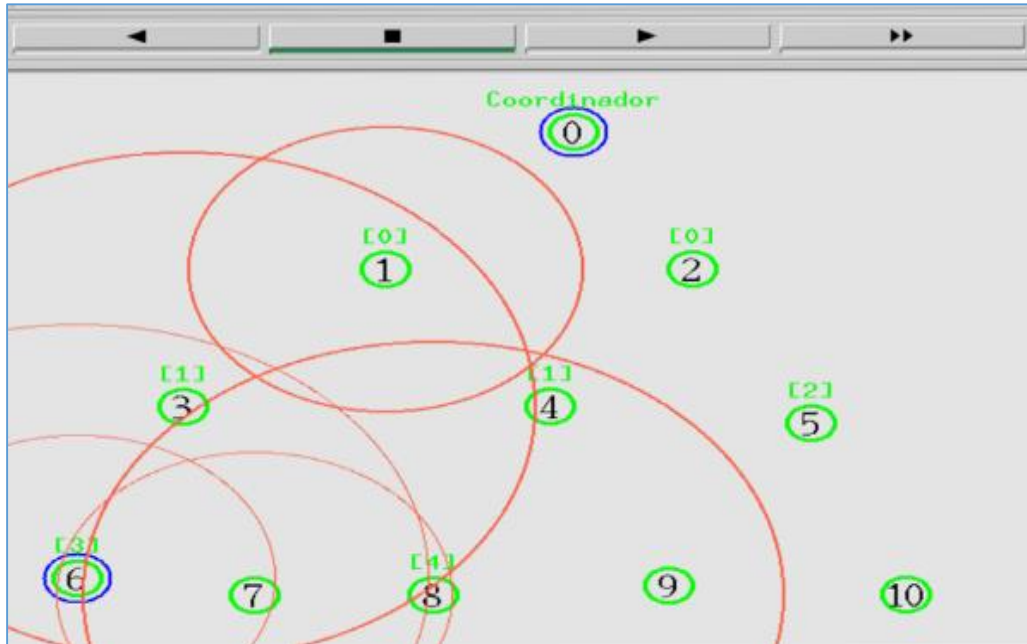
***** Estadísticas de la Topologia Estrella*****
Total Packets Enviados      :      1548
Total Packets Recibidos    :      1538
Total Packets Desechados   :       224
Total Packets Reenviados   :       541
Packet Delivery Ratio     :       99
Throughput de la red (KBps) :       0.7690
Promedio End-to-End Delay  :       0.016213630 ms
Total Energy Consumida     :      46662.835839
  
```

**Figura 63:** Resultado general de la simulación en NS2. (Topología estrella)

Fuente: Elaboración propia.

### ✓ Topología en Árbol:

En la Figura 64 se muestra el resultado del archivo .nam para la topología árbol.



**Figura 64:** Resultados del archivo .nam (Topología Arbol)

Fuente: Elaboración propia.

La Figura 64 nos permite observar la estructura de la Red Inalámbrica de Sensores WSN, bajo una topología en árbol la misma que al igual que la topología en estrella está compuesta por once nodos, 5 de ellos configurados como nodos finales, 5 como nodos routers y uno como nodo coordinador, así mismo se observan los enlaces inalámbricos entre estos.

Por tratarse de una topología física en árbol todos los datos provenientes de los nodos finales pasan por un nodo router que es a su vez el encargado de enviar los datos hasta el nodo coordinador.

La simulación de la red en estrella arrojó como resultado un archivo de trazas .tr como el de la Figura 65.

```

arbol.tr (~-/TESIS/TOPO_ARBOL) - gedit
Arbrir Guardar Deshacer
arbol.tr x
s -t 0.000640000 -Hs 0 -Hd -1 -Ni 0 -Nx 30.00 -Ny 40.00 -Nz 0.00 -Ne -1.000000 -
Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
r -t 0.001088031 -Hs 2 -Hd -1 -Ni 2 -Nx 35.00 -Ny 32.00 -Nz 0.00 -Ne -1.000000 -
Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
r -t 0.001088038 -Hs 1 -Hd -1 -Ni 1 -Nx 22.00 -Ny 32.00 -Nz 0.00 -Ne -1.000000 -
Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
s -t 0.141760000 -Hs 0 -Hd -1 -Ni 0 -Nx 30.00 -Ny 40.00 -Nz 0.00 -Ne -1.000000 -
Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
s -t 0.282560000 -Hs 0 -Hd -1 -Ni 0 -Nx 30.00 -Ny 40.00 -Nz 0.00 -Ne -1.000000 -
Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
s -t 0.421632000 -Hs 0 -Hd -1 -Ni 0 -Nx 30.00 -Ny 40.00 -Nz 0.00 -Ne -1.000000 -
Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 0 -Mt 0
s -t 0.502560000 -Hs 1 -Hd -1 -Ni 1 -Nx 22.00 -Ny 32.00 -Nz 0.00 -Ne -1.000000 -
Nl MAC -Nw --- -Ma 0 -Md ffffffff -Ms 1 -Mt 0

```

**Figura 65:** Resultados del archivo .tr (Topología Arbol)

**Fuente:** Elaboración propia.

En la Figura 66 se muestra el resultado que se obtuvo del simulador NS2 al aplicar al archivo de trazas arbol.tr (Figura 65) el filtro AWK respectivo.

```

***** Estadísticas de la Topología Árbol*****
Total Packets Enviados      :      2290
Total Packets Recibidos    :      1029
Total Packets Desechados   :      1893
Total Packets Reenviados   :      2224
Packet Delivery Ratio      :         44
Throughput de la red (KBps) :      0.5145
Promedio End-to-End Delay  :      0.341449245 ms
Total Energy Consumida     :      46645.452310

```

**Figura 66:** Resultado general de la simulación en NS2. (Topología Árbol)

**Fuente:** Elaboración propia.

En la Tabla 16 se muestran los resultados que se obtuvieron al comparar las métricas de evaluación de WSN que se definieron para cada topología tales como:

- ✓ Packet Delivery Ratio (%)
- ✓ Throughput de la Red (Kbps)
- ✓ End-to-End Delay (ms)
- ✓ Consumo de energía (J).

**Tabla 17:** Resultado de la comparación entre las métricas usadas en la simulación.

<b>Parámetros.</b>	<b>Topología Estrella</b>	<b>Topología Árbol</b>
<b>Packet Delivery Ratio (%)</b>	99	44
<b>Throughput de la Red (Kbps)</b>	0.7690	0.5145
<b>End-to-End Delay (ms)</b>	0.016213630	0.341449245
<b>Consumo de energía (J)</b>	46662.835839	46645.452310

Fuente: Elaboración propia.

### **6.1.1. Análisis de los resultados de la simulación de la Red inalámbrica de sensores.**

Como se puede observar en la tabla 16 la topología en estrella presenta mejores resultados en cuanto a eficiencia de la red; pues la relación entre los paquetes que se envían hacia el nodo coordinador y los paquetes que se reciben satisfactoriamente es de un 99% en comparación al 44% que presenta la topología en Árbol.

De igual forma durante el tiempo de simulación fluyen por la Red en topología estrella 0.77 Kbps; frente a los 0.51 Kbps que fluyen por la red configurada bajo una topología física en árbol, se observa que el volumen de tráfico o de información (Throughput) que fluye a través de la WSN es mayor en una red configurada en estrella.

El tiempo medio que tarda un paquete en llegar a su destino se cuantifico mediante el retardo de extremo a extremo (End-to-End Delay), como muestra la tabla 16 en topología estrella un paquete tarda en llegar al nodo coordinador en promedio 0.016ms en comparación con los 0.341ms que se tarda en topología en árbol.

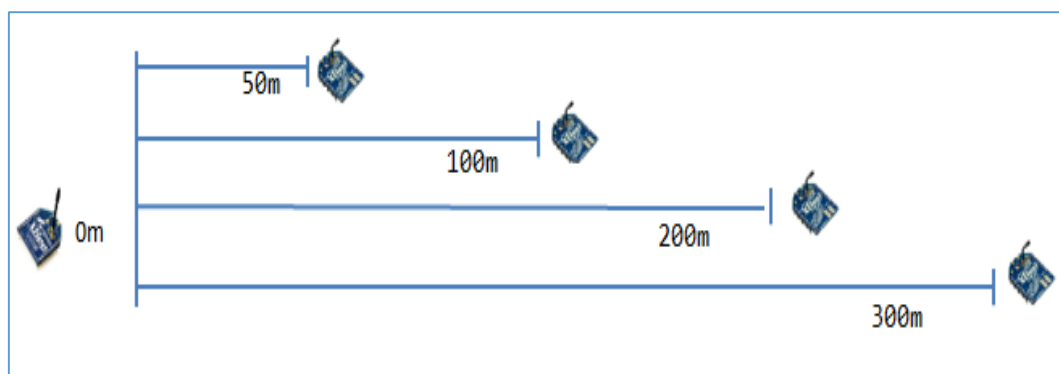
Finalmente se puede evidenciar que el consumo de energía es relativamente menor si tenemos configurada nuestra red bajo topología en árbol pues presenta un consumo total de 46645 J frente a los 46662 J que consume la red configurada en topología en estrella.

Al analizar los resultados obtenidos de cada topología de la red inalámbrica de sensores mediante las métricas de evaluación (pruebas de envío de paquetes eficiencia y consumo de energía) se determina a la topología estrella como la más adecuada para la implementación de una red inalámbrica de sensores ambientales eficiente en el Jardín Botánico Reinaldo Espinosa.

## 6.2. Pruebas y Resultados del prototipo de WSN.

### 6.2.1. Prueba de Recepción de Señal.

Para medir el nivel de recepción de señal en exteriores, entre los módulos XBee Pro S2B se realizó pruebas a 50 m, 100 m, 200 m, y 300 m de distancia entre los módulos tal como lo muestra la Figura 67. Para esta prueba se toman medidas de recepción de paquetes e indicador de fuerza de señal de recepción (RSSI: Receive Signal Strength Indication) a través de la pestaña “Range Test” del software XCTU. Esta configuración consiste en el envío de paquetes de 32 bytes de un dispositivo a otro el cual a su vez devuelve la trama de datos hacia el origen.

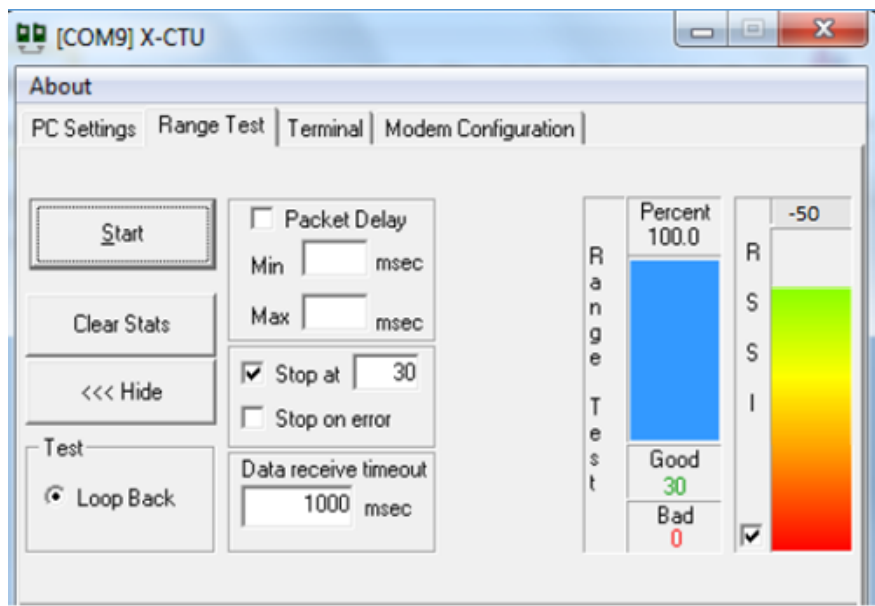


**Figura 67:** Diagrama para pruebas de recepción de señal en exteriores

Fuente: Elaboración propia.

Para medir el nivel de recepción de señal entre los módulos XBee en ambientes externos, se utilizó dos portátiles, dos placas de conexión serial, dos módulos XBee Pro S2B, el programa X-CTU y su herramienta Range-test.

✓ **50 metros.**



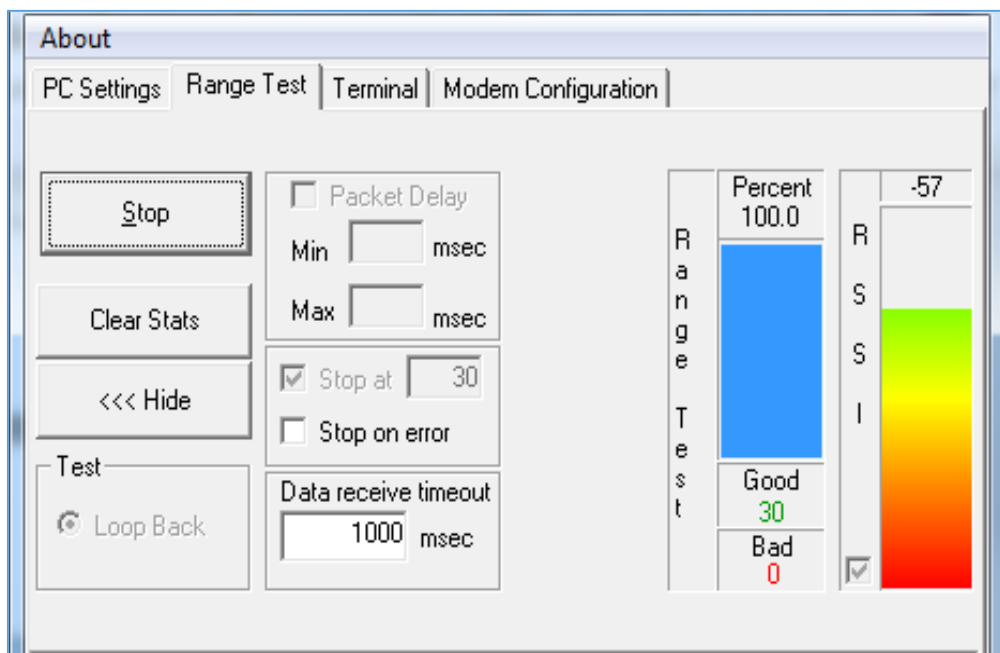
**Figura 68:** Prueba 50 metros exteriores

Fuente: Elaboración propia.

En la figura 68 se muestra el resultado de la prueba de recepción de señal que se realizó a 50 m de distancia entre dos dispositivos XBee Pro S2B, como se observa se mantiene un 100% de recepción de paquetes con un nivel indicador de fuerza de señal (RSSI) de -50dBm, lo que concuerda con el cálculo realizado en la sección 5.3.4, a una distancia de 50m el nivel de potencia de la señal de recepción calculado fue igual a -50.48 dBm.

✓ **100 metros.**

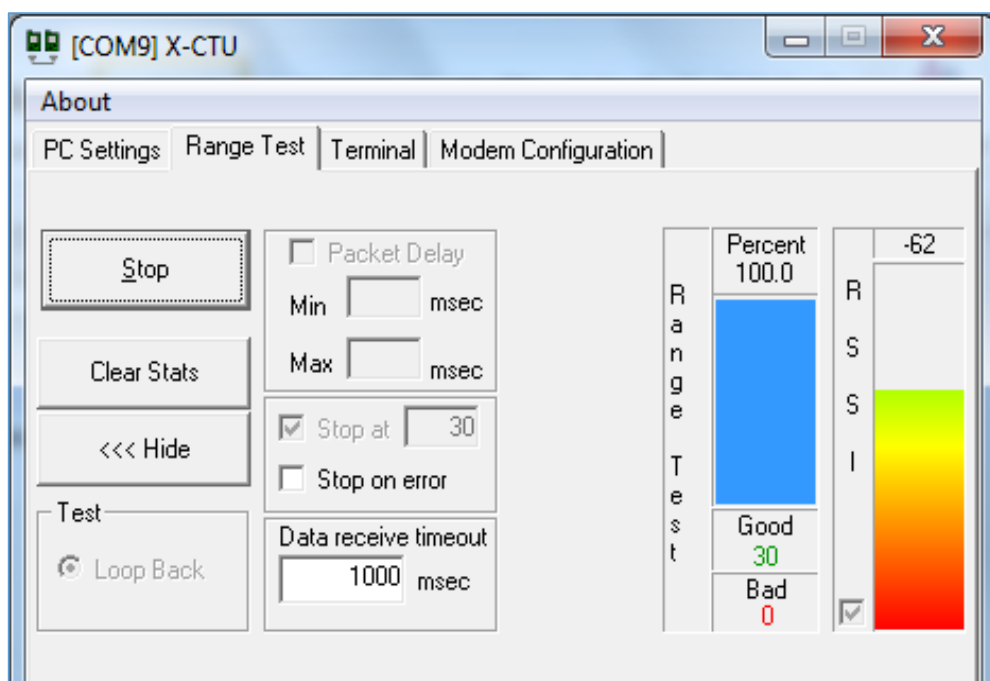
En la prueba realizada a una distancia de 100m, se mantiene un 100% de recepción de paquetes y un nivel de recepción de -57 dBm (nivel de potencia), tal como lo muestra la figura 69; lo que concuerda con el cálculo realizado en la sección 5.3.5, a una distancia de 100m el nivel de potencia de la señal de recepción calculado fue igual a -57.18 dBm.



**Figura 69:** Prueba 100 metros exteriores

Fuente: Elaboración propia.

✓ 200 metros



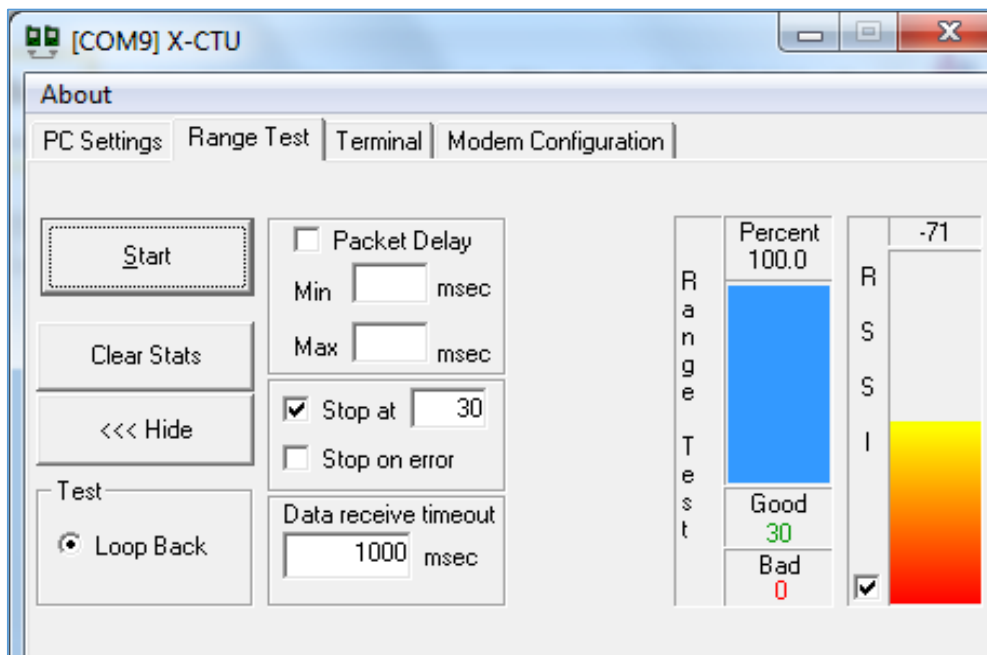
**Figura 70:** Prueba 200 metros exteriores

Fuente: Elaboración propia.



A la distancia de 200m, se mantiene un 100% de recepción de paquetes con un nivel de recepción de  $-62\text{dBm}$ , como se muestra en la figura 70; lo que concuerda con el cálculo realizado en la sección 5.3.6, a una distancia de 200m el nivel de potencia de la señal de recepción calculado fue igual a  $-62.98\text{ dBm}$ .

✓ **300 metros**

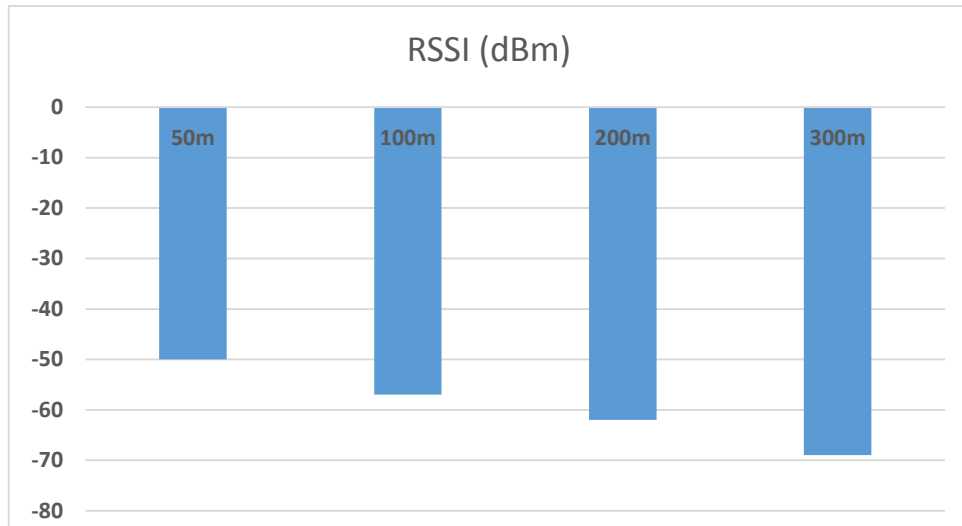


**Figura 71:** Prueba 300 metros exteriores

Fuente: Elaboración propia.

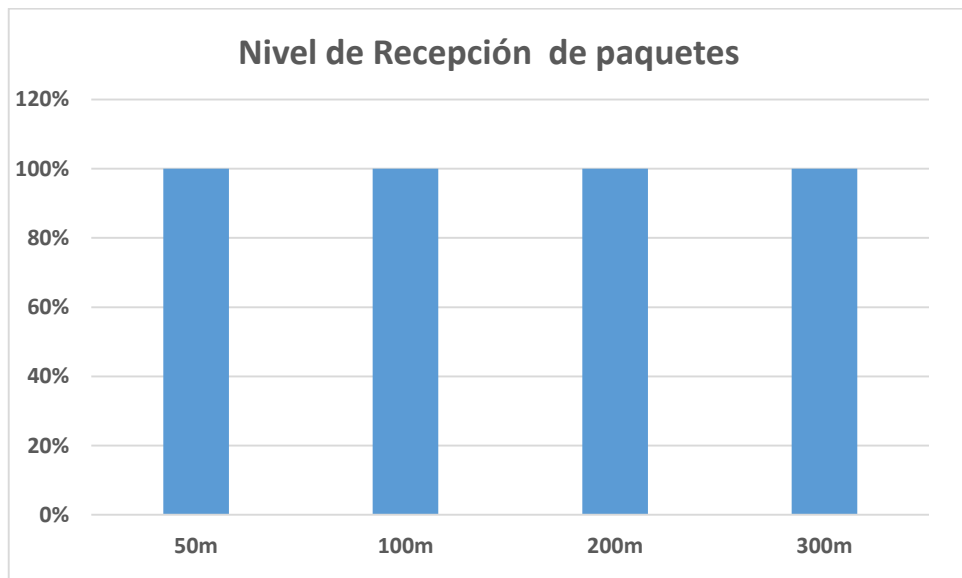
A una distancia de 300 se mantiene un 100% de recepción de paquetes con un nivel indicador de fuerza de señal de  $-71\text{dBm}$ . (Figura 71), lo que concuerda con el cálculo realizado en la sección 5.3.7, a una distancia de 300m el nivel de potencia de la señal de recepción calculado fue igual a  $-70.23\text{ dBm}$ .

Los resultados generales obtenidos al realizar la prueba de recepción de señal en exteriores son representados en las Figura 72 y 73.



**Figura 72:** Resultados de RSSI en exteriores.

Fuente: Elaboración propia.



**Figura 73:** Resultados del nivel recepción de paquetes en exteriores.

Fuente: Elaboración propia.

### 6.2.1.1. Interpretación de resultados de la prueba de recepción de señal.

Según datos del fabricante los módulos XBee-PRO S2B en ambientes externos presentan una potencia de salida de 63mW (18dBm), la sensibilidad del receptor cuando llega al valor de -102 dBm tendrá una tasa de error de 1% de paquetes perdidos, esto permite operar a 1600 metros en exteriores; sin embargo en la practica el alcance de estos módulos en ambientes externos depende de diversos factores ajenos a la potencia de transmisión y

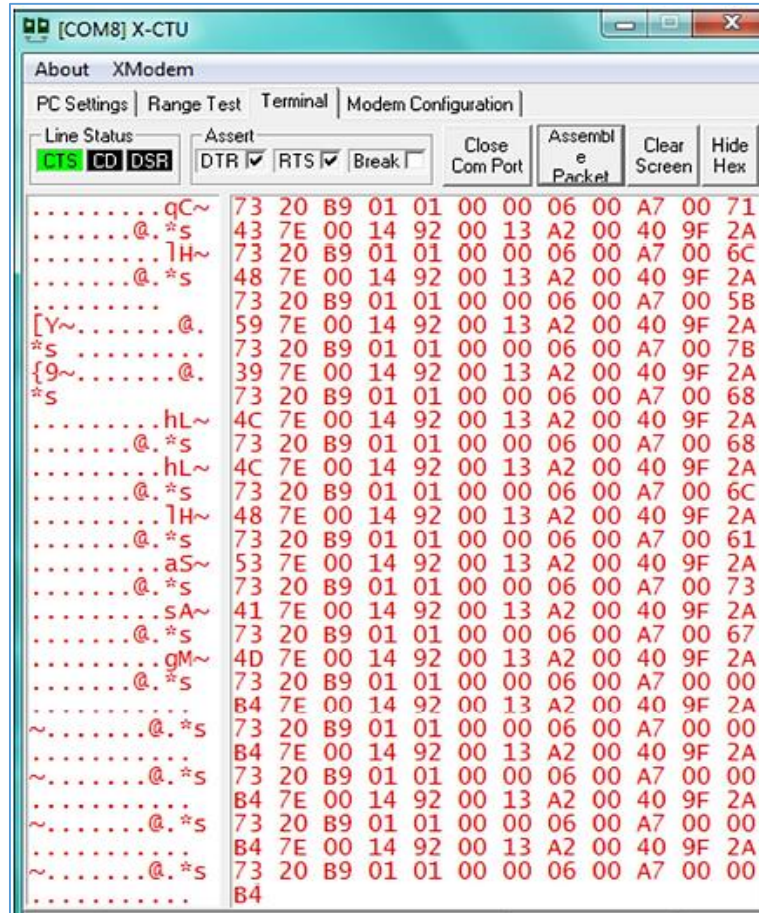
sensibilidad de recepción como la atenuación por espacio libre, ganancias de las antenas, atenuaciones de campo etc.

Las pruebas para determinar la potencia de recepción en exteriores se realizaron en el escenario en donde se desplegara la red inalámbrica de sensores (jardín botánico Reinaldo Espinosa), según las pruebas realizadas a una distancia de 50-100-200-y 300m el nivel de recepción de los paquetes en todos los casos es de un 100% (Figura 73) y el nivel de potencia de la señal de recepción en ningún caso es menor a la sensibilidad de recepción (-102 dBm) permitida por el dispositivo, los resultados obtenidos en esta prueba concuerdan con los cálculos efectuados en la sección 5.3, por tanto se corrobora que la eficiencia de la red no se ve afectada; al igual mediante esta prueba se constató que entre mayor es la distancia de comunicación entre dos módulos XBee disminuye el RSSI (Receive Signal Strength Indication), como se ve en los resultados de la Figura 72, esto en correspondencia a la ecuación 6 que determina una relación inversamente proporcional entre la distancia y el nivel de potencia de recepción de la señal, se puede decir que al aumentar la distancia entre el transmisor y receptor disminuye el nivel de potencia de recepción de la señal.

#### **6.2.2. Prueba de envío y recepción de datos entre el nodo sensor y el nodo coordinador.**

Para esta prueba se utilizó en un extremo el nodo sensor con su respectivo XBee Pro S2b configurado como (router/sensor), y al otro una PC conectada a la placa de desarrollo USB del módulo coordinador de la Red que cuenta con XBee Pro S2B configurado con el firmware de XBee (coordinador API).

Mediante el software X-CTU y pestaña “*Terminal*” se pudo observar la cadena de caracteres que el nodo sensor envió hacia el coordinador tal como lo muestra la Figura 85.



**Figura 74:** Tramas API que llegan al coordinador enviadas por el nodo sensor.

Fuente: Elaboración propia.

### 6.2.2.1. Interpretación de resultados de la prueba de envío y recepción de datos.

Se observa que el nodo router/sensor envía su trama de caracteres con datos provenientes de los sensores hacia el dispositivo coordinador de la Red, de esta forma se puede evidenciar que el nodo periférico trabaja correctamente, y que los nodos esclavos se encuentran direccionados correctamente hacia el nodo maestro.

### 6.2.3. Prueba General del funcionamiento del prototipo de WSN.

Para esta prueba los nodos que se diseñaron se colocaron en los exteriores del Jardín botánico Reinaldo Espinosa y para evitar que existan daños por factores externos como el sol, lluvia, viento, etc. en los componentes electrónicos que los conforman, se diseñó una caja para impermeabilizar; la antena del módulo de comunicación XBee Pro S2B se

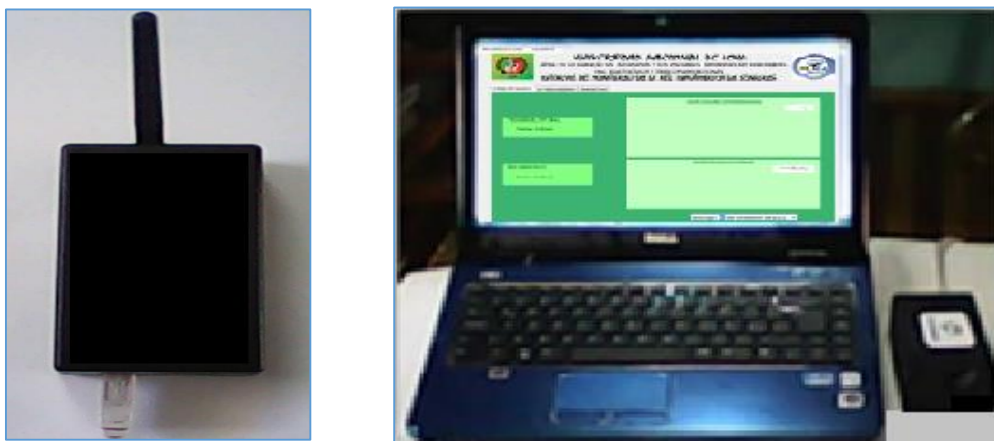
dejó fuera de la caja para lograr que la señal no se atenué y alcance mayor distancia, los sensores se ubicaron de tal forma que se obtengan los datos con mayor precisión. Cada nodo o dispositivo final queda tal como se ve en la Figura 86.



**Figura 75:** Nodo sensor.

Fuente: Elaboración propia.

Por otra parte el coordinador; estará ubicado junto a la máquina que tiene la aplicación de monitoreo. La Figura 87 muestra el dispositivo coordinador conectado a la pc.



**Figura 76:** Nodo Coordinador.

Fuente: Elaboración propia.

Una vez que la red se pone en ejecución; cada uno de los nodos recoge información de temperatura y humedad a través de los sensores que están conectados a la placa Arduino, este procesa la información y envía una trama de datos cada 10 minutos para que el

coordinador al recibirla a través de los módulos de comunicación XBee identifique de que nodo provienen, el nodo coordinador está directamente conectado a una computadora en donde corre una aplicación que es la encargada de procesar los datos provenientes de los dos nodos sensores y finalmente estos datos son presentados en tiempo real mediante una aplicación.

Como se observa en la Figura 88 una vez puesta en funcionamiento la WSN; la interfaz de monitoreo nos permite elegir el puerto de comunicación serial de donde se extraerá la información perteneciente a los nodos sensores.

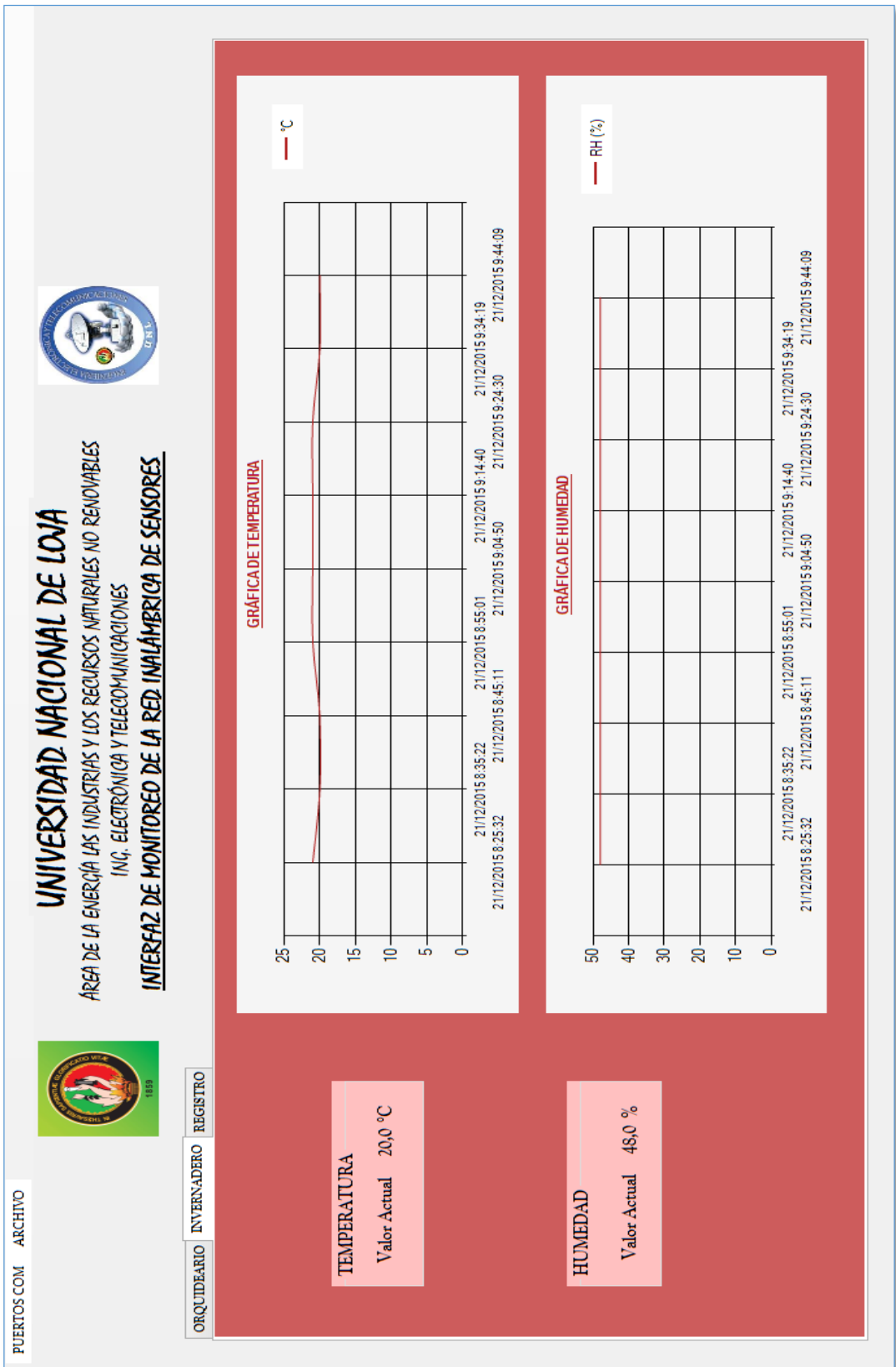


**Figura 77:** Interfaz de monitoreo (Selección del puerto COM)

Fuente: Elaboración propia.

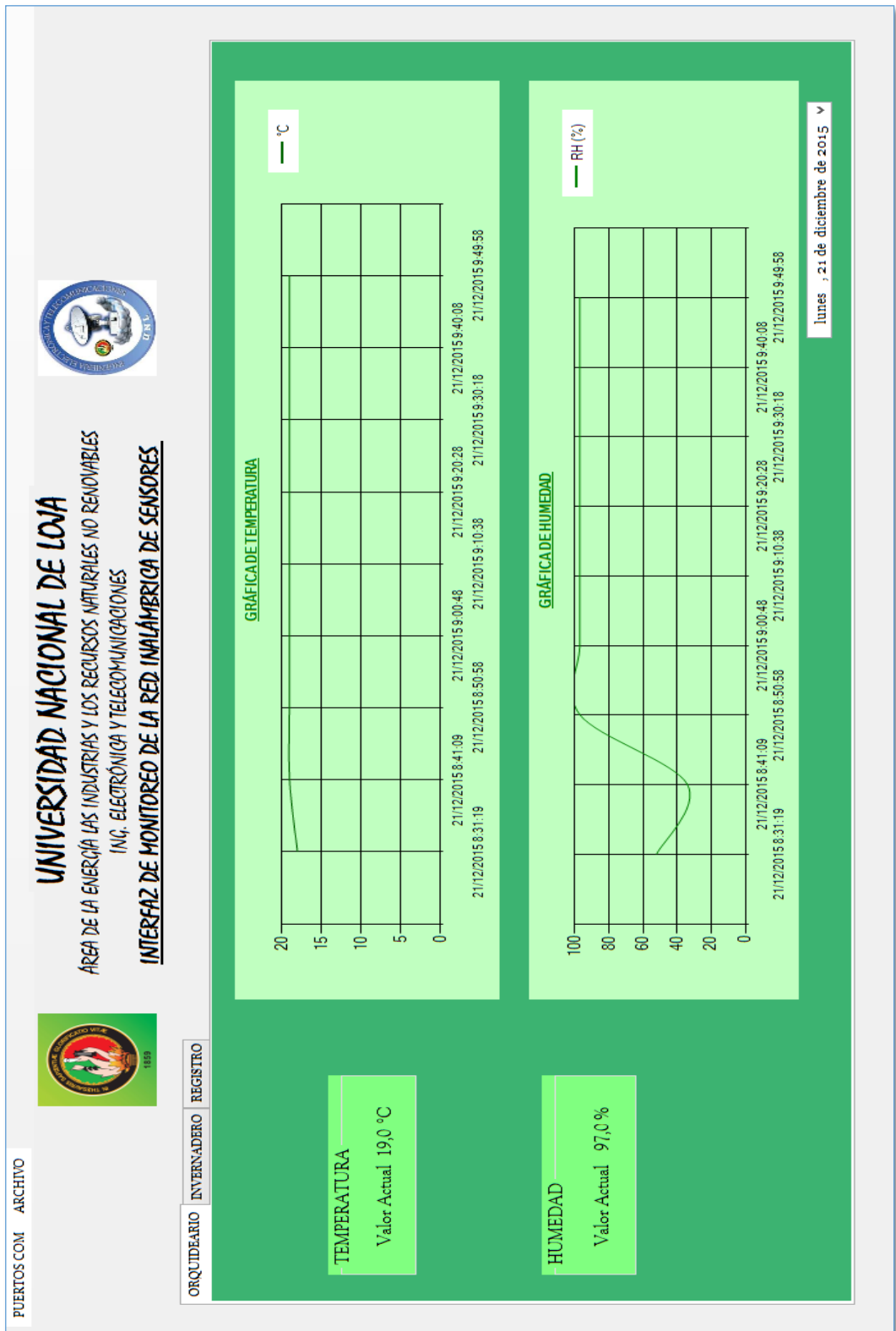
De igual forma se puede monitorear a través de la interfaz el valor actual de temperatura y humedad del sector en donde se ubica nuestro nodo sensor así como una gráfica del histórico de sus mediciones como se muestra en la Figura 89 y 90.

El sistema al iniciar el monitoreo guarda los datos y estos se visualizan mediante la pestaña de registro de la Interfaz en ella se registran parámetros como el lugar de ubicación del nodo, el valor de la temperatura, y humedad del suelo y la fecha y hora de la medición. (Figura 91), así mismo la interfaz permite guardar estos datos en un archivo fácil de visualizar para su posterior análisis mediante la pestaña Archivo opción Guardar información. (Figura 92)



**Figura 78:** Monitoreo de variables ambientales a través de la Interfaz gráfica. (Sector Invernadero).

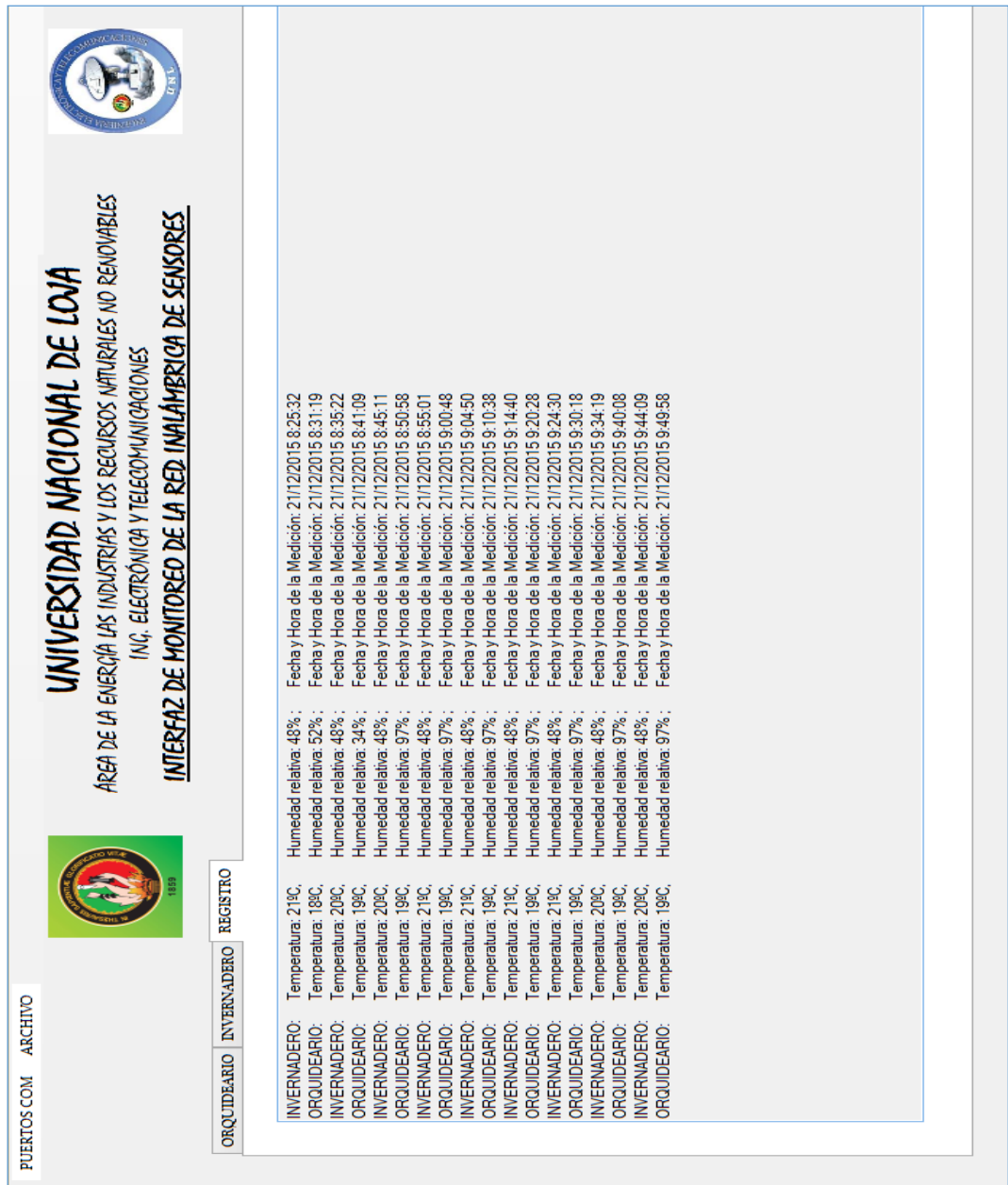
Fuente: Elaboración propia.



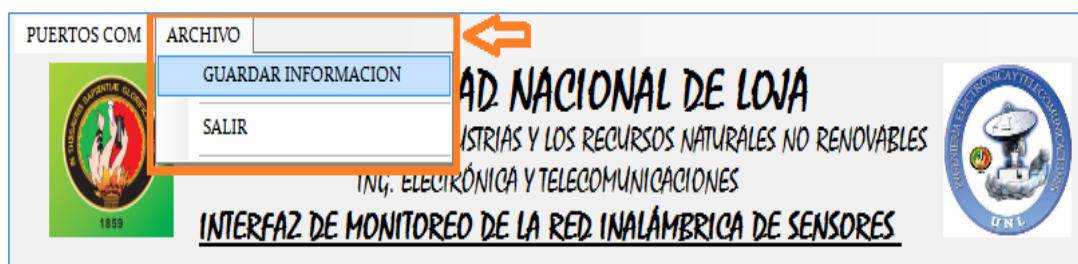
**Figura 79:** Monitoreo de variables ambientales a través de la Interfaz gráfica. (Sector Orquideario).

Fuente: Elaboración propia.





**Figura 80:** Registro de mediciones a través de la Interfaz gráfica.  
**Fuente:** Elaboración propia.



**Figura 81:** Pestaña para guardar el registro de mediciones  
**Fuente:** Elaboración propia.

#### 6.2.4. Costo del prototipo implementado

El análisis del costo del dispositivo está basado en los precios actuales de los elementos y dispositivos utilizados, estos se detallan en la Tabla 17, así mismo se presenta una estimación del costo del diseño e implementación del hardware y software del prototipo de WSN.

**Tabla 18.** Presupuesto del prototipo de WSN.

Descripción	Cantidad	Valor Unitario	Valor Total
Módulo XBee Pro S2B	3	\$92.00	\$276.00
Batería 12V/1.3 A/h	2	\$17.50	\$35.00
Placa Arduino Uno V1.8.1	2	\$35.00	\$70.00
XBee Shield Pro S2B	2	\$18.00	\$36.00
Sensor de temperatura LM35DZ	2	\$2.50	\$5.00
Sensor de Humedad Sparkfun SEN92355P	2	\$19.00	\$38.00
Antena XBee 5dBi	3	\$18.00	\$54.00
XBee Explorer USB	1	\$33.00	\$33.00
Caja para el montaje de los nodos Sensores	2	\$15.00	\$30.00
Caja para el montaje del nodo Coordinador	1	\$10.00	\$10.00
Cable USB a Mini USB	1	\$7.00	\$7.00

Cables y plug para Arduino	2	\$2.00	\$4.00
Diseño y construcción del hardware del prototipo de WSN	-----	\$200	\$200
Diseño y construcción del software del prototipo de WSN	-----	\$700	\$700
<b>TOTAL</b>			<b>\$1507,00</b>

## 7. DISCUSIÓN.

Con los resultados de este trabajo se confirmó la factibilidad de la implementación de una red inalámbrica para su aplicación en el área ambiental. El desarrollo de un prototipo de WSN bajo el estándar IEEE802.15.4 permitió realizar de manera eficiente el monitoreo de variables ambientales en el escenario de monitorización (Jardín Botánico Reinaldo Espinosa); al conocer y almacenar en tiempo real las mediciones de parámetros ambientales de las diferentes secciones del jardín botánico.

El estudio del estándar de comunicación inalámbrica IEEE 802.15.4 (ZigBee) y de las redes inalámbricas de sensores realizado previo al desarrollo de este sistema; permitió determinar las bases y criterios de diseño relevantes para el desarrollo del prototipo de WSN en el entorno de aplicación requerido.

Al identificar las ventajas en cuanto a latencia, consumo de energía, cobertura, tamaño de la red, costos, versatilidad de adaptación en diversos escenarios y aplicaciones del estándar IEEE 802.15.4 (ZigBee), frente a otros estándares de redes inalámbricas de área personal como IEEE 802.15.1 (Bluetooth) y 802.11 (WiFi), se corroboró los motivos por los que ZigBee es el estándar más utilizado en el diseño e implementación de redes inalámbricas de sensores, es evidente que la expansión de la tecnología ZigBee obedece al uso de bandas libres de frecuencia, bajo coste, capacidad de funcionamiento en tiempo real y principalmente bajo consumo de potencia lo cual impone uno de los principales criterios de diseño de las WSN pues deben gastar la menor cantidad de energía para asegurar que sus baterías perduren el máximo tiempo posible.

Por tanto se confirmó que el uso de ZigBee puede reducir los costos de instalación de una red WSN, además de permitir a los usuarios crear una red de forma rápida y llegar a lugares de difícil acceso gracias a que los dispositivos basados en este estándar son utilizados como elementos de uso primario cuando se trata de aplicaciones que requieren de un bajo consumo de recursos y energía. Existen muchas áreas de interés para el aprovechamiento de esta tecnología, algunas incluso que todavía no han sido exploradas; por ello uno de los inconvenientes para el diseño de redes inalámbricas de sensores bajo

este estándar es el desconocimiento generalizado que aún se tiene acerca de las aplicaciones, manejo y manipulación especializada de las redes inalámbricas de sensores.

Los resultados de la evaluación de las topologías físicas de Wireless Sensor Network (WSN) mediante el software Network Simulator 2 (NS2), permitieron definir la mejor topología física a utilizar para el monitoreo de variables ambientales en el Jardín Botánico Reinaldo Espinosa, se determinó que la topología en estrella es la indicada para transmitir los datos recogidos de los sensores en tiempo real logrando un buen desempeño de envío de paquetes, de consumo de energía y de cobertura, cuenta con un Packet Delivery Ratio del 99%, Throughput de 0,77 Kbps, y una latencia igual a 0,016 ms. El inconveniente que se presentó en esta etapa del trabajo, se dio en la elección del simulador pues son todavía escasos los proveedores de software que incluyen el estándar IEEE 802.15.4 dentro sus herramientas de simulación, en la mayoría de simuladores de WSN los módulos que implementan IEEE 802.15.4/ ZigBee, son fruto de contribuciones externas y están aún en fase de desarrollo, por lo que no se garantiza el correcto funcionamiento del estándar dentro de estos, de la misma forma la documentación y el soporte técnico de dichos módulos es escaso; para el desarrollo de la simulación es importante conocer previamente el área o escenario en el que se va a implementar la red WSN, y familiarizarse con la arquitectura del simulador y sus lenguajes de programación.

Posteriormente se logró desarrollar un prototipo de red WSN para medir variables ambientales de temperatura y humedad en dos secciones del jardín botánico Reinaldo Espinosa (invernadero-orquideario); éste se diseñó bajo topología en estrella, y contó con dos nodos finales y un nodo coordinador, basados en el estándar IEEE 802.15.4, arduino, un computador como gestor de datos, sensores, y el módulo de comunicación inalámbrica XBEE PRO S2B. En lo que respecta a la interfaz de usuario diseñada en Visual Studio, esta cumplió su objetivo al permitir el monitoreo de la red de sensores, así como usar la información para interpretarla y visualizarla gráficamente.

Gracias a las pruebas de campo que se realizaron se pudo corroborar el correcto funcionamiento de los nodos sensores, nodo coordinador y de la interfaz gráfica, pues el

prototipo permitió gestionar en tiempo real la información proporcionada por los sensores.

Mediante las pruebas de cobertura se logró verificar que los dispositivos de comunicación inalámbrica XBee PRO S2B son adecuados para implementar una Red Inalámbrica de sensores en el jardín botánico Reinaldo Espinosa, en cuyo escenario la máxima distancia a cubrir es de 225m y según los cálculos y pruebas de nivel de recepción de señal (RSSI), realizadas a distintas distancias (50m, 100m 200m 300m), el nivel de potencia de recepción no fue menor a la sensibilidad de recepción de estos dispositivos, por tanto no existió pérdida de paquetes; garantizando de esta manera la eficiencia de la WSN. El alcance del radioenlace en exteriores con presencia de pérdidas de propagación debidas a la vegetación, que se constató mediante pruebas de cobertura fue hasta los 300 metros, pero con posibilidades de alcanzar los 1600 metros (según datos del fabricante) siempre que se respete la condición de línea de vista entre el transmisor y el receptor.

El desempeño de los equipos en la teoría es muy diferente que cuando se los implementa físicamente, por lo tanto al tener una comparación, se podría determinar las variaciones que existen entre ellos y en qué afectarían al utilizarlos en ambientes reales.

En cuanto a la topología de red utilizada (estrella), en la práctica se pudo comprobar una alta eficiencia en la obtención de los datos en tiempo real, al igual que se pudo observar un menor costo al momento de implementarla. La desventaja radicó en la carga que recae sobre el nodo coordinador; la cantidad de tráfico que debió soportar en la aplicación de este trabajo es poca, pero pudo haber aumentado conforme se agregaran más nodos periféricos, lo que la hace poco recomendable si se tratara de desplegar la red en escenarios de monitorización de mayor tamaño.

Con el desarrollo del presente trabajo se pudo verificar que actualmente las WSN como tecnología han sido adoptadas he implementadas en distintas disciplinas y diversos proyectos que contribuyen principalmente a la vinculación con la sociedad, en el campo de la agricultura que es el campo de aplicación de esta investigación, una red de sensores con las características de WSN permite la fácil obtención, recolección y transmisión de

datos de distinta naturaleza, los mismos que al ser procesados y evaluados permitirán describir la situación actual del escenario en el que se despliega la red esto permitiría a expertos desarrollar modelos matemáticos que ayuden a pronosticar futuros cambios en el sistema, información importante para ciertos sectores dentro de la agricultura, también se podría pensar en sistemas de alertas tempranas activados por algún fenómeno ambiental que está siendo monitoreado y que permitirá precautelar la integridad física de cierta población.

## 8. CONCLUSIONES.

- ✓ La WSN constituyen una tecnología emergente prometedora y un campo de investigación activo en el ámbito de Redes inalámbricas para el desarrollo de la sociedad, su modelo de arquitectura de Red de bajo costo, fiable, de rápido y denso despliegue, con capacidades de auto organización y tolerante a fallas, les permite tener un amplio rango de aplicaciones que posibilitan la interacción del ser humano con el entorno.
- ✓ Se logró un gran aprendizaje sobre la tecnología ZigBee (estándar IEEE 802.15.4) se determinaron sus principales ventajas, desventajas y campo de aplicación frente a otras tecnologías WPAN como Bluetooth y Wi-fi. Concluyendo que esta tecnología puede reducir costos de instalación de una red WSN, además de permitir a los usuarios crear una red de forma rápida y llegar a lugares de difícil acceso.
- ✓ Al analizar los resultados de la simulación realizada en NS2 se determinó que la topología física adecuada para implementar una red inalámbrica de sensores para monitoreo de variables ambientales en el Jardín Botánico Reinaldo Espinosa es la topología estrella, cuenta con un Packet Delivery Ratio del 99%, Throughput de 0,77 Kbps, un delay de 0,016 ms, y un consumo de energía de 46662.835839 J similar al de la topología física en árbol.
- ✓ Se desarrolló un prototipo de red inalámbrica de sensores basada en el estándar IEEE 802.15.4 (ZigBee), para el monitoreo de variables ambientales (Temperatura y humedad relativa), apoyando con ello a que se desarrollen técnicas de agricultura de precisión.
- ✓ Al realizar las pruebas de recepción de la señal en exteriores y analizar los resultados, se determinó que los módulos de comunicación XBee PRO S2B son adecuados para implementar una red inalámbrica de sensores bajo topología estrella en el jardín botánico Reinaldo Espinosa, una de las principales cualidades de estos módulos es que están basados en el protocolo ZigBee lo que garantiza un



ahorro de energía en los dispositivos, por lo que es ideal el uso de baterías como fuente de alimentación de los nodos sensores.

- ✓ Se diseñó una interfaz HMI, que permite al usuario monitorear en tiempo real los datos proporcionados por el sistema de una forma eficiente, rápida, y sencilla.

## 9. RECOMENDACIONES.

- ✓ La herramienta NAM no es imprescindible dentro de una simulación en NS2 debido a que no aporta ningún tipo de datos o información con la que se pueda extraer conclusiones directas sobre el funcionamiento de una WSN, sin embargo se recomienda el uso de esta herramienta, sobre todo al principio, para entender que es lo que está sucediendo en la simulación y saber de esta forma cómo procesar los resultados obtenidos.
- ✓ Puesto que los archivos de trazas que se generan como resultado en el simulador NS2 son de gran tamaño y difíciles de entender a simple vista se recomienda que para su análisis e interpretación se investigue que representa cada campo de la trama de datos que se muestra en este archivo, de esta forma se puede extraer de la trama la información que es de relevancia para la aplicación.
- ✓ Durante el desarrollo de la red inalámbrica de sensores es importante realizar correctamente la configuración del módulo de comunicación inalámbrica del nodo coordinador debido a que es el único dispositivo que puede establecer y dar inicio a una red inalámbrica de sensores (WSN).
- ✓ Para que los módulos XBee funcionen adecuadamente es necesario regular los voltajes de alimentación a un voltaje constante de 3.3V; puesto que los dispositivos XBee son sensibles a fallas y un cambio brusco de voltaje puede dañar los módulos permanentemente.
- ✓ En la elección de los módulos de comunicación inalámbrica XBee se recomienda tomar en cuenta que si bien los módulos de la serie 1 y serie 2 tienen los mismos pines, estos no pueden comunicarse entre sí porque sus versiones no son compatibles.
- ✓ Cuando se trata de implementar una Red basada en dispositivos XBee se recomienda trabajar con modo de operación API y no AT; de tal manera que la

comunicación entre los dispositivos sea confiable y las tramas de datos que se envían nos proporcionen la mayor cantidad de información posible sobre la WSN.

- ✓ Para un trabajo a futuro se puede realizar una ampliación y mejora de este sistema, al aumentar diferentes tipos de sensores o actuadores dependiendo la aplicación a la que sea destinada la Red, de igual forma al contar con una Red Ip en el escenario de monitorización se puede interconectar a internet la WSN mediante diferentes opciones de gateways que ofrece el fabricante de los módulos XBee, de tal forma que se desarrolle un sistema de monitoreo desde cualquier ubicación geográfica.
  
- ✓ Es importante almacenar los datos que nos proporciona el prototipo de WSN; en la aplicación que se desarrolló se realiza un registro de estos datos mediante herramientas propias del software VS 2013, sin embargo existen softwares especializados en la creación de bases de datos tal es el caso de Workbench un programa que permite al diseñador crear bases de datos en MySQL, de esta forma la información de la red se encuentra almacenada de manera ordenada en una base de datos administrada que apoya actividades de agricultura de precisión tales como:
  - Irrigación: permite el aprovechamiento de lugares con escasos recursos de agua controlando el riego mediante las WSN.
  
  - Fertilización: se puede automatizar el proceso de dispersión de fertilizadores (cantidad, tiempo, lugar, etc.) para aprovechar el uso de la tierra, afectando el crecimiento de las plantas y su calidad.
  
  - Control de plagas: Esto mejora a reducir enfermedades en las plantas modificando las condiciones del ambiente en ciertas aplicaciones para que sean apropiadas para la planta.
  
  - Invernaderos: ya que se trata de ambientes cerrados, se puede modificar temperatura del aire, intensidad de la luz, humedad y temperatura de la tierra.

## **BIBLIOGRAFÍA.**

- [1] A. S. Tanenbaum, Computer Networks, Prentice-Hall, 2003. 3 Edición.
- [2] Redes De Computadoras un enfoque descendente, James F. Kurose 5 Edición.
- [3] Transmision por radio (7ª ed) Jose Hernando Rabanos; Luis Mendo Tomas; Jose Manuel Riera Salis , editorial universitaria ramon areces, 2013.
- [4] AKYILDIZ, I. AND VURAN, M., Wireless Sensor Networks (Advanced Texts In Communications And Networking)., 1a ed., New Jersey-EEUU., Ed. Mankrono Print Media., 2010., Pp. 413-502.
- [5] FALUDI, R., Building Wireless Sensor Networks., 1a ed., New York-Estados Unidos., Ed. BRIAN., 2010., 320p
- [6] GISLASON, D., Zigbee Wireless Networking., Washington- Estados Unidos., Ed. Elsevier., 2008., Pp.400-413

### BIBLIOGRAFIA DE INTERNET.

- [7] R. d. Computadoras, «Redes de Computadoras,» Marzo 2004. [En línea]. Available: <http://www.uoc.edu/masters/oficiales/img/922.pdf>. [Último acceso: 15 Marzo 2015].
- [8] «Introduccion a las Redes,» [En línea]. Available: <http://inech.weebly.com/uploads/1/2/6/2/12627663/redes.pdf>. [Último acceso: 15 marzo 2015].
- [9] J. R. Sánchez y H. M. V. Lendechy, «Las Redes Inalambricas,» Universidad Autonoma del Carmen, Septiembre-Octubre 2013. [En línea]. Available: <http://www.unacar.mx/contenido/difusion/acalan55pdf/contenido.pdf>. [Último acceso: 17 marzo 2015].
- [10] J. Garbarino, «Protocolos para redes inalambricas de sensores,» 7 noviembre 2011. [En línea]. Available: <http://materias.fi.uba.ar/7500/Garbarino.pdf>. [Último acceso: 18 marzo 2015].

- [11] S. R. M. Cantillo, «DESARROLLO DE APLICACIONES BASADAS EN WSN,» septiembre 2010. [En línea]. Available: <https://riunet.upv.es/bitstream/handle/10251/8592/PFC%20-%20DESARROLLO%20DE%20APLICACIONES%20BASADAS%20EN%20WSN.pdf?sequence=1>. [Último acceso: 20 marzo 2015].
- [12] M. G. M. J. Moreno, «ZIGBEE,» 23 mayo 2012. [En línea]. Available: <http://sx-de-tx.wikispaces.com/ZIGBEE>. [Último acceso: abril 2015].
- [13] B. L. R. P. Vasques y I. B. d. A. Coutinho, «ZIGBEE: Tipos de Dispositivos,» 2010. [En línea]. Available: [http://www.gta.ufrj.br/grad/10\\_1/zigbee/dispositivos.html](http://www.gta.ufrj.br/grad/10_1/zigbee/dispositivos.html). [Último acceso: Abril 2015].
- [14] J. Longares, «Introducción a Zigbee y las redes de sensores inalámbricas,» 7 Abril 2013. [En línea]. Available: <http://www.javierlongares.com/arte-en-8-bits/introduccion-a-zigbee-y-las-redes-de-sensores-inalambricas/>. [Último acceso: Abril 2015].
- [15] E. A. B. Hernandez, «Red ZigBee mediante modulos XBee,» Bucaramanga, 2010.
- [16] «ZigBee Wireless Protocolo,» [En línea]. Available: <http://ordenador.wingwit.com/Redes/local-networks/72109.html#.VlodB7jhDIV>. [Último acceso: abril 2015].
- [17] «TUTORIAL XBEE,» XBee.cl, [En línea]. Available: <http://xbee.cl/tutorial-xbee/>. [Último acceso: abril 2015].
- [18] D. p. A. O. R. p. P. A. E. M. I. M. LTDA., «Guía del Usuario XBEE,» w w w . o l i m e x . c l, Revisión Julio 2010.
- [19] sabasjimenez@ieee.org, «XBEE,» 26 septiembre 2011. [En línea]. Available: <http://tecnologicobj12.blogspot.com/2011/09/que-es-xbee.html>. [Último acceso: abril 2015].

- [20] A. Duarte, «Arduino y Xbee,» [En línea]. Available:  
<http://www.andresduarte.com/arduino-y-xbee> [Último acceso: mayo 2015].
- [21] R. P. Corbacho, «Elección de una Herramienta de simulacion,» [En línea]. Available:  
[http://bibliotecadigital.icesi.edu.co/biblioteca\\_digital/bitstream/10906/68447/1/eleccion\\_herramienta\\_simulacion.pdf](http://bibliotecadigital.icesi.edu.co/biblioteca_digital/bitstream/10906/68447/1/eleccion_herramienta_simulacion.pdf). [Último acceso: mayo 2015].
- [22] J. M. V. ´ a. Viejo, «Diseno y Desarrollo de una Aplicacion Para el Estudio Comparativo de Topologias de Red,» España, 2011.
- [23] NS. creators, «The Network Simulator - ns-2,» 19 septiembre 2014. [En línea]. [Último acceso: mayo 2015].
- [24] I. J. L. Muñoz, «Análisis de Protocolos con Net Simulator 2 (ns-2),» 2007. [En línea]. Available:  
<http://downloads.tuxpuc.pucp.edu.pe/linuxweek2007/martes/LW2007-NS2.pdf>. [Último acceso: mayo 2015].
- [25] M. Maracara, «Simulador NS-2,» 15 julio 2012. [En línea]. Available:  
<http://simuladors2.blogspot.com/>. [Último acceso: mayo 2015].
- [26] J. M. H. M., «NS2 - Network Simulator,» Valparaiso, 12 de mayo de 2004.
- [27] S. B. C., 3 mayo 2009. [En línea]. Available:  
<http://www.marblestation.com/?p=761>. [Último acceso: mayo 2015].
- [28] K. F. hkfall@ee.lbl.govi y Kannan Varadhan hkannan@catarina.usc.edui, The ns Manual, California: U. Berkeley, November 4; 2011.
- [29] J. C. Ugas, «Estudio y análisis de prestaciones de redes móviles mediante simulaciones NS-2 para validar modelos analíticos,» Barcelona- Cataluña, Noviembre 2009.
- [30] R. J. Balcells J. y e. Marcombo, «Sensores y Actuadores,» [En línea]. Available:  
<http://isa.uniovi.es/docencia/autom3m/Temas/Tema7.pdf>. [Último acceso: Junio 2015].

- [31] T. Instruments, «LM35 Precision Centigrade Temperature Sensors,» enero 2015. [En línea]. Available: <http://www.ti.com/lit/ds/symlink/lm35.pdf>. [Último acceso: junio 2015].
- [32] M. Angelou, «SENSORS / TEMPERATURE,» 29 julio 2012. [En línea]. Available: <https://learn.adafruit.com/tmp36-temperature-sensor>. [Último acceso: junio 2015].
- [33] L. T. Ojed, «ARDUINO UNO R3,» [En línea]. Available: <http://arduino.cl/arduino-uno/>.
- [34] D. I. I. B. R. East, «XBee®/XBee-PRO® ZB RF Modules,» marzo 2012. [En línea]. Available: <http://www.adafruit.com/datasheets/XBee%20ZB%20User%20Manual.pdf>. [Último acceso: junio 2015].
- [35] R. Faludi, Building Wireless Sensor Networks, Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo: Published by O'Reilly Media, 2011.

## 10. ANEXOS.

### ANEXO I: ABREVIATURAS.

<u>ACRÓNIMO</u>	<u>DESCRIPCIÓN</u>
<b>A/D</b>	Analógico/Digital.
<b>AODV</b>	Ad Hoc On-Demand Vector Routing.
<b>API</b>	Aplicación de Interfaz de programación.
<b>AT</b>	Comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el usuario y un terminal modem.
<b>CBR</b>	Constant Byte Rate, (tráfico de bytes a una velocidad constante).
<b>CSMA/CA</b>	Acceso múltiple por detección de portadora con evasión de colisiones.
<b>DC</b>	Corriente Directa.
<b>DH</b>	Dirección de Destino en Alto
<b>DL</b>	Dirección de Destino en Bajo.
<b>E/S</b>	Entradas/Salidas.
<b>FFD</b>	Dispositivo ZigBee de funcionalidad completa.
<b>RFD</b>	Dispositivo ZigBee de funcionalidad reducida.
<b>GND</b>	Ground (Conexión a tierra).
<b>HMI</b>	Interfaz Hombre-Máquina.
<b>IEEE</b>	Institute of Electrical and Electronics Engineers.
<b>I/O</b>	Entradas y salidas, analógicas y digitales.
<b>ISM</b>	Banda de frecuencia libre para fines industriales, científicos y médicos.
<b>MAC</b>	Control de acceso al medio.
<b>MANET</b>	Mobile Ad-Hoc Network.



<b>MIT</b>	Massachusetts Institute of Technology.
<b>NAM</b>	Network Animator.
<b>NS2</b>	Network Simulator 2.
<b>OSI</b>	Open System Interconnection (Sistema de interconexión abierta).
<b>PDA</b>	Personal digital assistant, asistente digital personal.
<b>PAN</b>	Personal Area Network (Red de Área Personal).
<b>PHY</b>	Capa Física.
<b>RF</b>	Radio Frecuencia.
<b>RXD ó Rx</b>	Recepción de datos.
<b>TCP/IP</b>	Transmission Control Protocol/Internet Protocol.
<b>TICs</b>	Tecnologías de la información y la comunicación.
<b>TXD ó Tx</b>	Transmisión de datos.
<b>UART</b>	Receptor/ Transmisor asíncrono universal.
<b>UDP</b>	User Datagram Protocol (Protocolo de datagramas del Usuario).
<b>USB</b>	Universal Serial Bus.
<b>VS</b>	Software de programación Visual Studio.
<b>WiFi</b>	Wireless Fidelity (Fidelidad Inalámbrica)
<b>WLAN</b>	Wireless Local Area Network (Red Inalámbrica de Área Local).
<b>WiMAX</b>	Worldwide Interoperability for Microwave Access (interoperabilidad mundial para acceso por microondas).
<b>WMAN</b>	Wireless Metropolitan Área Network (Red de área Metropolitana).
<b>WPAN</b>	Wireless Personal Area Network (Red de área Personal).

<b>WSN</b>	Wireless Sensor Network (Red inalámbrica de sensores).
<b>WWAN</b>	Wireless Wide Area Network (Red de área Extensa).
<b>ZC</b>	Coordinador ZigBee.
<b>ZDO</b>	Objeto del Dispositivo ZigBee.
<b>ZED</b>	Dispositivo Final ZigBee.
<b>ZR</b>	Router ZigBee.

## ANEXO II:

### CONSTRUCCIÓN DEL NODO SENSOR

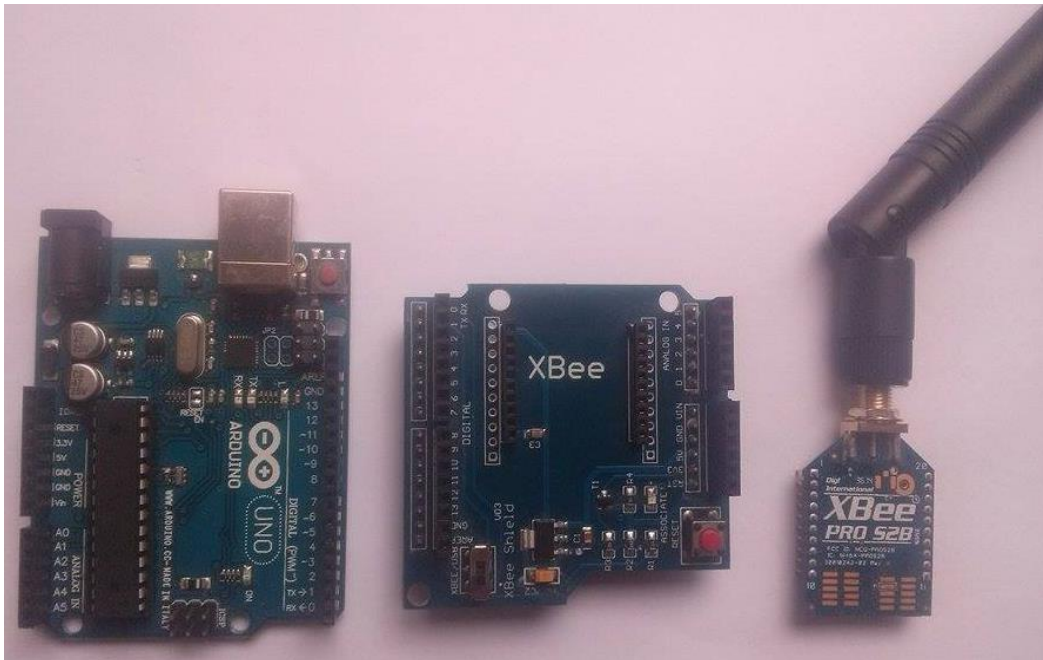


Figura 1. Módulos utilizados para la construcción de un nodo sensor

Fuente: Elaboración Propia

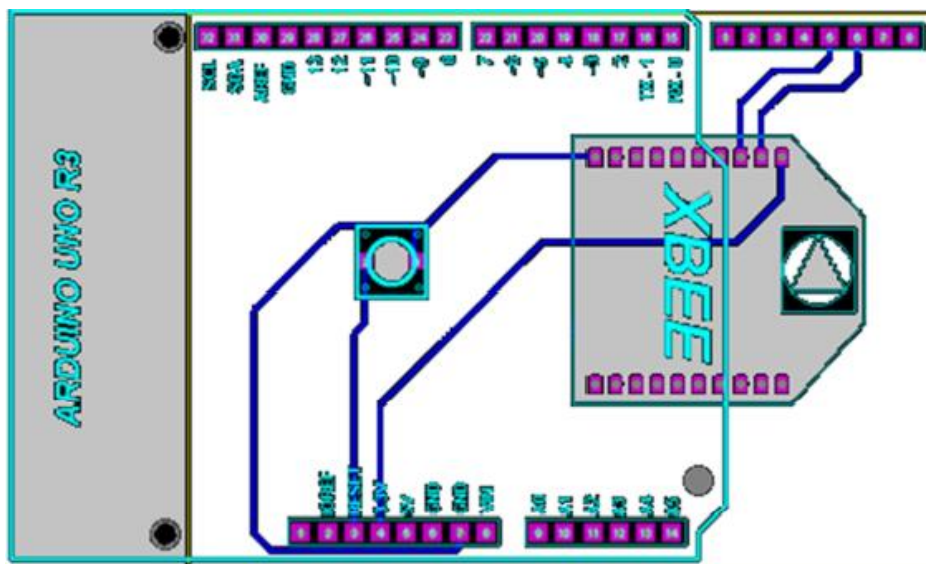


Figura 2. Esquema de conexión entre Arduino UNO XBee Shield y XBee S2B PRO

Fuente: Elaboración Propia



Figura 3. Montaje final del nodo Sensor.

Fuente: Elaboración Propia

### ANEXO III:

#### CONSTRUCCIÓN DEL NODO COORDINADOR

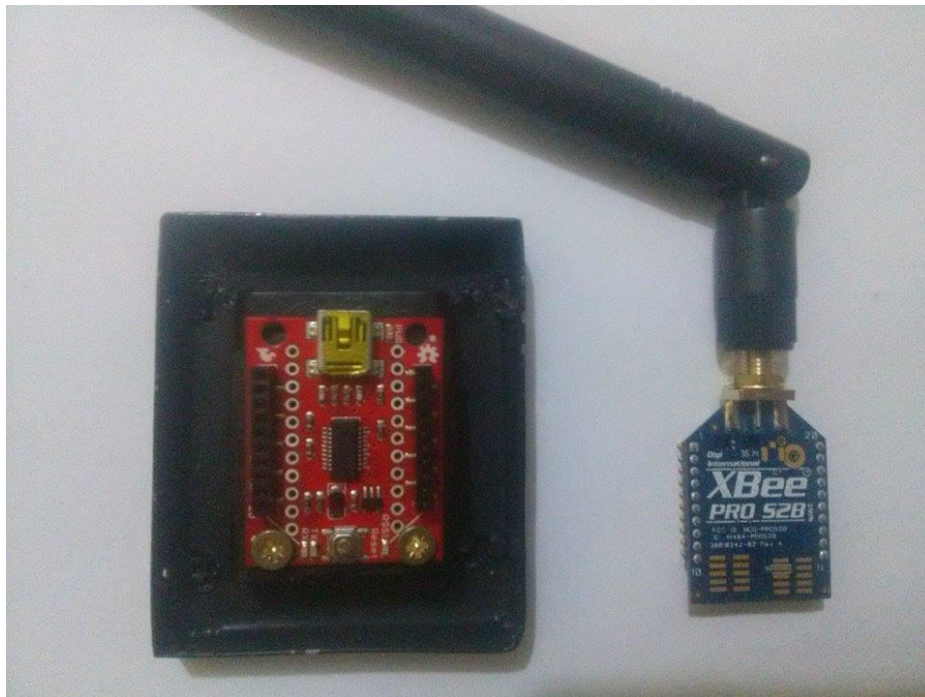


Figura 4. Módulos utilizados para la construcción de un nodo Coordinador

Fuente: Elaboración Propia



Figura 5. Montaje final del nodo Coordinador.

Fuente: Elaboración Propia

#### ANEXO IV

#### PROTOTIPO DE LA RED INALÁMBRICA DE SENSORES DESARROLLADA.



Figura 6. Gráfica de los elementos que conforman el Prototipo.

Fuente: Elaboración Propia



HOJA DE DATOS DEL MÓDULO XBEE PRO S2B

# 1. XBee®/XBee-PRO® RF Modules

The XBee and XBee-PRO RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices.

The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.



## Key Features

### Long Range Data Integrity

#### XBee

- Indoor/Urban: up to 100' (30 m)
- Outdoor line-of-sight: up to 300' (90 m)
- Transmit Power: 1 mW (0 dBm)
- Receiver Sensitivity: -92 dBm

#### XBee-PRO

- Indoor/Urban: up to 300' (90 m), 200' (60 m) for International variant
- Outdoor line-of-sight: up to 1 mile (1600 m), 2500' (750 m) for International variant
- Transmit Power: 63mW (18dBm), 10mW (10dBm) for International variant
- Receiver Sensitivity: -100 dBm

RF Data Rate: 250,000 bps

### Advanced Networking & Security

- Retries and Acknowledgements
- DSSS (Direct Sequence Spread Spectrum)
- Each direct sequence channels has over 65,000 unique network addresses available
- Source/Destination Addressing
- Unicast & Broadcast Communications
- Point-to-point, point-to-multipoint and peer-to-peer topologies supported

### Low Power

#### XBee

- TX Peak Current: 45 mA (@3.3 V)
- RX Current: 50 mA (@3.3 V)
- Power-down Current: < 10 µA

#### XBee-PRO

- TX Peak Current: 250mA (150mA for international variant)
- TX Peak Current (RPSMA module only): 340mA (180mA for international variant)
- RX Current: 55 mA (@3.3 V)
- Power-down Current: < 10 µA

### ADC and I/O line support

Analog-to-digital conversion, Digital I/O  
I/O Line Passing

### Easy-to-Use

- No configuration necessary for out-of box RF communications
- Free X-CTU Software (Testing and configuration software)
- AT and API Command Modes for configuring module parameters
- Extensive command set
- Small form factor

## Worldwide Acceptance

**FCC Approval (USA)** Refer to Appendix A [p64] for FCC Requirements. Systems that contain XBee®/XBee-PRO® RF Modules inherit Digi Certifications.

ISM (Industrial, Scientific & Medical) **2.4 GHz frequency band**

Manufactured under **ISO 9001:2000** registered standards

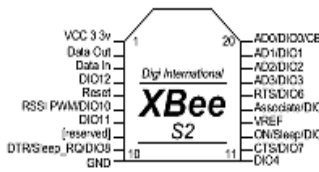
XBee®/XBee-PRO® RF Modules are optimized for use in the United States, Canada, Australia, Japan, and Europe. Contact Digi for complete list of government agency approvals.



**Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules**

Specification	XBee	XBee-PRO
<b>Performance</b>		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
<b>Power Requirements</b>		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 $\mu$ A	< 10 $\mu$ A
<b>General</b>		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
<b>Networking &amp; Security</b>		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
<b>Agency Approvals</b>		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW07215214	R201WW08215111 (Max. 10 dBm transmit power output)*
Australia	C-Tick	C-Tick

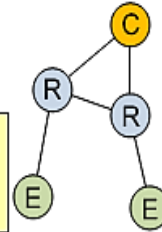
GUÍA DE CONFIGURACIÓN DEL MÓDULO XBEE PRO S2B



**XBee S2 Quick Reference Guide**

IEEE 802.15.4 = Zigbee Protocol. XBee is a microcontroller made by digi which uses the Zigbee protocol. The XBee uses 3.3V and has a smaller pin spacing than most breadboards/proto boards. Because of this, it is often useful to purchase a kit to interface the XBee with a breadboard.

Sept/2012 <http://tunnelsup.com>



<b>Coordinator</b> – 1 required in every network In charge of setting up the network Can never sleep
<b>Router</b> – multiple may exist Can relay signals from other routers/EPs Can never sleep
<b>End Point</b> – multiple may exist Cannot relay signals Can sleep to save power

<b>Specs</b>	Operating Voltage: 2.1 – 3.6V Operating Current: 40mA@3.3V Indoor range: 40 Meters Line of sight range: 120 Meters Max Analog Pin Reading: 1.2V	Digital I/O pins: 11 Analog input pins: 4 Mesh routable Self Healing network Firmware: ZB ZigBee	RF Data Rate: 250kbps Throughput speed: 35kbps Frequency: ISM 2.4GHz OK Temp: -40 to 85C
--------------	---	--	---

<b>XBee Modes</b>	<b>Transparent</b> – Communication through the XBee. If data is not generated from the XBee itself then both XBees should be set to AT. <b>Command</b> – Communication to the XBee. If one XBee is sensing data, that XBee should be in AT mode while the receiving one should be in API mode.
-------------------	---

<b>Arduino Connectivity:</b> Arduino TX connects to XBee RX (Data in) Arduino RX connects to XBee TX (Data out)
---

<b>XBee Setup</b>	Connect the XBee to a TTL Serial FTDI adapter – OR – Arduino hack: Connect RX to RX, TX to TX, RESET to ground to bypass the Arduino entirely and get serial to XBee. Use the free X-CTU software to configure the XBee. Baud: 9600 – FC: Hardware – Data Bits: 8 – Parity: None – Stop Bits: 1
-------------------	---

<b>Arduino Integration:</b> Data sent to Serial.print() will go out TX port of Arduino which is then connected to the RX port of XBee. If XBee is in AT mode it will transmit it wirelessly. Data received from XBee will be sent to the Serial.
---

<b>Basic Settings</b>	PAN ID – The network to communicate over. If 0, the XBee will join any. DH/DL – Destination Serial number. Used to send to a specific XBee's Serial. Set to 0 to send to just the Coordinator. Set to 0x0000000000FFFF to broadcast. JV – Router/EP should be set to 1 so it rejoins the network on startup
-----------------------	---

<b>Arduino Example: Read an analog value using API</b> // Remote XBee: AT, Base XBee: API if (Serial.available() >= 21) { // Make sure the frame is all there if (Serial.read() == 0x7E) { // 7E is the start byte for (int i = 1; i < 19; i++) { // Skip ahead to the analog data byte discardByte = Serial.read(); } int analogMSB = Serial.read(); // Read the first analog byte data int analogLSB = Serial.read(); // Read the second byte int analogReading = analogLSB + (analogMSB * 256); } }
---

<b>Pin Settings</b>	For pin settings to work, receiver XBee must be in API mode D0 – Set pin 0 to start sensing IR – Collect data on sensing pins every XX millisecs
---------------------	--

Byte	Example	Description
0	0x7e	Start byte – Indicates beginning of data frame
1	0x00	Length – Number of bytes (ChecksumByte# – 1 – 2)
2	0x10	
3	0x17	Frame type - 0x17 means this is a AT command Request
4	0x52	Frame ID – Command sequence number
5	0x00	64-bit Destination Address (Serial Number)
6	0x13	MSB is byte 5, LSB is byte 12
7	0xA2	
8	0x00	0x0000000000000000 = Coordinator
9	0x40	0x0000000000000FFF = Broadcast
10	0x77	
11	0x9C	
12	0x49	
13	0xFF	Destination Network Address
14	0xFE	(Set to 0xFFFFE to send a broadcast)
15	0x02	Remote command options (set to 0x02 to apply changes)
16	0x44 (D)	AT Command Name (Two ASCII characters)
17	0x02 (2)	
18	0x04	Command Parameter (queries if not present)
19	0XF5	Checksum

<b>Arduino Example: Change the pin setting on a remote Xbee</b> // Remote XBee: AT, Base XBee: API Serial.write(0x7E); // Sync up the start byte Serial.write((byte)0x0); // Length MSB (always 0) Serial.write(0x10); // Length LSB Serial.write(0x17); // 0x17 is the frame ID for sending an AT command Serial.write((byte)0x0); // Frame ID (no reply needed) Serial.write((byte)00); // Send the 64 bit destination address Serial.write((byte)00); // (Sending 0x0000000000000FFF (broadcast)) Serial.write((byte)00); Serial.write((byte)00); Serial.write((byte)00); Serial.write(0xFF); Serial.write(0xFF); Serial.write(0xFF); // Destination Network Serial.write(0xFE); // (Set to 0xFFFFE if unknown) Serial.write(0x02); // Set to 0x02 to apply these changes Serial.write('D'); // AT Command: D1 Serial.write('1'); Serial.write(0x05); // Set D1 to be 5 (Digital Out HIGH)
--



Byte	Example	Description	long chexsum = 0x17 + 0xFF + 0xFF + 0xFF + 0xFE + 0x02 + 'D' + '1' + 0x05; Serial.write( 0xFF - (chexsum & 0xFF)); // Checksum	
0	0x7e	Start byte - Indicates beginning of data frame	<b>Sleep Mode</b> Endpoints can sleep to save power. An endpoint that only wakes up every 5 minutes to send data may only be awake for 6 seconds a day. SM - 4 = Cyclic sleep SP - Sleep time (up to 28 secs) SN - Number of sleep cycles ST - Time awake	<b>Pin I/O Options</b> 0 - Disabled 1 - N/A 2 - ADC 3 - Digital IN 4 - Digital OUT, LOW 5 - Digital OUT, HIGH
1	0x00	Length - Number of bytes (ChecksumByte# - 1 - 2)		
2	0x14			
3	0x92	Frame type - 0x92 indicates this will be a data sample		
4	0x00	64-bit Source Address (Serial Number)		
5	0x13	MSB is byte 4, LSB is byte 11		
6	0xA2			
7	0x00			
8	0x40			
9	0x77			
10	0x9C			
11	0x49		<b>Digital Ch Mask</b> First Byte n/a n/a n/a D12 D11 D10 n/a n/a Second Byte D7 D6 D5 D4 D3 D2 D1 D0 Example: 0x00 0x0D = 0000 0000 0000 1101 Pins D3, D2 and D0	<b>Notes</b>
12	0x36	Source Network Address - 16 Bit		
13	0x6A			
14	0x01	Receive Opts. 01=Packet Acknowledged. 02=Broadcast packet		
15	0x01	Number of sample sets. Always set to 1 due to XBEE limitations		
16	0x00	Digital Channel Mask - Indicates which pins are set to DIO		
17	0x20			
18	0x01	Analog Channel Mask - Indicates which pins are set to ADC		
19	0x00	Digital Sample Data (if any) - Reads the same as Digital Mask		
20	0x14			
21	0x04	Analog Sample data (if any)		
22	0x25	There will be two bytes here for every pin set for ADC		
23	0xF5	Checksum(0xFF - the 8 bit sum of the bytes from byte 3 to this byte)	<b>Analog Ch Mask</b> (volt) n/a n/a n/a A3 A2 A1 A0 Example: 0x05 = 0000 0101 = Pin A2 and A0	



## LM35 Precision Centigrade Temperature Sensors

### 1 Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full –55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates from 4 V to 30 V
- Less than 60- $\mu$ A Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only  $\pm\frac{1}{4}$ °C Typical
- Low-Impedance Output, 0.1  $\Omega$  for 1-mA Load

### 2 Applications

- Power Supplies
- Battery Management
- HVAC
- Appliances

### 3 Description

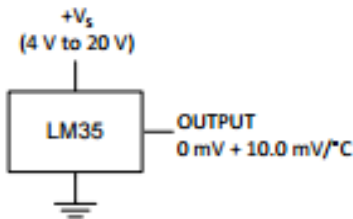
The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of  $\pm\frac{1}{4}$ °C at room temperature and  $\pm\frac{1}{4}$ °C over a full –55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60  $\mu$ A from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a –55°C to 150°C temperature range, while the LM35C device is rated for a –40°C to 110°C range ( $-10^\circ$  with improved accuracy). The LM35-series devices are available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D devices are available in the plastic TO-92 transistor package. The LM35D device is available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

#### Device Information<sup>(1)</sup>

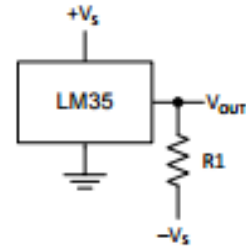
PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM35	TO-CAN (3)	4.699 mm × 4.699 mm
	TO-92 (3)	4.30 mm × 4.30 mm
	SOIC (8)	4.90 mm × 3.91 mm
	TO-220 (3)	14.986 mm × 10.16 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

### Basic Centigrade Temperature Sensor (2°C to 150°C)



### Full-Range Centigrade Temperature Sensor



Choose  $R_1 = -V_s / 50 \mu\text{A}$   
 $V_{\text{OUT}} = 1500 \text{ mV at } 150^\circ\text{C}$   
 $V_{\text{OUT}} = 250 \text{ mV at } 25^\circ\text{C}$   
 $V_{\text{OUT}} = -550 \text{ mV at } -55^\circ\text{C}$

## 6 Specifications

### 6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)<sup>(1)(2)</sup>

	MIN	MAX	UNIT	
Supply voltage	-0.2	35	V	
Output voltage	-1	6	V	
Output current		10	mA	
Maximum Junction Temperature, $T_{j\text{max}}$		150	°C	
Storage Temperature, $T_{\text{slg}}$	TO-CAN, TO-92 Package	-60	150	°C
	TO-220, SOIC Package	-65	150	

- (1) If Military/Aerospace specified devices are required, please contact the Texas Instruments Sales Office/ Distributors for availability and specifications.
- (2) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions.

### 6.2 ESD Ratings

			VALUE	UNIT
$V_{\text{(ESD)}}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 <sup>(1)</sup>	±2500	V

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

### 6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	MAX	UNIT
Specified operating temperature: $T_{\text{MIN}}$ to $T_{\text{MAX}}$	LM35, LM35A	-55	150	°C
	LM35C, LM35CA	-40	110	
	LM35D	0	100	
Supply Voltage (+ $V_s$ )		4	30	V

### 6.4 Thermal Information

THERMAL METRIC <sup>(1)(2)</sup>	LM35				UNIT
	NDV	LP	D	NEB	
	3 PINS		8 PINS	3 PINS	
$R_{\theta\text{JA}}$ Junction-to-ambient thermal resistance	400	180	220	90	°C/W
$R_{\theta\text{JC(top)}}$ Junction-to-case (top) thermal resistance	24	—	—	—	

## HOJA DE DATOS DE LA UNIDAD DE ENERGÍA.



PR121312Volt 1.3AH

### Battery Construction

Component	Positive Plate	Negative Plate	Container	Cover	Safety Valve	Terminal	Separator	Electrolyte
Raw material	Lead dioxide	Lead	ABS UL94HB/V0	ABS UL94HB/V0	Rubber	Copper	Fiberglass	Sulfuric acid

### Constant Current Discharge Rating Amperes @25°C (77°F)

Time	10min	15min	30min	1hr	3hr	5hr	10hr	20hr
Final Discharge Voltage V/cell	2.81	2.29	1.33	0.78	0.34	0.22	0.12	0.07

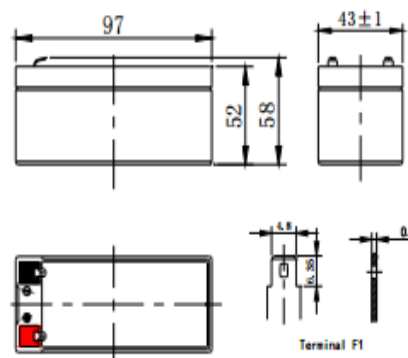
### Characteristics

Rated Capacity 25°C (77°F)	C20 1.75V/C		1.30 AH
	C10 1.75V/C		1.21 AH
	C5 1.70V/C		1.11 AH
	C1 1.60V/C		0.78 AH
Internal Resistance	Full charged battery 25°C (77°F)		120.0 mΩ
Capacity affected by temperature (20 hour rate)	40°C (104°F)		107%
	25°C (77°F)		100%
	0°C (32°F)		76%
	-10°C (14°F)		66%
Self Discharge	Batteries can be stored for more than 6 months at 25°C. Self-discharge ratio less than 3% per month at 25°C. Please charge batteries before using.		
Standard Terminal			F1
Max. Discharge Current 25°C (77°F)			13 A(5s)
Final Discharge Voltage V/cell	1.75V	1.70V	1.60V
Discharge Current (A)	(A) ≤0.15C	0.15C < (A) < 0.5C	(A) ≥0.5C
Charging (Constant Voltage)	Cycle	Initial Charging Current 0.3 A Or Small	
		14.5V~14.9V/ 25°C (77°F)	
	Float	13.6V~13.8V/ 25°C (77°F)	

### Specifications

Nominal Voltage		12 V	
Nominal Capacity (20hr)		1.3 AH	
Dimension	Length	97 mm	3.82 in
	Width	43 mm	1.69 in
	Height	52 mm	2.05 in
	Total Height (with terminals)	58 mm	2.28 in
Weight	Approx.	0.55kg	1.42 lbs

### Physical Dimensions: mm



# Model: SEN92355P

## Introduction

This Moisture Sensor can be used to detect the moisture of soil or judge if there is water around the sensor, let the plants in your garden reach out for human help. They can be very to use, just insert it into the soil and then read it. with help of this sensor, it will be realizable to make the plant remind you : hey, i am thirsty now, please give me some water.

Model: [SEN92355P](#)

## Features

- Easy to Use
- 2.0cmX6.0cm grove module
- feature3

## Specification

Item	Min	Typical	Max	Unit
Voltage	3.3	/	5	V
Current	0	/	35	mA

## Application Ideas

- Botanical gardening
- Water sensor
- Application3

## Schematic

