

**IMPLEMENTACIÓN Y CARACTERIZACIÓN DE UNA RED DE SENSORES  
INALÁMBRICOS PARA EL CONTROL DISTRIBUIDO DE ILUMINACIÓN EN  
EDIFICACIONES.**

**DANILO ESTEBAN OLIVA PATIÑO  
CLARA ELIZABETH VERDUGO MUÑOZ**

**UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE ELECTRÓNICA  
SAN JUAN DE PASTO  
2015**

**IMPLEMENTACIÓN Y CARACTERIZACIÓN DE UNA RED DE SENSORES  
INALÁMBRICOS PARA EL CONTROL DISTRIBUIDO DE ILUMINACIÓN EN  
EDIFICACIONES.**

**DANILO ESTEBAN OLIVA PATIÑO  
CLARA ELIZABETH VERDUGO MUÑOZ**

**Trabajo de grado presentado como requisito para optar al título de  
Ingeniero Electrónico**

**Director:  
PhD. ANDRES PANTOJA BUCHELI**

**UNIVERSIDAD DE NARIÑO  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE ELECTRÓNICA  
SAN JUAN DE PASTO  
2015**

## **NOTA DE RESPONSABILIDAD**

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor”.

Acuerdo 1. Artículo 324. Octubre 11 de 1966, emanado del honorable Consejo Directivo de la Universidad de Nariño.

**Nota de Aceptación**

---

---

---

---

---

---

Firma del Presidente del Jurado

---

Firma Jurado 1

---

Firma Jurado 2

San Juan de Pasto, marzo de 2015.

## **AGRADECIMIENTOS**

*Agradezco a Dios:*

## DEDICATORIA

*Dedicado a Dios:*

## CONTENIDO

	<b>Pág.</b>
INTRODUCCIÓN .....	19
1. DESCRIPCIÓN DEL SISTEMA.....	23
1.1 ESTÁNDAR IEEE 802.15.4.....	24
1.2 BANDAS DE OPERACIÓN .....	25
1.2.1 Físicos. ....	26
1.2.2 Lógicos.....	26
1.2.3 Modos de Operación.....	30
1.2.4 Descripción. ....	33
1.2.5 Características: .....	33
1.2.6 Arduino Mega 2560. ....	35
1.2.7 Características: .....	35
2. PRUEBAS PARA CALCULO DE DESEMPEÑO DE UNA WSN .....	37
2.1 CONFIGURACIÓN DE MÓDULOS PARA CONEXIONES 2 Y 3 NODOS	37
2.2 PRUEBAS Y ANÁLISIS.....	42
2.3 ANÁLISIS DE RESULTADOS.....	48
3. IMPLEMENTACION Y CARACTERIZACION DE UNA RED .....	54
3.1 ANÁLISIS DE TOPOLOGÍAS.....	54
3.1.1 Arquitectura Básica de una Red XBee.....	54
3.1.2 Topologías. ....	55
3.1.3 Topologías bajo el protocolo IEEE 802.15.4 .....	57
4. IMPLEMENTACION.....	61
4.1 PRUEBA 1 .....	62
4.1.1 Configuración módulos.....	62
4.1.2 Parámetros de las tramas. ....	62
4.1.4 Prueba.....	65
4.1.5 Resultados. ....	65

4.2	DISEÑO DE NODOS CENTRAL Y SENSORES.....	68
4.2.1	Nodo Central.....	69
4.2.2	Diseño de circuitos impresos.....	72
4.2.3	Caracterización del consumo de potencia.....	73
4.3	PRUEBA 2.....	73
4.3.1	Configuración módulos.....	73
4.3.3	Programación.....	76
4.3.4	Prueba.....	77
4.3.5	Resultados.....	78
	CONCLUSIONES.....	81
	RECOMENDACIONES.....	83
	BIBLIOGRAFÍA.....	84
	ANEXOS.....	88

## LISTA DE FIGURAS

	<b>Pág.</b>
Figura 1.	Diagrama de bloques nodo central ..... 23
Figura 2.	Diagrama de bloques nodo sensor ..... 24
Figura 3.	Bandas de operación del Estándar IEEE 802.15.4..... 25
Figura 4.	Dispositivos lógicos en red básica Xbee. [37]..... 27
Figura 5.	a. Vista superior módulo <i>Xbee</i> pro s1, b. Distribución de pines módulo <i>Xbee</i> pro s1 ..... 28
Figura 6.	a. Ubicación del número serial en el módulo <i>Xbee</i> . b. Número serial del módulo <i>Xbee</i> en el software X-CTU. .... 29
Figura 7.	a. <i>Xbee</i> Xplorer Dongle sin conectar con el módulo <i>Xbee</i> . b. <i>Xbee</i> Xplorer Dongle con el módulo <i>Xbee</i> conectado al PC para la comunicación serial. .... 30
Figura 8.	Configuración punto a punto de dos módulos <i>Xbee</i> pro s1 con X- CTU, a. Configuración parámetros módulo 1, b. Configuración parámetros módulo 2..... 38
Figura 9.	Canales disponibles para el protocolo IEEE 802.15.4 a 2.4 GHz [15]..... 40
Figura 10.	Configuración para pruebas con 3 módulos. .... 41
Figura 11.	Configuración tres módulos en X-CTU ..... 42
Figura 12.	Calidad de la señal y pérdida de paquetes para las pruebas realizadas con dos nodos ..... 50
Figura 13.	a. Comparación del porcentaje de calidad de la señal para las pruebas realizadas con el enlace directo y enlace de tres nodos, b. Comparación del porcentaje de pérdida de paquetes para las pruebas realizadas con el enlace directo y enlace de tres nodos.... 52
Figura 14.	Porcentaje calidad de la señal y porcentaje de perdida de paquetes para diferentes distancias entre los nodos 1 y 3, prueba

	tres nodos, b. Porcentaje calidad de la señal y porcentaje de pérdida de paquetes vs cantidad de obstáculos entre los nodos 1 y 3, prueba tres nodos. ....	53
Figura 15.	Tipos de topologías soportados por el estándar IEEE 802.15.4 [15].....	56
Figura 16.	Diagrama de la Metodología de diseño de una WSN. ....	60
Figura 17.	a. Porcentaje calidad de la señal y porcentaje de pérdida de paquetes para las distancia entre los enlaces 4 a 8, 3 a 6 y 4 a 6, b. Porcentaje calidad de la señal y porcentaje de pérdida de paquetes para los obstáculos entre los enlaces 4 a 8, 3 a 6 y 4 a 6.....	67
Figura 18.	Porcentaje calidad de la señal y porcentaje de pérdida de paquetes para las distancia entre los enlaces 8 a 4, 6 a 3 y 6 a 4, b. Porcentaje calidad de la señal y porcentaje de pérdida de paquetes para los obstáculos entre los enlaces 4 a 8 a 4, 6 a 3 y 6 a 4.....	68
Figura 19.	Tarjeta implementada nodo central, a. Vista superior, b. Vista inferior.....	70
Figura 20.	Sumador inversor con LM324.....	71
Figura 21.	Tarjeta implementada nodo sensor, a. Vista inferior, b. Vista superior.....	72
Figura 22.	Distancia vs calidad señal prueba 2 de la red implementada .....	79

## LISTA DE TABLAS

	<b>Pág.</b>
Tabla 1. Propiedades del estándar IEEE 802.15.4 .....	25
Tabla 2. Parámetros de bandas de frecuencia [34] .....	26
Tabla 3. Configuración de cada módulo punto a punto .....	39
Tabla 4. Canales de Frecuencia y su respectivo comando at.....	40
Tabla 5. Configuración módulos conexión tres nodos .....	42
Tabla 6. Pruebas módulos inalámbricos dos nodos .....	44
Tabla 7. Pruebas módulos inalámbricos tres nodos .....	48
Tabla 8. Desempeño de los módulos conexión de 2 nodos.....	49
Tabla 9. Desempeño de los módulos conexión de 3 nodos.....	51
Tabla 10. Nodos prueba final.....	61
Tabla 11. Transmisión/ recepción mensajes prueba con computadores .....	62
Tabla 12. Configuración módulos prueba 1 .....	62
Tabla 13. Parámetros trama tipo 0x81 Rx packet 16 bit address .....	63
Tabla 14. Parámetros trama tipo 0x00 tx request 64 bit address.....	64
Tabla 15. Características de la red prueba 1 .....	66
Tabla 16. Valores de resistencias para la etapa de amplificación .....	72
Tabla 17. Características del consumo de nodos sensores.....	73
Tabla 18. Configuración módulos prueba 2 .....	74
Tabla 19. Parámetros Trama tipo 0x83 Rx packet 16 bit address io .....	75
Tabla 20. Parámetros Trama tipo 0x17 remote at command.....	76
Tabla 21. Características de la red Prueba 2.....	78

## LISTA DE ANEXOS

	<b>Pág.</b>
ANEXO A: PLANOS DE LOS PISOS 3 Y 4 DEL BLOQUE TECNOLÓGICO DE LA UNIVERSIDAD DE NARIÑO CON LA UBICACIÓN DE LOS NODOS PARA CADA PRUEBA POR ENLACES.....	89
ANEXO B: CÓDIGO PARA REALIZAR LAS PRUEBAS DE TRES NODOS IMPLEMENTADO EN ARDUINO.....	90
ANEXO C: PLANOS DEL PISO 4 DEL EDIFICIO TECNOLÓGICO DE LA UNIVERSIDAD DE NARIÑO CON LA UBICACIÓN DE LOS NODOS PARA CADA TIPO DE TOPOLOGÍA.....	91
ANEXO D: PLANO DEL PISO 3 DEL EDIFICIO TECNOLÓGICO DE LA UNIVERSIDAD DE NARIÑO CON LA UBICACIÓN DE LOS NODOS PARA LA RED FINAL.....	92
ANEXO F: PROGRAMACIÓN DE LA RED FINAL IMPLEMENTADA EN ARDUINO .....	99
ANEXO G: ILUSTRACION CIRCUITOS IMPRESOS NODO CENTRAL Y SENSOR.....	106
ANEXO H: PRESUPUESTO DETALLADO DE LAS TARJETAS NODO CENTRAL Y SENSOR.....	107

## GLOSARIO

**ACK (Acknowledged):** Trama interna del Xbee Pro para determinar si un paquete fue o no entregado correctamente. Paquete enviado en modo API como respuesta a un comando o trama específica.

**ADC (Analog to Digital Converter):** convertidor analógico a digital, convierte una señal analógica a una señal digital.

**API (Application Programming Interface):** Modo de operación de los módulos Xbee y Zigbee, para transmitir información mediante tramas que se pueden emplear para realizar diferentes funciones tales como direccionamiento y configuración remota de módulos inalámbricos.

**ARTWORK (Ilustración):** Diseño ilustrado de las pistas de un circuito, realizado en un programa de diseño de circuitos impresos, para implementarse al pasarlo a una lámina de pertinax.

**BPSK (Binary Phase Shift Keying):** Modulación de desplazamiento de fase de 2 símbolos. Es el tipo de modulación más sencillo de todos, puesto que solo emplea 2 símbolos, con 1 bit de información cada uno.

**BROADCAST (difusión):** Forma de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo.

**BUFFER:** Es un circuito amplificador bastante usado en los circuitos electrónicos digitales. Sirve para amplificación de potencia, y acoplamiento de impedancias. Si el buffer es de entrada, se logra su manejo ("abrir" y "cerrar") con señales de muy poca corriente. Si tiene el buffer en las salidas, es posible manejar cargas de 40, 100 y 200 miliamperios, cuando en forma normal podría suministrar solamente 16 miliamperios.

**CHECKSUM (Suma de verificación):** Byte que contiene el resultado de la suma de los bytes en una trama del modo API de un Xbee, entre la longitud de la trama y el byte del checksum.

**DIGIMESH: DSSS (Direct Sequence Spread Spectrum)** espectro ensanchado de secuencia directa.

**EEPROM (Electrically Erasable Programmable Read-Only Memory)** Es un tipo de memoria no volátil que puede ser programada, borrada y reprogramada

eléctricamente, a diferencia de la EPROM que ha de borrarse mediante un aparato que emite rayos ultravioleta.

FFD (Full-Function Device): Dispositivo de funcionalidad completa. Nodo activo capaz de recibir mensajes en formato 802.15.4 que puede ser usado en dispositivos de red que actúen de interfaz con el usuario.

FIRMWARE: Bloque de instrucciones de máquina para propósitos específicos, grabado en una memoria, normalmente de lectura/escritura (ROM, EEPROM, flash, etc.), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo.

GT (Guard Time): Tiempo de guarda desde el último carácter recibido en forma serial en los módulos Xbee.

IDLE: Estado en el cual se encuentra un módulo Xbee cuando no está ni transmitiendo ni recibiendo, ni ahorrando energía ni en el modo de comandos.

ISM (Industrial, Scientific and Medical): Bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica. En la actualidad estas bandas han sido popularizadas por su uso en comunicaciones WLAN (e.g. Wi-Fi) o WPAN (e.g. Bluetooth).

LCD (Liquid Crystal Display): Pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. Se utiliza en dispositivos electrónicos de pilas, ya que utiliza cantidades muy pequeñas de energía eléctrica.

LED (Light-Emitting Diode): 'Diodo emisor de luz'; es un componente optoelectrónico pasivo y más concretamente, un diodo que emite luz.

LR-WPAN (Low-Rate Wireless Personal Area Network): Redes de área local personal de baja velocidad. Protocolo de comunicaciones que trata las necesidades de sistemas con poca transmisión de datos pero vidas útiles muy altas con alimentación limitada de baterías, etc. y una complejidad muy baja.

O-QPSK (Offset Quadrature Phase Shift Keying): Esquema de modulación usado en sistemas de comunicación inalámbrica. OQPSK es una variante del esquema de modulación básico QPSK.

PWM (Pulse Width Modulation): Modulación por ancho de pulsos de una señal o fuente de energía; es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (senoidal o cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

RF (Radio Frequency): Tasa de oscilación en el intervalo de alrededor de 3 kHz a 300 GHz, que corresponde a la frecuencia de las ondas de radio y las corrientes alternas que llevan señales de radio. RF por lo general se refiere a oscilaciones eléctricas en vez de oscilaciones mecánicas; Sin embargo, existen sistemas de RF mecánicas (RF MEMS ).

RFD (Reduced Function Device): Dispositivo de funcionalidad reducida. Nodo pasivo que tiene la capacidad y funcionalidad limitadas con el objetivo de conseguir un bajo coste y una gran simplicidad.

RP-SMA (Reverse Polarity Sub-Miniature version A): Es una variante del conector SMA donde su principal característica es el cambio de polaridad magnética (no confundir con polaridad eléctrica), con el susodicho que los conectores SMA y RP-SMA no son compatibles entre ellos, además de ser físicamente muy similares.

RSSI (Received Signal Strength Indicator): El indicador de fuerza de la señal recibida es una escala de referencia (en relación a 1 mW) para medir el nivel de potencia de las señales recibidas por un dispositivo en las redes inalámbricas (típicamente WIFI o telefonía móvil).

SRAM (Static Random Access Memory): Memoria estática de acceso aleatorio (o RAM estática); tipo de tecnología de memoria RAM basada en semiconductores, capaz de mantener los datos, mientras siga alimentada, sin necesidad de circuito de refresco.

ST (Sleep Time): Tiempo antes de que un módulo Xbee pasa a modo Sleep.

TCP (Transmission Control Protocol): Protocolo de Control de Transmisión, es uno de los protocolos fundamentales en Internet. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.

UART (Universal Asynchronous Receiver-Transmitter): Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo. Las funciones principales de un chip UART son: manejar las interrupciones de los dispositivos conectados al puerto serie y convertir los datos en formato paralelo, transmitidos al bus de sistema, a datos en formato serie, para que puedan ser transmitidos a través de los puertos y viceversa.

UNICAST: Envío de información desde un único emisor a un único receptor.

USB (Universal Serial Bus): Bus estándar industrial que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos.

WSN (Wireless Sensor Network): Redes de área local personal de baja velocidad. Protocolo de comunicaciones que trata las necesidades de sistemas con poca transmisión de datos pero vidas útiles muy altas con alimentación limitada de baterías, etc. y una complejidad muy baja.

X-CTU: Software utilizado para configurar y probar productos Digi RF módems, principalmente módulos Xbee y Zigbee.

RANGE TEST (Prueba de Rango): Prueba el rango de enlaces inalámbricos en diversas condiciones. Esta información es útil cuando la planificación y la implementación de una red real.

MESH (Malla): Red en topología malla implementada sobre una red inalámbrica LAN.

SNR (Signal to Noise Ratio): Relación señal a ruido es la relación que existe del cociente entre la potencia de una señal que se transmite y la potencia del ruido que interfiere con la misma. Este margen es medido en decibelios.

## RESUMEN

En este documento se presenta la implementación y caracterización de una red de sensores inalámbricos para el control distribuido de iluminación en edificaciones para un piso del nuevo edificio tecnológico de la Universidad de Nariño. En base a los estudios realizados de los módulos *Xbee*, principalmente sobre sus características de acoplamiento (conexión a la red y topología), adquisición, transmisión y direccionamiento de datos, se plantea una metodología de diseño para redes de sensores inalámbricos. Se realizaron pruebas en el edificio antes mencionado para caracterizar el comportamiento de los módulos ante obstáculos y variaciones de distancia y se analiza datos de alcance, pérdida de paquetes y potencia de transmisión. Con la información y resultados obtenidos se realizó una red en topología estrella utilizando una tarjeta Arduino Mega 2560 y un módulo *Xbee* configurado en modo API (Application Programming Interface) como nodo central y ocho nodos sensores o dispositivos finales. Se propone una metodología de diseño de WSN, a partir de ella se realiza la caracterización de la red y pruebas de direccionamiento entre algunos enlaces de la misma. En adición se presenta el diseño e implementación de 9 tarjetas didácticas, una correspondiente al nodo central y 8 a los nodos sensores para probar la red mediante un control PWM remoto de iluminación y obtener información de la comunicación en la red. Finalmente se concluye acerca del trabajo realizado.

## **ABSTRACT**

This document presents the implementation and characterization of a wireless sensor network for distributed lighting control in buildings for the Universidad de Nariño's new technological building. Based on studies of the Xbee modules, mainly on the characteristics of coupling (connection to the network and topology) acquisition transmission and routing of data. A design methodology for wireless sensor networks is proposed. Tests were performed on the aforementioned building to characterize the behavior of the modules to obstacles and distance variations and range data. Packet loss and transmission power is analyzed. With the information and results was performed on a network star topology using an Arduino Mega 2560 and a Xbee module configured in API (Application Programming Interface) mode as hub-eight sensor nodes or end devices. A design methodology of WSN is proposed. Therefrom the characterization and testing of the network routing links between some of the network is performed. In addition the design and implementation of 9 printed circuits, one corresponding to the central node and 8 to the sensor nodes is presented to test the network using a PWM lighting remote control and information communication network. Finally we conclude about the work done.

## INTRODUCCIÓN

En la actualidad los entornos y aplicaciones inteligentes se han hecho cada vez más comunes en campos como la industria, la domótica, la medicina, entre otros. El desempeño de cada aplicación o entorno depende de los datos provenientes del mundo real los cuales se adquieren de sensores de diferentes tipos ubicados distribuidamente y es necesario que estos datos sean transmitidos a una unidad de procesamiento para que se generen las acciones de control pertinentes. Adicionalmente, se ha visto necesario que la comunicación de datos se realice inalámbricamente de manera confiable, eficaz y con bajo consumo de energía. De esta manera, los desafíos en la interpretación, análisis y procesamiento de la información son grandes principalmente en entornos inteligentes, mientras que la confiabilidad en la transmisión de la información es proporcionada por las redes de sensores inalámbricas (WSN) [1].

Una definición de sensores inalámbricos y WSN se da en [2] como una red compuesta por un gran número de nodos sensores que además de tener la capacidad de sensor, tienen la capacidad de almacenar, procesar y transmitir datos y están densamente localizados dentro o muy cerca al fenómeno a monitorear, además resaltan la importancia de la presencia de una estación base y su comunicación con la internet.

En los últimos años se han utilizado las WSN en diferentes campos y tareas, principalmente para la transmisión de datos de sensores distribuidos especialmente en entornos tanto internos como externos. Esto ha generado un auge en el uso de tecnologías de comunicación tales como UWB (*Ultra Wide Band*), IrDA, bluetooth y Zigbee (ZB) [36], siendo la más popular la última de estas por su alcance, bajo coste, escalabilidad (más de 65536 terminales), bajo consumo de potencia, fácil integración puesto que se pueden fabricar nodos con pocos componentes electrónicos y la confiabilidad en la transmisión de datos que ofrece [3]. Entre las principales aplicaciones de las WSN tanto en entornos interiores como exteriores, se puede encontrar varios campos de aplicación como la medicina, aplicaciones militares, obras civiles, medio ambiente, agricultura, automoción, domótica, entre otras.

En la medicina podemos encontrar aplicaciones como el monitoreo de pacientes con enfermedades crónicas, personas vulnerables que requieren un cuidado especial [8] o que trabajan en condiciones extremas y necesitan ser monitoreadas; lo anterior se realiza mediante la adquisición de datos de vital importancia, como son los signos vitales y motores provenientes de sensores biomédicos tales como oxímetros, sensores de temperatura, entre otros; la transmisión y recepción inalámbricas y el procesamiento de datos se realiza con dispositivos embebidos

como Zigbee conjuntamente con Arduino y unidades de computo [6]; también se suele realizar con dispositivos más completos en funcionalidad y procesamiento que además de trabajar con la misma tecnología (Zigbee) disponen de unidades de procesamiento más complejas en la misma tarjeta, lo cual provee muchas ventajas [7]; en este caso las WSN cumplen un papel importante en la prevención o atención de cualquier tipo de afección tempranamente y con menores riesgos. Las aplicaciones militares son varias en torno a las WSN y exigen confiabilidad, robustez y seguridad en la comunicación de datos; se suelen aplicar en vehículos no tripulados, aviones, tanques de guerra, misiles, entre otros para realizar ataques o defensa coordinados. En este tipo de aplicaciones las WSN son de vital importancia ya que muchas veces de ellas dependen muchas vidas [31].

En obras civiles las WSN se pueden usar para garantizar la seguridad de las personas mediante la medición de variables como vibración, temperatura, aceleración, entre otras que permiten tener información acerca de la seguridad de una estructura [9]. En el medio ambiente tienen una gran utilidad, por ejemplo, para el monitoreo de incendios forestales mediante sensores de temperatura y módulos embebidos como el Mica2 que permiten enviar la información a una estación base remota y de esa manera poder detectar cambios que indiquen un incendio que conlleve a una rápida respuesta para mitigar dicho suceso [32]. En la actualidad la WSN son ampliamente usadas en la agricultura para medir, transmitir y procesar datos de variables como humedad y temperatura de manera que mediante sensores, módulos Xbee de bajo costo y elementos actuadores sea posible automatizar y monitorear el riego y el estado un cultivo [10]; además se suelen usar también en entornos *indoor* como invernaderos para monitorear temperatura y humedad relativa de ciertas plantas usando principalmente la tecnología Zigbee [11].

Un área que está muy ligada a las WSN es la domótica, en la que se realiza la automatización de procesos en entornos *indoor* y en algunos casos se busca la optimización y eficiencia energética, como se muestra en algunos ejemplos en [4]. En este trabajo se realiza un sistema domótico basado en sensores y dispositivos integrados Zigbee comerciales que permiten una fácil instalación y solamente requieren mínimas configuraciones; el sistema se implementa en una construcción residencial y se realizan pruebas con sensores de humo, infrarrojos y detectores de apertura.

En cuanto a los sistemas de iluminación, las WSN cumplen un papel de vital importancia para la solución de problemas como consumo de energía y eficiencia energética, además de la transmisión de datos de manera confiable en un sistema con tecnología de control inteligente principalmente computadores y módulos RF [12]. Estos sistemas de interconexión requieren el diseño de una red de comunicaciones con una topología y características que se adecuen al entorno de propagación y que además el sistema de comunicaciones cumpla con las propiedades características de una WSN, además de su fácil implementación a

partir de la combinación de sistemas embebidos como Arduino, módulos inalámbricos y protocolos de iluminación como DALI [13]. En algunos casos de control distribuido de iluminación se cuenta con un elevado número de sensores, gran cantidad de información y actuadores que requieren una señal de control proveniente de una unidad de procesamiento centralizada [14] por lo que se hace necesaria la comunicación de estos múltiples dispositivos con el controlador principal de manera inalámbrica evitando inconvenientes en la instalación, mantenimiento, pérdida de información o potencia en el cableado y desde luego generar un ahorro en el consumo de energía.

Es posible además utilizar diferentes tipos módulos en configuraciones que permiten obtener datos directamente sin emplear una unidad de procesamiento adicional en cada nodo, depende de las capacidades de los módulos; esto trae ventajas como reducir tanto el tamaño como los costos de la red significativamente.

Como se puede apreciar en los anteriores ejemplos las WSN son muy importantes en distintos tipos de aplicaciones y la tecnología Zigbee actualmente es una de las más utilizadas, empleando módulos Xbee o ZB conjuntamente con procesadores de algún tipo. En algunos casos se recurre a la utilización de sensores, actuadores, routers y otros elementos con la tecnología Zigbee incorporada que se encuentran disponibles en el mercado. Por ejemplo en [4] se usan elementos comerciales como Gateways y sensores inalámbricos Zigbee de humo, magnéticos, infrarrojos, entre otros; mientras que en [10] se lleva a cabo un diseño desde cero, puesto que se diseña cada nodo sensor y se realiza un montaje el cual da unas prestaciones propias a la aplicación que comúnmente es difícil encontrar comercialmente. Los diseños desde cero se presentan en la mayoría de aplicaciones a razón de que el costo se reduce considerablemente, puesto que al usar algunos componentes electrónicos y realizar el diseño de tarjetas y demás, resulta mucho más económico que adquirir equipos comerciales que tienen funcionalidades limitadas a su fabricación y un mayor costo. Adicionalmente trae ventajas en cuanto a la funcionalidad específica de cada dispositivo implementado.

La selección de los dispositivos a utilizar depende estrictamente de la aplicación que se requiera implementar. En los últimos años, los módulos Xbee han tomado mucha importancia en las WSN gracias a sus características, variedad de series y a sus modos de trabajo. De esta manera en la práctica se suelen emplear dispositivos Xbee con funcionalidades limitadas que se complementan con sistemas embebidos y sensores para cumplir con tareas específicas, empleando principalmente las funciones de una tarjeta de desarrollo para procesamiento y direccionamiento de datos y apoyándose con software de simulación [5].

Teniendo en cuenta la utilidad y el uso extendido de las WSN, en este trabajo se propone la implementación y caracterización de una red de sensores inalámbricos

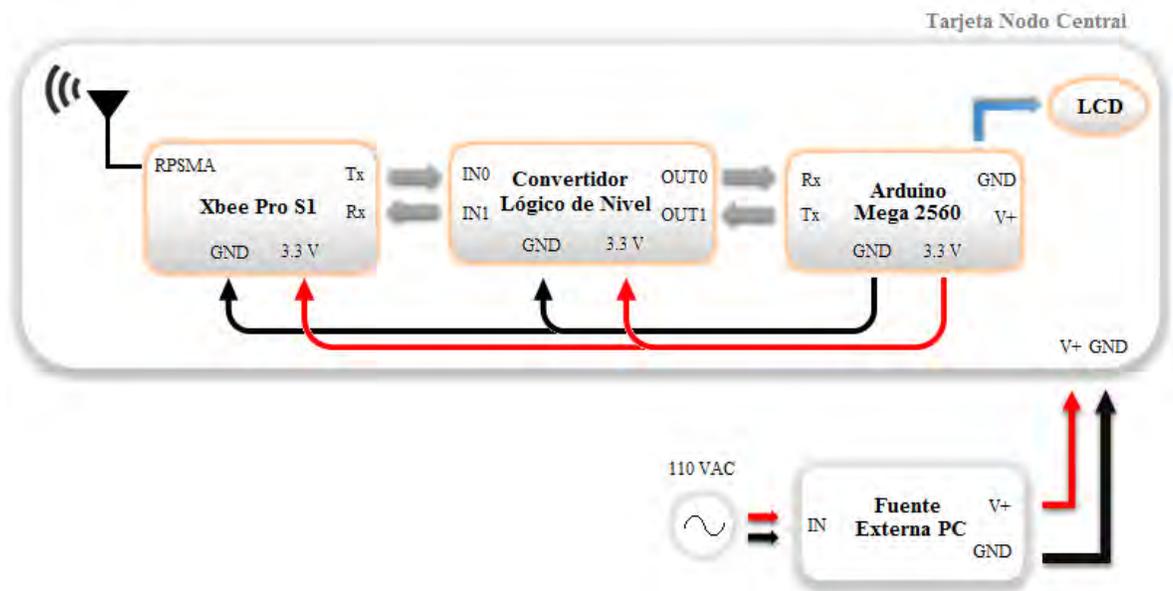
para el control de iluminación en edificaciones basada en la tecnología *Xbee* y *Arduino*. Luego de realizar una revisión bibliográfica y el análisis de desempeño de los módulos *Xbee Pro Serie 1* en entornos indoor con distintos tipos de obstáculos y condiciones de propagación, se plantea una metodología de diseño de WSNs para edificaciones. Posteriormente se aplica esta metodología para la implementación de una red con nodos casi autónomos a partir de módulos *Xbee Pro S1* con la que se busca emular una red para el control de iluminación; se espera que la red permita transmitir datos de un gran número de sensores distribuidos espacialmente hacia un controlador centralizado y que sea confiable, escalable o modificable si se requiere. Se obtuvo una metodología de diseño aplicable a diferentes tipos de aplicaciones y una red final confiable con la que es posible adquirir datos de varios nodos sensores y controlar remotamente la iluminación de algunos actuadores luminosos. Mediante la caracterización de la misma, se analiza su desempeño obteniendo buenos resultados, dando como aporte principal de la investigación, una posible solución al problema de transmisión inalámbrica de datos para control de iluminación en edificaciones y la base de nuevos estudios en este tema que actualmente tiene gran auge.

El principal aporte de esta investigación, es la implementación y caracterización de una red de sensores inalámbricos para el control de iluminación en edificaciones; para ello se realiza una revisión bibliográfica, un análisis de desempeño de los módulos *Xbee pro serie 1* en entornos *indoor* con distintos tipos de obstáculos y condiciones de propagación y se propone una metodología de diseño WSNs para edificaciones; una vez implementada la red se realiza un análisis de desempeño a la red en general, teniendo en cuenta parámetros como pérdida de paquetes, potencia de transmisión y alcance.

A manera de resumen del presente documento, en la primera sección se describe el estándar IEEE 802.15.4 y el funcionamiento en general de los dispositivos que trabajan bajo este protocolo. Posteriormente, se describe el hardware y software empleados en el proyecto. En la sección 3 se presenta la configuración y las pruebas de los módulos *Xbee Pro S1* en dos pisos del edificio Tecnológico de la Universidad de Nariño, para analizar su desempeño con los resultados obtenidos. En la siguiente sección se realiza el análisis de topologías de redes en general y bajo el protocolo IEEE 802.15.4. En la sección 5 se plantea una metodología de diseño de WSN. En seguida se presenta las pruebas y caracterización de la red final implementada con sus respectivas configuraciones y resultados. Finalmente se presenta las conclusiones respecto a los resultados del trabajo realizado.

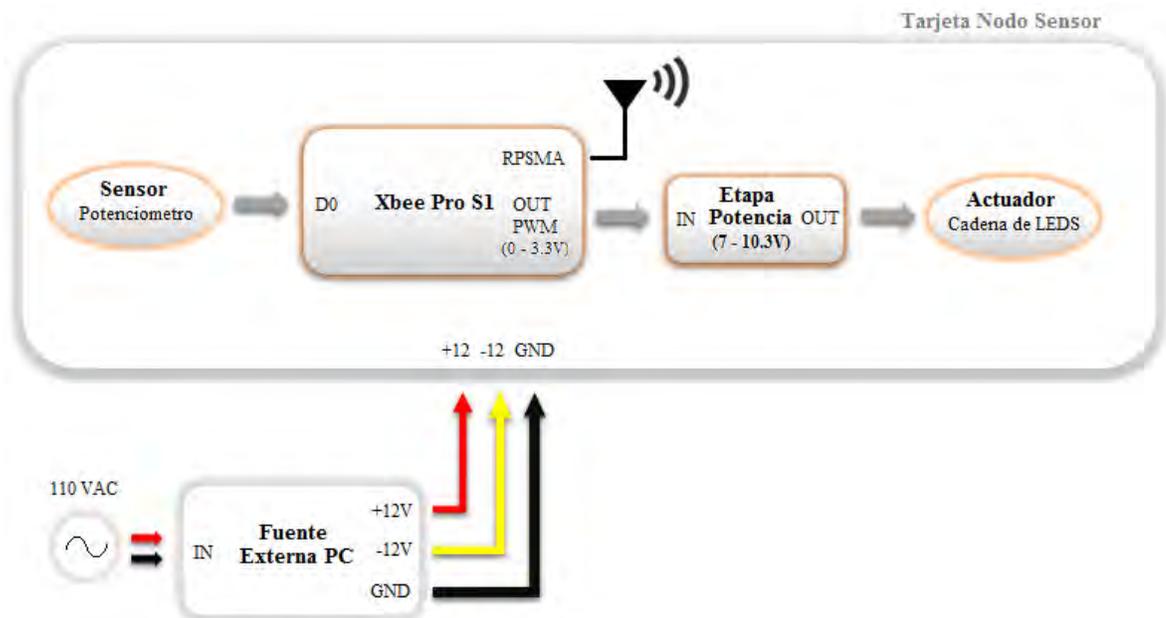
## 1. DESCRIPCIÓN DEL SISTEMA

Para realizar las pruebas del control PWM de iluminación con la red de sensores inalámbricos basada en el protocolo 802.15.4, se propone un sistema compuesto de tarjetas que representan nodos sensores y un nodo central. En la Figura 1 se presenta el diagrama de bloques correspondiente al nodo central; el diseño del nodo sensor se indica con el diagrama de la Figura 2.



**Figura 1. Diagrama de bloques nodo central**

En el nodo sensor, en la entrada analógica del módulo *Xbee* se debe ubicar el sensor de iluminación, para la prueba se utiliza un potenciómetro; de igual manera, para visualizar la respuesta del módulo al control aplicado desde el nodo central se puede ubicar cualquier actuador, para el caso se utiliza una cadena de LEDs con su etapa de potencia, para mostrar la aplicación en control de iluminación.



**Figura 2. Diagrama de bloques nodo sensor**

Cada uno de los dispositivos mostrados en las figuras anteriores se explica con detalle en esta sección más adelante.

Para el diseño e implementación de la red, inicialmente se hace un breve estudio del estándar IEEE 802.15.4 y posteriormente se describe el hardware empleado. 802.15.4 es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos [15]. La comunicación en la red de sensores inalámbricos está basada en este estándar que describe las bandas de operación, dispositivos físicos y lógicos y topologías soportadas.

### 1.1 ESTÁNDAR IEEE 802.15.4

El estándar IEEE 802.15.4 es un protocolo de comunicación desarrollado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) [15], en el que se apoyan las WSNs para su intercambio de datos. Su diseño fue contemplado para cubrir las necesidades de redes inalámbricas de área personal de baja tasa de transferencia de datos (LR-WPAN), que sean de bajo coste y bajo consumo de potencia, permitiendo que los nodos puedan ser alimentados con baterías sin la necesidad de recargarlas durante largos periodo de tiempo. La Tabla 1, muestra las características principales de este estándar.

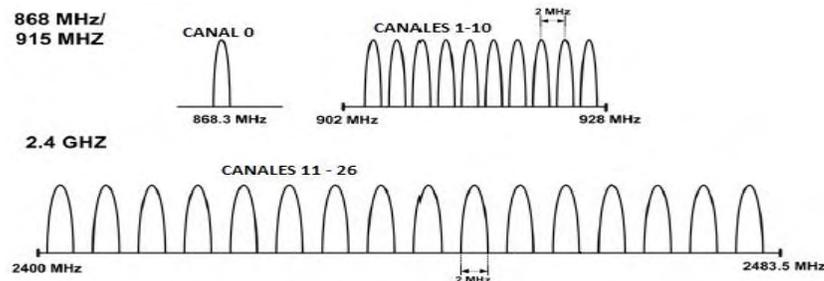
**Tabla 1. Propiedades del estándar IEEE 802.15.4**

Propiedad	Rango		
Bandas de Frecuencia	868 MHz	915 MHz	2.4 GHz
Velocidad de transmisión de datos	20kb/s	40kb/s	250 kb/s.
Canales	11	11	16
Modulación	BPSK	BPSK	O-QPSK
Alcance	10 – 20 m.		
Latencia	Abajo de los 15 ms.		
Direccionamiento	Cortos de 8 bits o 64 bits IEEE		
Temperatura	El rango de temperatura industrial: -40° a +85° C		
Topologías	Punto-punto, estrella, malla		
Máximo número de nodos	65000 nodos		

## 1.2 BANDAS DE OPERACIÓN

El estándar opera sobre las bandas libres de frecuencia reservadas internacionalmente para uso en áreas de tipo industrial, científica y médica (ISM). Particularmente, en la banda ISM de 2.4 GHz; se definen hasta 16 canales, cada uno con un ancho de banda de 2 MHz espaciados cada 3 MHz. En la Figura 3 se muestra la distribución de canales para las bandas ISM usadas en el estándar.

Se utiliza la técnica de espectro ensanchado de secuencia directa (DSSS), lográndose tasas de transmisión de hasta 250 Kbps en rangos de cobertura que oscilan entre algunos metros y un par de kilómetros, dependiendo de las condiciones del entorno. La Tabla 2, muestra las características más detalladas de las bandas de operación.



**Figura 3. Bandas de operación del Estándar IEEE 802.15.4**

**Tabla 2. Parámetros de bandas de frecuencia [34]**

Banda de Frecuencia	Modulación	Tasa de Transferencia (Kbps)
868 MHz	BPSK	20
915 MHz	BPSK	40
2.4 GHz	O-QPSK	250

En las comunicaciones inalámbricas las bandas de operación sirven, entre otras cosas, para evitar interferencias, y por tanto, se puede aplicar algoritmos para el acceso múltiple por frecuencia a los nodos que estén presentes en una WSN. No obstante, se tiene el limitante en cantidad de nodos, ya que el número de bandas es muy pequeño; esto brindaría una gran ventaja en el desempeño de una WSN que requiera muy pocos nodos.

## **Dispositivos**

**1.2.1 Físicos.** De acuerdo a la funcionalidad y complejidad del hardware de los nodos, se definen en el estándar IEEE 802.15.4 dos clases de dispositivos físicos: Dispositivos de funcionalidad completa (FFD) y Dispositivos de funcionalidad reducida (RFD).

Los dispositivos de funcionalidad completa (FFD) tienen una cantidad de memoria y recursos suficientes para ejecutar todas las funciones del estándar, incluso pueden efectuar operaciones adicionales como la de comunicarse con otras redes. En las aplicaciones suele ser un computador o un sistema embebido que hace las veces de estación base o nodo coordinador de la red o los enrutadores.

Por otra parte, los dispositivos de funcionalidad reducida (RFD) son de capacidad y funcionalidad limitadas con el fin de conseguir un bajo coste a partir de su simplicidad. En las aplicaciones reales estos suelen ser los sensores o actuadores presentes en la red [16].

**1.2.2 Lógicos.** De acuerdo al rol que los nodos desempeñan en la red y las responsabilidades que se les asigna se definen en el estándar IEEE 802.15.4 tres tipos de dispositivos lógicos: Coordinador, Enrutador y Dispositivo final.

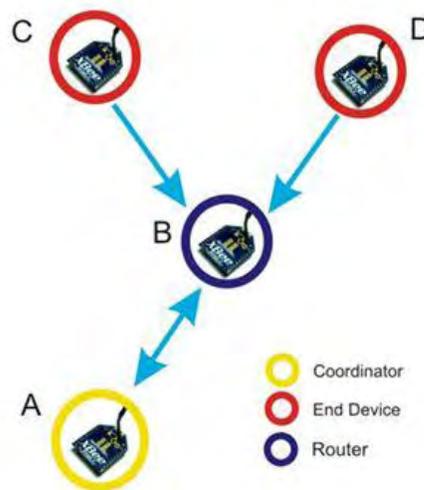
**Coordinador.** Es un dispositivo FFD único y obligatorio que debe tener la red para que pueda ser definida e inicializada. Permite vincular enrutadores y dispositivos finales y puede realizar funciones de enrutador, de tal forma que puede participar en el encaminamiento de paquetes y ser origen y/o destino de información. Dada su importancia siempre debe estar activo.

Enrutadores. Son dispositivos FFD que mantienen la información sobre la red. Interconecta los dispositivos que están más alejados del coordinador según la topología de red usada, siempre seleccionando la mejor ruta para transmitir los datos y tienen la capacidad de retransmitir paquetes de otros enrutadores o dispositivos finales. Dependiendo del tamaño y topología de la red pueden o no ser implementados.

Dispositivos finales. Son dispositivos RFD que no pueden retransmitir datos a otros destinatarios. Deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador o un enrutador y por tanto no pueden comunicarse directamente entre sí. Dada su función pueden reducir el consumo de energía de la red pues no es necesario que permanezcan activos todo el tiempo.

En la figura 4, se muestra los 3 tipos de dispositivos lógicos, en una red básica con módulos Xbee.

En los siguientes apartes se describen los principales componentes empleados para las pruebas y la implementación de la red final.



**Figura 4. Dispositivos lógicos en red básica Xbee. [37]**

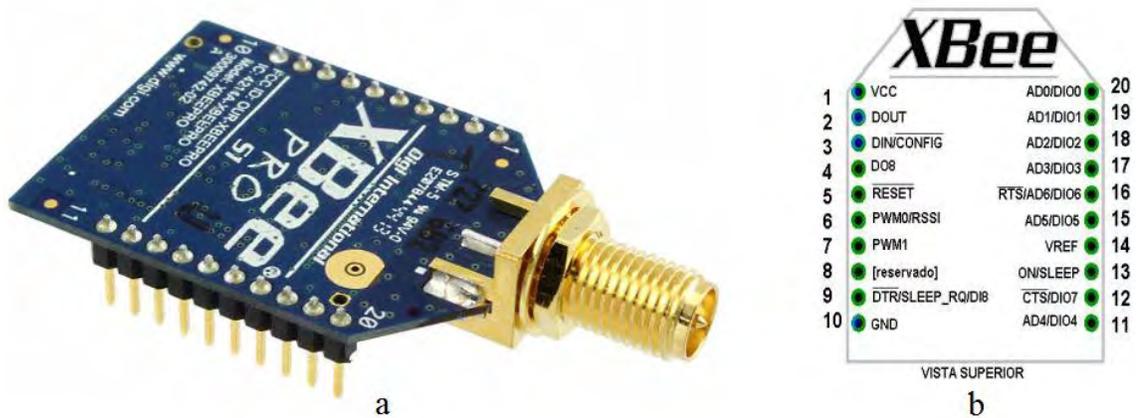
### **Módulos Xbee:**

Los módulos Xbee son dispositivos RF embebidos diseñados para trabajar bajo el estándar 802.15.4 que satisfacen las necesidades de bajo consumo de energía y bajo costo, características de una WSN.

Existen en el mercado diferentes tipos de módulos Xbee que generalmente se diferencian en la serie a la que pertenecen; los hay desde la Serie 1 hasta la serie 9 excepto por la 7. Cada serie ofrece características algunas propias de cada serie

y algunas comunes, entre las que se destacan sus diversas topologías de red, presencia de entradas analógicas, digitales, salidas PWM, retardos pequeños, frecuencia de trabajo a 2,4 GHz o a 900 MHz, entre otras.

El Xbee Pro Serie 1 viene estándar con firmware 802.15.4 para topologías de red punto-punto o estrella. Este firmware ofrece entradas ADC (conversión analógica a digital), y E/S digitales y analógicas. El Firmware Xbee 802.15.4 es significativamente más rápido que ZigBee y el rendimiento también es más alto, en la práctica es de alrededor de 80 Kbps [17]. En la Figura 5 se muestra el módulo Xbee pro s1 y su distribución de pines. Cada módulo tiene un número serial en la parte posterior que es único, viene asignado por la fábrica, este también se puede consultar cuando se conecta el módulo al computador. En la Figura 5, a se indica la ubicación del número serial en el módulo y en Figura 6 b el número serial en el software X-CTU.

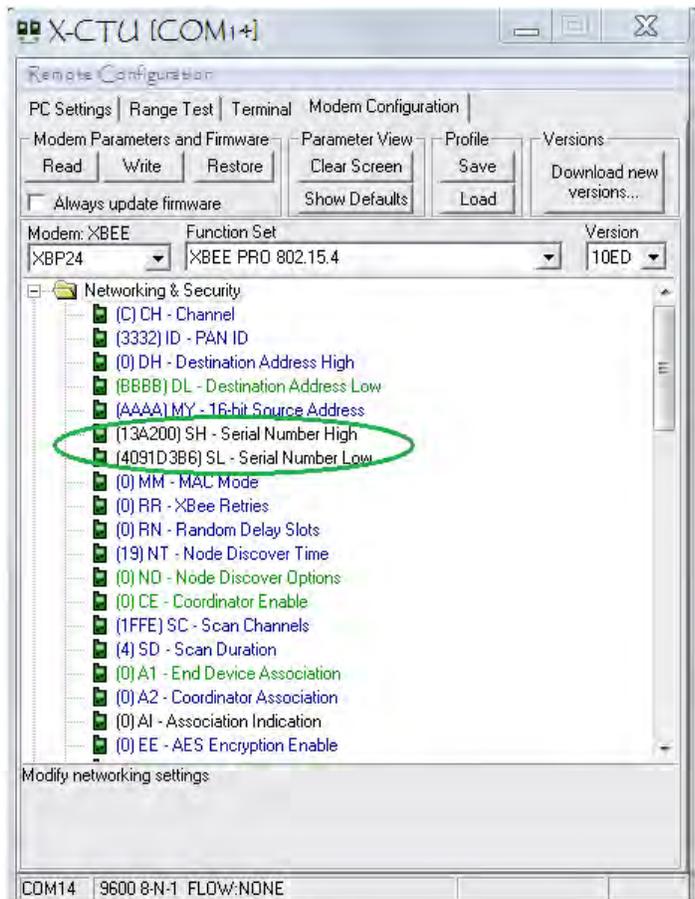


**Figura 5. a. Vista superior módulo Xbee pro s1, b. Distribución de pines módulo Xbee pro s1**

Los módulos Xbee pueden ser programados según las necesidades del usuario, por ejemplo, configurar un módulo como coordinador y otros como dispositivos finales para poder formar una red; diferentes modos de operación, para ahorro de energía (modo sleep), para conexión rápida (modo transparente), para un envío de información en tramas con características del enlace (modo API). Las configuraciones son realizadas mediante selección de parámetros del módulo cuando este se encuentra conectado al pc por el hardware. El software utilizado es X-CTU, en la Figura 6 b se puede ver un pantallazo. En la Figura 7 se muestra el hardware para la conexión USB para configurar cada módulo con el software X-CTU.

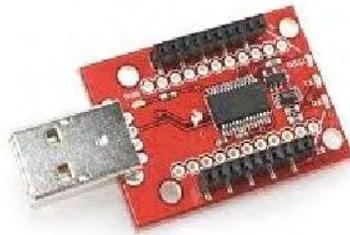


a



b

**Figura 6. a. Ubicación del número serial en el módulo Xbee. b. Número serial del módulo Xbee en el software X-CTU. Los seriales mostrados corresponden a módulos diferentes.**



a



b

**Figura 7. a. Xbee Xplorer Dongle sin conectar con el módulo Xbee. b. Xbee Xplorer Dongle con el módulo Xbee conectado al PC para la comunicación serial.**

**1.2.3 Modos de Operación.** Los módulos *Xbee*, pueden operar en 8 modos en los cuales varían las condiciones de transmisión y recepción. A continuación se presenta una descripción general de los modos de operación [18].

#### ***Modo recibir/transmitir.***

Este modo permite que al módulo le llegue algún paquete RF a través de la antena (modo Receive) o cuando se manda información serial al buffer del pin 3 (UART Data in) que luego será transmitida (modo Transmit); se encuentra por defecto en los módulos, es recomendable utilizarlo si se desea un modo sencillo de comunicación entre diferentes nodos, ya sea entre dos o múltiples nodos.

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

Además es posible enviar información por dos modos. Unicast y Broadcast. Por el primero, la comunicación es desde un punto a otro, y es el único modo que permite respuesta de quien recibe el paquete RF, es decir, quien recibe debe enviar un **ACK** (paquete de datos interno del módulo, invisible al usuario que indica que la información recibida) a la dirección de origen. Quien envió el

paquete, espera recibir un **ACK**, pero en caso de que no le llegue, reenviará el paquete hasta 3 veces o hasta que reciba el **ACK**. En el modo Broadcast la comunicación es entre un nodo y a todos los nodos de la red. En este modo, no hay confirmación por **ACK**.

### ***Modo de Bajo Consumo (Sleep Mode).***

Permite que el módulo RF entre en un modo de bajo consumo de energía cuando no se encuentra en uso. Para poder entrar en modo de sueño, se debe cumplir una de las siguientes condiciones: Sleep\_RQ (pin 9) está en alto y tiene activa la opción sleep mode (SM= 1, 2 o 5); El módulo está en reposo (no hay transmisión ni recepción de datos) durante el tiempo definido por el parámetro ST (Time before Sleep). [ST sólo está activado cuando SM= 4, 5].

La configuración de los ciclos de sueño se realiza principalmente con el comando SM. Por defecto, los modos de sueños están deshabilitados (SM=0), permaneciendo el módulo en estado de reposo/recepción. En este estado el módulo está siempre preparado para responder a un comando, ya sea, por el puerto serial o la interfaz RF.

### ***Modo de Comando.***

Permite ingresar comandos AT al módulo Xbee, para configurar, ajustar o modificar parámetros, como la dirección propia o la de destino, así como su modo de operación, entre otras funciones. 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.

Para ingresar a este modo se debe esperar un tiempo dado por el comando GT (Guard Time, por defecto ATGT=0x3E84 equivale a 1000 ms) luego ingresar **+++** y luego esperar otro tiempo GT. Como respuesta el módulo entregará un **OK**. El módulo Xbee viene por defecto con una velocidad de 9600 bps. En caso de no poder ingresar al modo de comandos, es posible que sea debido a la diferencia de velocidades entre el módulo y la interfaz que se comunica vía serial.

### ***Modo Transparente***

En este modo todo lo que ingresa por el pin 3 (Data in), es guardado en el buffer de entrada y luego transmitido, y todo lo que ingresa como paquete RF, es guardado en el buffer de salida y luego enviado por el pin 2 (Data out). El modo Transparente viene por defecto en los módulos Xbee.

Este modo está destinado principalmente a la comunicación punto a punto, donde no es necesario ningún tipo de control. También se usa para reemplazar alguna

conexión serial por cable, ya que es la configuración más sencilla posible y no requiere una mayor configuración.

En este modo, la información es recibida por el pin 3 del módulo *Xbee*, y guardada en el buffer de entrada. Dependiendo de cómo se configure el comando RO, se puede transmitir la información apenas llegue un carácter (RO=0) o después de un tiempo dado sin recibir ningún carácter serial por el pin 3. En ese momento, se toma lo que se tenga en el buffer de entrada, se empaqueta, es decir, se integra a un paquete RF, y se transmite. Otra condición que puede cumplirse para la transmisión es cuando el buffer de entrada se llena, esto es, más de 100 bytes de información.

### ***Modo de operación API***

Este modo es más complejo, pero permite el uso de tramas o frames con cabeceras que aseguran la entrega de los datos, al estilo TCP. Extiende el nivel en el cual la aplicación del cliente, puede interactuar con las capacidades de red del módulo.

Cuando el módulo *Xbee* se encuentra en este modo, toda la información que entra y sale, es empaquetada en tramas, que definen operaciones y eventos dentro del módulo, dependiendo del tipo de trama que se transmita.

Así, una trama de Transmisión de Información (información recibida por el pin 3 o DIN) incluye:

- Trama de información RF transmitida.
- Trama de comandos (equivalente a comandos AT).

Mientras que una trama de Recepción de Información incluye:

- Trama de información RF recibida.
- Comando de respuesta.
- Notificaciones de eventos como Reset, Associate, Disassociate, etc.

Este modo API, provee alternativas para la configuración del módulo y ruteo de la información en la capa de aplicación del cliente. Un cliente puede enviar información al módulo *Xbee* y estos datos serán contenidos en una trama cuya cabecera tendrá información útil referente el módulo.

El módulo puede enviar paquetes de datos contenidos en tramas a otros módulos de destino con información importante como la dirección de origen, RSSI (potencia de la señal de recepción) e información de la carga útil de los paquetes recibidos.

Entre las opciones que permite el modo de operación API, se tienen:

- Transmitir información a múltiples destinatarios, sin entrar al modo de Comandos.
- Recibir estado de éxito/falla de cada paquete RF transmitido.
- Identificar la dirección de origen de cada paquete recibido.
- Configuración remota de dispositivos.

Este modo de operación de los *Xbee* se utiliza más adelante en las pruebas de la red implementada; en esa sección se describe cada una de las tramas empleadas en la comunicación de los nodos que interfieren en el sistema.

### ***Idle***

Cuando el módulo no se está en ninguno de los otros modos, se encuentra en éste. Es decir, si no está ni transmitiendo ni recibiendo, ni ahorrando energía ni en el modo de comandos, entonces se dice que se encuentra en un estado al que se le llama IDLE.

### **Software de configuración X-CTU:**

**1.2.4 Descripción.** X-CTU es una aplicación gratuita multiplataforma diseñado para permitir a los desarrolladores interactuar con módulos Digi RF a través de una interfaz gráfica fácil de usar. Incluye herramientas que hacen que sea fácil de configurar y probar los módulos de RF *Xbee*.

#### **1.2.5 Características:**

- Puede administrar y configurar varios dispositivos de RF, incluso de forma remota (over-the-air) los dispositivos conectados, utilizando el modo API.
- El proceso de actualización del firmware restaura la configuración del módulo, la manipulación de forma automática los cambios de modo y la velocidad de transmisión.
- Puede guardar sus sesiones de consola y cargarlos en un PC que ejecuta X-CTU diferente.
- Generador de tramas: Genera fácilmente cualquier tipo de marco API para guardar su valor.
- Intérprete Marcos: Decodificar un marco de principios activos y de ver sus valores de trama específicos.
- Recuperación: Recuperar módulos de radio que han dañado el firmware o están en el modo de programación.
- Sesión de la consola de carga: Carga de una sesión de consola guardada en cualquier X-CTU PC en funcionamiento.

- Prueba de Rango: Realizar una prueba de rango entre 2 módulos de radio de la misma red.
- Explorer Firmware: Navegar a través de firmware de la biblioteca de X-CTU.
- Un proceso de actualización le permite actualizar automáticamente la aplicación en sí y la biblioteca firmware de radio sin necesidad de descargar ningún archivo extra.[19]

### **Xbee Xplorer Dongle:**

a) Para poder configurar los parámetro de los módulos y obtener los enlaces deseados en la red es necesario conectarlos uno a uno al computador y mediante el software X-CTU modificarles las características según las necesidades, como dirección de origen, de destino, canal de trabajo, PAN ID, entre otros; para la comunicación con el computador es necesario utilizar un hardware que permita estas configuraciones mediante el puerto USB del PC.

El *Xbee explorer* USB permite conectar directamente a un computador y utilizar cualquier módulo *Xbee* mediante un puerto USB. Al conectar un módulo *Xbee* al PC tendremos acceso a los pines Tx/Rx del *Xbee* y estará listo para funcionar.

Permite la conexión con el software X-CTU para realizar las configuraciones necesarias en cada módulo. Es ideal para establecer una base inalámbrica desde un ordenador y así poder conectar sin cables a una placa que utilice un módulo *Xbee*.

Esta unidad funciona con todos los módulos *Xbee* incluyendo la Serie 1 y Serie 2.5, la versión estándar y Pro. Tiene LEDs de estado para encendido de la tarjeta y estado de la comunicación serial. (Tx/Rx)[20]. En el presente trabajo se utiliza estos dispositivos para la configuración de los módulos y para realizar una de las pruebas de la red, para las demás se implementan circuitos impresos que hacen las veces de coordinador y nodos sensores.

### **Arduino:**

Es una plataforma de hardware libre para la creación de proyectos, basada en software y hardware flexibles y fáciles de usar. Puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores.

El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse de manera autónoma, es decir, sin necesidad de conectar a un ordenador y también tienen la posibilidad de comunicarse con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP) [21].

**1.2.6 Arduino Mega 2560.** Es una placa electrónica basada en el microprocesador ATmega2560. Tiene 54 entradas/salidas digitales (15 proporcionan salida PWM), 16 entradas analógicas, 4 UARTs, un cristal oscilador de 16 MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset.

Puede ser alimentado vía la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente. El Mega2560 difiere de todas las placas anteriores ya que no utiliza el chip controlador de USB a serial FTDI. En su lugar, ofrece las ATmega16U2 programado como convertidor USB a serie.

### 1.2.7 Características:

- Microcontrolador: ATmega2560
- Voltaje de funcionamiento: 5V, 3.3V
- Voltaje de entrada (recomendado): 7-12V
- Voltaje de entrada (limite): 6-20V
- Pines E/S digitales: 54 (15 proporcionan salida PWM)
- Pines de entrada analógica: 16
- Intensidad por pin: 40 mA
- Intensidad en pin 3.3V: 50 mA
- Memoria Flash: 256 KB de las cuales 8 KB las usa el gestor de arranque.
- SRAM: 8 KB
- EEPROM: 4 KB
- Velocidad de reloj: 16 MHz

Todas las características de la tarjeta se pueden consultar en [22].

En la WSN se tiene un nodo coordinador, encargado de direccionar la información y tomar las decisiones, es aquí donde se implementa el método de control deseado, para esta aplicación PWM; por tanto es necesario un dispositivo controlador que acompañe al módulo inalámbrico. Para las pruebas en este proyecto se escoge el Arduino Mega 2560 porque a diferencia de un microcontrolador, permite una fácil conexión ya sea directamente a un computador únicamente mediante un cable USB; a los módulos y demás componentes a través de cables o *jumper*, ofrece la alimentación adecuada a los módulos Xbee; da la posibilidad de conectar hasta 4 módulos por sus puertos seriales, además de múltiples sensores en sus entradas analógicas en caso de ser necesarios, esto da cabida a comprobaciones en las pruebas de los diferentes tipos de tramas y a la versatilidad del sistema para posibles modificaciones; se explica más a fondo en la sección V. En este proyecto es posible utilizar también un Arduino con menos prestaciones como el Arduino UNO o Leonardo, sin embargo el proyecto quedaría

limitado a modificaciones y no hubiera sido posible las comprobaciones durante las pruebas.

### **Convertidor lógico de nivel ADAF-395:**

Este dispositivo es un convertidor de nivel basado en el chip TXB0108 bidireccional de 8 canales, en el presente trabajo se emplea para realizar conversiones de voltaje de 3.3V a 5V y viceversa; se puede conectar a un protoboard directamente y puede ser utilizada con señales serial, I2C, SPI y con cualquier otro tipo de señal digital. Necesita ser alimentada tanto del lado de alto voltaje como del lado de bajo voltaje, los pines están identificados como entradas (Input) y salidas (output) [23].

El voltaje de trabajo de los módulos *Xbee pro s1* está entre 3.0 y 3.4 V y los terminales Dout y Din del mismo se deben conectar a los pines del Arduino TX1 y TX2 respectivamente; es por eso que se utiliza este convertidor digital como protección para el módulo *Xbee* puesto que si se sobrepasa su voltaje de trabajo el modulo se podría dañar; este convertidor se alimenta con 5V y 3.3V, aunque es posible alimentarlo con voltajes más bajos y así realizar otras conversiones. Para más información acerca de este dispositivo se puede consultar su hoja de características.

## **2. PRUEBAS PARA CALCULO DE DESEMPEÑO DE UNA WSN**

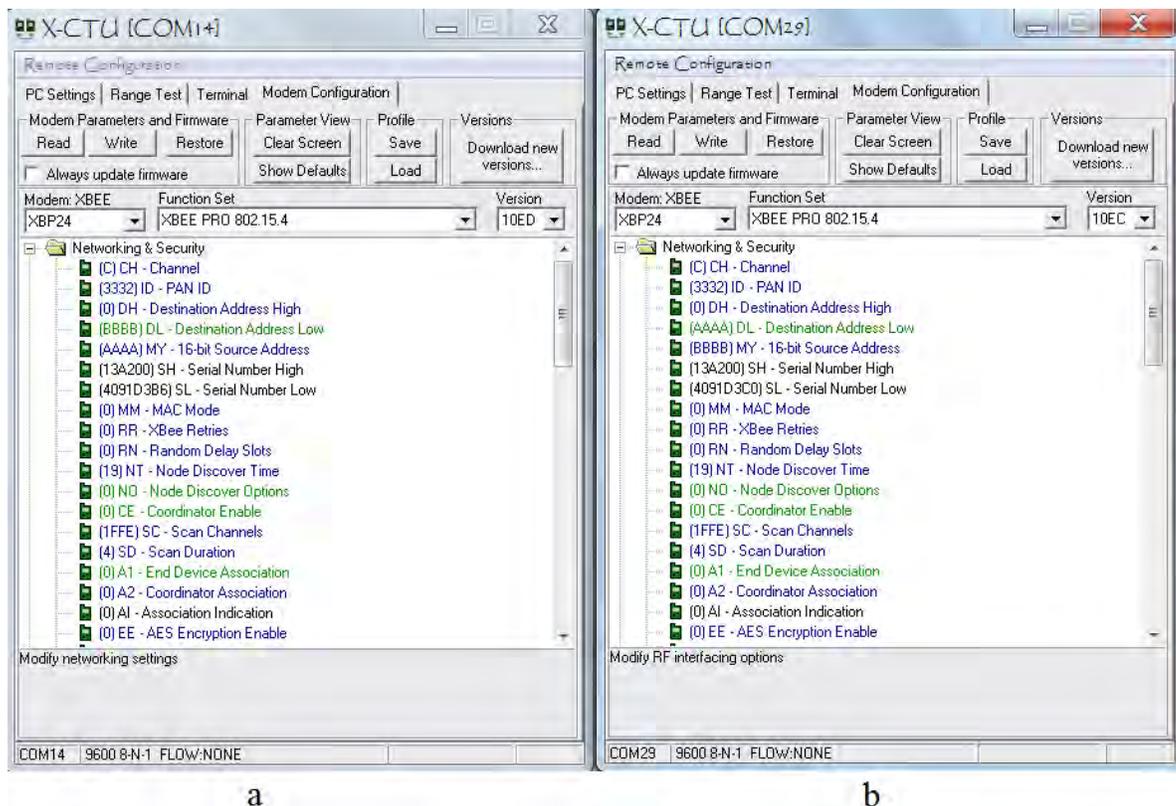
Para verificar el desempeño de los módulos *Xbee* y posteriormente el de la red, se realizan pruebas bajo diferentes condiciones de propagación en el entorno destinado para la aplicación, que son los pisos 3 y 4 del edificio tecnológico de la Universidad de Nariño, con el fin de determinar el alcance de los módulos en dicho entorno y el comportamiento de estos a diferentes tipos y cantidad de obstáculos.

Los parámetros de desempeño que se tendrán en cuenta son tasa de pérdida de paquetes, calidad de la señal (RSSI), alcance y obstáculos. Se realizaron dos tipos de pruebas, una en la que se enlazan dos módulos punto a punto utilizando 2 módulos *Xbee* conectados a 2 computadores mediante los *Xbee Explorer Dongle* y la segunda en la que se emplean 2 computadores cada uno conectado a un módulo *Xbee* y un tercer módulo conectado a una tarjeta Arduino Mega que hace las veces de repetidor y reenvía un mensaje de cualquiera de los otros dos módulos.

### **2.1 CONFIGURACIÓN DE MÓDULOS PARA CONEXIONES 2 Y 3 NODOS**

A continuación se describe las configuraciones realizadas para llevar a cabo las pruebas.

Para las pruebas con dos nodos se configuraron los módulos para comunicación punto a punto en la interfaz X-CTU. La Figura 8 y la Tabla 3, muestran las configuraciones para este caso.



**Figura 8. Configuración punto a punto de dos módulos Xbee pro s1 con X-CTU, a. Configuración parámetros módulo 1, b. Configuración parámetros módulo 2.**

Para que dos Xbee se comuniquen deben pertenecer a la misma subred (PAN) y al mismo canal, además de una correspondencia en las direcciones de origen del uno y destino del otro. En la Tabla III se puede entender mejor como configurar dos módulos para que se comuniquen punto a punto. La dirección de destino (destination address) del módulo 1 debe ser igual a la dirección de origen (source address) del módulo 2, así los datos que se deseen enviar del dispositivo 1 llegarán al 2. Igual para obtener el enlace en el otro sentido, de 2 hacia 1; la misma dirección de origen de 1 se usará como dirección de destino del módulo 2.

**Tabla 3. Configuración de cada módulo punto a punto**

Propiedad	Parámetro Modulo 1	Parámetro Modulo 2
Modem	XBP24	XBP24
Function set	XBEE PRO 802.15.4	XBEE PRO 802.15.4
Firmware versión	10ED	10EC
PAN ID 16 bits	3332	3332
Canal	C	C
Destination Address High - 64 bits	0	0
Destination Address Low - 64 bits	BBBB	AAAA
16 bits Source Address	AAAA	BBBB
Serial Number High	13A200	13A200
Serial Number Low	4091D3B6	4091D3C0

Para la configuración de los módulos se utilizó en su mayoría las configuraciones por defecto, de este modo, el identificador de PAN está configurado como **3332** y el canal como **C**; este canal es uno de los 16 disponibles para el estándar IEEE 802.15.4 como se muestra en la Figura 9, posee 3 MHz de ancho de banda y se encuentra comprendido entre las frecuencias 2,4075 GHz y 2,4125 GHz con frecuencia central en 2,41 GHz. En la Figura 5 se indican a los canales disponibles para el protocolo IEEE 802.15.4.

A partir del ancho de banda del canal se puede calcular el ruido térmico del mismo mediante la Ecuación (1).

$$N = KTB [W] \quad (1)$$

Dónde:

*N*: Es el ruido del canal en Watios [*W*]

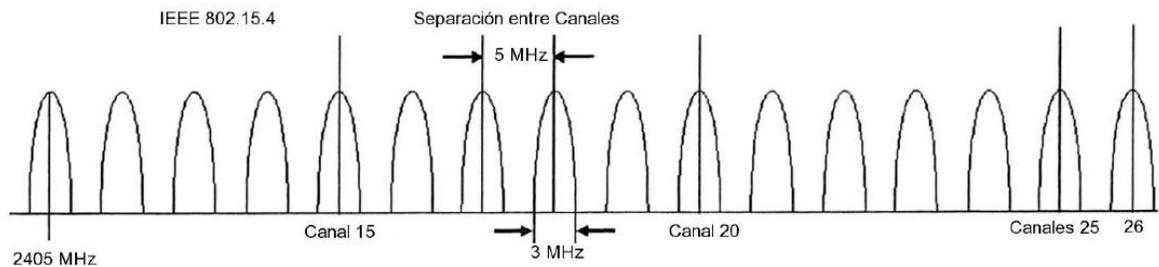
*K*: La constante de Boltzmann  $1,38 * 10^{-23}$  [*Joules/°K*]

*T*: Temperatura [*°K*]

*B*: Ancho de banda [*Hz*]

Teniendo una temperatura de aproximadamente 20 °C se obtiene una potencia del ruido de  $1.21 \cdot 10^{-14}$  W; usando la Ecuación (2), podemos obtener la potencia en dBm.

$$P(\text{dBm}) = 10 \cdot \log_{10} \left( 1000 \cdot \frac{P(\text{W})}{1\text{W}} \right) = 10 \cdot \log_{10} \left( \frac{P(\text{W})}{1\text{W}} \right) + 30 = -109.16 \text{ dBm} \quad (2)$$



**Figura 9. Canales disponibles para el protocolo IEEE 802.15.4 a 2.4 GHz [15]**

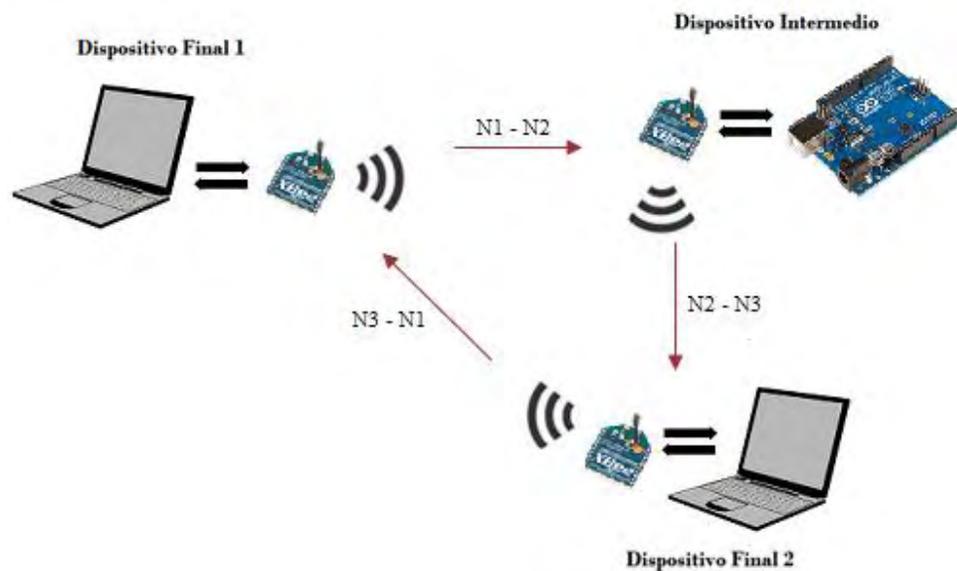
En la Tabla 4, se muestra los diferentes canales disponibles, las frecuencias correspondientes a cada uno y el comando AT para cada uno de ellos.

**Tabla 4. Canales de Frecuencia y su respectivo comando at**

canal	Hexadecim al	Frecuencia [GHz]			Comando AT
		Inferior	Central	Superior	
11	0x0B	2,4025	2,4050	2,4075	ATCH0B
<b>12</b>	<b>0x0C</b>	<b>2,4075</b>	<b>2,4100</b>	<b>2,4125</b>	<b>ATCH0C</b>
13	0x0D	2,4125	2,4150	2,4175	ATCH0D
14	0x0E	2,4175	2,4200	2,4225	ATCH0E
15	0x0F	2,4225	2,4250	2,4275	ATCH0F
15	0x10	2,4225	2,4250	2,4275	ATCH10
17	0x11	2,4325	2,4350	2,4375	ATCH11
18	0x12	2,4375	2,4400	2,4425	ATCH12
19	0x13	2,4425	2,4450	2,4475	ATCH13
20	0x14	2,4475	2,4500	2,4525	ATCH14
21	0x15	2,4525	2,4550	2,4575	ATCH15
22	0x16	2,4575	2,4600	2,4625	ATCH16
23	0x17	2,4625	2,4650	2,4675	ATCH17
24	0x18	2,4675	2,4700	2,4725	ATCH18
25	0x19	2,4725	2,4750	2,4775	ATCH19
26	0x1A	2,4775	2,4800	2,4825	ATCH1A
Frecuencia Base		2,405			GHz

Para las pruebas con 3 nodos se configura los módulos de manera unidireccional en pares como se muestra en la Figura 10, utilizando en el nodo intermedio un dispositivo controlador (Arduino) que toma por la entrada serial los paquetes recibidos del dispositivo final 1 y los almacena, guarda la potencia del enlace N1-N2; comunica nuevamente el mensaje al *Xbee* por el puerto serial para que lo envíe al dispositivo final 2. Cuando se solicite por el nodo 3 la potencia del primer enlace, el Arduino se la enviará mediante el serial del *Xbee*.

La configuración de cada módulo se realiza también en X-CTU como se puede observar en la Tabla 5, En la Figura 10, se tiene un pantallazo de las modificaciones de los parámetros para los tres módulos utilizados. Para que la comunicación se efectuó fue necesario programar un código en Arduino que permite enviar datos entre los 3 nodos, de esta manera se obtiene la tasa de pérdida de datos, la potencia de envío de los módulos y el alcance de la comunicación entre dos dispositivos finales.



**Figura 10. Configuración para pruebas con 3 módulos.**

**Tabla 5. Configuración módulos conexión tres nodos**

Propiedad	Parámetro Modulo 1	Parámetro Modulo 2	Parámetro Modulo 3
Modem	XBP24	XBP24	XBP24
Function set	XBEE PRO	XBEE PRO	XBEE PRO
	802.15.4	802.15.4	802.15.4
Firmware versión	10ED	10ED	10ED
PAN ID 16 bits	3332	3332	3332
Canal	C	C	C
Destination Address High - 64 bits (DH)	0	0	0
Destination Address Low - 64 bits (DL)	BBBB	CCCC	AAAA
16 bits Source Address (MY)	AAAA	BBBB	CCCC
Serial Number High	13A200	13A200	13A200
Serial Number Low	4091D3B9	40BA731F	40B32A35



**Figura 11. Configuración tres módulos en X-CTU**

## 2.2 PRUEBAS Y ANÁLISIS

Se realizaron pruebas con los módulos inalámbricos *Xbee pro s1* en los pisos 3 y 4 del edificio tecnológico de la Udenar y se obtuvieron los siguientes datos de desempeño:

- Alcance.
- tasa de pérdida de paquetes.
- potencia de transmisión.

Las pruebas realizadas se efectuaron en diferentes disposiciones tanto en línea de vista, como con uno, dos o tres obstáculos. Además como medida adicional se sometió a los módulos a una prueba final en un entorno exterior para comparar los resultados con el mayor alcance obtenido en el ambiente *indoor*, esto con el fin de comprobar el comportamiento de los módulos en comparación a la hoja de datos de los mismos.

En los planos 1, 2 y 3 del anexo A se muestra la ubicación y disposición de los nodos para cada prueba realizada en el edificio tecnológico de la universidad de Nariño.

El plano 1 corresponde a las pruebas realizadas con dos nodos en el piso 4, el plano 2 a las realizadas en el piso 3 tanto de dos nodos como de tres y finalmente, el plano 3 indica las pruebas de tres nodos realizadas en el cuarto piso.

Para cada prueba se repitió la toma de los datos tres veces, y en el proceso de determinación de la pérdida de paquetes, se utiliza la interfaz X-CTU en la opción *range test* enviado una muestra cada 300 *ms*, un total de 1000 paquetes.

Las tablas 6 y 7 indican los datos obtenidos en las pruebas de los módulos con diferentes características en el entorno de propagación *indoor*.

La tabla 6, corresponde a comunicación punto a punto y la VII a enlaces con tres nodos.

**Tabla 6. Pruebas módulos inalámbricos dos nodos**

PRUEBA	CANTIDAD DE OBSTÁCULOS	DISTANCIA [m]	RSSI [dBm] (-100, -36)*		n	TASA DE PERDIDA DE PAQUETES** %	
			N <sub>1</sub>	N <sub>2</sub>		N <sub>1</sub>	N <sub>2</sub>
1	0	17.00	-58.67	-56.00	1.24	0.00	0.00
2	0	74.00	-71.33	-68.33	0.86	0.50	0.47
3	0	220.10	-75.00	-75.67	0.36	2.40	5.07
4	1	15.00	-65.33	-65.00	0.72	0.40	0.23
5	1	17.00	-60.67	-62.67	0.87	0.40	0.50
6	1	17.00	-54.67	-54.67	1.32	1.23	0.93
7	2	21.45	-77.00	-73.00	0.30	0.97	4.27
8	2	25.40	-79.00	-78.33	-0.02	2.43	0.13
9	3	29.04	-88.00	-89.00	-0.71	13.97	13.10
10	3	34.84	-94.33	-99.00	-1.44	11.87	13.93

Relaciona cada prueba con la intensidad de la fuerza de la señal (RSSI) vista por cada nodo, así como la tasa de pérdida de paquetes y distancia entre transmisor y receptor.

\*rango de valores en el que se recibe satisfactoriamente los paquetes de datos, tomado de la hoja de características del Xbee s1. La intensidad es solicitada mediante el comando ATDB, la respuesta se tiene en hexadecimal 0x24 - 0x64 (x-1dBm).

\*\*Se obtiene de la diferencia entre 100% y el porcentaje de paquetes correctamente recibidos.

La prueba 2 se realizó a una distancia de 74 m que fue la máxima encontrada con línea de vista en el piso 4 del edificio, obteniendo en la fuerza de la señal (RSSI) valores considerados aceptables ya que se encuentran en la mitad del rango en el que se reciben satisfactoriamente los paquetes de información. Se obtuvo una tasa de pérdida de paquetes baja, es decir, que un porcentaje muy alto de los mensajes transmitidos llegan correctamente.

La prueba 3 en ambiente *outdoor* con línea de vista, se realizó con el fin de conocer el alcance máximo de comunicación entre dos Xbee pro s1; se obtuvo que a una distancia de 220.1m la tasa de pérdida de paquetes en ambos nodos es baja, es decir que alrededor del 96.3% de los mensajes enviados fueron recibidos correctamente. Se tiene una señal aceptable como se puede ver en la columna RSSI.

Las pruebas 1, 5 y 6 se realizaron con igual distancia entre los módulos, en ambiente *indoor* con diferentes condiciones; en línea de vista, con una interferencia (muro), y con interferencia parcial (muro estructural) respectivamente. Se evidencia que en la prueba 6 se tiene mejor calidad de la señal debido a los rebotes que se pudieron ocasionar, pero por la existencia de metal en el muro ocurre un aumento en el porcentaje de mensajes mal recibidos.

En línea de vista se obtienen los mejores resultados en cuanto a tasa de pérdida de paquetes, todos los mensajes se reciben correctamente. En la prueba 6 (con un muro) se tiene una señal con calidad más baja con respecto a los otros dos con

iguales condiciones, la pérdida de paquetes no tiene un aumento significativo en comparación a la prueba sin obstáculos.

La prueba 8 es realizada con dos obstáculos (muros) a una distancia mayor a las anteriores, se evidencia la disminución en la fuerza de la señal recibida, igualmente la pérdida de paquetes es más significativa que en caso anteriores. El peor de los casos encontrados al realizar los enlaces entre 2 nodos es la prueba 10, esta se realizó con 3 obstáculos a una distancia cercana (35 m). Los valores de RSSI obtenidos por ambos nodos fueron los más bajos de todas las pruebas, la tasa de pérdida de paquetes muestra que alrededor de 12% de los mensajes enviados fueron recibidos con errores. La potencia recibida por el nodo 2 en esta prueba, es muy baja, -99 dBm, para comparar con la potencia del ruido  $P_N = -109.16 \text{ dBm}$  se calcula la relación señal a ruido (SNR) con la Ecuación 3.

$$SNR = \frac{P_s}{P_N} \quad (3)$$

$$SNR = 0.91$$

Entre más bajo sea el valor de SNR mejor es la transmisión de la señal; para el caso de la prueba 10, la relación señal a ruido indica que la potencia de la señal recibida es baja, muy cercana al ruido, esto puede ser lo que ocasione la tasa de pérdida de paquetes alta para este caso. Por tanto, la prueba 10 muestra la necesidad de enlaces con 3 nodos que mejoren la calidad de la señal y se obtenga menor relación señal a ruido y tasa de pérdida de paquetes baja.

La ecuación de Friis relaciona la potencia transmitida con la recibida entre dos módulos de transmisión (en este caso), encontrados a una distancia  $d$  uno del otro. La Ecuación (4) calcula las pérdidas de propagación a partir de la distancia entre los puntos, la frecuencia de trabajo y una constante  $n$  que es el exponente de pérdidas de propagación, el cual depende del entorno, si es en interiores o exteriores, cantidad y tipo de obstáculos existentes, entre otros.

Con los datos obtenidos de las pruebas de dos nodos es posible encontrar el exponente de pérdidas de propagación ( $n$ ), como se indica a continuación:

Las pérdidas de propagación a partir de la ecuación de Friis (4):

$$L = 32.45 + 10 * n * \log_{10}d + 20 * \log_{10}f \quad (4)$$

La potencia de recepción depende de la transmitida, la ganancia de cada antena y las pérdidas de propagación, como se muestra en la Ecuación 5.

$$P_{rx} = P_{tx} + G_1 + G_2 - L \quad (5)$$

Dónde:

$L$ : Pérdidas de propagación

$n$ : Exponente de pérdidas de propagación

$d$ : Distancia [Km] entre transmisor y receptor

$f$ : Frecuencia [MHz]

$P_{rx}$ : Potencia de recepción

$P_{tx}$ : Potencia de transmisión

$G_n$ : Ganancia de la antena  $n$ , para este caso, las antenas son iguales

Entonces  $G_1 = G_2 = G$

Despejando  $n$  de la Ecuación 4:

$$n = \frac{L - 32.45 - 20 \cdot \log_{10} f}{10 \cdot \log_{10} d} \quad (6)$$

Obteniendo  $L$  de la Ecuación 5

$$L = P_{tx} + 2G - P_{rx} \quad (7)$$

Reemplazando (7) en (6) tenemos la Ecuación 8 que sirve para encontrar el exponente de pérdidas de propagación que es la constante deseada:

$$n = \frac{P_{tx} + 2G - P_{rx} - 32.45 - 20 \cdot \log_{10} f}{10 \cdot \log_{10} d} \quad (8)$$

Las antenas utilizadas en el receptor y transmisor tienen ganancias iguales ( $G = 2 \text{ dBi}$ ). La frecuencia de trabajo para los módulos Xbee pro s1 es de  $2.4 \text{ GHz}$ . La potencia de transmisión utilizada por los módulos Xbee pro s1 es de  $18 \text{ dBm}$ [17].

Para la potencia de recepción y las distancias se utilizarán los datos consignados en las columnas 3 y 5 de la Tabla V, así para cada prueba realizada con diferentes condiciones, se obtendrá un valor del exponente de pérdidas; estos datos se encuentran en la columna 6 de la Tabla V.

En el caso de la prueba 10, donde la potencia recibida en el nodo 2 es de  $-99 \text{ dBm}$ , el exponente de pérdidas de paquetes es el más bajo encontrado, esto es debido a que la cantidad de obstáculos y la distancia entre los dos módulos inalámbricos perjudican en alto grado la propagación de la señal.

En las pruebas de 3 nodos el propósito es transmitir una señal desde el nodo 1 hasta el nodo 3, que sobrepase más obstáculos y mayor distancia que en las pruebas con dos nodos.

En la Tabla VII, la prueba 11 la señal recorre una distancia total de 30.13 m sobrepasando en total 7 obstáculos, el porcentaje de pérdida de paquetes visto por el nodo final es realmente bajo, 1.9%, es decir que el 98.1% de los paquetes enviados son recibidos correctamente; en cuanto a potencia, en el último intervalo se obtuvo -76 dBm que es aceptable considerando que el límite inferior es -100 dBm [18] y la potencia del ruido en ese canal es de -109.16 dBm.

Comparando las pruebas 11 y 9, la distancia total recorrida por el mensaje es similar, pero se puede diferenciar que en el caso de tres nodos, la pérdida de paquetes es menor, y la potencia es mayor teniendo en cuenta que sobrepasan más obstáculos.

En la prueba 13, el mensaje recorre la mayor distancia de todas las pruebas de tres nodos. Se puede notar, con ayuda de la tabla VII, que en este enlace interviniendo tres nodos no es mejor que si solo se tuvieran el nodo 1 y el 3, sin necesidad del 2; ya que en los dos casos se tienen 4 obstáculos. Sin embargo, pero la tasa de pérdida de paquetes entre el enlace de dos nodos es menor. (De 1.73% a 2.60% para el caso de tres nodos), y la potencia de la señal recibida en el enlace directo es mayor que en la que se utiliza un nodo intermedio. -75.00 dBm a -82.33 dBm.

Por lo anterior es necesario tener en cuenta las necesidades de cobertura, porque si se requiere comunicación con tres nodos en la cual el intermedio también cumpla la función de sensor/actuador, es probable que un enlace de tres nodos sea el adecuado. Como se puede notar en el plano 3, esta prueba tiene una distribución favorable al momento de cubrir un área extensa.

La prueba 14, realizada en el tercer piso, presenta el mejor comportamiento en cuanto a paquetes recibidos, con 0% erróneos (es decir, todos los paquetes enviados por el nodo 1 son recibidos correctamente por el nodo 3). En cuanto al RSSI, es de -82.23 que es bueno comparado con el de enlace directo -98.00, en el cual se tuvo una pérdida del 79.3%, es decir, que la mayoría de los paquetes recibidos eran erróneos.

**Tabla 7. Pruebas módulos inalámbricos tres nodos**

PRUEBA	DISTANCIA [m] / CANTIDAD DE OBSTÁCULOS (I)								RSSI [dBm] (-100, -36)*			TASA DE PERDIDA DE PAQUETES** %	
	$N_1 - N_2$	I	$N_2 - N_3$	I	$N_1 - N_3$	I	$N_3 - N_1$	I	$N_1$	$N_2$	$N_3$	$N_1$	$N_3$
11	18.50	2	11.63	5	30.13	7	22.48	3	-81.00	-78.77	-76.00	1.47	1.90
12	22.90	2	19.37	3	42.27	5	40.96	5	-86.67	-58.00	-85.00	13.90	5.00
13	24.11	3	23.67	1	47.78	4	34.57	4	-75.00	-90.33	-82.33	1.73	2.60
14	20.30	2	18.51	4	38.81	6	34.33	8	-98.00	-75.33	-82.23	79.30	0

Relaciona cada prueba con la intensidad de la fuerza de la señal (RSSI) vista por cada nodo, así como la tasa de pérdida de paquetes y distancia entre transmisor y receptor.

\*rango de valores en el que se recibe satisfactoriamente los paquetes de datos, tomado de la hoja de características del Xbee Pro S1. La intensidad es solicitada mediante el comando ATDB, la respuesta se da en hexadecimal en el rango [0x24 - 0x64] (x-1dBm).

\*\*Se obtiene de la diferencia entre 100% y el porcentaje de paquetes correctamente recibidos.

Columnas:  $N_1 - N_2$  es la distancia entre el dispositivo final 1 y el intermedio.  $N_2 - N_3$  Distancia entre el dispositivo intermedio y el final 2.  $N_1 - N_3$  Distancia total que recorre la señal desde el nodo 1 pasando por el intermedio hasta el final 2.  $N_3 - N_1$  Distancia entre el nodo final 3 y el nodo 1 sin pasar por el intermedio (línea recta). En la figura 4 se pueden aclarar la definición de las distancias.

En el plano 2, pruebas realizadas en el tercer piso, se puede ver la ubicación de los nodos para el último caso analizado. La razón principal para obtener la calidad mala en la señal de dos puntos, seguramente es la ubicación del nodo 3, ya que es una sección pequeña con paredes cercanas alrededor, en donde se pueden presentar fenómenos como refracción y reflexión, que al sumarse a la condición de la distancia entre los dos nodos, se presentan errores en los mensajes recibidos.

En el Anexo B se encuentra el código en Arduino utilizado para las pruebas de tres nodos.

### 2.3 ANÁLISIS DE RESULTADOS

Para indicar de una manera más general el desempeño de los módulos en ambientes *indoor*, se tiene unas nuevas tablas, en las que la potencia de transmisión RSSI es representada en porcentaje. La principal razón de representar RSSI de esta manera y no en dBm es porque así se puede observar más claramente la calidad de la transmisión mediante este parámetro, entre mayor sea este número mejor será la calidad de la señal. Para efectos de convertir RSSI a porcentaje se utiliza la siguiente ecuación:

$$RSSI(\%) = \frac{P_{dBm} - P_{min}}{P_{max} - P_{min}} * 100 \quad (9)$$

Donde  $P_{min} = -100$ ,  $P_{max} = -36$  y  $P_{dBm}$  corresponde a los valores consignados en la columna RSSI [dBm] de las tablas de prueba.

En la Tabla 8, se muestra para cada prueba un solo valor de RSSI en porcentaje que califica la calidad de la señal y la tasa de pérdida de paquetes que se obtiene del enlace. La Figura 12 tiene la gráfica comparativa de la calidad de la señal y la tasa de pérdida de paquetes para cada prueba. A partir de la Tabla VIII y la Figura 11 se puede caracterizar el comportamiento de los módulos Xbee Pro S1 ante variaciones de distancia e obstáculos; en dichos ítems se puede observar que estos responden mejor ante las variaciones de distancia que a los obstáculos entre enlaces, sin embargo de acuerdo a lo experimentado en las pruebas los datos fueron enviados correctamente en todos los casos excepto en las pruebas 9 y 10 con 3 obstáculos, por lo tanto esto se deberá tener en cuenta al momento de desplegar la red final.

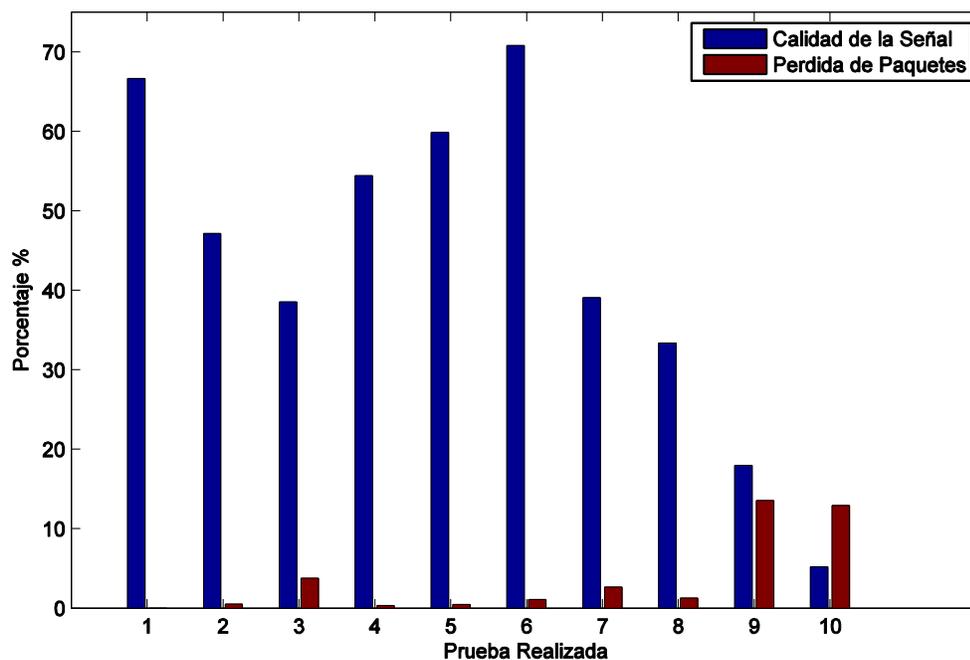
**Tabla 8. Desempeño de los módulos conexión de 2 nodos**

PRUEBA	OBSTÁCULOS	DISTANCIA [m]	RSSI* %	TASA DE PERDIDA DE PAQUETES** %
1	0	17.00	66.66	0.00
2	0	74.00	47.14	0.49
3	0	220.10	38.54	3.74
4	1	15.00	54.43	0.32
5	1	17.00	59.89	0.45
6	1	17.00	70.83	1.08
7	2	21.45	39.06	2.62
8	2	25.40	33.84	1.28
9	3	29.04	17.97	13.54
10	3	34.84	5.21	12.90

Relaciona cada prueba con la intensidad de la fuerza de la señal (RSSI) vista por cada nodo, así como la tasa de pérdida de paquetes y distancia entre transmisor y receptor.

\*Tomado a partir de la columna RSSI [dBm] (tabla v) utilizando la ecuación 1 para determinar la calidad de la señal mediante porcentaje dentro del rango de valores en el cual se recibe satisfactoriamente los paquetes de datos.

\*\*Se obtiene de la diferencia entre el 100% y el porcentaje de paquetes correctamente recibidos.



**Figura 12. Calidad de la señal y pérdida de paquetes para las pruebas realizadas con dos nodos**

La tabla 9, muestra para cada prueba dos valores de distancia/obstáculos, RSSI y tasa de pérdida de paquetes, la primera columna de cada caso corresponde al enlace directo, entre el nodo 1 y 3 sin utilizar el nodo intermedio; la segunda a en enlace con los tres nodos. En la figura 10, se tiene las gráficas de barras comparativas entre el enlace directo y el de tres nodos para cada prueba.

**Tabla 9. Desempeño de los módulos conexión de 3 nodos**

PRUEBA	DISTANCIA [m] / CANTIDAD DE OBSTÁCULOS (I)				RSSI* %		TASA DE PERDIDA DE PAQUETES** %	
	$N_3 - N_1$	I	$N_1 - N_3$	I	$N_1$	$N_3$	$N_1$	$N_3$
	11	22.48	3	30.13	7	29.69	37.50	1.47
12	40.96	5	42.27	5	20.83	23.44	13.90	5.00
13	34.57	4	47.78	4	39.06	27.61	1.73	2.60
14	34.33	8	38.81	6	3.13	27.77	79.30	0

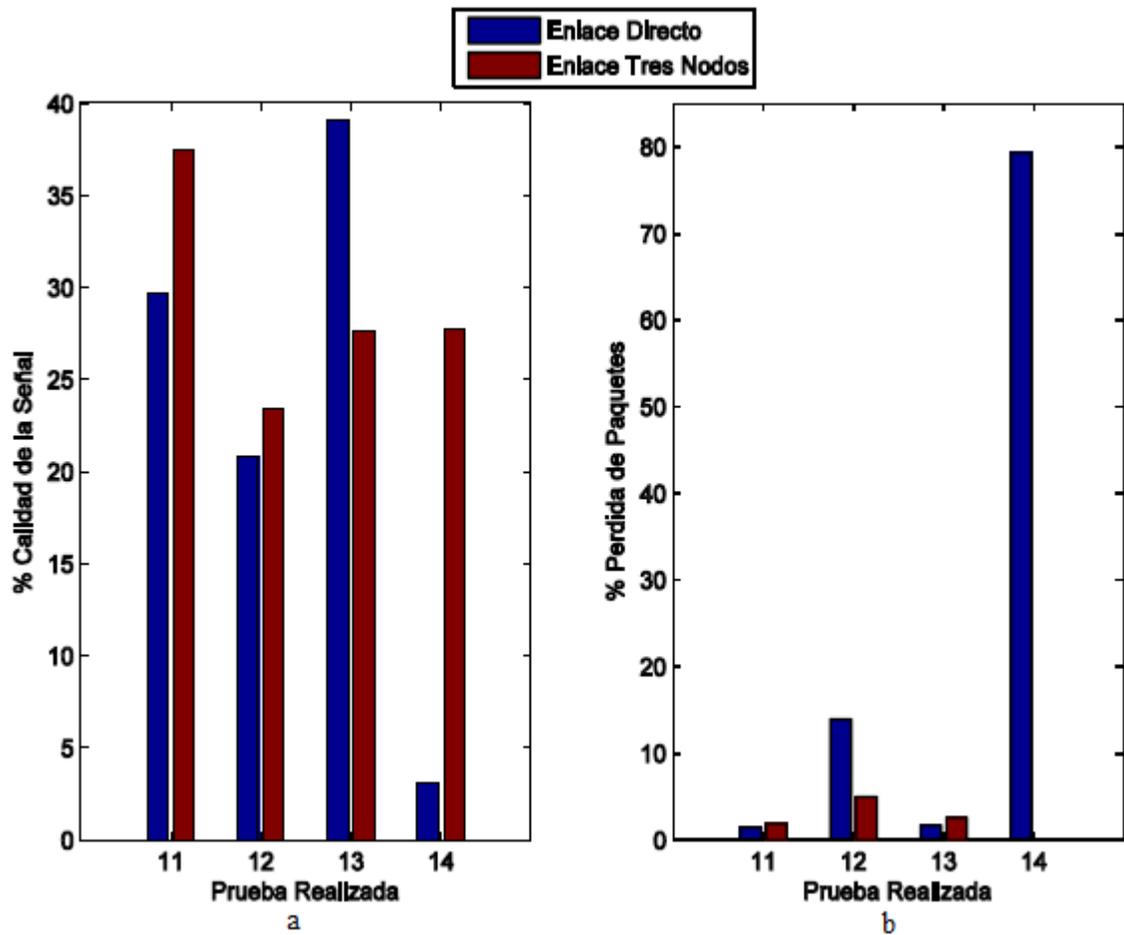
Relaciona cada prueba con la intensidad de la fuerza de la señal (RSSI), así como la tasa de pérdida de paquetes y distancia entre transmisor y receptor.

\* Tomado a partir de la columna RSSI [dBm] (tabla VI) utilizando la ecuación 1 para determinar la calidad de la señal mediante porcentaje dentro del rango de valores en el cual se recibe satisfactoriamente los paquetes de datos.

\*\*Se obtiene de la diferencia entre 100% y el porcentaje de paquetes correctamente recibidos.

En la Figura 13, se puede comparar las características del enlace de 2 nodos con el de 3, donde las ubicaciones y los dispositivos de 1 y 3 son los mismos para ambos casos, en la pérdida de paquetes se nota claramente la mejoría con el enlace de tres nodos, ocurren menos errores a un mayor alcance que con el enlace directo (prueba 14) y la calidad de la señal es más alta.

En términos generales, se puede concluir que en ciertos casos, cuando se requiere cobertura más exigente, mayor alcance y pérdida de paquetes baja se puede usar enlaces de tres nodos; el inconveniente con el caso del Xbee pro s1 es que requiere de un controlador externo para comportarse como enrutador a diferencia de otros módulos, por ejemplo, el Xbee digimesh 2.4 GHz que puede configurarse como enrutador, optimiza la ruta para llevar el mensaje de un punto a otro, especial para trabajar con topología en malla. [34]



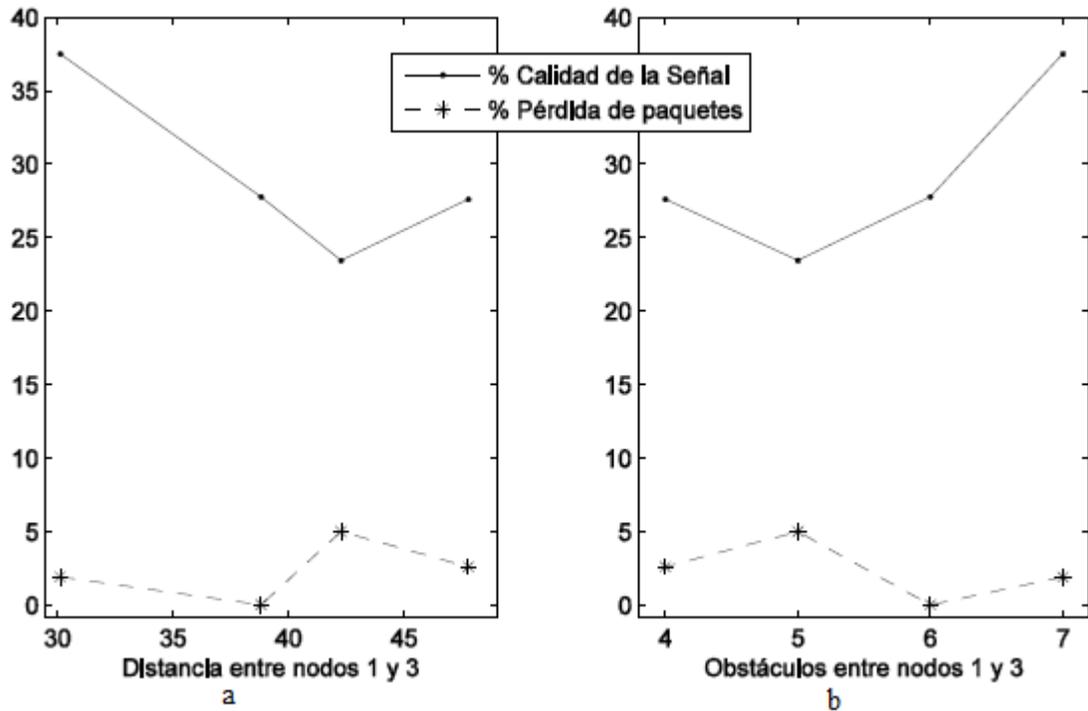
**Figura 13. a. Comparación del porcentaje de calidad de la señal para las pruebas realizadas con el enlace directo y enlace de tres nodos, b. Comparación del porcentaje de pérdida de paquetes para las pruebas realizadas con el enlace directo y enlace de tres nodos.**

En la primera gráfica de la Figura 11, se tiene los porcentajes para pérdida de paquetes y calidad de la señal a partir de la distancia entre los nodos; el peor caso se presenta en la prueba 13, con una distancia de 42.27 m y 5 obstáculos. Aunque no es la mayor distancia en las pruebas realizadas, esta es una de las más críticas porque presenta obstáculos metálicos, como muros estructurales y puertas, además los nodos se encuentran en ambientes cerrados, aulas de clase, en el Anexo A plano 3 se tiene la disposición de los módulos para cada prueba.

En cuanto a la gráfica de obstáculos mostrada en la Figura 14, el mejor de los casos ocurre cuando hay más obstáculos (prueba 11). Como se puede ver en la Tabla 9, esto es compensado, con una distancia entre módulos menor que en las

otras pruebas; obteniendo así un porcentaje de pérdida de paquetes bajo y la calidad de la señal es mejor comparada con las otras pruebas realizadas.

Por tanto, para la ubicación de los nodos en una red debe tenerse en cuenta tanto la distancia como la cantidad y clase de las obstáculos; así sí se presentan muros estructurales, por ejemplo, se puede compensar la pérdida que esto generaría acercando un poco más los nodos que se desea comunicar y sus enrutadores.



**Figura 14. Porcentaje calidad de la señal y porcentaje de pérdida de paquetes para diferentes distancias entre los nodos 1 y 3, prueba tres nodos, b. Porcentaje calidad de la señal y porcentaje de pérdida de paquetes vs cantidad de obstáculos entre los nodos 1 y 3, prueba tres nodos.**

### 3. IMPLEMENTACION Y CARACTERIZACION DE UNA RED

En esta sección se indican los pasos y componentes utilizados para la implementación de la WSN; además se presentan las pruebas realizadas a la red y se explica la programación utilizada para cada una de ellas; esto con el propósito de lograr la caracterización de la red de sensores inalámbricos para el control de iluminación a partir de los valores de distancia, obstáculos, pérdida de paquetes y calidad de la señal tomados adquiridos durante las pruebas.

En este proyecto, para la implementación de la WSN, primero se realiza un análisis de las topologías soportadas tanto por el estándar IEEE 802.15.4 como por los módulos inalámbricos disponibles.

#### 3.1 ANÁLISIS DE TOPOLOGÍAS

**3.1.1 Arquitectura Básica de una Red XBee.** Los módulos Xbee son versátiles a la hora de establecer diversas topologías de red y, dependiendo la serie que se elija pueden crearse distintos tipos de redes. Una red Xbee la forman básicamente 3 tipos de elementos.

- Un único dispositivo Coordinador.
- Dispositivos Enrutadores.
- Dispositivos finales (Nodos sensores).

En la Figura 14 tipos de topologías, también se puede aclarar cuáles son los tres tipos de elementos que hacen parte de una red Xbee.

##### *a. El coordinador*

Es el nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el Coordinador puede formar una red, permitiendo unirse a él a dispositivos Enrutadores y Dispositivos finales. Una vez formada la red, el Coordinador hace las funciones de Enrutador, esto es, participar en el enrutado de paquetes y ser origen y/o destinatario de información.

### *b. Los Enrutadores*

Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta, para enrutar un paquete de información. Lógicamente un enrutador debe unirse a una red *Xbee* antes de poder actuar como Enrutador retransmitiendo paquetes de otros Enrutadores o de Dispositivos finales.

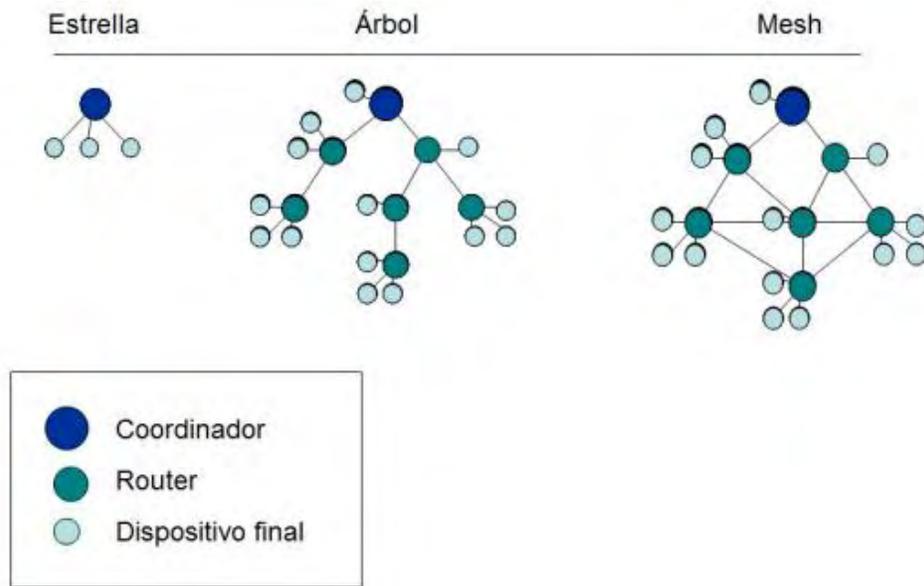
### *c. Dispositivos Finales*

Los dispositivos finales no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador o un Enrutador, es decir, no puede enviar información directamente a otro dispositivo final. Normalmente estos equipos van alimentados a baterías. El consumo es menor al no tener que realizar funciones de enrutamiento.

Para configurar los módulos como algún tipo de elemento anteriormente mencionado, es necesario conectar cada módulo al computador y modificar los parámetros para activar cada función. Algunos módulos, como el *Xbee pro s1* no presentan la función de enrutador, por tanto, para que este pueda comportarse así es necesario acompañarlo de un controlador que tenga puerto serial, (microcontrolador, arduino, fpga, entre otros), para y redireccionar la información. En la tabla XII se indica como pueden ser configurados los módulos como dispositivos finales y coordinador para una misma red.

**3.1.2 Topologías.** Los dispositivos finales no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador o un Enrutador, es decir, no puede enviar información directamente a otro dispositivo final. Normalmente estos equipos van alimentados a baterías. El consumo es menor al no tener que realizar funciones de enrutamiento.

Para configurar los módulos como algún tipo de elemento anteriormente mencionado, es necesario conectar cada módulo al computador y modificar los parámetros para activar cada función. Algunos módulos, como el *Xbee pro s1* no presentan la función de enrutador, por tanto, para que este pueda comportarse así es necesario acompañarlo de un controlador que tenga puerto serial, (microcontrolador, arduino, fpga, entre otros), para y redireccionar la información. En la tabla 12, se indica como pueden ser configurados los módulos como dispositivos finales y coordinador para una misma red.



**Figura 15. Tipos de topologías soportados por el estándar IEEE 802.15.4 [15]**

*a. Topología en estrella*

A la red en estrella se le puede tipificar como la topología más sencilla de todas. Esta topología tiene un coordinador conectado a una red con uno o más dispositivos finales o sensores. Cabe recordar que un dispositivo final se comunica solamente con un enrutador o coordinador, y nunca directamente con otro dispositivo que sea dispositivo final. La ventaja de esta topología es que se puede aislar una terminal defectuosa sin afectar la comunicación en los demás dispositivos de la red, sin embargo si el nodo central falla la red deja de funcionar. Los módulos que operan bajo estándar 802.15.4 que soporta la topología estrella. [15]

*b. Topología en árbol*

La topología en árbol (cluster tree) es una red mucho más compleja que la red en estrella y es similar a la usada para la conexión en redes de PC, en la cual los dispositivos se bifurcan igual que las ramas de un árbol. Posee un nodo raíz que hace las veces de coordinador y enrutadores que se conectan a los dispositivos finales. Tiene la desventaja de que si el nodo raíz falla, toda la red falla o si falla un enrutador una o más ramas pueden quedar desconectadas de la red.

*c. Topología en malla*

La topología de malla a diferencia de otras (como la topología en árbol y la topología en estrella), no requiere de un servidor o nodo central, así, un error en

un nodo, sea importante o no, no implica la caída de toda la red y permite que si en un momento dado un nodo del camino falla y cae, pueda seguir la comunicación entre todos los demás adicionalmente y con esto se reduce el mantenimiento. Los nodos pueden seguir comunicándose debido a que el coordinador es el encargado de rehacer y gestionar todos los caminos.

Las redes de malla son auto routeables. La red puede funcionar, incluso cuando un nodo desaparece o la conexión falla, ya que el resto de los nodos evitan el paso por ese punto. En consecuencia, la red malla, se transforma en una red muy confiable [24].

Los tres tipos de topologías anteriormente mencionados se plantearon en el plano del cuarto piso del edificio tecnológico de la Udenar para mostrar la ubicación de cada uno de los nodos.

En el Anexo C se encuentran tres planos. El plano 1 corresponde a la topología tipo estrella que, tiene un nodo central y ocho finales (sensor/actuador) ubicados en diferentes secciones del piso. El siguiente plano muestra la ubicación de los nodos para la topología tipo árbol, que tiene un nodo central, tres que cumplen la función de enrutador/final y cuatro que son solamente nodos sensor/actuador. En el plano 3 se tiene la ubicación de los nodos para la topología tipo malla, en este caso, cada uno de los nodos puede cumplir cualquier función, ya sea coordinador, enrutador o dispositivo final.

Para decidir que topología de red se va a implementar es necesario realizar pruebas con los módulos disponibles, los Xbee pro s1 permiten redes peer to peer (par a par o punto a punto) o de topología en estrella, puede verse en la Figura 14; para poder implementar otras topologías se requiere módulos configurados como enrutadores, lo cual no es posible con Xbee pro s1. Sin embargo, si es posible formar un módulo enrutador con un Xbee s1 y un controlador, el inconveniente se encuentra en los costos que esto conllevaría además de la programación para lograr una óptima ruta para la recepción del mensaje.

**3.1.3 Topologías bajo el protocolo IEEE 802.15.4.** Dependiendo de los requisitos de la aplicación, una red inalámbrica que trabaje bajo el protocolo IEEE 802.15.4 opera en cualquiera de las dos topologías: topología en estrella o topología punto a punto. En la topología en estrella, se establece la comunicación entre los dispositivos finales y un controlador central, llamado el coordinador PAN (Personal Área Network). El coordinador es el controlador principal de la red. Un dispositivo final suele ser usado como el punto de inicio o punto de terminación de la red de comunicaciones. Un coordinador de PAN también puede tener una aplicación específica y puede ser utilizado para iniciar, terminar, o enrutar la comunicación alrededor de la red. Todos los dispositivos que operan en una misma red con cualquiera topología tienen direcciones únicas. Un dispositivo final

utilizará o bien la dirección extendida para la comunicación directa en la red o la dirección corta que fue asignada por el coordinador cuando se asocia el dispositivo. El coordinador de la PAN será a menudo alimentado por red eléctrica, mientras que los dispositivos finales son en general alimentados con pilas. [15].

Para el presente proyecto se dispone de los módulos Xbee Pro serie1. La Serie 1 viene con firmware 802.15.4 que permite redes peer to peer (par a par o punto a punto) o de topología en estrella; esta última se muestra en la Figura 14. Este firmware ofrece ADC (conversión analógica-digital), entradas y salidas digitales.

### **Metodología de diseño:**

Como uno de los principales resultados de este trabajo, se propone una metodología de diseño de redes inalámbricas, que se plantea de manera muy práctica y sencilla; en ella se definen las características de la red que se va a implementar en este proyecto y los pasos que se siguen y podrían aplicarse a otro tipo de trabajos con redes inalámbricas de sensores. Para el diseño e implementación de una WSN se debe determinar algunos requerimientos básicos de manera que el resultado final cumpla con la funcionalidad de la aplicación, el presupuesto y el consumo de potencia. A continuación se describe un listado de aspectos importantes al momento de fijar dichos requerimientos:

- **Aplicación y entorno:** Inicialmente se determina la aplicación específica de la red y el entorno en el cual se implementara la misma, por ejemplo Control de Iluminación (*Outdoor/Indoor*), Control de un sistema de riego (*Outdoor/Indoor*), domótica (*Indoor*); adicionalmente se requiere saber si el sistema será autónomo o si se requiere una interfaz hombre-máquina (HMI). Para el caso de este proyecto la aplicación consiste en el control PWM de iluminación distribuido en un entorno *Indoor*.
- **Tipo de sensores y actuadores:** Se define qué tipo de sensores se van a usar para determinar el número y tipo de entradas que se requieren, por ejemplo si son uno o varios sensores y si son de tipo analógico o digital; además que tipo de salidas se necesitan, por ejemplo salidas analógicas, PWM y/o digitales. Para este trabajo los sensores son de tipo analógico y se necesita que además también sean actuadores por lo que necesariamente se necesita que el modulo a elegir tenga al menos una entrada analógica y una salida PWM o analógica.
- **Plano o bosquejo del entorno:** obtener un plano o bosquejo del espacio en el que se desea implementar la red es importante, con el fin de dimensionar cada sección donde se pondrán los nodos, la distancia entre ellos y la ausencia o presencia de obstáculos; ya que esto influye en la selección de los módulos inalámbricos y la topología de la red; adicionalmente se suele requerir realizar pruebas de conectividad en el entorno. Se pudo adquirir los planos del edificio tecnológico de la Universidad de Nariño gracias al Fondo de construcciones de

la universidad, gracias a ello se pudo facilitar la disposición de elementos y la cantidad y tipo de obstáculos que se puedan presentar.

- Definir la topología de red: dependiendo de los anteriores aspectos se puede escoger una topología; a partir de las características de los dispositivos disponibles y sobre todo la aplicación se determina cual topología es posible o se requiere implementar; se debe tener en cuenta que no todos los módulos inalámbricos permiten configurarse para todas las topologías.

Para seleccionar el tipo de topología se debe tener en cuenta las siguientes características:

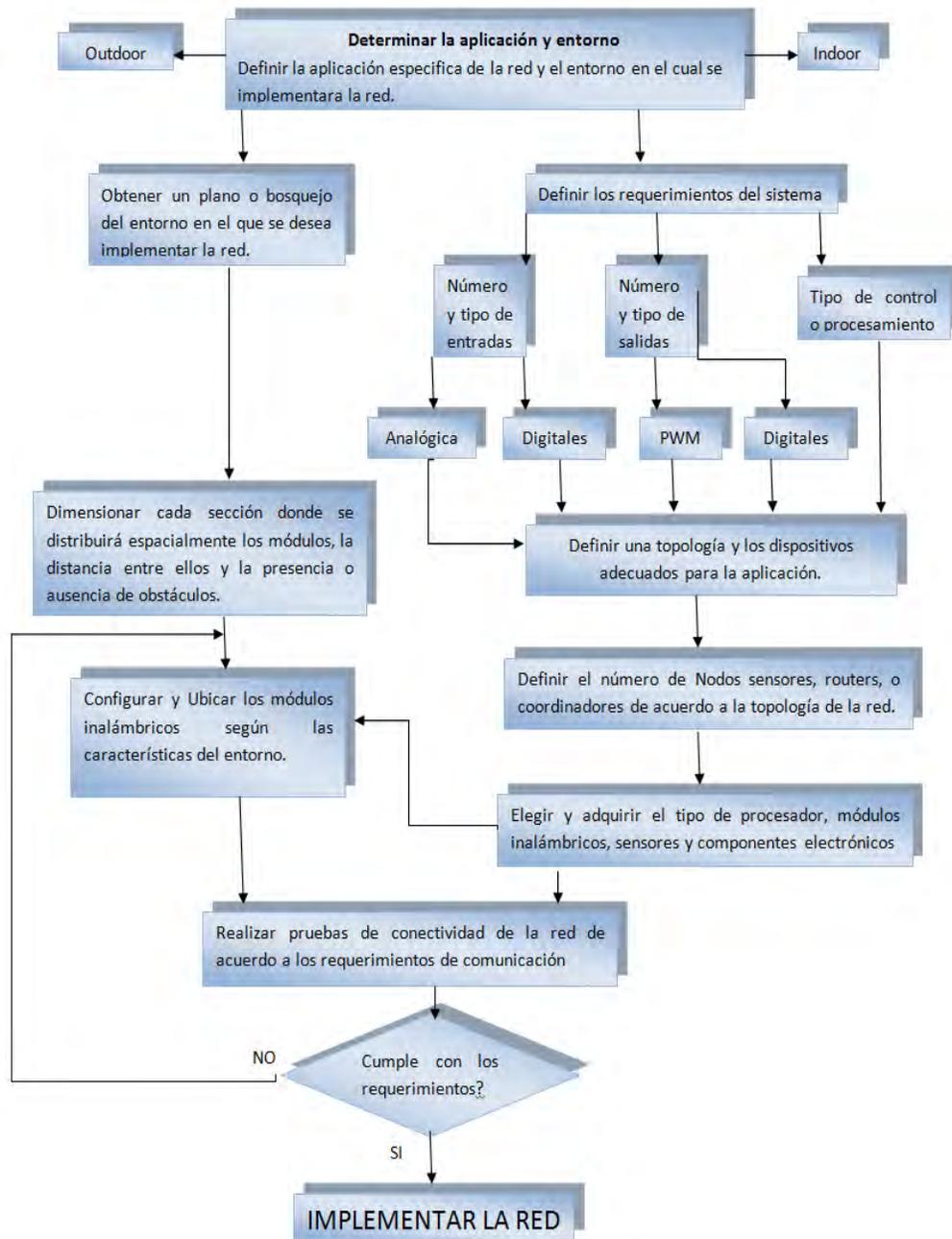
**Malla:** Cuando se utilice un método de optimización y un control no centralizado, escogiendo módulos inalámbricos adecuados, que permitan ser configurados como nodos finales y enrutadores; esta topología no requiere nodo coordinador. Esta topología se aplica cuando se requiere el funcionamiento de la red a pesar del fallo de un nodo cualquiera.

**Estrella:** Cuando se requiere un control centralizado; el funcionamiento de la red no se ve afectado en gran medida si uno de los nodos finales falla, sin embargo esta topología requiere un nodo coordinador que tiene a su cargo el control de la red y envía la información a cada uno de los nodos finales y si este se ve afectado o se daña, la red deja de funcionar.

**Árbol:** Cuando es necesario un control centralizado, pero a diferencia de la topología tipo estrella, se necesita enrutar la información, por tanto puede soportar un método de optimización. Al requerir un nodo coordinador, enrutadores y nodos finales; si falla un nodo diferente a uno final, la red se puede ver afectada en gran medida o puede dejar de funcionar; adicionalmente los dispositivos escogidos deben poder configurarse en los tipos de dispositivos lógicos.

Una vez conocidos todos los parámetros y requerimientos anteriormente mencionados se procede a la selección y/o diseño de dispositivos de control, alimentación y comunicación. En este punto es importante que los elementos elegidos cumplan con requerimientos de alcance, topología y configuración necesarios.

Posteriormente al diseño se realizan pruebas con los dispositivos que se implementaran en la red para verificar el funcionamiento y realizar la corrección de posibles errores; si las pruebas se realizan con éxito finalmente se implementa la red con todas sus partes y componentes; en caso contrario se debe redefinir la disposición de los elementos de la red y realizar pruebas nuevamente. En la Figura 16 se presenta la metodología de diseño resumida mediante un diagrama de flujo.



**Figura 16. Diagrama de la Metodología de diseño de una WSN.**

## 4. IMPLEMENTACION

A continuación se presenta la implementación y caracterización de la red final para el control PWM de iluminación en un ambiente *indoor*. La red se configura en topología estrella, con ocho dispositivos finales o nodos sensor-actuador y un nodo central, éste último se encarga de direccionar la información y configurar los nodos remotamente. Los módulos *Xbee* pro S1 por si solos, sin un coordinador, no son capaces de realizar tareas como direccionamiento y configuración. Debido a esto, para implementar otra topología de red es necesario el uso de más dispositivos de coordinación.

Las pruebas de dicha red se plantearon para el piso cuarto del edificio tecnológico de la Universidad de Nariño, pero se realizaron en el piso tercero ya que por el momento en el cuarto no se cuenta con la disponibilidad de instalaciones eléctricas en su totalidad. La distribución espacial de los dos pisos es la misma por tanto no hay inconveniente en cuanto a obstáculos o distancias entre nodos de la misma red.

En el Anexo D se tiene el plano con la ubicación de los nodos para la red final, y en la Tabla 10, se relacionan los obstáculos existentes y la distancia entre cada nodo final y el nodo central.

**Tabla 10. Nodos prueba final**

NODO	CANTIDAD DE OBSTÁCULOS	DISTANCIA [m]
1	1	6.73
2	0	18.83
3	2	24.04
4	2	13.91
5	1	9.80
6	2	18.81
7	0	10.47
8	3	25.72

Tabla que relaciona los obstáculos existentes y la distancia entre cada nodo final y el nodo central.

Se realizaron dos programaciones diferentes para probar el funcionamiento de la red, la primera por medio de envío de mensajes entre computadores mediante X-CTU direccionando la información, y la segunda usando unas tarjetas para prueba sin ayuda de computadores. Estas se explican con más detalle a continuación.

## 4.1 PRUEBA 1

La red se encuentra distribuida espacialmente como se muestra en el plano del Anexo C, en cada uno de los nodos finales es posible ubicar un computador con un *Xbee*. Mediante el software X-CTU se realiza el envío de cualquier mensaje, colocando previamente el identificador del nodo al que se quiere transmitir. En el nodo receptor aparecerá entonces el mensaje seguido de un símbolo correspondiente, en hexadecimal, a la potencia recibida en dBm. En la Tabla 11, se muestra ejemplos de la transmisión y recepción de mensajes mediante X-CTU.

**Tabla 11. Transmisión/ recepción mensajes prueba con computadores**

NODO 1	NODO 2	NODO 3	NODO 4
2MENSAJE MENSAJE NUEVO8	MENSAJE\$		1MENSAJE NUEVO
		4MENSAJE DE PRUEBA 2	MENSAJE DE PRUEBA 2:
4ENVIO A NODO 4	3ULTIMO	ULTIMO]	ENVIÓ A NODO 4\$
EJEMPLO DE MENSAJES TRANSMITIDOS/RECIBIDOS CON LA RED FINAL EN LA PRUEBA CON COMPUTADORES.			

**4.1.1 Configuración módulos.** La configuración de los módulos para esta prueba se indica en la tabla XII, el canal y el identificador de la red deben ser iguales en todos los *Xbee* para lograr la comunicación. El *Xbee* del nodo central se configura como coordinador y en modo API, esto es necesario para poder recibir la información por medio de tramas.

**Tabla 12. Configuración módulos prueba 1**

PARAMETRO	NODO								
	CENTRAL	1	2	3	4	5	6	7	8
MY	BBBB	DEDE	AAAA	ABAB	ABCD	BABA	FEFE	ACDE	CFCF
DL	0	BBBB							
AP	1*	0	0	0	0	0	0	0	0
CE	1**	0	0	0	0	0	0	0	0

Configuración de los módulos *Xbee* para la primera prueba de la red.

\*El nodo central debe configurarse en modo API (AP=1).

\*\*El nodo central se asigna como coordinador (CE=1).

**4.1.2 Parámetros de las tramas.** En el modo API se tienen muchos tipos de tramas, para este caso se empleó las tramas de tipo *0x81, Rx (Receive) Packet 16 bit address* para dispositivos que trabajan en el protocolo 802.15.4, que son los *Xbee pro s1*. Los parámetros de dicha trama se muestran en la Tabla XIII. En la

Tabla 13, se describe las tramas de tipo *0x00 Tx Request 64-bit address* con las que se envía el mensaje a cada dispositivo final.

El delimitador de inicio (7E) indica que una trama ha sido recibida, la longitud cuenta el número de bytes desde tipo de trama hasta el final sin contar el *checksum*; el tipo de trama, como se dijo anteriormente, dice que estructura tiene la trama. La dirección MY del Xbee que transmite el dato se puede ver en “dirección de origen de 16 bits”. La potencia de la señal transmitida se tiene en hexadecimal en el parámetro RSSI; finalmente se indica el *checksum* el cual se calcula según la Ecuación 10.

$$\begin{aligned}
 &checksum \\
 &= 0xFF \\
 &- (suma\ de\ bytes\ despues\ de\ la\ longitud\ hasta\ antes\ del\ checksum) \quad (10)
 \end{aligned}$$

**Tabla 13. Parámetros trama tipo 0x81 Rx packet 16 bit address**

PARAMETROS DE LA TRAMA (FRAME PARAMETERS)	VALOR (FIELD VALUE)*	VALOR (FIELD VALUE)**	POSICION
Delimitador de Inicio (Start delimiter)	7E	7E	
Longitud (Length)	00 06	00 06	0
Tipo de trama (Frame type)	81	81	1
Dirección de origen de 16 bits DL (16-bits source address DL)	AB AB	CF CF	2-3
RSSI	24	2 <sup>a</sup>	4
Opciones (Options)	00	00	5
Dato RF (RF data)	6D	41	6
Suma de prueba (Checksum)***	97	75	7
<b>Trama</b>	<b>7E 00 06 81 AB</b> <b>AB 24 00 6D 97</b>	<b>7E 00 06 81 CF</b> <b>CF 2A 00 41 75</b>	

\*Trama vista por el coordinador cuando del dispositivo 3 (MY=ABAB) se escribe “m”, potencia 24 (-36dBm).

\*\*Trama vista por el coordinadora cuando del dispositivo 8 (MY=CFCF) se escribe “A”. Potencia 2A (-42dBm).

\*\*\*Suma de bytes de la trama, desde “tipo de trama” hasta “dato RF”, sirve para corroborar que la trama se envía correctamente; en caso de que el checksum esté mal la trama no se recibe.

**Tabla 14. Parámetros trama tipo 0x00 tx request 64 bit address**

<b>PARAMETROS DE LA TRAMA (FRAME PARAMETERS)</b>	<b>VALOR (FIELD VALUE)*</b>	<b>VALOR (FIELD VALUE)**</b>
Delimitador de Inicio (Start delimiter)	7E	7E
Longitud (Length)	00 13	00 19
Tipo de trama (Frame type)	00	00
Identificador de trama (Frame ID)	01	01
Dirección de destino de 64 bits DL (64-bits source address DL)	00 00 00 00 00 00 AA AA	00 00 00 00 00 00 DE DE
Opciones (Options)	04	04
Dato RF (RF data)	6D 65 6C 73 61 6A 65 24	6D 65 6C 73 61 6A 65 20 6E 75 65 76 6F 38
(Checksum)	A3	DA
<b>Trama</b>	<b>7E 00 13 00 01 00 00 00 00 00 00 AA AA 04 6D 65 6C 73 61 6A 65 24 A3</b>	<b>7E 00 19 00 01 00 00 00 00 00 DE DE 04 6D 65 6C 73 61 6A 65 20 6E 75 65 76 6F 38 DA</b>

\*Trama creada por el coordinador para enviarle al nodo 2 (MY AAAA) “mensaje” seguido de “\$” que corresponde a la potencia de la señal recibida -36dBm.  
\*\*Trama creada por el coordinador para enviarles al nodo 1 (MY DEDE) “mensaje nuevo” seguido de “8” que corresponde a la potencia de la señal recibida -56dBm.

**4.1.3 Programación.** A continuación se tiene una descripción general de la programación implementada en el Arduino del nodo central:

- Definición de variables
- Inicializar LCD, puerto serial.
- Leer puerto serial y almacenar por posiciones las tramas a partir del byte 7E.
- Comparar las posiciones 3 y 4 de la trama con las direcciones que identifican cada nodo.
- Guardar los bytes correspondientes al mensaje recibido.
- Con el primer byte del mensaje seleccionar a que nodo se enviará la información guardada anteriormente.
- Cuando lea doble “enter” envía el mensaje y la potencia de transmisión del nodo de origen al nodo destino anteriormente seleccionado.

Para el direccionamiento del mensaje se utiliza una función que asocia la dirección de origen MY con un número asignado a cada nodo, correspondiente a los datos de la Tabla 17.

El envío del mensaje se realiza con otra función que construye y envía las tramas con la información de dirección, mensaje y potencia de transmisión del nodo de origen. Las tramas utilizadas son las tipo 0x00 Tx Request 64-bit address mostradas en la Tabla XIV.

El código realizado en Arduino con la programación anteriormente descrita se encuentra en el Anexo E. ç

**4.1.4 Prueba.** Para la primera prueba de la red implementada se utilizaron 4 computadores, cada uno de ellos con un módulo Xbee y un USB xplorer para la conexión y se ubicaron en sus nodos correspondientes según el plano del Anexo D.

Los nodos 8, 4, 3 y 6 son utilizados para obtener los datos para caracterizar la red. Para la prueba se envía entre cada enlace un total de 5 mensajes, de los cuales se guarda la potencia para posteriormente analizarla en la tabla que caracteriza la red. Para la pérdida de paquetes se envía mensajes consecutivos entre los nodos escogidos, después se realiza la comparación entre el mensaje enviado y el recibido para calcular el porcentaje de error.

**4.1.5 Resultados.** En la Tabla XV se encuentran los datos de la prueba 1 realizada a la red. En el enlace entre los nodos 4 y 8 se presenta un caso crítico de pérdida de paquetes con un porcentaje de error alto, esto es debido a que en este enlace se tiene la distancia más larga con 3 obstáculos que atraviesan un sector con estructura metálica y en adición, una trama de escritura del mensaje grande en donde se pierde información. En la siguiente prueba (enlace 8-4) no se pierden paquetes porque se tienen múltiples tramas de lectura, una por cada carácter. Esta situación posteriormente puede ocasionar fallas en la respuesta del actuador.

Los enlaces entre los nodos 3 y 6 presentan un comportamiento similar al anterior, debido a que la comunicación del central y el nodo final 3 debe pasar (simulando una línea recta) por algunas estructuras metálicas que generan pequeñas pérdidas de las tramas de recepción del dato.

Por último, el enlace entre nodos 4 y 6 presenta las mejores respuestas debido a su distancia corta y a obstáculos no estructurales.

**Tabla 15. Características de la red prueba 1**

ENLACE	CANTIDAD DE OBSTÁCULOS	DISTANCIA TOTAL [m]	RSSI* %	TASA DE PERDIDA DE PAQUETES** %
4 – 8	5	39.63	38.44	43.57
8 – 4	5	39.63	32.19	3.33
3 – 6	4	42.85	26.56	10.71
6 – 3	4	42.85	36.56	0.67
4 – 6	4	32.72	32.50	0.00
6 – 4	4	32.72	57.50	0.00

Relaciona cada prueba con la intensidad de la fuerza de la señal (RSSI) vista por cada nodo, así como la tasa de pérdida de paquetes y distancia entre transmisor y receptor.

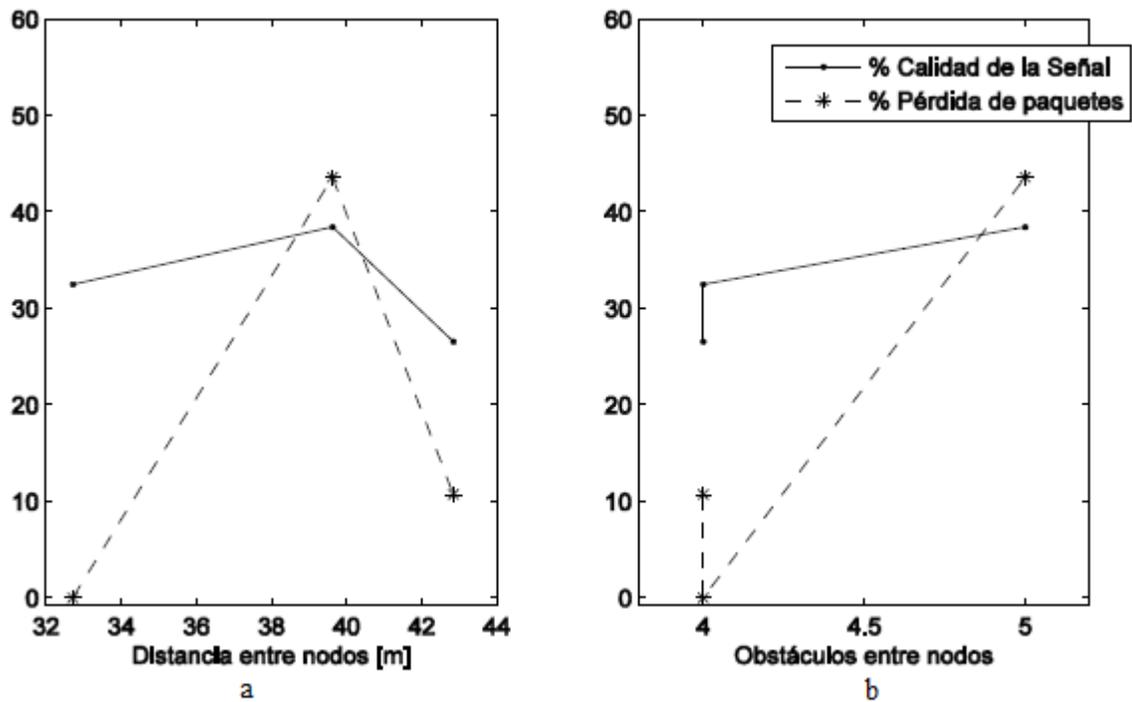
\*Utilizando la ecuación 1 para determinar la calidad de la señal mediante porcentaje dentro del rango de valores en el cual se recibe satisfactoriamente los paquetes de datos.

\*\*Se obtiene de la diferencia entre el 100% y el porcentaje de paquetes correctamente recibidos.

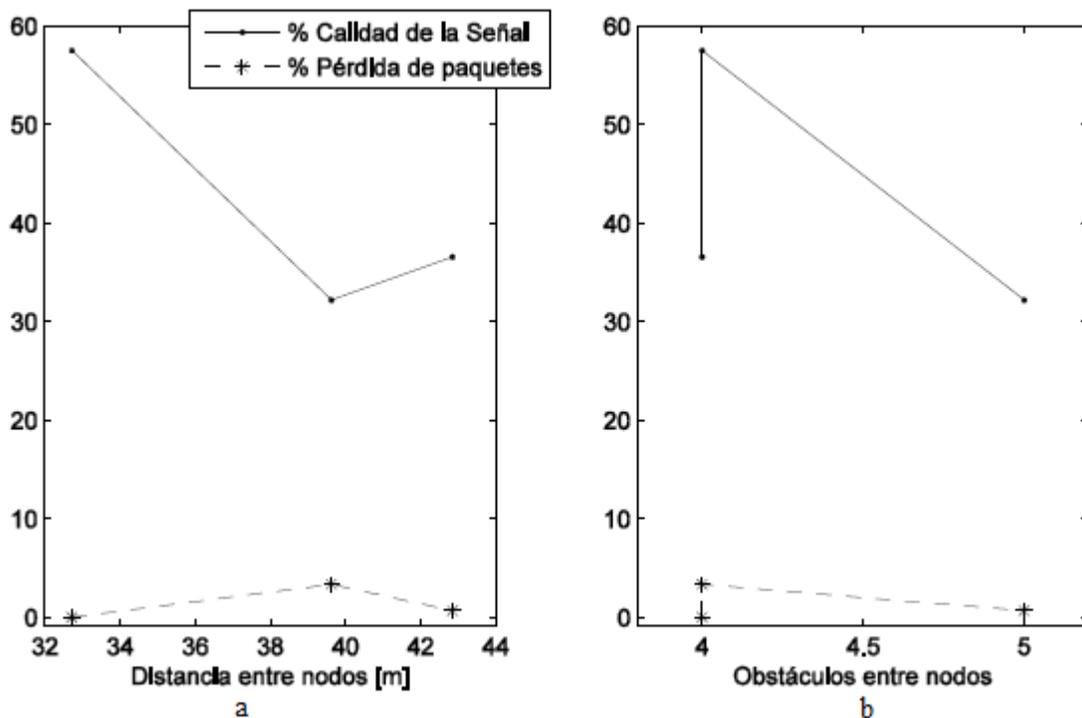
En la Figura 17 se muestra las gráficas para las características de calidad de la señal y pérdida de paquetes en los enlaces 4 - 8, 3 - 6 y 4 - 6, dependiente de la distancia y número de obstáculos entre los nodos mencionados. Se puede notar que a la distancia máxima entre nodos no se obtiene un porcentaje alto de pérdida de paquetes comparado con el máximo registrado que corresponde al enlace 4 – 8 que además, presenta 5 obstáculos. Con ayuda de la segunda gráfica de la Figura 16 se mira el caso con porcentaje de pérdida de paquetes alto y una calidad de la señal buena comparada con las registradas en los otros dos casos; para 4 obstáculos se tienen dos valores registrados uno por cada enlace con esta característica, la pérdida de paquetes es mínima cuando la distancia entre los nodos es la menor (enlace 4 - 6, distancia 32.72 m).

Para los enlaces 8 - 4, 6 - 3 y 4 - 6 se tienen gráficas de la Figura 18, que indican porcentajes de calidad de la señal y pérdida de paquetes según el número de obstáculos y la distancia entre los nodos que intervienen en esta prueba.

Por tanto, se demuestra que para la calidad de la señal y la recepción de mensajes correctamente debe tenerse en cuenta, tanto la distancia entre los puntos de enlace, la cantidad y tipo de obstáculos.



**Figura 17. a. Porcentaje calidad de la señal y porcentaje de pérdida de paquetes para las distancia entre los enlaces 4 a 8, 3 a 6 y 4 a 6, b. Porcentaje calidad de la señal y porcentaje de pérdida de paquetes para los obstáculos entre los enlaces 4 a 8, 3 a 6 y 4 a 6.**



**Figura 18. Porcentaje calidad de la señal y porcentaje de pérdida de paquetes para las distancia entre los enlaces 8 a 4, 6 a 3 y 6 a 4, b. Porcentaje calidad de la señal y porcentaje de pérdida de paquetes para los obstáculos entre los enlaces 4 a 8 a 4, 6 a 3 y 6 a 4.**

## 4.2 DISEÑO DE NODOS CENTRAL Y SENSORES

En la sección anterior se definieron los parámetros necesarios para el diseño de una red, teniendo en cuenta esto se diseña el nodo coordinador y 8 nodos sensores para la topología soportada por los módulos y aplicada al control de iluminación. Además se definió la ubicación de cada nodo en el plano del piso elegido y se realizó la programación para efectuar las pruebas pertinentes de la red.

El siguiente paso consiste en diseñar unas tarjetas que permitirán que la red emule el control PWM de iluminación de manera inalámbrica a partir de elementos sencillos, pero que permita modificarse para ser adaptada a otras aplicaciones. Adicionalmente dichas tarjetas permiten que la red sea autónoma y solo requiera de alimentación para poder funcionar y se fabrican de manera que sean didácticas y puedan ser reutilizadas por estudiantes para realizar pruebas y prácticas de laboratorio ya que se ensamblaron con headers que permiten que las salidas y entradas de los módulos y del Arduino queden accesibles para conectar otros dispositivos, además de poder usar la tarjeta del nodo central como un shield

LCD-*Xbee* para el Arduino Mega. A continuación se describe las partes de la red más detalladamente.

**4.2.1 Nodo Central.** El nodo central o coordinador está compuesto por un Arduino Mega 2560, un módulo *Xbee* Pro S1, un convertidor lógico de nivel, una pantalla LCD 16x2, una base adaptadora *Xbee* a DIP, headers, un potenciómetro y un LED indicador. En el Anexo G se encuentra el diagrama esquemático del nodo coordinador. En la Figura 19 se muestra la tarjeta del nodo central implementada. A continuación se describe las partes principales y sus funciones:

***Arduino Mega 2560:***

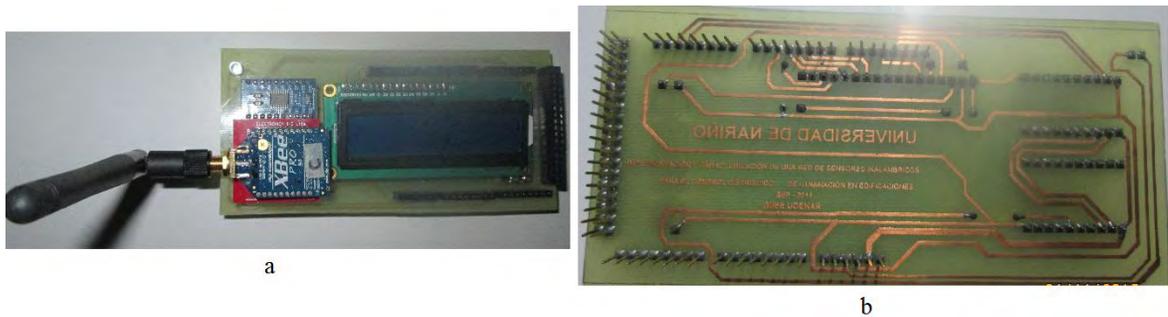
Interpreta y procesa la información recibida por el puerto serial 1 y conjuntamente con el módulo *Xbee* se encarga del direccionamiento de la información hacia los nodos sensores. Adicionalmente permite visualizar alguna información en la pantalla LCD usando la librería *LyquidCristal.h*. Es posible incluir un algoritmo de control de iluminación más complejo en el código de programación, ya que el control que se realizó fue muy sencillo y consiste básicamente en la variación de la luminosidad de una cadena de LEDs mediante una señal PWM y que depende del nivel de voltaje de un sensor.

***Xbee Pro S1:***

Se comunica mediante el puerto serial 1 con el Arduino y se encarga de recibir las señales de cada nodo sensor en forma de tramas y enviar otros tipos de tramas a los mismos para activar cada actuador (cadena de LEDs) ubicados en los diferentes nodos.

***Convertidor Lógico de Nivel:***

Realiza conversiones de voltaje de 3.3 V a 5 V y viceversa, debido a que el nivel de voltaje del puerto serial del Arduino es de 5 V y el voltaje de trabajo del *Xbee* es de 3.3 V. Se debe proteger este último de niveles mayores a 3.3 V conectando el puerto serial del Arduino a los pines *Din* y *Dout* del módulo.



**Figura 19. Tarjeta implementada nodo central, a. Vista superior, b. Vista inferior.**

### **Nodo Sensor/Actuador:**

Este nodo está compuesto por un módulo Xbee, un regulador de voltaje LM1117-3.3v, un amplificador operacional LM324, una cadena de LEDs, conectores (Headers y borneras) y resistores. Un nodo sensor actuador se encarga de recibir los valores analógicos de un sensor, enviar estos datos al nodo central en forma de tramas y recibir mensajes de configuración de este para modificar una salida PWM del módulo Xbee. En el anexo G se encuentra el diagrama esquemático del nodo sensor. En la Figura 21 se muestra una de las tarjetas de nodo sensor implementada. A continuación se describe con más detalle las partes de un nodo sensor.

#### ***Xbee Pro S1:***

Envía los valores del sensor leídos en su entrada analógica 0 y recibe los mensajes y configuraciones del nodo central para modificar el valor de su salida PWM0 y con ello variar el valor de voltaje que ingresa al amplificador operacional.

#### ***Cadena de LEDs:***

Cumple la función de actuador y permite visualizar el cambio en la configuración del módulo Xbee y varía su luminosidad dependiendo del valor de la salida PWM de este.

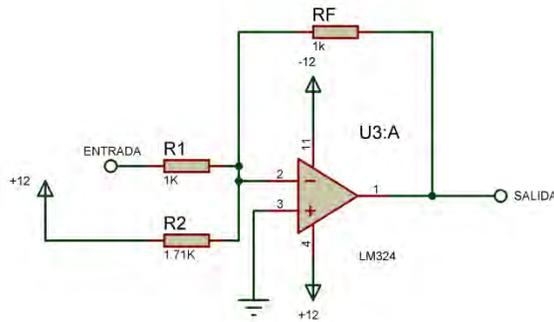
#### ***Regulador 3.3v + capacitores:***

El circuito es alimentado con 12V por lo tanto se requiere este regulador para bajar el voltaje al que trabaja el Xbee; se utilizó el regulador integrado LM1117-3.3v, sus características se pueden consultar en [A].

### **Amplificador Operacional:**

Se realizó una etapa de amplificación y suma con el integrado LM324. Se ingresa el voltaje de la salida PWM0 del Xbee y un nivel de 12V mediante una configuración sumador inversor se enciende la cadena de LEDs. El rango de trabajo seleccionado para ésta fue de 7V a 10.3V.

En la Figura 20 se tiene el diagrama del amplificador sumador inversor utilizado para la etapa de potencia.



**Figura 20. Sumador inversor con LM324**

La Ecuación 11 corresponde al sumador inversor.

$$v_0 = -R_f \left( \frac{v_1}{R_1} + \frac{v_2}{R_2} \right) \quad (11)$$

Donde  $v_1$  es el voltaje de referencia que se encuentra en el rango de 0 y 3.3v.  $v_2 = 12v$  Ya que es el mismo voltaje de alimentación de la tarjeta.

Para la amplificación,  $v_1 = 0V \Rightarrow v_0 = -7V$  y se le asigna un valor estándar a la resistencia de referencia  $R_f = 1K\Omega$ , por tanto en la ecuación se tiene:

$$-7 = -1K \left( \frac{12}{R_2} \right) \quad R_2 \cong 1.71K\Omega$$

Y para la resistencia  $R_1$ ,  $v_1 = 3.3V \Rightarrow v_0 = -10.3V$

$$-10.3 = -1K \left( \frac{3.3}{R_1} + \frac{12}{1.71K} \right) \quad R_1 \cong 1K\Omega$$

Los valores de resistencias calculados y utilizados para la etapa de amplificación se indican en la Tabla 16.

Tabla 16. Valores de resistencias para la etapa de amplificación

RESISTENCIA	VALOR CALCULADO	VALOR NOMINAL*
$R_F$	1K $\Omega$	1K $\Omega$
$R_1$	1.71K $\Omega$	1.5K $\Omega$ + 220 $\Omega$
$R_2$	1K $\Omega$	1K $\Omega$

\*valor comercial de las resistencias adquiridas.

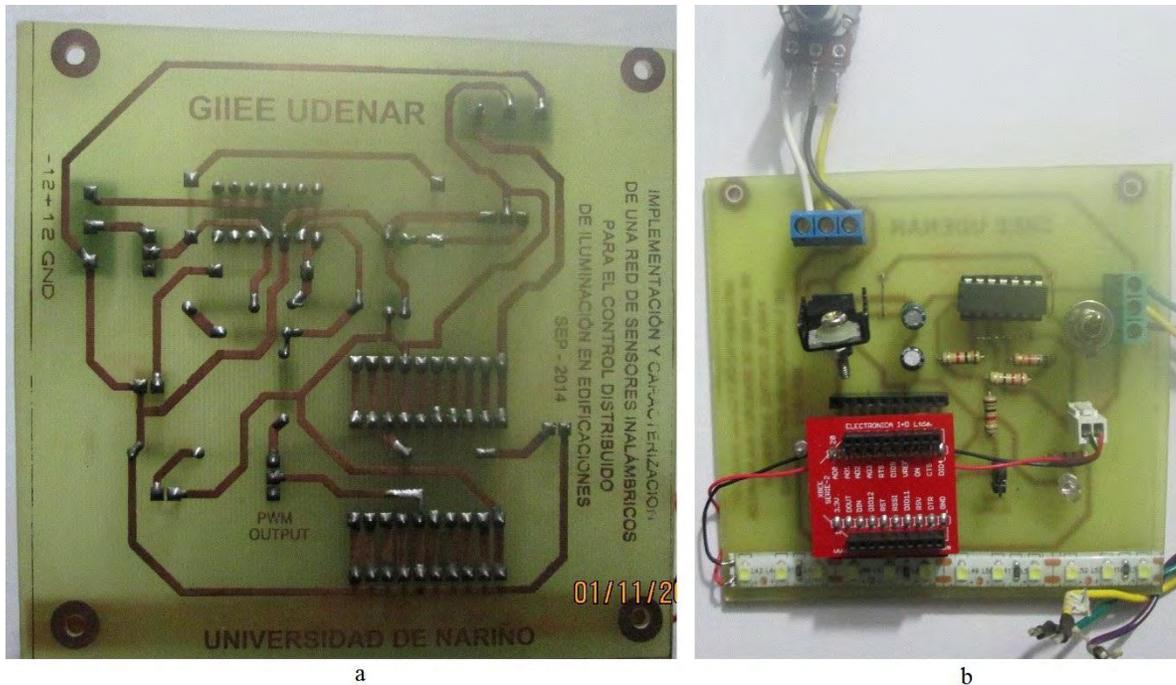


Figura 21. Tarjeta implementada nodo sensor, a. Vista inferior, b. Vista superior.

**4.2.2 Diseño de circuitos impresos.** Para el diseño de circuitos impresos se utilizó el software Proteus 8 Profesional y su componente Ares que permiten realizar el diagrama esquemático y el diagrama de circuito impreso. En el Anexo F se presentan los diagramas esquemáticos de los circuitos para los dos tipos de nodos y en el Anexo G se puede observar las ilustraciones (*artwork*) de los circuitos impresos.

Se usó la *técnica de transferencia de tonner* que consiste en la impresión en papel fotográfico del *artwork* y con una plancha la transferencia en la placa de pertinax, posteriormente se sumerge las placas en cloruro férrico y se ensambla cada uno de sus componentes. En el Anexo H se encuentra el presupuesto detallado de las tarjetas fabricadas.

**4.2.3 Caracterización del consumo de potencia.** En la Tabla 17, se indica las características de consumo para cada uno de los módulos sensores. Para obtener estos valores se mide la corriente en diferentes condiciones y se calcula la potencia con el voltaje entregado por cada fuente.

**Tabla 17. Características del consumo de nodos sensores**

NODO	VOLTAJE (V)		CORRIENTE (mA)		POTENCIA (W)	
	V+	V-	Xbee* Tx	Máxima Xbee** Tx/Rx	Xbee Tx	Máxima Xbee Tx/Rx
1	12.0	-11.8	83.4	86.3	1,00	1,04
2	11.4	-10.3	83.3	85.7	0,95	0,98
3	11.7	-10.9	80.0	82.5	0,94	0,97
4	11.7	-11.0	82.8	85.4	0,97	1,00
5	11.7	-11.7	83.5	86.5	0,98	1,01
6	11.9	-11.7	77.9	80.6	0,93	0,96
7	11.7	-11.3	78.3	80.7	0,92	0,94
8	11.2	-10.4	77.9	80.3	0,87	0,90

\*Sin Activar la cadena de LEDS, Transmitiendo la trama de muestras

\*\*Medida cuando la cadena de LEDS está encendida al 100%

## 4.3 PRUEBA 2

La segunda prueba de la red se realiza utilizando las tarjetas de nodo sensor/actuador, distribuidas en el tercer piso del edificio tecnológico de la universidad de Nariño como se muestra en el plano del Anexo D. Con esta prueba se trata de simular el comportamiento de la red en el escenario real; en este existirán variaciones de diferentes tipos y mediante los potenciómetros de los nodos sensores (Figura 20) que emulan los sensores de iluminación; mediante unas pequeñas modificaciones en la programación de Arduino es posible direccionar tramas y modificar una salida PWM en cualquier nodo sensor/actuador. El fin principal de esta prueba es verificar si los enlaces de cada nodo sensor con el nodo central tienen una comunicación correcta; de esta manera se configura la red de manera que la información de cada sensor sea enviada por intervalos de tiempo diferentes hasta el nodo central, para que éste active las salidas PWM de cada módulo. El rango de entrada analógica es de 0 a 3.3v al igual que el rango de voltaje de la salida PWM.

**4.3.1 Configuración módulos.** Para esta prueba los módulos se configuran como se muestra en la tabla 18, a diferencia de la prueba anterior, en esta es necesario configurar la entrada D0 como conversor analógico digital y el pin RSSI (6) del módulo Xbee como salida PWM (PWM0). Adicionalmente se configura el

parámetro IR de los módulos sensor/actuador para definir el tiempo de muestreo de los mismos.

Los módulos se configuran en el mismo canal y pan ID, el modo API se encuentra habilitado únicamente en el nodo central; las características de dirección son únicas en cada Xbee para evitar obstáculos y poder direccionar correctamente la información.

**Tabla 18. Configuración módulos prueba 2**

PARAMETRO	NODO								
	CENTRAL	1	2	3	4	5	6	7	8
SL*	40B32 7FB	40B32 A4B	40B32 7FA	4091D 3C1	4091D 3B9	40995 A7A	4091D 3C0	40BA7 31F	40B32 A35
MY	BBBB	DEDE	AAAA	ABAB	ABCD	BABA	FEFE	ACDE	CFCF
DL	0	BBBB							
AP	1**	0	0	0	0	0	0	0	0
CE	1***	0	0	0	0	0	0	0	0
D0****	0	2	2	2	2	2	2	2	2
P0*****	0	2	2	2	2	2	2	2	2
IR*****	--	3E8							

Configuración de los módulos Xbee para la primera prueba de la red.

\*Dirección de 64bits, (Serial number low). Viene de fábrica.

\*\*El nodo central debe configurarse en modo API (AP=1).

\*\*\*El nodo central se asigna como coordinador (CE=1).

\*\*\*\*Habilita como ADC la entrada D0.

\*\*\*\*\*Habilita como PWM el pin 6(RSSI) del módulo.

\*\*\*\*\*Tiempo de muestreo (ms).

**4.3.2 Parámetros de las Tramas.** En esta prueba se utilizan dos tipos de tramas, la tipo *0x83 Received packet 16 bit address IO* que es recibida por el nodo central y contiene las muestras o valores sensados por cada dispositivo final y la tipo *0x17 Remote AT command* que en esta caso, es enviada desde el nodo central a cada uno de los nodos sensores para configurar el ciclo útil de la salida PWM en el estado deseado. En la tabla 19 se muestra dos ejemplos de trama tipo 0X83 y en la tabla 20 se consignan dos ejemplos de trama tipo 0X17 en la cual se puede apreciar cada parámetro de la trama y las posiciones que tiene cada byte y/o Word.

**Tabla 19. Parámetros Trama tipo 0x83 Rx packet 16 bit address io**

<b>PARAMETROS DE LA TRAMA (FRAME PARAMETERS)</b>	<b>VALOR (FIELD VALUE)*</b>	<b>VALOR (FIELD VALUE)**</b>	<b>POSICION</b>
Delimitador de Inicio (Start delimiter)	7E	7E	
Longitud (Length)	00 0A	00 0A	0-1
Tipo de trama (Frame type)	83	83	2
Dirección de origen de 16 bits DL (16-bits source address DL)	DE DE	CF CF	3-4
RSSI	24	24	5
Opciones (Options)	00	00	6
Numero de muestras	01	01	7
Indicador de canal*	02	02	8
	00	00	9
Dato RF (RF data)	00 B2	00 B0	10-11
Suma de prueba (Checksum)	B8	B6	12
<b>Trama</b>	<b>7E 00 0A 83 DE DE 24 00 01 02 00 00 B2 B8</b>	<b>7E 00 0A 83 CF CF 24 00 01 02 00 00 B0 B6</b>	

\*Trama vista por el coordinador del valor sensado por el dispositivo 1 (MY=DEDE), potencia 24 (-36dBm).  
 \*\*Trama vista por el coordinador del valor sensado por el dispositivo 8 (MY=CFCF), Potencia 24 (-36dBm).

**Tabla 20. Parámetros Trama tipo 0x17 remote at command**

<b>PARAMETROS DE LA TRAMA (FRAME PARAMETERS)</b>	<b>VALOR (FIELD VALUE)*</b>	<b>VALOR (FIELD VALUE)**</b>
Delimitador de Inicio (Start delimiter)	7E	7E
Longitud (Length)	00 11	00 11
Tipo de trama (Frame type)	17	17
Identificador de Trama (Frame ID)***	00	00
Dirección de 64 bits de destino SH + SL	00 13 A2 00 40 B3 2A 4E	00 13 A2 00 40 91 D3 C1
Dirección de 16 bits de destino DL (16-bits source address DL)	DE DE	AB AB
Opciones de comando remoto (aplicar cambios)	02	02
Comando AT (M0)	4D 30	AD 30
Valor de parámetro (0-1023)	03 FF	02 00
Suma de prueba (Checksum)	8B	F7
<b>Trama</b>	<b>7E 00 11 17 00 00 13 A2 00 40 B3 2A 4E DE DE 02 4D 30 03 FF 8B</b>	<b>7E 00 11 17 00 00 13 A2 00 40 91 D3 C1 AB AB 02 4D 30 02 00 F7</b>

\*Trama enviada por el coordinador al dispositivo 1 (MY=DEDE) para poner la salida PWM0 en su valor máximo (03 FF)  
\*\*Trama enviada por el coordinador al dispositivo 1 (MY=DEDE) para poner la salida PWM0 en su valor medio (02 00)  
\*\*\*Identifica los datos de la trama para enviar una respuesta; si se configura como 0x00 la respuesta no es requerida.

**4.3.3 Programación.** A continuación se tiene una descripción general de la programación implementada en el Arduino del nodo central para la prueba 2:

- Definición de variables
- Inicializar LCD, puerto serial.
- Leer puerto serial y almacenar por posiciones las tramas a partir del byte 7E.
- Comparar las posiciones 3 y 4 de la trama con las direcciones que identifican cada nodo de origen.
- Guardar los bytes en las posiciones 10 y 11 correspondientes a las muestras de los sensores y el byte 5 que es la potencia de transmisión.
- Con los bytes 3 y 4 de dirección define el nodo de destino para configuración remota.
- Con los bytes correspondientes a las muestras guardadas asigna el valor PWM que se enviara al nodo de destino.

- Envía el PWM como comando AT remoto para configurar la salida PWM del nodo de destino.
- Se muestra en la LCD el nodo de origen, el valor del sensor en cada uno y la potencia de transmisión.

Para el direccionamiento del mensaje se utiliza una función que asocia la dirección de origen *MY*, *SH* y *SL* con un número asignado a cada nodo, correspondiente a los datos de la tabla XVIII.

El envío del comando AT remoto se realiza con otra función que construye y envía las tramas con la información de dirección y PWM.

El código realizado en Arduino con la programación descrita arriba se encuentra en el anexo E; tiene como fin emular y caracterizar la red implementada, permitiendo que la tarjeta Arduino establezca una comunicación serial con el módulo *Xbee* coordinador y se pueda leer e interpretar los datos provenientes de cada uno de los nodos sensores y la potencia de transmisión de cada uno de ellos, además de mostrar la información que recibe y generar y enviar una trama de control. Los nodos sensores están provistos de un potenciómetro encargado de emular un sensor de iluminación y una cadena de LEDs que emula a un actuador luminoso, cuya intensidad de luz está dada por la configuración remota que se realiza desde el nodo central; como se mencionó anteriormente. En base a esta prueba se puede establecer las modificaciones necesarias a la red ya sea en cuanto a programación o ubicación de los dispositivos.

**4.3.4 Prueba.** La prueba de la red en funcionamiento se realizó en el piso 3 del edificio tecnológico de la Universidad de Nariño, cada nodo final se compone de una tarjeta sensor/actuador y una fuente de computador, ya que esta entrega los voltajes requeridos (12v, -12v, GND) tanto para la alimentación como para la etapa de potencia de la cadena de LEDs. En esta prueba no es necesario el uso de computadores ya que cada nodo tendría su propia alimentación y el procesamiento de la información se realiza en el nodo central con lo cual la red se hace autónoma.

En la pantalla LCD del nodo central se puede visualizar el valor de voltaje sentido y la potencia de la señal transmitida por cada nodo final; se almacena los datos de potencia por enlace mostrados en la LCD; 10 datos por cada enlace para obtener un porcentaje promedio que pueda caracterizar la red.

Para la ejecución de esta prueba surge un inconveniente que afecta significativamente el desempeño de la red, que es la escasez de tomas eléctricas; por esta razón los nodos sensores debieron ser ubicados en sitios un poco alejados o que tenían 2 o 3 obstáculos; por ello muchos datos fueron recibidos erróneamente y en la mayoría de los casos los actuadores no se activaban, es

decir las tramas de configuración remota eran recibidas de forma incorrecta por los nodos sensores o simplemente no se recibían.

**4.3.5 Resultados.** En la Tabla 21 se muestran las características de la red obtenidas durante la prueba 2, se puede notar que la mejor calidad de la señal se presenta en el enlace 7, esto es debido a la ausencia de obstáculos y la distancia corta entre el nodo final 7 y el central; como se experimentó en las pruebas iniciales la distancia y principalmente los obstáculos afectan significativamente la comunicación en la red; por ello entre los peores casos encontrados, están los nodos 3, 6, y 8 que no tienen un porcentaje alto de la calidad de la señal, como se ve en la Tabla 25 correspondiente a los datos de la prueba 1, estos nodos presentan pérdida de paquetes alto, por tanto es inevitable que ocurran errores al momento de enviar el valor sensado y la trama de control para encender correctamente el actuador.

**Tabla 21. Características de la red Prueba 2**

ENLACE	OBSTÁCULOS	DISTANCIA [m]	RSSI* %
1	1	6.73	63.54
2	0	18.83	65.63
3	2	24.04	41.67
4	2	13.91	52.60
5	1	9.80	40.63
6	2	18.81	43.75
7	0	10.47	90.63
8	3	25.72	43.23

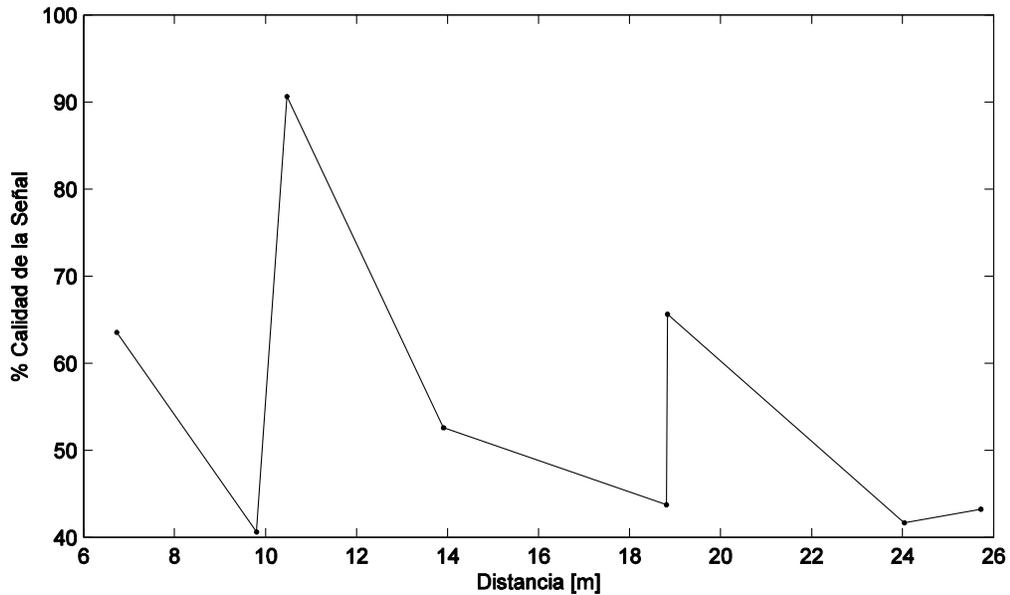
Relaciona cada enlace con la intensidad de la fuerza de la señal (RSSI), y distancia entre nodo final y central.

\*Se utiliza la ecuación 1 para determinar la calidad de la señal mediante porcentaje dentro del rango de valores en el cual se recibe satisfactoriamente los paquetes de datos.

En cuanto a la potencia de transmisión y sensibilidad de recepción para los transceptores 802.15.4 se establece la cantidad mínima de energía necesaria para transmitir en -3 dBm, (0,5 mW) y la sensibilidad mínima en el receptor en -92 dBm (6,3 \* E-10 mW). Algunos módulos de radio como los Waspote tienen una potencia de transmisión variable desde 0 dBm (1 mW); Xbee 802.15.4 de 20 dBm (100 mW) y para el Xbee Pro, ambos valores son superiores al mínimo establecido en la norma 802.15.4. En cuanto a la sensibilidad de recepción, el

Xbee muestra -92 dBm ( $6,3 \cdot E-10$  mW) y -100 dBm ( $1 \cdot E-10$  mW) el Xbee Pro [35].

Esto significa que se detecta cualquier paquete que llega con energía tan baja como  $0,000000000063$  mW para el Xbee y  $0,00000000001$ mW para el Xbee Pro. [35]



**Figura 22. Distancia vs calidad señal prueba 2 de la red implementada**

En la Figura 19 se puede apreciar el porcentaje de calidad de la señal en función de la distancia del nodo central a los nodos sensores, fácilmente se puede apreciar que la calidad de la señal decrece con la distancia siendo el nodo 8 el peor de los casos; como se había mencionado antes la calidad de la señal se ve afectada además por los obstáculos que se puedan presentar en los enlaces; por esta razón la calidad de la señal para los nodos 1, 3, 4, 5, 6 y 8 se ve muy afectada. En el plano del Anexo D se puede apreciar con más claridad la ubicación, distancia e obstáculos para esta prueba. Se observa que el nodo 1, 4 y 5 a pesar de estar muy cerca del nodo central, estos presentan una baja calidad de la señal debido a que los muros que hacen interferencia son de tipo estructural y metálico, adicionalmente se puede notar que los dos nodos que tienen línea de vista presentan la mejor calidad de la señal y según la prueba de la sección V-C, la menor pérdida de paquetes.

A partir de los resultados obtenidos se debe replantear la ubicación de los nodos sensores, puesto que se presentan demasiados datos erróneos, sobre todo en los enlaces que presentan menor calidad de la señal; adicionalmente las tramas de

control PWM no se reciben y por lo tanto el actuador de iluminación (cadena de LEDS) no se activa en ninguna de las tarjetas, lo que era esperado porque se presentaban demasiados obstáculos para algunos nodos. Por falta de tiempo y de recursos para poder conectar los nodos mediante baterías o extensiones eléctricas, no se realiza pruebas adicionales de la red, sin embargo al disminuir las distancias e obstáculos de algunos nodos se pueden obtener mejores resultados en la comunicación de la red.

Con el análisis de las pruebas realizadas se recomiendan distancias máximas entre los nodos de transmisión y recepción para la cantidad de obstáculos existentes, esto aplicaría para los módulos *Xbee pro S1*, en la tabla XXII se puede asociar mejor esta recomendación. Para tres obstáculos no se puede recomendar una distancia entre dos puntos porque se puede presentar una pérdida de paquetes considerable.

Obstáculos*	Distancia [m]
0	70
1	17
2	12
3	>8

\*Obstáculos no metálicos

## CONCLUSIONES

Los módulos utilizados en la implementación de la red son muy limitados en cuanto a topologías ya que, son diseñados principalmente para conexiones punto a punto. Para formar una red con estos *Xbee* es necesario utilizar algún dispositivo de control, ya que por sí solos no pueden enrutar la información y por tanto, tampoco permite crear varias topologías.

Para la topología en estrella escogida fue necesario utilizar un Arduino mega 2560 que junto con un módulo *Xbee* pro s1 formaron lo que se denominó nodo central, encargado de organizar la información adquirida por cada nodo y enviar la respuesta al actuador; para realizar otras topologías es necesario enrutadores que podrían ser formados con un módulo *Xbee* pro s1 y con un controlador que podría ser de la línea de Arduino o un microcontrolador, el inconveniente principal es su alto costo y la complejidad en la programación para lograr una ruta óptima para la comunicación deseada. Por tanto, se recomienda usar gamas más altas de *Xbee* para lograr el enrutamiento de la información por sí mismo, sin necesidad de agregar controladores adicionales; algunos ejemplos de las series que se pueden emplear para este fin son la S2, S3B, ZB o Digimesh.

Gracias a las pruebas realizadas en su totalidad, se determinó que las estructuras metálicas presentes en el edificio y los muros estructurales del tercer piso interfieren notablemente en la señal ocasionando en algunos casos, porcentaje alto de pérdida de paquetes, errores en los mensajes y en las respuestas de los actuadores. Por tanto, se sugiere, para un trabajo futuro utilizar módulos en los que pueda implementarse otra topología que permita optimizar las rutas de la señal de acuerdo a la calidad de la misma; evitando atravesar las obstáculos mencionadas.

En base a las pruebas realizadas, se concluye que se puede ubicar los nodos en el espacio determinado, de manera que la cantidad y tipo de obstáculos se vea compensado con distancias más cortas entre nodos; para el caso del presente trabajo, se requiere de distancias cortas entre el nodo sensor y el nodo coordinador; con el fin de disminuir las pérdidas de propagación y a su vez, mejorar la calidad de la señal recibida.

El modo de operación API es una herramienta muy poderosa en las WSN con *Xbee* o *Zigbee* ya que permiten obtener información como direcciones de origen, potencia o incluso el estado en que cada dispositivo se encuentra, de todos los nodos conectados en red y una de las características más importantes es que permite la configuración remota de dispositivos mediante el envío de tramas muy sencillas de construir.

Inicialmente se había propuesto la implementación y pruebas de la red en el piso 4 del edificio mencionado anteriormente, sin embargo este piso no estuvo disponible, lo que conlleva a que las pruebas se realizaran en el piso 3 y se dificultara la disposición de nodos en dicha planta. Debido a la disponibilidad de tomas eléctricas y al acceso limitado a las diferentes aulas del piso en cuestión, se tuvo la necesidad de buscar algunos lugares adecuados para probar la red en funcionamiento. Sin embargo, los resultados no fueron los esperados porque algunos nodos quedaron muy alejados del nodo central y otros con obstáculos que deterioraban notoriamente la comunicación en la red.

Se obtuvo una red básica que tiene un desempeño aceptable y que permite acoplar un algoritmo de control, para el caso, control PWM, la inclusión de diferentes tipos de configuraciones remotas y la vinculación de muchos más dispositivos a partir de algunas modificaciones al código de programación. Adicionalmente mediante la modificación de las tarjetas implementadas y el uso de otras series de módulos Xbee, es posible implementar diferentes tipos de redes con otras topologías que permitan hacer la red más compleja y confiable. Además es posible la conexión de cualquier tipo de sensor con su respectivo ajuste de nivel de voltaje directamente a las entradas del módulo utilizado.

El presente proyecto deja campo abierto a múltiples investigaciones sobre redes de sensores inalámbricos. Se recomienda probar nuevos dispositivos o nuevas versiones de módulos Xbee, tales como los programables, que pueden llegar a trabajar de manera autónoma sin necesidad de un dispositivo de control ya que tienen mayores capacidades en cuanto a procesamiento y almacenamiento de información; o de series superiores al utilizado en este proyecto que tienen la posibilidad de enrutar la información entre otras funciones y por tanto permitirán optimizar la red.

## RECOMENDACIONES

Se recomienda usar gamas más altas de módulos *Xbee* que soporten otras configuraciones y topologías, por ejemplo módulos a partir de la serie 3 que permiten programarse internamente para realizar tareas similares a las de un microcontrolador, lo que permitiría tener nodos autónomos y mayores prestaciones. Tener precauciones con el voltaje y la corriente de alimentación de los módulos *Xbee*, puesto que son muy sensibles a perturbaciones, además de tener especial cuidado con las conexiones de sus pines.

Es muy importante realizar las configuraciones pertinentes para el funcionamiento de los módulos, se debe tener en cuenta principalmente las configuraciones de entradas/salidas (I/O Settings), los parámetros IR (Tiempo de muestreo), PAN ID (Identificador de PAN) y CH (canal), además de las conexiones necesarias como el voltaje de referencia  $V_{ref}$  en caso de utilizarse las entradas analógicas. Al realizar la configuración de los módulos *Xbee*, es recomendable elegir un canal distinto al de su configuración predeterminada o no colocar un canal fijo, para que el módulo coordinador elija uno de los canales disponibles. Esto es necesario, especialmente cuando existen redes WI-FI cercanas que puedan interferir con la red, adicionalmente se requiere realizar un análisis de interferencias en el entorno mediante un analizador de espectros y así poder elegir el canal adecuado. Es recomendable tener en cuenta la metodología de diseño para lograr un correcto funcionamiento de la red que cumpla con los requerimientos y facilitar el proceso de diseño e implementación. Para el trabajo realizado se empleó fuentes conectadas a la red eléctrica y una de las características principales del estándar 802.15.4, es el bajo consumo de energía, por lo que se recomienda utilizar baterías recargables, y si es posible hacer uso de algún tipo de energía renovable.

Para elegir la topología adecuada puede compensarse la cantidad y tipo de obstáculos con la distancia entre los nodos, además se recomienda seguir la tabla que se muestra a continuación para la distribución espacial de los nodos sensores, de manera que se ubiquen de acuerdo a las distancias recomendadas con los respectivos obstáculos para que se obtenga una buena comunicación en una red.

<b>Obstáculos*</b>	<b>Distancia [m]</b>
<b>0</b>	70
<b>1</b>	17
<b>2</b>	12
<b>3</b>	9
<b>*Obstáculos no metálicos</b>	

## BIBLIOGRAFÍA

- [1] He Danping, A novel methodology for planning reliable wireless sensor networks, Tesis Doctoral, Centro de Electronica Industrial, Universidad Politecnica de Madrid, Centro de Electronica Industrial, 2014.
- [2] Dargie, Waltenegus, Poellabauer. Fundamentals of wireless sensor networks: theory and practice. John Wiley & Sons, 2010.
- [3] Alonso, N. Redes de comunicaciones industriales. Editorial UNED, 2013.
- [4] Araque, C., Sánchez, C, Diseño e Implementación de un Sistema Domótico Inalámbrico Basado en el Protocolo de Redes de Comunicación Zigbee y Sistema de Supervisión HMI para la Seguridad y Eficiencia de Consumo Energético en Hogares Ecuatorianos, Trabajo de grado, Ingeniería Electrónica e Instrumentación, Universidad de las Fuerzas Armadas. Ecuador, 2012.
- [5] Carvajal, S., Villafuerte M. Evaluación de las topologías físicas de WSN mediante la implementación de un prototipo de medición de variables ambientales para el GEAA-ESPOCH. Trabajo de grado, Ingeniería en Electrónica, Telecomunicaciones y redes, Escuela Superior Politécnica de Chimborazo. Ecuador, 2014.
- [6] Flores, E., Redes de Sensores Inalámbricas Aplicado a la Medicina. Trabajo de grado, Máster en Tecnologías de la Información y Comunicaciones en Redes Móviles, Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicaciones, Universidad de Cantabria, España. 2012.
- [7] Barneda, I., Aguilo, J., Soler, V. Zigbee aplicado a la transmisión de datos de sensores biomédicos. Trabajo de grado, Escuela Técnica Superior de Ingeniería de la UAB, Universidad Autónoma de Barcelona, España. 2008.
- [8] Gutiérrez, M., Clasificador de Actividades, Trabajo de grado, Ingeniería en Informática, Universidad Pontificia Comillas, España. 2009.
- [9] Flórez, D., Sánchez, C., Páez C. Diseño e implementación de una red de sensores para la adquisición de variables relacionadas con la vigilancia estructural de puentes. Artículo en línea, Revistas científicas, Pontificia Universidad Javeriana

Vol. 13, No.- 2, Colombia, 2010. Disponible en:  
<http://revistas.javeriana.edu.co/index.php/iyu/article/view/964/562>

[10] Castro, N., Chamorro L. Diseño e Implementación de una Red de Sensores Inalámbricos para el Control de Riego por Goteo, Trabajo de grado, Ingeniería Electrónica, Universidad de Nariño, 2012.

[11] Barón Juan, Aplicación de tecnología Zigbee para el monitoreo de variables ambientales en invernaderos, Artículo en línea, Universidad Nacional Abierta y a Distancia, Colombia, 2013. Disponible en: [http://uea2013.frbb.utn.edu.ar/wp-content/uploads/PF\\_4.pdf](http://uea2013.frbb.utn.edu.ar/wp-content/uploads/PF_4.pdf)

[12] Lanyun, S., Hongli, L., Xiuzhen, Z. The economical electrical power control system of building illumination based on the rf wireless correspondence network. Chinese Control and Decision Conference (CCDC). IEEE, 2011.

[13] Zorrilla, C. Diseño de un sistema de gestión inalámbrica de iluminación LED. Trabajo de grado, Máster en Tecnologías de la Información y Telecomunicaciones, Universidad Politécnica de Cartagena, 2012.

[14] España N., Diseño e Implementación de un Control Distribuido Basado en Dinámicas de Replicadores, Trabajo de grado, Ingeniería Electrónica, Universidad de Nariño, 2014.

[15] Estándar IEEE 802.15.4-2006 - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), disponible en:  
<http://standards.ieee.org/getieee802/download/802.15.4g-2012.pdf>

[16] Acedo, A., Molina, M., Ortigoza, R. Revisión del proceso de identificación de nodos en las Wireless Sensor Networks. Septiembre, Revista Electrónica de Estudios Telemáticos, Universidad Rafael Belloso Chacin, 2008.

[17] The Major Differences in the Xbee Series 1 vs. the Xbee Series 2. Artículo en línea disponible en:  
<http://www.digi.com/support/kbase/kbaseresultdetl?id=2213>

[18] Manual de Usuario: Product Manual v1.xEx - 802.15.4 Protocol. Documento en línea disponible en:  
<https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Manual.pdf>

- [19] Descripción software X-CTU, disponibles en: <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/xctu>
- [20] Descripción producto Xbee Xplorer Dongle sparkfun, disponible en: <https://www.sparkfun.com/products/8687>
- [21] Descripción Tarjeta Arduino, disponible en: <http://www.arduino.cc/es/>
- [22] Hoja de características Arduino Mega 2560, disponible en: <http://www.atmel.com/Images/doc2549.pdf>
- [23] Descripción del producto Convertidor de nivel lógico bidireccional ADAF-395, disponible en: [http://www.didacticaselectronicas.com/index.php?page=shop.product\\_details&flypage=flypage.tpl&product\\_id=1879&category\\_id=25&option=com\\_virtuemart&Itemid=30](http://www.didacticaselectronicas.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=1879&category_id=25&option=com_virtuemart&Itemid=30)
- [24] López, D., Gala S. Red inalámbrica para instrumentación de procesos. Trabajo de grado, Ingeniería Electrónica Industrial, Universidad Técnica de Catalunya 2012.
- [25] Bonivento, A., Carloni, L., Sangiovanni-Vincentelli, A. Platform based design for wireless sensor networks. Mobile Networks and Applications 11.4 (2006), IEEE, pag. 469-485.
- [26] Wang, Q., Yang, W. Energy consumption model for power management in wireless sensor networks. Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on. IEEE, 2007.
- [27] Lee, M., Eom, K., Kang, H., Shin, C., Yoe, H. Design and implementation of wireless sensor network for ubiquitous glass houses. Computer and Information Science, 2008. ICIS 08. Seventh IEEE/ACIS International Conference on. IEEE, 2008.
- [28] Pan, M., Yeh, L., Chen, Y., Lin, Y., Tseng, Y. A WSN-based intelligent light control system considering user activities and profiles. Sensors Journal, IEEE 8.10 2008: 1710-1721.

[29] Yang, H., He, Y. Wireless sensor network for orchard soil and climate monitoring. Computer Science and Information Engineering, 2009 WRI World Congress on. Vol. 1. IEEE, 2009.

[30] Planos nuevo edificio tecnológico Universidad de Nariño. Fondo de construcciones de la Universidad de Nariño.

[31] Cobos E. "Estudio de las redes sensoriales como una nueva alternativa de comunicación inalámbrica". Trabajo de grado, Escuela Politecnica del Ejercito, 2007, pp. 127-128.

[32] Yanjun Li; Zhi Wang; Yeqiong Song, "Wireless Sensor Network Design for Wildfire Monitoring," WCICA 2006, The Sixth World Congress on Intelligent Control and Automation, vol.1, 2006, pp.109,113.

[33] Hoja de características Xbee Digimesh 2.4, disponible en: [http://www.digi.com/pdf/ds\\_xbeedigimesh24.pdf](http://www.digi.com/pdf/ds_xbeedigimesh24.pdf)

[34] Campos, G., "Caracterización del Consumo en Redes Zigbee/802.15.4". Trabajo de grado, Ingeniería de Telecomunicación, Universidad de Málaga, España, 2009.

[35] Wireless Sensor Networks, 802.15.4 vs ZigBee, artículo en línea disponible en: <http://sensor-networks.org/index.php?page=0823123150>

[36] Dargie, Waltenegus, and Poellabauer Christian. Fundamentals of wireless sensor networks: theory and practice. John Wiley & Sons, 2010.

[37] Mesh networking and P2P distributed voip, Artículo en línea disponible en: <http://freeasinfreenet.blogspot.com/2011/06/xbee-mesh.html>

# **ANEXOS**

**ANEXO A: PLANOS DE LOS PISOS 3 Y 4 DEL BLOQUE TECNOLÓGICO DE  
LA UNIVERSIDAD DE NARIÑO CON LA UBICACIÓN DE LOS NODOS PARA  
CADA PRUEBA POR ENLACES**

## ANEXO B: CÓDIGO PARA REALIZAR LAS PRUEBAS DE TRES NODOS IMPLEMENTADO EN ARDUINO

```

//Código utilizado para las pruebas de tres nodos,
//con # envia la potencia de la señal al nodo 3

#include <stdlib.h>
#include <stdio.h>
#include <SoftwareSerial.h>

char sentence[100];
char valor;
char db[3];

void setup()
{
  Serial1.begin(9600);
}

void loop()
{
  if(Serial1.available()>0)
  {
    char cadena = Serial1.read();

    if(cadena == '#')
    {
      delay(1000);
      Serial1.write("+++");
      delay(1100);
      leer();
      if(sentence[0]=='O'
      &&sentence[1]=='K')
      {
        Serial1.write("ATDB");
        Serial1.write('\r');
        delay(100);

        db[0] = Serial1.read();
        db[1] = Serial1.read();

        delay(100);
        Serial1.write("ATCN");
        Serial1.write('\r');
        delay(100);
        leer();
        Serial1.write(db);
      }
    }
    else
    {
      Serial1.write(cadena);
    }
  }
}

void leer()
{
  static int i=0;
  while(Serial1.available())
  {
    char ch=Serial1.read();
    if(ch!='\r')
    {
      sentence[i]=ch;
      i++;
    }
    else
    {
      sentence[i]='\0';
      i=0;
    }
  }
}

```

**ANEXO C: PLANOS DEL PISO 4 DEL EDIFICIO TECNOLÓGICO DE LA  
UNIVERSIDAD DE NARIÑO CON LA UBICACIÓN DE LOS NODOS PARA  
CADA TIPO DE TOPOLOGÍA.**

**ANEXO D: PLANO DEL PISO 3 DEL EDIFICIO TECNOLÓGICO DE LA  
UNIVERSIDAD DE NARIÑO CON LA UBICACIÓN DE LOS NODOS PARA LA  
RED FINAL**

## ANEXO E: CÓDIGOS DE LA RED FINAL IMPLEMENTADA EN ARDUINO

### PRUEBA 1: COMPUTADORES

```

//Codigo para probar la red con envio de mensaje
//direccionando mediante x-ctu

byte origen[100];
char datos1[100], datos2[100],
datos3[100], datos4[100];
char datos5[100], datos6[100],
datos7[100], datos8[100];
char msj[100], msj1[100],msj2[100],
msj3[100], DB, m1;
char msj4[100], msj5[100],msj6[100],
msj7[100];
int dato, j=0,k=0, f=0, p=0, dir=0,
dir1=0, dir2=0, dir3=0;
int dir4=0, dir5=0, dir6=0, dir7=0,
dir8=0;
int m=0, n=0, l=0, g=0, L=0,
adress=0, adress1=0, t=0;
unsigned long dec=0, ld=0;
int rt=0, rt1=0, rt2=0, rt3=0, h1=0,
h2=0, h3=0, h4=0, rs=0;

#include <LiquidCrystal.h>
LiquidCrystal LCD(12, 11, 5, 4, 3, 2);

void setup()
{
  Serial1.begin(9600);
  LCD.begin(16, 2);
  LCD.print("Modo Prueba PCs");
}

void loop()
{
  if(Serial1.available() >=10)
  {
    if(Serial1.read()==0x7E)
    {
      for(int i=0; i<=8; i++)
      {
        origen[i]= Serial1.read();
      }
      DB=origen[5];
    }

    //nodo 1__DEDE
    if(origen[3]==222 &&
origen[4]==222)
    {
      if(origen[7]==13)
      {
        m=j;
        j=0;
        dir=datos1[0];

        if(dir>='2' && dir<='8')
        {
          for(int x=0; x<=m; x++)
          {
            msj[x]=datos1[x+1];
            datos1[x]=0;
          }

          direccion(dir);
          enviar(msj,DB,m+1);
        }
      }
    }
    else
    {
      datos1[j]=(origen[7]);
      msj[j]='\0';
      j++;
    }
  }

  //nodo 2__AAAA
  else if(origen[3]==170 &&
origen[4]==170)
  {
    if(origen[7]==13)
  
```

```

    {
        n=k;
        k=0;
        dir1=datos2[0];
        if(dir1=='1' || (dir1>='3' &&
dir1<='8'))
        {
            for(int y=0; y<=n; y++)
            {
                msj1[y]=datos2[y+1];
                datos2[y]='\0';
            }
            direccion(dir1);
            enviar(msj1,DB,n+1);
        }
    }

    else
    {
        datos2[k]=(origen[7]);
        msj1[k]='\0';
        dir1=0;
        k++;
    }
}

//nodo 3___ABAB
else if(origen[3]==171 &&
origen[4]==171) //ABAB
{
    if(origen[7]==13)
    {
        l=f;
        f=0;
        dir2=datos3[0];

        if(dir2=='1' || dir2=='2' ||
(dir2>='4' && dir2<='8'))
        {
            for(int w=0; w<=l; w++)
            {
                msj2[w]=datos3[w+1];
                datos3[w]='\0';
            }

            direccion(dir2);
            enviar(msj2,DB,l+1);
        }
    }

    else
    {
        datos3[f]=(origen[7]);
        msj2[f]='\0';
        dir2=0;
        f++;
    }
}

//nodo 4___ABCD
else if(origen[3]==171 &&
origen[4]==205)
{
    if(origen[7]==13)
    {
        g=p;
        p=0;
        dir3=datos4[0];

        if(dir3!='4' && dir3<='8')
        {
            for(int z=0; z<=g; z++)
            {
                msj3[z]=datos4[z+1];
                datos4[z]='\0';
            }
            direccion(dir3);
            enviar(msj3,DB,g+1);
        }
    }
    else
    {
        datos4[p]=(origen[7]);
        msj3[p]='\0';
        dir3=0;
        p++;
    }
}

//nodo 5___BABA

```

```

else if(origen[3]==186 &&
origen[4]==186)
{
if(origen[7]==13)
{
n=k;
k=0;
dir4=datos5[0];
if(dir4!='5' && dir4<='8')
{
for(int y=0; y<=n; y++)
{
msj4[y]=datos5[y+1];
datos5[y]='\0';
}
direccion(dir4);
enviar(msj4,DB,n+1);
}
}
else
{
datos5[k]=(origen[7]);
msj4[k]='\0';
dir4=0;
k++;
}
}

//nodo 6___FEFE
else if(origen[3]==254 &&
origen[4]==254)
{
if(origen[7]==13)
{
n=k;
k=0;
dir5=datos6[0];
if(dir5!='6' && dir5<='8')
{
for(int y=0; y<=n; y++)
{
msj5[y]=datos6[y+1];
datos6[y]='\0';
}
}
}
else
{
datos6[k]=(origen[7]);
msj6[k]='\0';
dir6=0;
k++;
}
}

direccion(dir5);
enviar(msj5,DB,n+1);
}
}

else
{
datos6[k]=(origen[7]);
msj5[k]='\0';
dir5=0;
k++;
}
}

//nodo 7___ACDE
else if(origen[3]==172 &&
origen[4]==222)
{
if(origen[7]==13)
{
n=k;
k=0;
dir6=datos7[0];
if(dir7!='7' && dir7<='8')
{
for(int y=0; y<=n; y++)
{
msj6[y]=datos7[y+1];
datos7[y]='\0';
}
}
direccion(dir6);
enviar(msj6,DB,n+1);
}
}
else
{
datos7[k]=(origen[7]);
msj6[k]='\0';
dir6=0;
k++;
}
}
}

```

```

//nodo 8___CFCF
else if(origen[3]==207 &&
origen[4]==207)
{
    if(origen[7]==13)
    {
        n=k;
        k=0;
        dir7=datos8[0];
        if(dir8!='8' && dir8<'8')
        {
            for(int y=0; y<=n; y++)
            {
                msj7[y]=datos8[y+1];
                datos8[y]='\0';
            }
            direccion(dir7);
            enviar(msj7,DB,n+1);
        }
    }

    else
    {
        datos8[k]=(origen[7]);
        msj7[k]='\0';
        dir7=0;
        k++;
    }
}
}

void escribir_LCD(int nodo,float
dat,int db)
{
    //LCD.clear();
    LCD.home();
    LCD.print("Nodo ");
    LCD.setCursor(5,0);
    LCD.print(nodo);
    LCD.setCursor(6,0);
    LCD.print(": ");

    LCD.setCursor(0, 1);

    LCD.print(dat);
    LCD.setCursor(4, 1);
    LCD.print('V');

    LCD.setCursor(8,1);
    LCD.print(-db,DEC);
    LCD.setCursor(11,1);
    LCD.print("dBm");
    delay(500);
}

void direccion(int ad)
{
    switch (ad) {
        case '1':
            // DEDE
            adress=222;
            adress1=222;
            break;

        case '2':
            // AAAA
            adress=170;
            adress1=170;
            break;

        case '3':
            // ABAB
            adress=171;
            adress1=171;
            break;

        case '4':
            // ABCD
            adress=171;
            adress1=205;
            break;

        case '5':
            // BABA
            adress=186;
            adress1=186;
            break;
    }
}

```

```

    case '6':
    // FEFE
    adress=254;
    adress1=254;
    break;

    case '7':
    // ACDE
    adress=172;
    adress1=222;
    break;

    case '8':
    // FEFE
    adress=207;
    adress1=207;
    break;

    default:
    adress=0;
    adress1=0;
    break;
}
}

void enviar(char data[100],char db,
unsigned long L)
{
    ld= L + 10;

    dectohex(ld);

    Serial1.write(0x7E);

    Serial1.write(h4*16+h3); //Calcular
longitud en hexadecimal
    Serial1.write(h2*16+h1);

    Serial1.write((byte)0x00); //tipo de
trama de escritura

    Serial1.write((byte)0x01); //no
requiere respuesta del modulo

    Serial1.write((byte)0x00);
//Direccion destino
    Serial1.write((byte)0x00);
    Serial1.write((byte)0x00);
    Serial1.write((byte)0x00);
    Serial1.write((byte)0x00);
    Serial1.write((byte)0x00);

    Serial1.write((char)adress);
    Serial1.write((char)adress1);

    Serial1.write((byte)0x04); //opciones

    int z=L-3;
    for(k=0; k<=z; k++)
    {
        Serial1.write((char)data[k]);
//escribe el mensaje
        m1=m1 + data[k];
    }

    Serial1.write((byte)db); //escribe la
potencia

//calculo checksum
    long Checksum= 0x01 + adress +
adress1 + 0x04 + m1 + db;
    long cs=0xFF-(Checksum & 0xFF);

    dectohex(cs);
    Serial1.write((h2*16+h1)); //escribe
checksum en hexadecimal

    for(k=0; k<=z; k++)
    {
        m1='\0';
    }
    adress=0;
    adress1=0;
}

void dectohex(unsigned long dec)
{
    if(dec>=16)

```

```

{
    rt=dec/16;
    h1=dec-rt*16;
    if(rt>=16)
    {
        if(rt==16)
        {
            h2=1;
            h3=0;
            h4=0;
        }
        else
        {
            rt1=rt/16;
            h2=rt-rt1*16;
            if(rt1>=16)
            {
                if(rt1==16)
                {
                    h3=0;
                    h4=1;
                }
                else
                {
                    rt2=rt1/16;
                    h3=rt1-rt2*16;
                    if(rt2>=16)
                    {
                        if(rt2==16)
                        {
                            h4=1;
                        }
                    }
                }
            }
        }
    }
    else
    {
        rt3=rt2/16;
        h4=rt2-rt3*16;
    }
    else
    {
        h4=rt2;
    }
    else
    {
        rt2=rt1/16;
        h3=rt1-rt2*16;
    }
    else
    {
        rt1=rt/16;
        h2=rt-rt1*16;
    }
    else
    {
        h1=dec;
        h2=0;
        h3=0;
        h4=0;
    }
}

```

## ANEXO F: PROGRAMACIÓN DE LA RED FINAL IMPLEMENTADA EN ARDUINO

### PRUEBA 2: SENSOR/ACTUADOR

```

#include <LiquidCrystal.h>
//Inicializar los pines para la libreria
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
byte origen[100];

float datos1,datos11,datos12,pwm1,x;
float datos2,datos21,datos22,pwm2;
float datos3,datos31,datos32,pwm3;
float datos4,datos41,datos42,pwm4;
float datos5,datos51,datos52,pwm5;
float datos6,datos61,datos62,pwm6;
float datos7,datos71,datos72,pwm7;
float datos8,datos81,datos82,pwm8;
int j=0,k=0, adht=0, adlt=0, H=0;

char msj[100], msj1[100],msj2[100],
msj3[100], m1;
int dato, f=0, p=0, dir=0, dir1=0, dir2=0,
dir3=0, adl=0, adl1=0, adl2=0, adl3=0;
int m=0, n=0, l=0, g=0, L=0, adress=0,
adress1=0, adh=0, adh1=0, adh2=0,
adh3=0;
unsigned long dec=0, ld=0;
int rt=0, rt1=0, rt2=0, rt3=0, h1=0, h2=0,
h3=0, h4=0, rs=0;

void setup()
{
  // This code will only run once, after
  each powerup or reset of the board
  Serial.begin(9600);
  Serial1.begin(9600);
  //Serial2.begin(9600);
  lcd.begin(16, 2);
  lcd.print("PRUEBA RED 1");
}

void loop()
{
  if(Serial1.available() >=14)
  {
    if(Serial1.read()==0x7E)
    {
      for(int i=0; i<=12; i++)
      {
        origen[i]= Serial1.read();
        //Serial.print(origen[i]);
        //Serial.println();
      }

      if(origen[3]==222 &&
origen[4]==222)//DEDE
      {
        datos11=origen[10];
        datos12=origen[11];
        datos1=datos11*256.0+datos12;
        pwm1=datos1;
        datos1=datos1/1023.0*3.3;
        Serial.println();
        Serial.print("Sensor 1: ");
        Serial.print(datos1);
        Serial.println();
        Serial.println();
        Serial.print(pwm1);
        escribir_lcd(1,datos1,origen[5]);
        direccion(2);
        enviar(pwm1);
      }

      if(origen[3]==170 &&
origen[4]==170)//AAAA
      {
        datos21=origen[10];
        datos22=origen[11];
        datos2=datos21*256+datos22;
        pwm2=datos2;
        datos2=datos2/1023.0 *3.3;
        Serial.println();
        Serial.print("Sensor 2: ");
        Serial.print(datos2);
      }
    }
  }
}

```

```

        Serial.println();
        Serial.println();
        Serial.print(pwm2);
        escribir_lcd(2,datos2,origen[5]);
        direccion(3);
        enviar(pwm2);
    }

    if(origen[3]==171 &&
origen[4]==171)//ABAB
    {
        datos31=origen[10];
        datos32=origen[11];
        datos3=datos31*256+datos32;
        pwm3=datos3;
        datos3=datos3/1023.0 *3.3;
        Serial.println();
        Serial.print("Sensor 3: ");
        Serial.print(datos3);
        Serial.println();
        Serial.println();
        Serial.print(pwm3);
        escribir_lcd(3,datos3,origen[5]);
        direccion(1);
        enviar(pwm3);
    }

    if(origen[3]==171 &&
origen[4]==205)//ABCD
    {
        datos41=origen[10];
        datos42=origen[11];
        datos4=datos41*256.0+datos42;
        pwm4=datos4;
        datos4=datos4/1023.0*3.3;
        Serial.println();
        Serial.print("Sensor 4: ");
        Serial.print(datos4);
        Serial.println();
        Serial.println();
        Serial.print(pwm4);
        escribir_lcd(4,datos4,origen[5]);
        direccion(5);

        enviar(pwm4);
    }

    if(origen[3]==186 &&
origen[4]==186)//BABA
    {
        datos51=origen[10];
        datos52=origen[11];
        datos5=datos51*256.0+datos52;
        pwm5=datos5;
        datos5=datos5/1023.0*3.3;
        Serial.println();
        Serial.print("Sensor 5: ");
        Serial.print(datos5);
        Serial.println();
        Serial.println();
        Serial.print(pwm5);
        escribir_lcd(5,datos5,origen[5]);
        direccion(6);
        enviar(pwm5);
    }

    if(origen[3]==254 &&
origen[4]==254)//FEFE
    {
        datos61=origen[10];
        datos62=origen[11];
        datos6=datos61*256.0+datos62;
        pwm6=datos6;
        datos6=datos6/1023.0*3.3;
        Serial.println();
        Serial.print("Sensor 6: ");
        Serial.print(datos6);
        Serial.println();
        Serial.println();
        Serial.print(pwm6);
        escribir_lcd(6,datos6,origen[5]);
        direccion(7);
        enviar(pwm6);
    }

    if(origen[3]==172 &&
origen[4]==222)//ACDE
    {
        datos71=origen[10];

```

```

    datos72=origen[11];
    datos7=datos71*256.0+datos72;
    pwm7=datos7;
    datos7=datos7/1023.0*3.3;
    Serial.println();
    Serial.print("Sensor 7: ");
    Serial.print(datos7);
    Serial.println();
    Serial.println();
    Serial.print(pwm7);
    escribir_lcd(7,datos7,origen[5]);
    direccion(8);
    enviar(pwm7);
}

    lcd.print(nodo);
    lcd.setCursor(6,0);
    lcd.print(": ");

    lcd.setCursor(0, 1);
    lcd.print(dat);
    lcd.setCursor(4, 1);
    lcd.print('V');

    lcd.setCursor(8,1);
    lcd.print(-db,DEC);
    lcd.setCursor(11,1);
    lcd.print("dBm");
    delay(500);

}

if(origen[3]==207 &&
origen[4]==207)//CFCF
{
    datos81=origen[10];
    datos82=origen[11];
    datos8=datos81*256.0+datos82;
    pwm8=datos8;
    datos8=datos8/1023.0*3.3;
    Serial.println();
    Serial.print("Sensor 8: ");
    Serial.print(datos8);
    Serial.println();
    Serial.println();
    Serial.print(pwm8);
    escribir_lcd(8,datos8,origen[5]);
    direccion(1);
    enviar(pwm8);
}

}

}

void direccion(int ad)
{
    switch (ad) {
        case 1:
            // DEDE
            adress=222;
            adress1=222;

            adh=0;adh1=19;adh2=162;adh3=0;
            //0013A200
            adl=64;adl1=179;adl2=42;adl3=78;
            //40B32A4E
            break;

        case 2:
            // AAAA
            adress=170;
            adress1=170;

            adh=0;adh1=19;adh2=162;adh3=0;
            //0013A200
            adl=64;adl1=179;adl2=39;adl3=250;
            //40B327FA
            break;

        case 3:
    }
}

void escribir_lcd(int nodo,float dat,int
db)
{
    lcd.clear();
    lcd.home();
    lcd.print("Nodo ");
    lcd.setCursor(5,0);

```

```

// ABAB
adress=171;
adress1=171;

    adh=0;adh1=19;adh2=162;adh3=0;
//0013A200

adl=64;adl1=145;adl2=211;adl3=193;
//4091D3C1
    break;

case 4:
// ABCD
adress=171;
adress1=205;

    adh=0;adh1=19;adh2=162;adh3=0;
//0013A200

adl=64;adl1=145;adl2=211;adl3=185;
//4091D3B9
    break;

case 5:
// BABA
adress=186;
adress1=186;

    adh=0;adh1=19;adh2=162;adh3=0;
//0013A200
    adl=64;adl1=153;adl2=90;adl3=122;
//40995A7A
    break;

case 6:
// FEFE
adress=254;
adress1=254;

    adh=0;adh1=19;adh2=162;adh3=0;
//0013A200

adl=64;adl1=145;adl2=211;adl3=192;
//4091D3C0
    break;

case 7:
// ACDE
adress=172;
adress1=222;

    adh=0;adh1=19;adh2=162;adh3=0;
//NO DETERMINADO

adl=64;adl1=145;adl2=211;adl3=182;
//NO DETERMINADO
    break;

case 8:
// CFCF
adress=207;
adress1=207;

    adh=0;adh1=19;adh2=162;adh3=0;
//NO DETERMINADO
    adl=64;adl1=179;adl2=42;adl3=53;
//NO DETERMINADO
    break;

default:
adress=0;
adress1=0;
break;
}
}

void enviar(float x)
{
// Id= 17; //15 bytes mas la longitud del
mensaje - 1
//
// dectohex(ld);

//Serial.write(0x7E);
Serial1.write(0x7E);

```

```

//Serial.write(0x00); //CALcular la
longitud msb
Serial1.write(0x00);

//Serial.write(0x11); //CALcular la
longitud lsb
Serial1.write(0x11);

//Serial.write((byte)0x17); // tipo de
trama
Serial1.write((byte)0x17);

//Serial.write((byte)0x00); //no requiere
respuesta
Serial1.write((byte)0x00);

//Serial.write((char)adh); //Direccion sh
//Serial.write((char)adh1);
//Serial.write((char)adh2);
//Serial.write((char)adh3);

Serial1.write((char)adh); //Direccion sh
Serial1.write((char)adh1);
Serial1.write((char)adh2);
Serial1.write((char)adh3);

//Serial.write((char)adl); //Direccion sl
//Serial.write((char)adl1);
//Serial.write((char)adl2);
//Serial.write((char)adl3);

Serial1.write((char)adl); //Direccion sl
Serial1.write((char)adl1);
Serial1.write((char)adl2);
Serial1.write((char)adl3);

//modo broadcast ff ff
//Serial.write((char)adress);
Serial1.write((char)adress);

//Serial.write((char)adress1);
Serial1.write((char)adress1);

//Serial.write((byte)0x01); //opciones
aplicar cambios AT
Serial1.write((byte)0x02);

//Serial.write((char)77);
Serial1.write((char)77);

//Serial.write((char)48);
Serial1.write((char)48);

dectohex(x);

//Serial.write(h4*16+h3); //
Serial1.write(h4*16+h3);

//Serial.write(h2*16+h1); //
Serial1.write(h2*16+h1);

adht=adh+adh1+adh2+adh3;
adlt=adl+adl1+adl2+adl3;
H=h4*16+h3+h2*16+h1;
long Checksum=
0x17+0x00+adht+adlt+adress+adress1
+0x02+77+48+H;

long cs=0xFF-(Checksum & 0xFF);

dectohex(cs);
// //Serial.print(0xFF-(Checksum &
0xFF),HEX); //escribe checksum
//Serial.write((h2*16+h1));
Serial1.write((h2*16+h1));
}

void dectohex(unsigned long dec)
{
if(dec>=16)
{
rt=dec/16;
h1=dec-rt*16;
if(rt>=16)
{
if(rt==16)

```

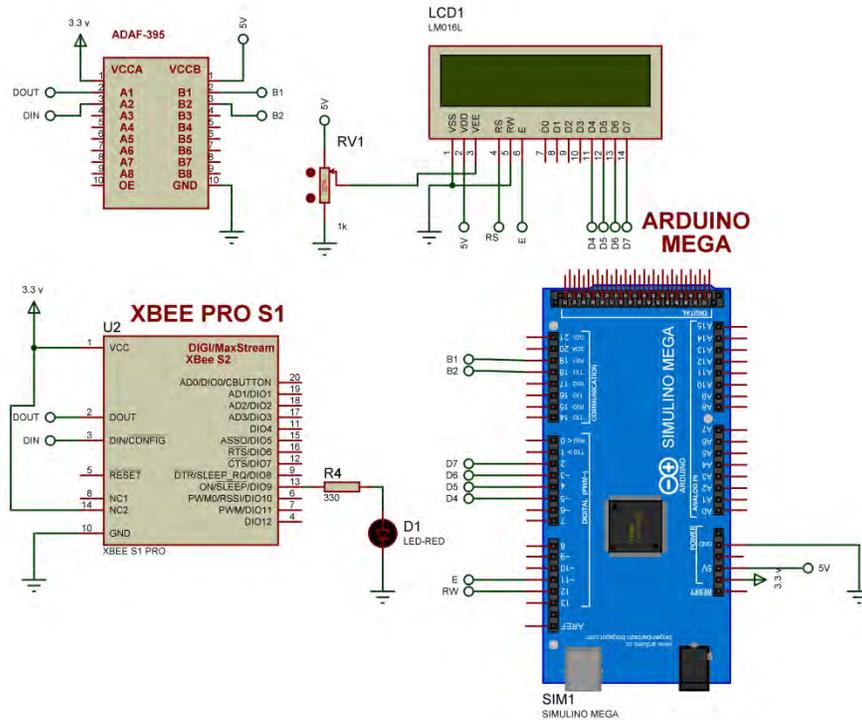
```

{
    h2=1;
    h3=0;
    h4=0;
}
else
{
    rt1=rt/16;
    h2=rt-rt1*16;
    if(rt1>=16)
    {
        if(rt1==16)
        {
            h3=0;
            h4=1;
        }
        else
        {
            rt2=rt1/16;
            h3=rt1-rt2*16;
            if(rt2>=16)
            {
                if(rt2==16)
                {
                    h4=1;
                }
                else
                {
                    rt3=rt2/16;
                    h4=rt2-rt3*16;
                }
            }
        }
    }
}
}
else
{
    h4=rt2;
}
}
else
{
    rt2=rt1/16;
    h3=rt1-rt2*16;
}
}
else
{
    rt1=rt/16;
    h2=rt-rt1*16;
}
}
else
{
    h1=dec;
    h2=0;
    h3=0;
    h4=0;
}
}
}

```

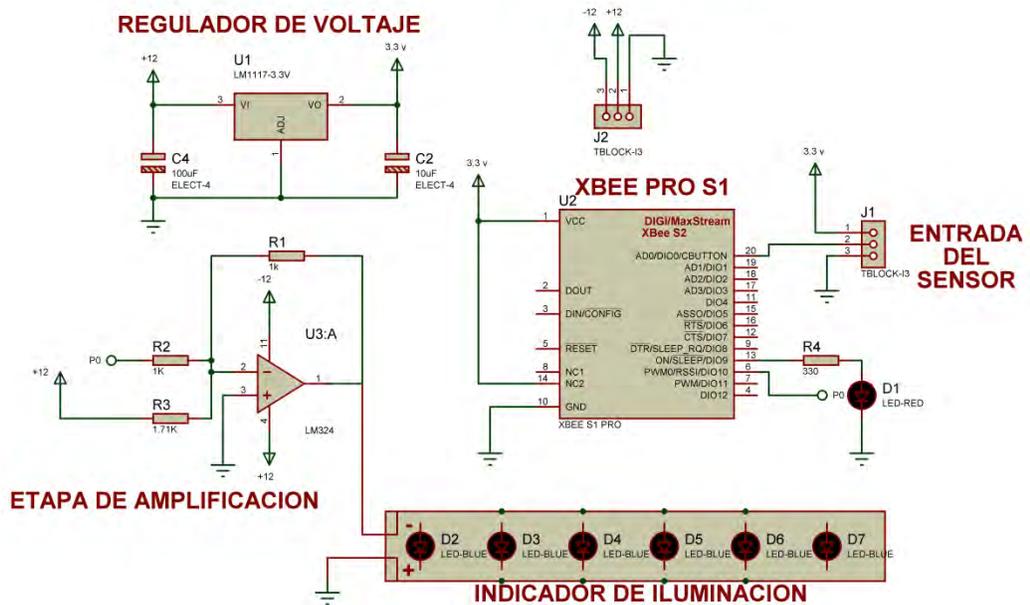
# ANEXO G: DIAGRAMAS ESQUEMATICOS NODO CENTRAL Y SENSOR

## DIAGRAMA ESQUEMÁTICO NODO CENTRAL



## DIAGRAMA ESQUEMÁTICO NODO SENSOR

### ALIMENTACION DEL NODO



**ANEXO G: ILUSTRACION CIRCUITOS IMPRESOS NODO CENTRAL Y  
SENSOR**

## ANEXO H: PRESUPUESTO DETALLADO DE LAS TARJETAS NODO CENTRAL Y SENSOR

### PRESUPUESTO NODO CENTRAL

DETALLE	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
Fabricación circuitos impresos	1	4.000	4.000
Headers 40 pines	2	500	1.000
Headers pin largo	12	1200	14.400
Led	1	100	100
Potenciómetro deslizante	1	700	700
Resistencia	1	25	25
Convertidor de nivel	1	38.000	38.000
LCD 16x2	1	12.000	12.000
Arduino Mega 2560	1	120.000	120.000
Módulo xbee pro s1	1	120.000	120.000
Base módulo xbee	1	9.000	9.000
Antena duck 2.4GHz	1	16.000	16.000
<b>TOTAL TARJETA NODO CENTRAL</b>		<b>\$ 335.225</b>	

### PRESUPUESTO NODO SENSOR

DETALLE	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
Fabricación circuitos impresos	1	4.000	4.000
Regulador LM1117 3.3v	1	3.000	3.000
Amp op LM324	1	700	700
Tira Leds 10 cm	1	1400	1400
Headers 40 pines	1	500	500
Base DIP 16	1	200	200
Bornera 3 pines	2	200	400
Conector SIL2	1	200	200
Interruptor	1	800	800
Led	2	100	200
Resistencias	5	25	125
Condensadores	2	150	300
Disipador	1	1000	1000
Potenciómetro	1	500	500
Cable metro	0.5	400	200
Módulo xbee pro s1	1	120.000	120.000
Base módulo xbee	1	9.000	9.000
Antena duck 2.4GHz	1	16.000	16.000
<b>TOTAL TARJETA NODO SENSOR</b>		<b>\$ 158.525</b>	

