

UACM

Universidad Autónoma
de la Ciudad de México

Nada humano me es ajeno

Colegio de Ciencia y Tecnología

Sistema de recolección
de datos para zonas remotas

T E S I S

Que para obtener el título de:
Licenciada en Ingeniería en Sistemas
Electrónicos y de Telecomunicaciones

P r e s e n t a:

Maricruz González Aguirre

D i r e c t o r a:

M. en C. Magali Cortez Vázquez

Ciudad de México, julio de 2021.

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS[©]

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

RESUMEN del trabajo recepcional de **Maricruz González Aguirre**, presentado como requisito parcial para la obtención del grado de **LICENCIADA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES**. Ciudad de México, julio 2021.

SISTEMA DE RECOLECCIÓN DE DATOS PARA ZONAS REMOTAS

En este trabajo recepcional se realiza el diseño e implementación de un sistema de recolección de datos que almacena la información recabada en un servidor web. La recolección de datos se realiza usando una red inalámbrica de sensores (RIS por sus siglas en español o WSN por sus siglas en inglés, *Wireless Sensor Network*), una red de área local (LAN, *Local Area Network*) y una red celular 3G. La WSN es implementada con tarjetas de desarrollo Arduino Uno y transceptores Xbee, basados en el protocolo Zigbee, mientras que la LAN es implementada con un *router* de la familia Cisco y las tarjetas de desarrollo Raspberry Pi 2 y Arduino Ethernet. Por último, la red celular se implementa con una tarjeta Arduino Uno y una tarjeta de comunicación GPRS SIM 900. Lo más relevante de este trabajo es la implementación de un sistema que puede usarse en zonas alejadas y que requieren de la recolección de diversos datos como son temperatura y humedad, entre otros. Así, se recaba dicha información y se envía a un nodo concentrador. Luego, el nodo concentrador puede comunicarse con el sitio más próximo con servicio de Internet, enviar esos datos al servidor web y, mediante tecnología celular, dar aviso de algún dato de alarma a cualquier lugar donde se requiera de la información.

En este documento se presenta el diseño, la implementación, pruebas de desempeño del sistema y un análisis de sus posibles usos en plantaciones o cultivos.

Palabras clave: Wireless Sensor Network, LAN, Zigbee y tecnología celular.

Agradecimientos

A la Santísima Trinidad y Virgen de Guadalupe, así como a mi amada familia directa; papis y hermanas, por su generoso amor incondicional.

A mis Maestras (os) pedagogos por convicción, especialmente a mi directora de tesis M. en C. Magali Cortez, por ser una extraordinaria Maestra. Impactó en mi formación de manera positiva, incitando al aprendizaje continuo y eficaz. De manera similar, agradezco a mis lectores por el tiempo otorgado, por enriquecer mi trabajo y mi vida. La sabiduría adquirida durante toda mi estancia en la UACM desarrolló en mí, diversas habilidades, no sólo para la licenciatura sino para la vida misma, por ello, me apasiona indagar y resolver problemas de cualquier índole.

A la Universidad Autónoma de la Ciudad de México por el apoyo con las becas: estudios de licenciatura en mis últimos 2 semestres, elaboración de trabajo recepcional y para la impresión y/o empastado de trabajo recepcional o tesis. Así mismo, agradezco a quien apoyó a admirable proyecto educativo.

A mí por ser una mujer única, que disfruta de la vida sanamente, de su familia, y que le apasiona el aprender. Por hacer lo que quiere cuando lo desea. Por tratar de dar el mejor ejemplo de vida, a la familia, los amigos, los compañeros de la escuela y del trabajo, los vecinos y así, contribuir a mi colonia, alcaldía, ciudad y por qué no, al país.

A ti que lees esta tesis te quiero decir que nunca es tarde para hacer lo que desees, uno define sus prioridades y actúa con base en ellas. Todo es posible, sólo inhala, exhala, sonríe, sé constante, disfruta de cada momento y cuando veas, estarás por consecuencia viviendo una maravillosa vida sin remordimientos.

Dedicatoria

A mi adorable familia: mi mami María del Carmen Aguirre, mi papi Cruz González, mi hermana Esme con mi dulce corazoncito Quetzalli y mi “hermanita” Isa con la mini Dara.

Contenido

RESUMEN.....	II
Agradecimientos	III
Dedicatoria.....	IV
Capítulo I Introducción	1
1.1 Introducción.....	1
1.2 Motivación del proyecto	2
1.3 Descripción y propuesta de solución del problema	3
1.4 Objetivos.....	4
Objetivo general	4
Objetivos específicos	4
1.5 Metodología	4
Capítulo II Marco Teórico.....	6
2 Tecnologías de comunicación.....	6
2.1 Redes de sensores	6
2.1.1 Topología.....	6
2.1.1.1 Nodo sensor	7
2.1.1.2 Nodo concentrador	8
2.1.2 Hardware para redes de sensores	8
2.1.3 Software para redes de sensores	9
2.1.3.1 Software para configurar los nodos	9
2.1.4 Protocolos de comunicación	12
2.1.4.1 La especificación Zigbee	13
2.1.4.2 Encaminamiento en redes de malla.....	16

2.1.4.3	Tramas para la recolección de datos	17
2.2	Redes de área local	20
2.2.1	Topología	20
2.2.2	Hardware para LAN	21
2.2.3	Estándar IEEE 802.3	21
2.2.4	Protocolo de Internet.....	21
2.3	Sistemas de telefonía celular	23
2.3.1	Tecnología 3G	24
2.3.2	Arquitectura	25
2.3.3	Servicio de mensajes en 3G	27
2.4	Subsistema de almacenamiento de datos.....	27
2.4.1	Necesidades técnicas del subsistema de almacenamiento de datos	27
2.4.2	Software para el subsistema de almacenamiento de datos.....	29
2.4.2.1	LAMP	29
2.4.2.2	Linux-Sistema Operativo.....	29
2.4.2.3	Apache – Servidor Web	30
2.4.2.4	MySQL – Gestor de Base de Datos	31
2.4.2.5	PHP – Lenguajes de programación	32
Capítulo III.	Implementación del sistema de recolección de datos	33
3.1.	Red de sensores.....	35
3.1.1	Nodo sensor	36
3.1.2	Nodo concentrador	40
3.1.3	Nodo router en modo alertador	42
3.2.	Transmisión de datos en la red de sensores.....	45
3.2.1	Escenarios de prueba para la transmisión de datos.....	48

3.3 Subsistema de almacenamiento de datos.....	53
3.3.1 Servidor web.....	53
3.3.2 Almacenamiento en la base de datos	54
3.4 Costo del sistema de recolección.....	60
Capítulo IV. Pruebas de funcionamiento del sistema completo	61
4.1 Pruebas realizadas	61
4.2 Análisis de resultados	65
4.2.1 Conectividad.....	65
4.2.2 Subsistema de almacenamiento	67
4.3 Factibilidad para su implementación para su uso en plantaciones o cultivos	67
4.3.1 Caso de ejemplo 1: tierra de cultivo	70
4.2.4 Caso de ejemplo 2: incendio en un bosque	71
Conclusiones	73
Anexo A.....	75
Códigos en Arduino del nodo concentrador y nodo alertador.....	75
Código del nodo concentrador	75
Código del nodo alertador para realizar el envío de un SMS y una llamada	79
Referencias.....	81

Índice de tablas

TABLA I. Tramas utilizadas.....	19
TABLA II. Estructura de la trama API[16]	19
TABLA III. Estructura y tipos de servidor.....	29
TABLA IV. Dirección de cada elemento de la RIS.....	35
TABLA V. Configuración de los módulos End Device	39

Tabla VI. Parámetros del nodo Xbee Coordinador	41
TABLA VII. Configuración del Router 1	43
TABLA VIII. Estructura de tramas para encendido y apagado de LED.....	48
TABLA IX. ID de interés para los nodos sensor	53
Tabla X. Instalación de servidor tipo LAMP.....	53
Tabla XI. Direcciones IP de los elementos de la LAN	56
Tabla XII. Costo del material utilizado en este proyecto.....	60
Tabla XIII. Velocidad de transmisión de las tramas utilizadas	66
Tabla XIV. Costo del sistema de monitoreo para caso de ejemplo 1	70
Tabla XV. Costo del sistema de monitoreo para caso de ejemplo 2.....	71

Índice de figuras

Figura 1. Topología: Malla.	7
Figura 2. Modo consola en XCTU.	10
Figura 3. Modo red en XCTU, calidad de conexión entre radios.	11
Figura 4. Modo de red en XCTU estados de los radios según el color.	11
Figura 5. Modo red en tabla.	12
Figura 6. Protocolo IEEE 802.15.4 y Zigbee.	13
Figura 7. Equipos para Zigbee e IEEE 802.15.4.	15
Figura 8. Topologías de red para Zigbee.	16
Figura 9. Red LAN para el subsistema de almacenamiento de datos.	23
Figura 10. Técnicas TDMA y CDMA [23].	25
Figura 11. Arquitectura de una red UMTS [24].	26
Figura 12. Sistema de recolección.	34
Figura 13. RIS.....	35

Figura 14. Representación gráfica del nodo sensor.	36
Figura 15. Nodo sensor ED1 real.	36
Figura 16. Nodo sensor ED2 real.	36
Figura 17. Calibración del sensor de humedad.	37
Figura 18. Configuración de la velocidad de muestreo del módulo Xbee ED1.	40
Figura 19. Nodo concentrador.	41
Figura 20. Esquema gráfico del nodo alertador.	42
Figura 21. Nodo alertador real.	42
Figura 22. Envío de mensaje exitoso por parte del <i>router</i> alertador.	44
Figura 23. Llamada realizada por el <i>router</i> alertador.	45
Figura 24. Envío de instrucción para encendido de LED con trama solicitud de comando remoto 0x17.	46
Figura 25. Orden de encendido de un LED por parte del coordinador. (a) Orden de encendido de LED, (b) respuesta de encendido de LED, (c) orden de apagado de LED, (d) respuesta de apagado de LED.	47
Figura 26. Ejemplo ilustrativo de intercambio de tramas	47
Figura 27. Escenario uno: sin línea de vista en el interior de un edificio.	49
Figura 28. Escenario dos con línea de vista.	50
Figura 29. Escenario dos con línea de vista en la mayoría de los nodos.	51
Figura 30. Escenario tres: sin línea de vista en exteriores.	51
Figura 31. RIS óptima con 1 coordinador, 4 <i>router</i> y 2 ED.	52
Figura 32. LAN para el envío de datos al servidor web.	54
Figura 33. Creación de WLAN con Arduino Yún.	55
Figura 34. Arquitectura de la red LAN, Conexión del servidor web vía Wifi.	56
Figura 35. Creación de la base de datos. (a) Nombre de la base de datos: sensor, (b) Tabla de tamaño 4 con nombre: VALORES.	57

Figura 36. Archivo <code>iot.php</code>	58
Figura 37. Carpeta con dos archivos: <code>iot.php</code> y <code>config.php</code>	58
Figura 38. Envío de datos del nodo coordinador al servidor web con método GET ..	58
Figura 39. Prueba de funcionamiento de la base de datos: incrustación de datos vía URL.	59
Figura 40. Datos guardados en la base de datos.	59
Figura 41. Prueba de inserción de datos vía URL.	60
Figura 42. RIS utilizada para probar el funcionamiento del sistema completo.	61
Figura 43. Registro en la base de datos de temperatura y humedad.	62
Figura 44. Registro de elevación de temperatura por ID 37 y cambio en valor de humedad de ID 43.	62
Figura 45. Registro en la base de datos de temperatura elevada.	63
Figura 46. Teléfono inteligente con registro. (A) Mensaje SMS (B) Llamada.	63
Figura 47. Consulta de la base de datos desde un teléfono inteligente.	64
Figura 48. Ejemplo de zona aledaña a la Ciudad de México.	68
Figura 49. Zona comunal con oficina y área de cultivo.	69

Capítulo I Introducción

En este proyecto se pretende implementar una red heterogénea que permita la convivencia de varias tecnologías inalámbricas para la recolección de información y su envío a un nodo de concentración, permitiendo así, el uso de dicho sistema en zonas donde la tecnología de comunicación inalámbrica de infraestructura no está disponible.

1.1 Introducción

Actualmente, existen diversas tecnologías inalámbricas, como las redes inalámbricas de sensores (RIS), las redes inalámbricas de área personal, las redes de tecnología celular, por mencionar algunas, que proporcionan diversos servicios. Por ejemplo, las redes inalámbricas de sensores se utilizan en gran parte para la supervisión y monitoreo de campos agrícolas, o para aplicaciones del sector militar, la geofísica, entre otros. Las RIS están formadas por equipos con mínimo gasto de energía y de reducidas dimensiones. Estas redes permiten adquirir y ceder información del sitio donde están distribuidas, mediante la comunicación inalámbrica a varios saltos usando los nodos que conforman la red. De esta manera, aunque el rango de comunicación de un nodo es pequeño puede comunicarse a grandes distancias a través de sus nodos vecinos más próximos. Por otro lado, las redes inalámbricas ya sea de área de vecinos (WNAN, Wireless Neighborhood Area Network), área local (WLAN, Wireless Local Area Network) o área personal (WPAN, Wireless Personal Area Network) son muy utilizadas en áreas pequeñas, ya que facilitan el intercambio de información entre equipos. Ejemplos de estas tecnologías son Wifi, Bluetooth y Zigbee. Por su lado, WiMAX unido con las tecnologías celulares intentan cubrir áreas metropolitanas de mayor cabida [1].

A pesar de que la tecnología celular se proyectó únicamente para voz, debido a que en esa época se tenían limitaciones tecnológicas, la tecnología celular actual es capaz de ofrecer otro tipo de servicios tales como video, audio y datos. Sin embargo, este tipo de servicios requieren cada vez de mayor ancho de banda y es un servicio que no está disponible generalmente en comunidades rurales o alejadas de las manchas citadinas [2]. No obstante, su utilización es primordial en estos días, por

ello, su evolución es constante. En el caso de la velocidad de transmisión, por ejemplo, avanzó de 10 Mbps para 3G a 100 Mbps para 4G [3].

Por otra parte, aunque en la Ciudad de México las zonas verdes son escasas, se practican en ellas trabajos agropecuarios, de cultivo, reforestación, chaponeo, entre otras, sin contar con servicios de energía eléctrica pública. En este sentido, es importante mejorar su uso y productividad con la introducción de mejoras tecnológicas, dado que son los últimos pulmones de la Ciudad, y de eso depende tanto la subsistencia de agricultores como del bosque en sí. Por ello, la implementación de tecnología inalámbrica de consumo muy bajo de energía es una alternativa viable para el propósito mencionado. Por ejemplo, el diseño de un sistema que incorpore una estación que permita la gestión y el envío de datos específicos recabados, todo de forma inalámbrica, permitiría un monitoreo más eficaz de las zonas, lo que representaría un ahorro en comparación con un sistema cableado. Además, dado que la recolección de datos se haría de manera inalámbrica también reduciría el personal necesario para supervisar este tipo de zonas. Un sistema como el anterior puede brindar diversidad de aplicaciones, por ejemplo, aplicaciones en las cuales se requiera monitorear hortalizas o tierras de cultivo, en las que se pueden obtener ciertos datos como temperatura, humedad, entre otros. Estos datos se pueden recolectar de manera dinámica por drones, o de manera estática, reenviándola hasta un nodo concentrador.

1.2 Motivación del proyecto

Debido a la falta de comunicación en zonas remotas y al avance tecnológico que se tiene actualmente, es importante ayudar a dichas zonas para el uso de sus recursos naturales y facilitar sus labores de trabajo. Esto último, ya que la mayoría de las personas encargadas de las zonas aledañas de la Ciudad de México, los comuneros o ejidatarios, son adultos mayores que presentan dificultades para desplazarse por el bosque. Es por ello, que se pensó en utilizar una RIS para ayudar al monitoreo de las áreas de cultivo para disminuir el desplazamiento físico de las personas a cargo, evitar la aparición de plagas y de ser posible, prevenir grandes incendios.

1.3 Descripción y propuesta de solución del problema

Como se mencionó anteriormente, las redes inalámbricas son una de las principales tecnologías de comunicación, sin embargo, este tipo de comunicación es limitado en ciertas zonas, por ejemplo, zonas rurales, ejidales, comunales, entre otras [2]. Por ejemplo, la tecnología celular requiere cierta infraestructura cuya instalación no resulta viable para los prestadores de dichos servicios dado que la demanda es baja en estas zonas; entonces resulta ser un gasto y no una inversión para ellos. Por otro lado, las redes Wifi, aunque son más asequibles tienen un radio de cobertura más limitado; mientras que las redes de sensores, aunque tienen un radio de cobertura mucho más amplio requieren de cierto nivel de conocimiento para su configuración, lo que limita su uso. Lo anterior confirma que dichas regiones quedan limitadas para percibir algún tipo de servicio de comunicación.

Por ello, se plantea usar tecnologías de comunicación inalámbrica para monitorear áreas de cultivo de zonas rurales, ejidales o comunales, o donde la comunicación es limitada o está ausente. Cabe mencionar que las redes inalámbricas se fundamentan en un bajo consumo de energía, capaces de conseguir información de su entorno, procesarla en sitio, y transmitirla mediante enlaces inalámbricos hasta un nodo coordinador central [1], lo anterior usando alimentación por baterías.

La estrategia de recolección de datos que se plantea reúne diferentes tecnologías de comunicación como son: Ethernet, la especificación Zigbee y la tecnología celular, lo que posibilita optimizar las ventajas de cada una de ellas de manera conjunta. Por otro lado, es un sistema que ayudaría de manera significativa a zonas de conservación que están actualmente incomunicadas.

La propuesta del presente trabajo pretende evitar el trabajo exhaustivo de humanos en áreas verdes para monitorear y administrar el sistema, puesto que el sistema de recolección de datos de una red inalámbrica es auto administrable.

1.4 Objetivos

Objetivo general

Diseñar, implementar y probar una red inalámbrica heterogénea que use las tecnologías necesarias para recolectar datos de una zona geográfica de difícil acceso, que permita su envío a un lugar determinado.

Objetivos específicos

- a) Interconectar varias tecnologías que permitan la transmisión y el almacenamiento de datos que sean recolectados de sitios remotos a fin de asegurar su resguardo y consulta posteriores.
- b) Generar un sistema de alerta que se active cuando en la recolección de datos se detecte un dato fuera del umbral deseado, que sea capaz de enviar un mensaje SMS y realizar una llamada al administrador de la red.
- c) Validar el uso de la red inalámbrica heterogénea.

1.5 Metodología

La metodología propuesta para el desarrollo del presente trabajo se divide en 3 etapas. La primera etapa involucra una revisión bibliográfica de redes de sensores, redes LAN, servidores web y su interconexión; así como el hardware y software existente en la UACM-SLT. La segunda etapa consta de la implementación de la RIS, posteriormente el envío de datos a través de una red cableada y el almacenamiento en un servidor web. La tercera y última etapa involucra las pruebas de desempeño del sistema de recolección de datos completo en dos escenarios: un ambiente urbano y uno semejante a una zona remota, así como el análisis de los resultados para verificar la viabilidad del uso del sistema en zonas de cultivo.

Por otro lado, para determinar la infraestructura de telecomunicaciones necesaria para el sistema de recolección de datos para una zona remota se consideró lo siguiente:

- ✓ Verificar del estado actual del sitio remoto a monitorear, esto es, la existencia de al menos un inmueble con suministro eléctrico, la distancia de las tierras de

cultivo al inmueble, así como la superficie a monitorear y el hardware existente.

- ✓ Considerar el capital disponible
- ✓ Determinar el dominio de conocimiento y manejo de las redes inalámbricas de sensores y de las redes de área local.
- ✓ Establecer qué es lo que se desea monitorear para definir los sensores a utilizar y la cantidad.
- ✓ Determinar las necesidades de almacenamiento

Con base en ello se propone la solución a la problemática planteada.

Capítulo II Marco Teórico

En este capítulo se explica la teoría, hardware y software utilizados para desarrollar el proyecto. Se parte con las tecnologías de comunicación, elementos que las conforman y su funcionamiento, así como los protocolos que rigen dichas tecnologías. Se continúa con la red de área local, conformación, topología y estándar empleado. Posteriormente, se expone el funcionamiento del sistema de telefonía celular, su tecnología y utilidad en modo alarma para el presente trabajo. Finalmente se describe el subsistema para realizar el almacenamiento de datos.

2 Tecnologías de comunicación

2.1 Redes de sensores

Las redes de sensores son utilizadas actualmente dado su mínimo consumo de energía, su capacidad de procesamiento y el avance en el desarrollo de hardware con microcontroladores de tamaño pequeño. Lo anterior, unido con las nuevas tecnologías inalámbricas, da como efecto la capacidad de extender diversos nodos para conformar redes de sensores enfocadas en diferentes ámbitos, para lograr recolectar datos y efectuar operaciones. Las peculiaridades más significativas de las redes de sensores son: la baja tasa de transferencia, la alimentación por baterías, escalabilidad y gran cantidad de nodos que se pueden conectar [4]. La tecnología utilizada para la Red de Sensores en este trabajo fue Zigbee, dado que es una tecnología diseñada para desarrollar redes de tipo malla, además de que se cuenta con dispositivos con dicha tecnología en la UACM y que más adelante se describirá. En los siguientes párrafos se explica la topología de la red de sensores, hardware, software, protocolos de comunicación y la especificación de Zigbee.

2.1.1 Topología

Una red inalámbrica de sensores (RIS) está compuesta por nodos sensores que interactúan entre sí, permitiendo monitorizar parámetros físicos en un entorno determinado y entregando dicha información a nodos coordinadores [5]. La topología de la red de sensores establecida en este trabajo es de tipo malla con comunicación Ad-Hoc. En la figura 1 se muestra un ejemplo gráfico de la tecnología de malla, en

ella, se puede ver la conexión que existe entre nodos para garantizar la comunicación entre sí.

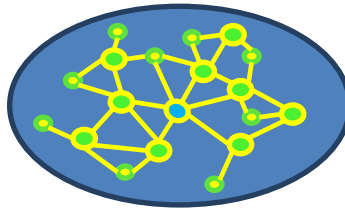


Figura 1. Topología: Malla.

2.1.1.1 Nodo sensor

El nodo sensor es el elemento final de la red, son unidades pequeñas conformadas por un microcontrolador, uno o varios sensores, un módulo de energía y un módulo de comunicación. Dichos nodos tienen la función de recolectar y transmitir los datos a un nodo concentrador. Los nodos son elementos repartidos espacialmente, que emplean sensores para capturar diversas condiciones en diferentes puntos, como son: la presión, la humedad, la vibración, el movimiento, la temperatura y el sonido [6]. Su configuración efectúa dos labores primordiales: la primera, adquirir la lectura del sensor, es decir, detectar el valor de voltaje que representa el valor del parámetro que está detectando el sensor; la segunda, digitalizar, empaquetar y transmitir dicha lectura a un nodo coordinador siempre que éste lo requiera [7].

En este trabajo se utilizan dos sensores por cada nodo sensor, uno de humedad y otro de temperatura, ambos analógicos.

El sensor de humedad utilizado es compatible con Arduino con número de parte MD0021, es un sensor simple y fácil de usar para probar la humedad del suelo. Se emplea el convertidor analógico digital (CAD), el cual ofrece un rango de salida digital para la humedad que va desde la magnitud 0 para la ausencia de agua hasta valores mayores a 111 cuando existe gran cantidad de agua en el suelo. Al existir escasez de agua en la tierra arroja valores cada vez más pequeños, mientras que, al tener agua, los valores arrojados serán cada vez mayores.

En el caso del sensor de temperatura se empleó el LM35, soporta temperaturas desde -55 °C hasta 150 °C. Para indicar la temperatura se emplea el CAD, el cual

cuenta con 1024 valores que oscilan de 0 a 1023, los cuales se referencian a un nivel de voltaje.

2.1.1.2 Nodo concentrador

El nodo concentrador es de real importancia puesto que es el encomendado de inicializar la red, intervenir en el encaminamiento de paquetes, establecer el canal de comunicación y ser el origen y/o receptor de la información, además de enviar todos los datos recabados a un servidor [7]. El nodo concentrador es implementado en el presente proyecto con tecnología Zigbee y una tarjeta de desarrollo para el procesamiento. Cabe señalar que en la topología Zigbee el nodo concentrador se configura en modo coordinador por las funciones que desarrolla en la red de sensores.

2.1.2 Hardware para redes de sensores

Existe variedad de hardware para la implementación de las redes de sensores. Por ejemplo, para implementar la comunicación inalámbrica de los nodos de la red existen módulos como: Xbee de Digi, CC2430 de Texas Instruments, MC13213 de Freescale, MICA Z de Crossbow, MR F24J4OMA de Microchip, Tmote Sky, Jn5121 de Jennic e IRIS 2.4 de Crossbow [8]. Para implementar el procesamiento en el nodo sensor se pueden utilizar diversas tarjetas de desarrollo como: Arduino, Raspberry, Intel Edison, NodeMCU, entre otros. Además, en el mercado se pueden encontrar diversidad de sensores según el entorno a monitorear. El hardware utilizado para redes de sensores del presente trabajo queda regido por dos tarjetas de desarrollo: la Arduino Uno para los nodos sensor y la Arduino Ethernet para el nodo concentrador. Este tipo de nodos cuentan con comunicación inalámbrica usando los módulos Xbee Serie 2 de Digi en modo Coordinador (C), Router (R) y End Device (ED). La alimentación de los nodos sensor fue atendida por baterías de 5 V y por vía USB para el nodo coordinador. Asimismo, los sensores empleados son analógicos, miden humedad y temperatura.

2.1.3 Software para redes de sensores

El software para redes de sensores es muy amplio dado que, la oferta de sistemas operativos involucra a una diversidad de microprocesadores. Existen sistemas operativos de código abierto disponibles como: TinyOS, FreeRTOS y Contiki, por mencionar los que gozan de una mayor aceptación [5]. Por otro lado, también existe software que únicamente nos permite configurar a los nodos sin hacer mayor cambio en su funcionamiento, a diferencia de las funcionalidades con un sistema operativo.

2.1.3.1 Software para configurar los nodos

El software utilizado para la configuración de los módulos Xbee es XCTU. Se trata de una aplicación multiplataforma gratuita que provee una interfaz gráfica para la interacción y configuración con los módulos de radiofrecuencia del fabricante DIGI, contiene una terminal con la cual se pueden mandar y recibir datos por medio del puerto al que está conectado el radio Xbee [1].

En XCTU versión 6.3.8.2 se tienen tres maneras de trabajo: configuración, consola y red. Dependiendo de la tarea a realizar serán las opciones que se utilicen. En nuestro caso, empleamos las tres. La más utilizada fue configuración para la conformación de la red. La de consola funcionó para ver el comportamiento entre radios, esto es, el envío de mensajes para la entrega y recepción de paquetes, con ello, se pudo ver cuándo la entrega era exitosa o un intento fallido. Por ejemplo, cuando se envió una trama API con la orden de encendido de LED de un radio específico a otro, se pudo corroborar la entrega vía consola (ver figura 2). En dicha figura se puede ver en primera instancia del lado superior izquierdo (Radio Modules), el radio activado como Coordinador conectado al puerto COM4 del ordenador, posterior a ello, se observa la activación del modo consola, representado por el ícono con nombre *Close* en color verde. Justo debajo del ícono de consola se observan las tramas (*Frames Log*) enviadas en color azul y las tramas recibidas en color rojo. En la sección *Frames Details* se puede ver a detalle cada trama, fue en esta sección que se corroboró la entrega exitosa de mensajes.



Figura 2. Modo consola en XCTU.

Finalmente, el modo de trabajo de red se empleó para visualizar la red en general, los enlaces y la calidad de comunicación entre radios. En la figura 3 se muestra cómo se ve en XCTU el modo de trabajo de red. Un enlace con alta calidad de conexión bilateral entre radios se indica con una etiqueta de valor 255 con línea color verde, el valor disminuirá conforme sea más débil la calidad de conexión y el color de la línea también cambiará, dicho valor se muestra junto a la línea que conecta a los radios. Sabiendo esto, en la figura 3 se aprecia una red formada por un Coordinador, un Router y un End Device, en dicha red se ve un enlace con calidad alta de 255/255 bilateral en color verde entre el coordinador y el *router*. También se muestra un registro de enlace entre el coordinador y el End Device en color gris, en este caso es de tal color porque desconecté intencionalmente al End Device para marcar la diferencia en la calidad de conexión.

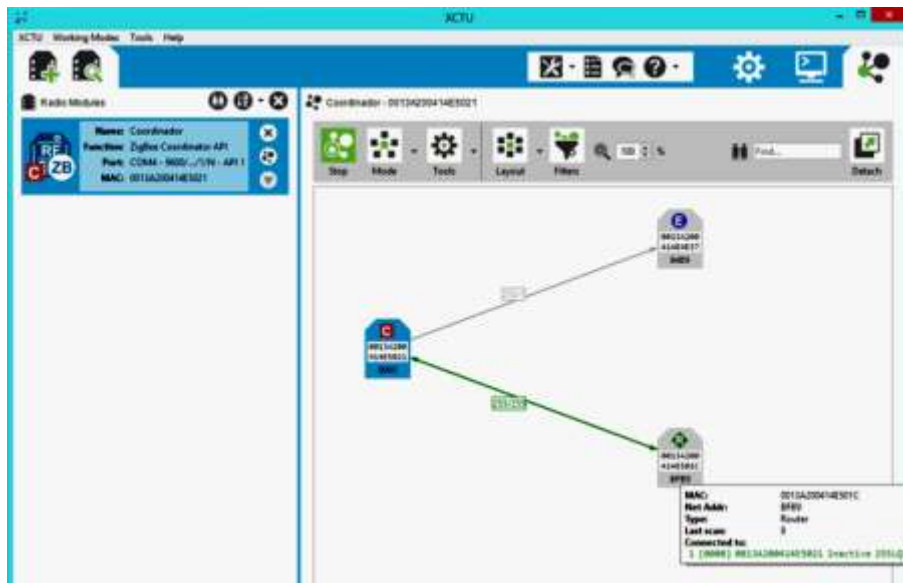


Figura 3. Modo red en XCTU, calidad de conexión entre radios.

El radio seleccionado siempre será de tono azul, un radio conectado en red y activo será de tono verde, un radio en tono gris representa una conexión por descubrir y rojo cuando el radio ha abandonado la red, ver figura 4.

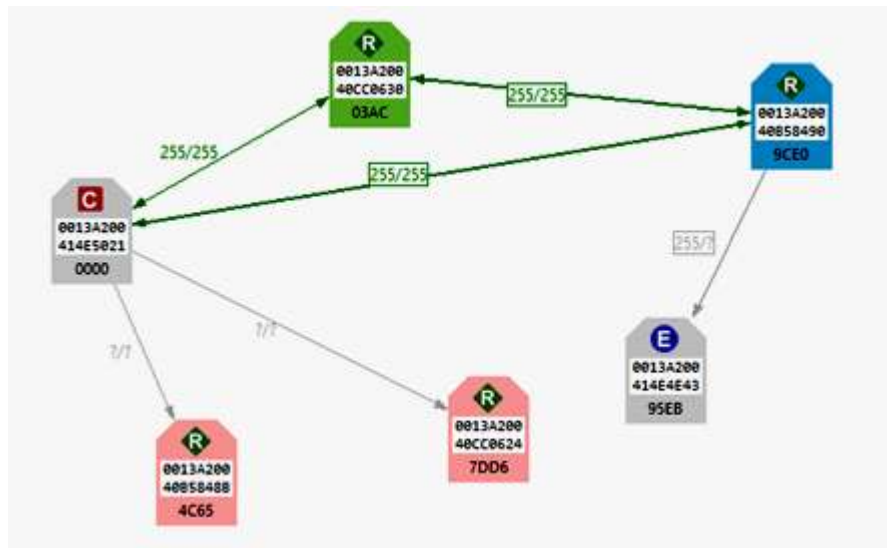


Figura 4. Modo de red en XCTU estados de los radios según el color.

Se puede ver en modo tabla la red formada y, al seleccionar al elemento deseado, se podrá ver qué radios están conectados a él y el valor de la calidad de conexión. En la figura 5 se muestran los radios conectados al radio Coordinador seleccionado, se muestran un Router y un End Device activos con alta calidad de conexión.

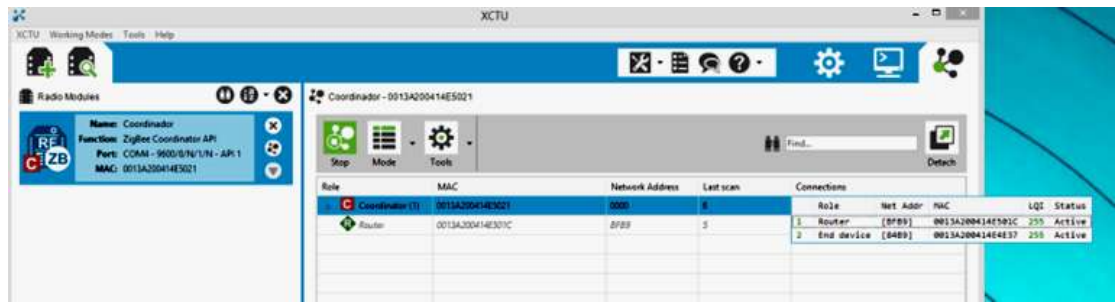


Figura 5. Modo red en tabla.

Por otro lado, se utiliza el software de Arduino para la configuración de la tarjeta de desarrollo Arduino Uno y de Arduino Ethernet, que forman parte del nodo sensor. Arduino es una plataforma de prototipos de hardware de código abierto compatible con un software también de código abierto. Se introdujo por primera vez en 2005 y se diseñó con el objetivo de hacer que el hardware y software resulte fácil de usar y esté disponible para la mayor audiencia posible [9].

2.1.4 Protocolos de comunicación

Los protocolos de comunicación son los que definen el conjunto de reglas y el lenguaje que facilitan la comunicación entre un emisor y un receptor, con la finalidad de que puedan intercambiar información y entenderse. Lo anterior, surge a partir de que varios fabricantes crean sus nodos para ser usados con protocolos de comunicación estandarizados por alguna organización específica o propietarios; por ello, los protocolos de comunicación resultan ser una parte fundamental en las redes de sensores [10].

En nuestro caso, nos interesamos por la especificación de Zigbee que resulta del vínculo de desarrollos basados en el protocolo IEEE 802.15.4, dado que estos logran la compatibilidad entre diferentes fabricantes y posibilitan un mínimo consumo de energía. IEEE 802.15.4 define las especificaciones de la capa física (PHY) y la subcapa de control de acceso al medio (MAC), de equipos fijos, móviles y portátiles que se comunican de modo inalámbrico, por medio de transmisiones que exijan poca tasa de transferencia de datos. IEEE 802.15.4 permite dos topologías de red, *peer to peer* o estrella [8]. Zigbee en cambio, define además de las 2 capas antes mencionadas la capa de red (NWK) y la de aplicación (APL), véase figura 6.

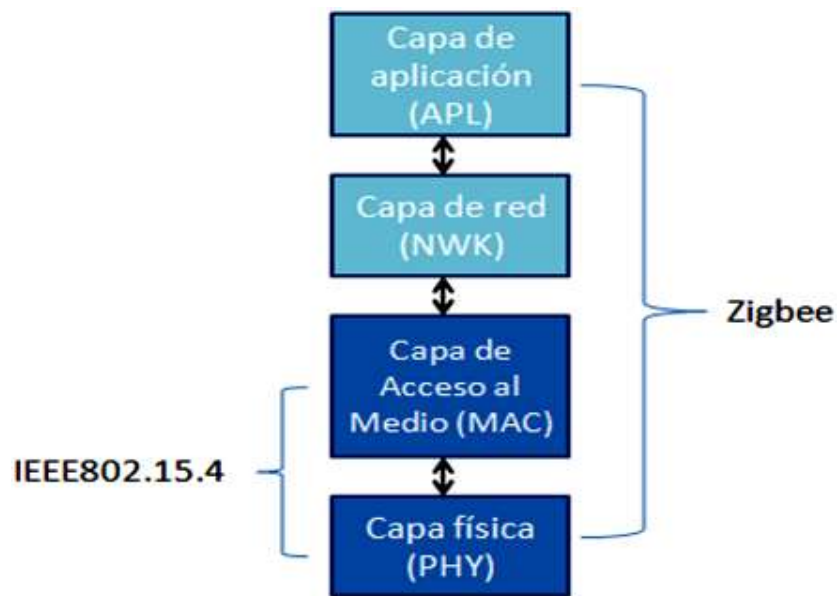


Figura 6. Protocolo IEEE 802.15.4 y Zigbee.

2.1.4.1 La especificación Zigbee

Zigbee es un estándar creado por la Alianza Zigbee conformada por diversas compañías de fabricantes de semiconductores y desarrolladores de software, incluso instaladores y fabricantes de equipos originales (OEM, Original Equipment Manufacture). Asimismo, ZigBee es una especificación que determina un conjunto de protocolos para la integración de las redes inalámbricas de baja velocidad de datos y longitud corta. Opera en las bandas de 868 MHz, 915 MHz y 2.4 GHz y puede transferir datos hasta 250 Kbps [11]. Los elementos pueden ser alimentados por baterías AA; dichos elementos permanecen en estado latente la mayor parte del tiempo, esto es, durmiendo para gastar menor cantidad de energía.

ZigBee se desarrolló con las siguientes características:

- Ultra bajo consumo energético que permita utilizar equipos a batería
- Bajo costo de equipos, mantenimiento e instalación de ellos
- Alcance corto menor a 50 metros
- Ciclo efectivo de transmisión optimizado menor a 0.1 %
- Velocidad de transmisión menor que 250 kbps. Típica: menor que 20 kbps

De acuerdo con su área de cobertura Zigbee, dadas sus características, es una red inalámbrica de área de vecindario WNAN.

Las redes WPAN, no se pensaron para reemplazar a un equipo cableado, más bien para abastecer una comunicación en el espacio operativo personal (POS *Personal*

Operating Space) sin necesidad de infraestructura. El POS, es el terreno esférico de 10 m de radio a la redonda del equipo. A las redes WPAN se les divide a su vez en redes de velocidad baja, media y alta. El estándar IEEE 802.15.4 es un ejemplo de red de baja velocidad, puesto que podría utilizarse para transferir video desde una cámara a un monitor contiguo. Bluetooth es un ejemplo de estándar de media velocidad, dado que podría transmitir música de alta calidad desde un equipo de audio a auriculares inalámbricos. Asimismo, se emplea para enlazar ratones, teclados y demás periféricos a ordenadores. ZigBee, con una velocidad de datos máxima de 250 kbps, se considera una red personal inalámbrica de baja velocidad y, en entorno óptimo, llega a ser considerada parte de una red de área local por su cobertura [11].

Las aplicaciones utilizadas en ZigBee presentan un tráfico catalogado en uno de los siguientes tipos:

- a) Datos periódicos (tráfico continuo): La aplicación define una tasa de datos. Es un caso típico de sensores donde por ejemplo un sensor requiere transferir la temperatura cada 10 segundos.
- b) Datos intermitentes (tráfico por eventos): En este caso la aplicación junto a otros estímulos externos al equipo define la tasa de datos. Por ejemplo, en un sistema domótico, los interruptores de luces transmiten únicamente ante la presencia de movimiento, mientras tanto están desactivados (denominado modo dormir). Esto permite consumir una energía mínima de la batería.

El estándar 802.15.4 define 2 tipos de equipos con la finalidad de minimizar el costo del sistema:

- a) FFD (Full Function Device): son equipos capaces de funcionar en cualquier topología, puede ser dispositivo, coordinador o un coordinador de una PAN. Este tipo de equipo puede comunicarse con FFDs y RFDs.
- b) RFD (Reduced Function Device): pueden solamente ser miembros de una red con topología estrella, así como comunicarse únicamente con el coordinador de la red. Son equipos sencillos con baja exigencia de procesamiento y memoria.

ZigBee exige que en sus redes exista al menos un equipo de función completa (FFD) a fin de que sea el coordinador de red, mientras que los nodos finales de la estrella pueden ser de función reducida para reducir costos como se muestra en la figura 7. Además, todos los equipos tienen modo de direccionamiento de 64 bits, pero se pueden usar direcciones de 16 bits para poder reducir el tamaño del encabezado del paquete.

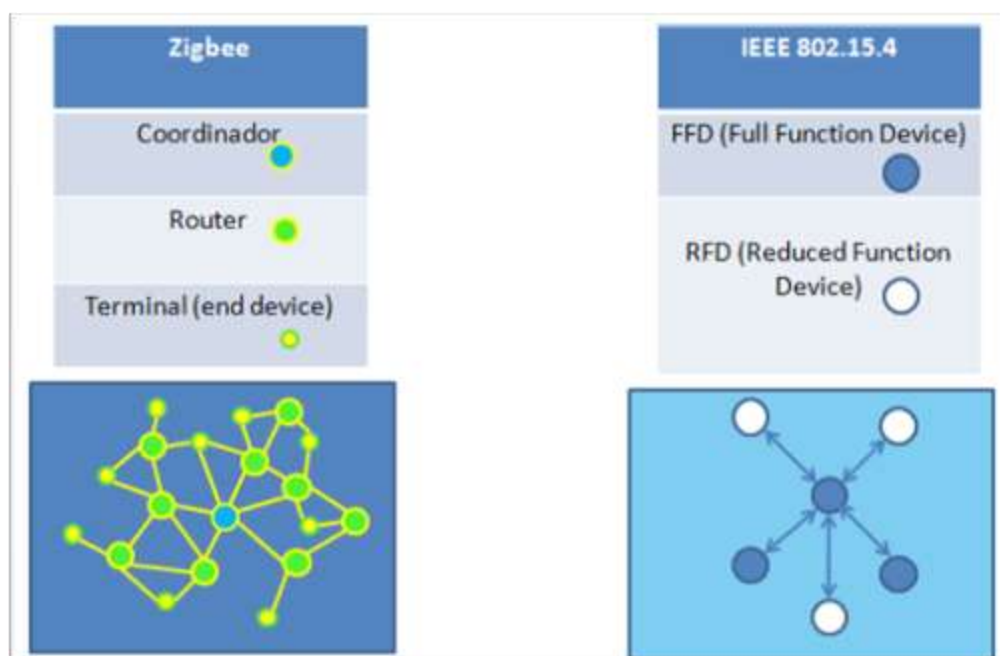


Figura 7. Equipos para Zigbee e IEEE 802.15.4.

ZigBee tolera tres topologías de red: estrella, malla, y árbol, dichas topologías se muestran en la figura 8 [12].

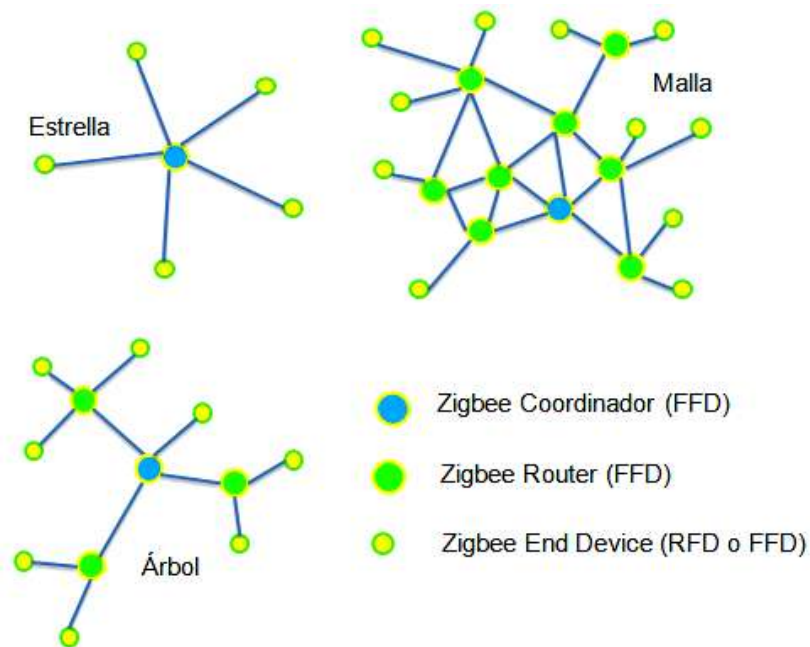


Figura 8. Topologías de red para Zigbee.

Zigbee permite hasta 65,000 nodos por red. Es ideal para aplicaciones orientadas a domótica que requieren conexiones inalámbricas de corto alcance y baja tasa de transferencia [13].

2.1.4.2 Encaminamiento en redes de malla

En este apartado se describe un poco la comunicación que realiza Zigbee para el envío óptimo entre nodos vecinos, desde un origen hasta un destino.

El encaminamiento debe entenderse como la ruta que se le da a algo hacia un punto determinado. En el ámbito de las redes de sensores, se pueden describir como los mecanismos necesarios para realizar la transmisión de datos desde un origen a un destino. En una RIS el protocolo de encaminamiento permite que la red sea apta para generar de manera autónoma las rutas necesarias para enviar información de un nodo a otro de manera eficiente y correcta, logrando así, la entrega de datos.

Algunos requisitos que deben presentar los protocolos de encaminamiento de las RIS son: eficiencia en cuanto a ancho de banda, convergencia rápida de rutas, escalabilidad, tolerancia a fallos y calidad de servicio [14].

Zigbee emplea el enrutamiento vector distancia Ad hoc AODV (*Ad hoc On Distance Vector routing*) para realizar el encaminamiento en las redes de malla. AODV es un

protocolo de encaminamiento bajo demanda, que basa el descubrimiento de rutas en un ciclo de petición-respuesta de ruta que realiza un nodo y utiliza el número de saltos desde el origen hasta el destino como métrica para tomar las decisiones. Esto es, cuando un nodo necesita enviar datos a un nodo destino, el nodo origen envía una solicitud en *broadcast* de petición de ruta, con el objetivo de descubrir un camino hasta el destino. Los nodos intermedios reenviarán dicho paquete y cualquier nodo que tenga un camino válido hasta el destino, o el mismo nodo destino, contestará a la fuente de forma *unicast* con un paquete de respuesta de ruta. Cuando el origen haya recibido ese paquete de respuesta, se inicia el envío de datos al destino. Las rutas son sostenidas, pero si ocurre algún error durante el tiempo de validez de las rutas, se propaga un mensaje de error para que los nodos implicados eliminen las rutas perjudicadas [15].

Los mensajes usados durante el descubrimiento de rutas son: petición de ruta, respuesta de ruta y error de ruta.

2.1.4.3 Tramas para la recolección de datos

Los radios Xbee admiten dos modos de operación: Interfaz de programación de aplicaciones (API) y transparente (AT). En el modo API, la comunicación es vía puerto serie en paquetes organizados (tramas) para el envío y recepción de dichos paquetes. De esa manera se permite establecer una comunicación compleja entre los módulos sin tener que definir su propio protocolo. Por el contrario, el modo AT presenta varias limitaciones, por ejemplo: un radio XBee simplemente transmite datos en serie al XBee de recepción, según lo identificado por la dirección del dispositivo; mediante comandos y si un Xbee requiere transmitir un mensaje a diferentes dispositivos, debe actualizar su configuración para establecer un nuevo destino. Además, un nodo en modo AT no puede identificar el origen de un mensaje inalámbrico que recibe. Si necesita distinguir entre los datos que provienen de diferentes dispositivos, los dispositivos que envían deben incluir información adicional. Para hacer esto, debe definir un protocolo robusto que incluya toda la información que cree que necesita en sus transmisiones.

El modo API se habilita configurando el parámetro AP en el software XCTU. Las principales ventajas del modo API frente al de AT se enuncian a continuación:

- 1.- Al existir diversidad de tramas para distintos propósitos (como comunicación y configuración), permite la configuración de los dispositivos sin ingresar al modo comando, ya sea para Xbee locales o remotos en la red.
- 2.- Puede utilizarse el modo API para transmitir a múltiples dispositivos los mensajes deseados. Dado que los datos del destino están incluidos como parte de la estructura de tramas API.
- 3.- La trama API incluye el origen del mensaje, por lo que es sencillo identificar de dónde provienen los datos. Es decir, se identifica la dirección de origen de cada paquete recibido.
- 4.- Permiten la administración de transmisión inalámbrica de datos a uno o varios destinos.
- 5.- Se recibe el estado (éxito o falla) de cada paquete transmitido, el reconocimiento o acuse de recibido (*ACK acknowledgement*) y número de reintentos. Al enviar un paquete, la radio transmisora recibe un ACK, lo que indica que el paquete se entregó con éxito. La radio transmisora reenviará el paquete si no recibe un ACK.
- 6.- Cuenta con la opción de muestras de entrada y salida (E/S), esta característica permite a un XBee recibir datos de E/S de uno o más XBee remotos.
- 7.- Es posible realizar una gestión y diagnóstico avanzado de la red.
- 8.- Los paquetes de recepción (RX) contienen la dirección de origen de la radio transmisora.
- 9.- Es posible enviar información fácilmente a múltiples radios y enviar paquetes de difusión.
- 10.- Se puede obtener el indicador de la fuerza de la señal recibida RSSID (*Received Signal Strength Indicator*) de un paquete.
- 11.- Los paquetes incluyen una suma de comprobación para la integridad de datos.

En el presente proyecto se utilizaron los comandos AT y tramas API mostrados en la tabla I.

TABLA I. Tramas utilizadas.

Número de trama	Nombre de trama	Utilizada por:
0x17	Solicitud de comando remoto <i>Remote Command Request</i>	El coordinador
0x97	Respuesta de comando remoto <i>Remote Command Response</i>	Los <i>routers</i> o ED, que reciben una orden
0x92	Zigbee IO indicador de recepción de muestra de datos	El coordinador
0x90	Paquete recibido de Zigbee <i>Packet received from Zigbee</i> (AO=0) API ZB_RX_RESPONSE	Coordinador
0x08	Comando AT <i>AT Command</i>	Cualquier módulo conectado a la computadora. Se utiliza para consultar o establecer parámetros del módulo.
0x88	Respuesta de comando AT <i>AT Command Response</i>	Los <i>routers</i> o ED, que reciben una orden 0x08
0x10	Solicitud de transmisión Zigbee API ZB_TX_REQUEST	Los <i>routers</i> o ED, que envían información al coordinador
0x8B	Estado de transmisión Zigbee API ZB_TX_STATUS_RESPONSE	El coordinador al completar una trama de tipo 0x10, el módulo envía un mensaje de estado de transmisión. Este mensaje indicará si el paquete se transmitió con éxito o si hubo una falla.

La estructura de la trama API se muestra en la tabla II:

TABLA II. Estructura de la trama API [16]

Delimitador de inicio	de	Longitud	Datos de la trama	Suma de comprobación
1		2 3	4 5 6 8 9 ... n	n + 1
0x7E		MSB LSB	API - Estructura específica (depende del tipo de trama)	Byte único

Los paquetes de datos estructurados en modo API se denominan tramas. Se envían y se reciben a través de la interfaz en serie del dispositivo y contienen el mensaje inalámbrico en sí mismo, así como cierta información adicional, como el destino, el origen de los datos o la calidad de la señal.

Cuando un dispositivo está en modo API, todos los datos que ingresan y salen del módulo a través de la interfaz en serie están contenidos en tramas que definen operaciones o eventos dentro del dispositivo.

Cualquier información recibida a través de la interfaz serie antes del delimitador de inicio es desechada silenciosamente por el XBee. Si la trama no se recibe correctamente o si la suma de comprobación falla, los datos también se descartan y el módulo indica la naturaleza de la falla respondiendo con otra trama [16].

En el capítulo III se describe ampliamente el uso de las tramas API para el desarrollo del proyecto.

2.2 Redes de área local

En este apartado se describe a las redes de área local, la topología empleada, así como el hardware existente y utilizado en el presente trabajo. También se describe el estándar IEEE 802.3 y el protocolo de Internet (IP).

Las redes de área local son redes de expansión geográfica limitada (unos cuantos km) y de propiedad privada. Son utilizadas para enlazar estaciones de trabajo u ordenadores personales, con el propósito de intercambiar información y compartir recursos. Además, sintetiza la administración de la red. Utilizan tecnología de expansión por medio de un cable simple al que están conectados todos los ordenadores. Operan a velocidades entre 10 Mbps - 1 Gbps, tienen bajo retardo y experimentan pocos errores, por lo que son consideradas como redes con comunicación más segura [17].

2.2.1 Topología

Se comprende por topología de una red local a la distribución física en la que se encuentran listos los ordenadores que la componen.

Existen tres tipos: anillo, bus y estrella. La topología adaptada en la red de área local implementada en el presente trabajo es de tipo estrella. Esta topología se caracteriza por tener un punto central; nodo central al que se conectan todos los equipos. De esta característica se desprende la desventaja de esta topología, la cual es que la máxima vulnerabilidad se encuentra precisamente en el nodo central, ya que, si éste falla, toda la red fallaría. La topología en estrella es empleada en redes Ethernet y ArcNet [17].

2.2.2 Hardware para LAN

La construcción de una red de área local en cuanto a hardware está regida por: equipos de red (Hub, NIC, switch, *router*), cableado estructurado (par trenzado de cobre UTP/STP, fibra óptica o cable coaxial) y conectores (RJ11, RJ12, RJ45, AUI, DB15, BNC, SC, ST coaxial, DB25 y DB9) [18]. En nuestro caso la red de área local queda implementada con el protocolo IEEE 802.3, usando un *router* de la familia Cisco y cableado UTP categoría 5e con conectores RJ45. Cabe señalar que la tarjeta de desarrollo con la que se realiza el proyecto cuenta ya con un puerto Ethernet.

2.2.3 Estándar IEEE 802.3

Ethernet es la tecnología de comunicaciones dominante por cable para LAN con más de 3 mil millones de interfaces instaladas en todo el mundo [19]. Es la red que se basa en el protocolo IEEE 802.3, por lo que el protocolo de acceso al medio es CSMA/CD, en el que las estaciones están a la escucha del canal permanentemente y, cuando lo encuentran libre de señal, efectúan sus transmisiones.

Cualquier estación conectada a una red IEEE 802.3 debe contar con una tarjeta de red que obedezca al estándar, con los elementos electrónicos y tener instalado el software apropiado para la recepción y generación de tramas (controlador). El protocolo se encarga de revisar las tramas que le llegan desde el canal, de unir los datos de información dándoles la forma de una trama y también hallar los posibles errores en destino, entre otros. También es el encargado de negociar con el sistema operativo del ordenador en que se instala los recursos que requiera [20].

En nuestro caso, como ya se mencionó anteriormente, se utiliza el protocolo IEEE 802.3 para Ethernet con topología estrella, puesto que se contaba con equipo de esa tecnología y funcionó de forma efectiva.

2.2.4 Protocolo de Internet

El protocolo IP, funciona en el nivel de red del modelo OSI, el cual permite encaminar los datos hacia otros equipos. También recibe el nombre de nivel Internet en el modelo TCP/IP. Considerando este último modelo, la información que proviene de la capa de aplicación se pasa del nivel de transporte al nivel de Internet en datagramas

IP, donde se le añaden las cabeceras necesarias para su nivel. Posteriormente se envía a niveles inferiores. Es en este nivel donde se usa el algoritmo de encaminamiento, al admitir un datagrama determina, en función de su dirección, si debe procesarlo y transitarlo al nivel superior, o bien dirigirlo hacia otro equipo.

IP (*Internet Protocol*) es un protocolo no orientado a la conexión, con mensajes de un tamaño variable limitado. Cada datagrama se gestiona de forma independiente, por lo que dos datagramas pueden tomar distintos caminos para llegar al mismo destino, provocando así, una llegada en distinto orden o repetidos. Es un protocolo no seguro, es decir, no corrige los problemas precedentes, ni tampoco informa de la existencia de ellos. Este protocolo además de recibir la información del nivel superior añade la información necesaria para su gestión (direcciones IP, *checksum*) [17].

En una red TCP/IP los ordenadores se reconocen mediante un número denominado dirección IP. Dicha dirección debe encontrarse dentro del rango de direcciones destinadas al organismo o empresa a la que pertenece, estos rangos son atribuidos por un organismo central de Internet, el NIC (*Network Information Center*).

Una dirección IP está compuesta por 32 bits agrupada en octetos, enseguida se muestra un ejemplo de una dirección IP:

11000000 10101000 01111101 01100111

Para que se entienda mejor utilizamos las direcciones IP en formato decimal, simbolizando el valor decimal de cada octeto y separando con puntos, por ejemplo, la dirección anterior, se escribe como 192.168.125.103.

La dirección de un ordenador se compone de dos partes donde la longitud puede cambiar:

- Bits de red: son los bits que determinan la red a la que pertenece el equipo.
- Bits de host: son los bits que distinguen a un equipo de otro dentro de una red.

En el presente trabajo se usa una LAN para almacenar los datos recolectados por la RIS en un servidor web. Esta LAN la integran dos equipos: el nodo coordinador con IP 192.168.125.103 y servidor web con la IP 192.168.125.105; ambos equipos se conectaron al *router* de Cisco. Así, la red de área local quedó integrada por el *router*

que unió tanto al nodo concentrador de la RIS como al servidor web, como se muestra en la figura 9.

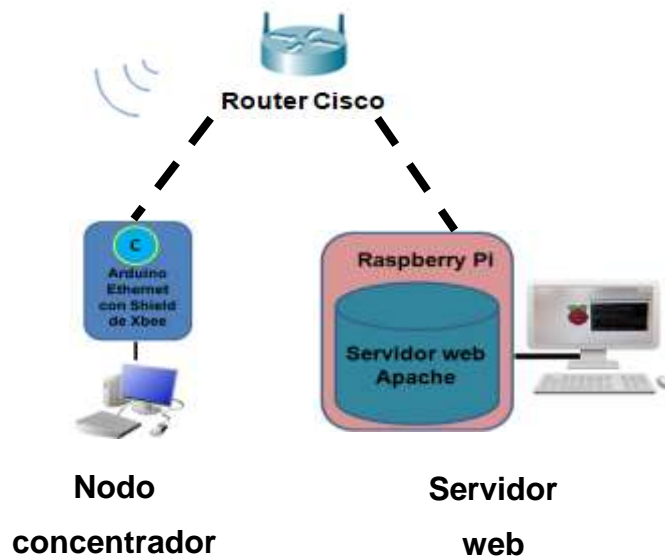


Figura 9. Red LAN para el subsistema de almacenamiento de datos.

2.3 Sistemas de telefonía celular

En este apartado se aborda la tecnología celular 3G, su arquitectura y el servicio de mensajes. Se pensó en dicha tecnología por la aplicación del sistema a implementar, pero más, por la cobertura proporcionada. Los sistemas de telefonía celular son de gran uso puesto que cubren una amplia zona geográfica como ciudades, para transmitir datos mediante señales de radio. Dichos sistemas entran en la clasificación de redes inalámbricas de área amplia (WWAN, *Wireless Wide Area Network*) con cobertura de más de 50 km [12].

La telefonía celular es un sistema de comunicaciones que usa conexiones inalámbricas. La noción fundamental de telefonía celular es la comunicación de dos equipos móviles alejados, cuyo medio de transmisión es el aire y el mensaje es enviado mediante ondas electromagnéticas a través de radio estaciones. La telefonía móvil o telefonía celular la conforman dos partes:

- A) Una red de repetidoras constituida por un grupo de antenas organizadas en celdas en la superficie terrestre, que teóricamente poseen una forma

hexagonal y cubren algunos kilómetros. Un conjunto de celdas forma un clúster.

- B) Terminales móviles, estos equipos solicitan el acceso a la red de repetidoras para que los usuarios accedan a los servicios.

Cada celda está compuesta de una estación base, ésta recibe solicitudes de conexión de los equipos móviles. Una vez efectuada la conexión con la estación base, ésta transfiere la información (voz y datos) a un centro de conmutación.

Al momento que un usuario se encuentra unido a una celda en una comunicación, por ejemplo, una llamada de voz, la comunicación se realiza dentro del área de cobertura de la celda. Cuando el usuario decide moverse y sale del área de cobertura de dicha celda, el equipo móvil ingresa al área de cobertura de otra celda y la comunicación continúa estable. Este proceso de transición de una celda a otra adyacente se conoce con el nombre de handover o handoff, se produce cuando la calidad del enlace es insuficiente en una de las estaciones [21]. El proceso descrito corresponde a un proceso llamado soft handover.

2.3.1 Tecnología 3G

La tecnología 3G (Tercera Generación) es la empleada en el presente trabajo puesto que se utiliza un módulo SIM 900 (tecnología GSM/GPRS) con una tarjeta SIM (*Subscriber Identity Module*), para comunicarnos vía tecnología celular; dicho módulo permite enviar y recibir llamadas y mensajes SMS, así como conectarse a Internet. Lo anterior se logra al adaptar el módulo SIM 900 con una tarjeta de desarrollo Arduino Uno previamente configurada, así, se obtienen algunas funciones de un teléfono móvil como realizar llamadas y enviar mensajes.

Para entender mejor el funcionamiento de dicha tecnología en los siguientes párrafos se explica brevemente un poco de historia, desarrollo y características de la tecnología 3G.

La tecnología 3G surge a partir de la necesidad de avanzar tanto en el ámbito técnico como en el comercial, puesto que tecnologías de 2G como GSM (*Global System for Global Communications*) con variaciones como GPRS (*General Packet Radio Service*) o EDGE (*Enhanced Data Rates for GSM Evolution*), sólo posibilitaba el

acceso a Internet con anchos de banda muy pequeños (teórico 500 Kbps, práctico 250 Kbps), que no era suficiente para enviar voz y datos a través de un teléfono celular. Fue entonces que se tuvo que mover de tecnología para conseguir velocidades tolerables de transmisión de datos, lo que llevó a la aparición del Sistema Universal de Telecomunicaciones Móviles (UMTS, *Universal Mobile Telecommunications System*).

Esta tecnología modifica la forma de acceso múltiple al medio ya que cambia de TDMA, utilizado por GSM, a WCDMA (*Wideband Code Division Multiple Access*). WCDMA se eligió ya que al utilizar códigos se permite mayor cantidad de accesos simultáneos con enlaces de mayor ancho de banda, lo que es más robusto que usar divisiones de tiempo en un segmento reducido del espectro que sólo permitía 8 accesos por portadora, como lo hacía GSM. En UMTS cada portadora WCDMA tiene un ancho de banda de 5 MHz, lo cual fue elegido puesto que esto permitía alcanzar velocidades de transmisión de hasta 2 Mbps dependiendo de las condiciones del canal radio [22]. En la figura 10 se visualizan las técnicas TDMA y CDMA.

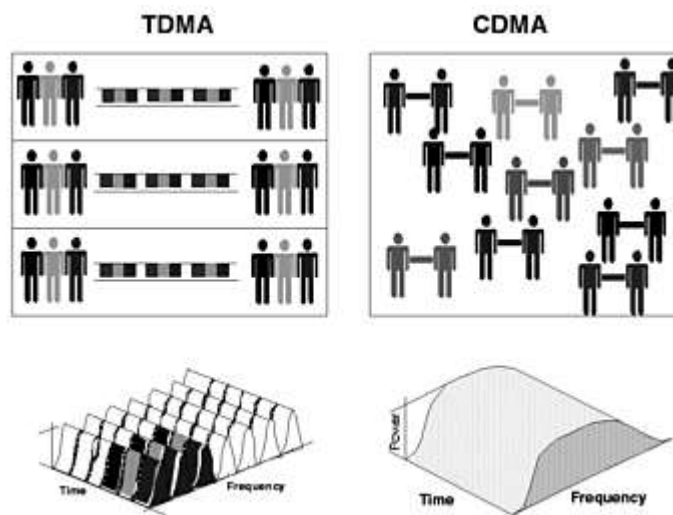


Figura 10. Técnicas TDMA y CDMA [23].

2.3.2 Arquitectura

En relación con la arquitectura de red, el orden del UMTS es casi análogo a la de GSM con la conmutación y desarrollo de paquetes. La red de telefonía móvil requiere ciertas conexiones y elementos a fin de posibilitar la comunicación entre los

diferentes equipos de usuario. Para obtener la comunicación entre terminales es necesario que se interconecten desde y hacia las estaciones de telefonía fija. De ahí radica lo importante de saber cuáles son los elementos que conforman la red y conocer sus funciones.

La arquitectura de red UMTS (3G) está compuesta por equipos móviles (UE, *User End*) de los usuarios finales, una estación base (Torre A) está encargada de la transmisión y recepción de las señales de radio hacia los terminales móviles o desde éstos hacia las BTS, también de operar con las distintas celdas del área en la que se localice. Las estaciones. El RNC (*Radio Network Controller*) controla los diferentes nodos conectados a la estación base y cumple el cometido de distinguir las conexiones según estén orientadas a voz o datos, para así gestionar la movilidad entre celdas y los recursos radio. El core o núcleo de redes de telecomunicaciones (CN) provee diversos servicios a usuarios que se conectan mediante la red de acceso y se encuentra conectado a la RNC mediante las respectivas interfaces. Se encarga del enrutamiento del tráfico móvil tanto de voz como de datos. Según el tipo de tecnología que se esté utilizando las funcionalidades del core varían (véase figura 11).

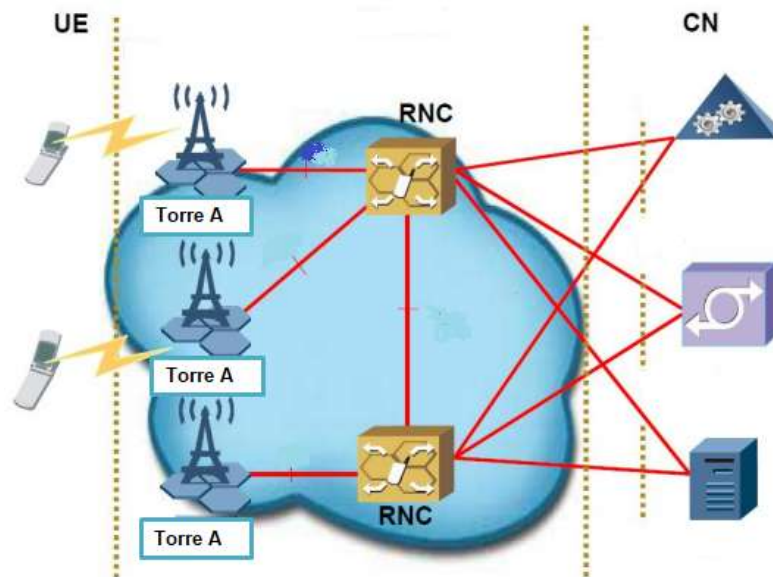


Figura 11. Arquitectura de una red UMTS [24].

2.3.3 Servicio de mensajes en 3G

Los principales servicios proporcionados por la tercera generación en telefonía celular, además del servicio de llamadas de voz, fueron el servicio de mensajes cortos (SMS, *Short Message Service*), las llamadas en espera y buzón de voz. Los mensajes cortos fueron un acontecimiento extraordinario ya que se obtuvieron los 15 billones de SMS mensuales para el año 2000. La alta calidad de voz digital y alternativas de bajo costo para realizar llamadas, como la mensajería de texto fueron algunas de las grandes ventajas del GSM [25]. Los SMS y las llamadas son servicios implementados en el presente trabajo como un sistema modo alarma, dado que, al detectar una medición fuera del umbral en las lecturas de los sensores de humedad y temperatura, se realiza el envío de un SMS o la llamada en caso necesario.

2.4 Subsistema de almacenamiento de datos

En esta sección se aborda el subsistema de almacenamiento de datos. Se explican las necesidades técnicas para almacenar la información en un sitio remoto una vez que los datos se hayan recolectado de la red. Posteriormente, se detalla cada uno de los elementos involucrados para la implementación del sistema de almacenamiento de datos, considerando los recursos de hardware y el software con que cuenta la UACM.

2.4.1 Necesidades técnicas del subsistema de almacenamiento de datos

El subsistema de almacenamiento se diseñó específicamente para garantizar la disponibilidad y almacenamiento de la información con bajo coste.

Para lograr esto, se sabe que un servidor web tiene componentes fácilmente sustituibles que presentan las mismas funciones, pero que se pueden implementar de diversas maneras, por ejemplo: el sistema operativo puede ser Linux/Unix, Windows, MacOS, los servidores web pueden ser Apache, Microsoft, Nginx, el gestor de base de datos puede ser MySQL, Microsoft Access, mongoDB; y lenguajes de programación; como: php, perl, ruby, java, python, c++ y swift.

En nuestro caso lo primero a revisar fueron las capacidades y características del hardware y sistema operativo disponibles en la UACM. Así, se optó por instalar un tipo de sistema web específico.

Existen dos tipos de webs: estáticas y dinámicas. Las webs estáticas son páginas básicamente informativas que tienen secciones y contenido fijo, en ellas, no se utilizan bases de datos ni requieren programación y son consideradas webs económicas dado que el tiempo de programación es mucho menor que el de las páginas dinámicas. Contrariamente, las webs dinámicas, permiten modificar su contenido de forma fácil y constante, se construyen usando lenguajes de programación, exigen base de datos para almacenar la información.

Las webs estáticas están alojadas en un servidor y creados en lenguaje HTML, pues están pensadas principalmente para mostrar información permanente a las que es posible acceder en cualquier instante de tiempo. Con respecto al contenido de las webs dinámicas, se genera de forma activa cada vez que se realiza una solicitud.

Como hemos dicho antes, en el caso específico del monitoreo de la RIS presenta un comportamiento dinámico, por lo que, resulta necesaria una infraestructura especial, que consta de un sistema operativo, un servidor de tipo web, un gestor de bases de datos y uno o más lenguajes de programación; facilidades que ofrecen, por ejemplo, el servidor con clasificación LAMP con software libre, abierto y gratuito. Existen además otros tipos de servidor web, los cuales son: XAMPP (con distribución de Apache que contiene MariaDB, PHP y Perl), WAMP (Windows, Apache, MySQL y php), MAMP (Mac OS, Apache, MySQL y PHP), entre otros. Sin embargo, al considerar el equipo físico existente, el sistema operativo, el capital preestablecido, el comportamiento de los datos, así como el dominio en base de datos y popularidad, se optó por el de tipo LAMP. La tabla III muestra la estructura y algunos tipos de servidor [26].

Aunado a lo anterior, se utiliza el hardware para poder implementarlo, esto es, CPU, monitor, teclado, ratón, módulo con comunicación WiFi y cable HDMI.

TABLA III. Estructura y tipos de servidor

Sistema Operativo	Tipo de Servidor web	Gestor de base de datos	Lenguaje de programación	Tipo
Linux/Unix	Apache	PostgreSQL MongoDB SQLite Cassandra	C++, C, ruby, java, Python	--
		MySQL	Php	LAMP
		MariaDB + MySQL	Php + Perl	XAMPP
Windows	Microsoft	Microsoft Access Microsoft SQL Server Oracle Database OrientDB	SQL, python, visual Basic, java	--
	NginX	MySQL	Php-CGI	NMP
	Apache		Php	WAMP
Mac OS	NginX	SQLite POSTGRESQL MongoDB CouchDB FoundationDB Airtable	Swift, C++, Python, JavaScript, Perl	--
		LibreOffice Base		MAMP

2.4.2 Software para el subsistema de almacenamiento de datos

2.4.2.1 LAMP

LAMP es el acrónimo para un paquete de software formado por los siguientes elementos: sistema operativo Linux, un servidor web Apache, una base de datos MySQL y el lenguaje de scripts PHP [27].

2.4.2.2 Linux-Sistema Operativo

Existe variedad de sistemas operativos libres y privados. Sin embargo, en el presente trabajo se emplea una tarjeta de desarrollo Raspberry Pi 2, la cual está diseñada y precargada para trabajar con un sistema operativo Raspbian (Unix/Linux) dadas las características de su hardware.

La Raspberry Pi funciona como un ordenador personal pequeño y económico. A pesar de que carece de capacidad para expansión de memoria y no se pueden colocar unidades de almacenamiento como CD, DVD y discos duros, tiene todo lo que requiere un ordenador personal simple. Es decir, tiene dos puertos USB, un puerto Ethernet, módulo Wifi, HDMI e incluso un conector de audio para sonido. La Raspberry Pi tiene una unidad microSD 2 que puede usar para iniciar el ordenador en cualquiera de los sistemas operativos. Todo lo que necesita es un monitor HDMI (o DVI con un adaptador HDMI a DVI), puertos USB para teclado y ratón, y una fuente de alimentación de 5 V [9].

En el presente trabajo, la Raspberry Pi 2 se utiliza para instalar el paquete de software LAMP para obtener un servidor web y lograr el monitoreo de la RIS en tiempo real con ambiente gráfico sencillo.

2.4.2.3 Apache – Servidor Web

Como se mencionó anteriormente, el servidor web creado en el presente trabajo es de Apache. Un servidor de este tipo se caracteriza principalmente por ser gratuito, eficiente, fiable y fácilmente configurable con código fuente abierto. El servidor web Apache es desarrollado por el Proyecto Servidor Apache (*Apache Server Project*). Dicho proyecto es a la vez manejado por un grupo de voluntarios emplazados alrededor del mundo, que a través de Internet desarrollan y planean el servidor y la documentación referenciada a éste. Estos voluntarios son conocidos como el grupo Apache.

Apache, al ser gratuito es uno de los servidores Web más usados por sus garantías suficientes para la instalación de servidores virtuales confiables o sitios Web. Uno de los atributos de los servidores Web es su capacidad de autenticación, puesto que, controlan el acceso de usuarios y estaciones de trabajo a determinados sitios Web, lo que cumple sin excepción Apache. De esta manera se mantiene restringido el acceso por Internet a usuarios indeseables, lo que imposibilita conocer la información de una organización a extraños. A través de los servidores virtuales, Apache oculta la existencia de un solo servidor real y muestra la presencia de varios servidores,

haciendo pensar a los usuarios que tienen un ordenador servidor para cada uno de ellos cuando en realidad corresponde al mismo ordenador físico [28].

El servidor Apache está disponible para las plataformas como Unix, Macintosh, Microsoft Windows y demás, en el que se ejecuta el protocolo de transferencia de hipertexto (HTTP, *Hypertext Transfer Protocol*). En nuestro caso queda regido en Unix, por tratarse de software abierto Raspbian, sistema operativo de la Raspberry Pi 2 empleada.

2.4.2.4 MySQL – Gestor de Base de Datos

Es común el uso de soluciones de sistemas abiertos y gratuitos para el avance de aplicaciones web en la actualidad. También la unión del lenguaje PHP y el sistema gestor de bases de datos MySQL, ya que permite la creación de una forma competente y sencilla de legítimos sitios web dinámicos.

MySQL utiliza el lenguaje de consulta estructurado (SQL, *Structured Query Language*), el cual es el lenguaje más utilizado y estandarizado para acceder a bases de datos relacionales. Soporta la sintaxis estándar del lenguaje SQL para la ejecución de consultas de creación, selección y manipulación de datos.

MySQL está disponible en más de veinte plataformas diferentes incluyendo las distribuciones más usadas de Linux, UNIX, Microsoft Windows y sistema operativo Mac. Además, resulta sencillo encontrar gran cantidad de software que lo soporte o desarrollado sobre MySQL. Lo anterior, hace posible una diversidad de aplicaciones en sistemas abiertos para la administración de las bases de datos a través de un servidor web.

Parte de lo mencionado hace a MySQL uno de los sistemas gestores de bases de datos más utilizados hoy en día, incluidas grandes corporaciones como: Finance, google, CISCO, Yahoo!, MP3.com, Motorola, NASA, Silicon Graphics, Texas Instruments; por mencionar algunas [29].

Como ya se expuso, el uso de MySQL es muy amplio, pero en nuestro caso resultó óptimo para la gestión de la base de datos creada, dado que es visiblemente

amigable, sencillo y claro al observar los registros de la RIS: fecha, hora, nodo, temperatura y humedad.

2.4.2.5 PHP – Lenguajes de programación

PHP es un lenguaje interpretado por el servidor caracterizado por su potencia, robustez, modularidad y versatilidad. Los programas escritos en PHP son embebidos directamente en el código HTML y ejecutados por el servidor web a través de un traductor, antes de transferir al cliente un resultado en forma de código HTML puro. Por ser un lenguaje de sistema abierto es posible descargarlo de Internet de manera gratuita en su sitio oficial¹.

PHP resulta ser un lenguaje flexible y bastante sencillo de aprender; principalmente para programadores adaptados con lenguajes como C, Perl o Java, debido a las semejanzas de sintaxis. Tal es el caso de los dos métodos para el envío de datos a un servidor web vía navegador (*browser*): GET y POST. El primer método permite enviar la información al servidor web de forma visible, esto es, se escriben en la dirección de “localización uniforme de recursos” (URL, *Uniform Resource Locator*) tanto los datos que se envían como el servidor web que los almacena. Contrariamente, el método POST no permite enviar la información al servidor web de forma visible. En el caso del proyecto al no tratarse de datos sensibles se optó por emplear el primer método.

Aunado a la riqueza del lenguaje PHP, se crearon opciones para la administración de las bases de datos, para que fuesen más intuitivas y fáciles de manejar, permitiendo la gestión remota de un servidor web. Una herramienta de las más conocidas es phpMyAdmin, disponible a través de Internet de manera gratuita y desarrollada en PHP por la colectividad de usuarios sin ánimo de lucro. Dicha herramienta garantiza la inspección de las bases de datos con una interfaz simple e intuitiva a la vez que potente [30]. Con respecto a la seguridad brindada por phpMyAdmin, se puede señalar que el acceso es limitado y seguro, puesto que se debe contar con una contraseña para poder acceder a la base de datos.

¹ <http://www.php.net/>

Capítulo III. Implementación del sistema de recolección de datos

Una de las principales tareas a realizar por el sistema de recolección es recabar todos los datos proporcionados por la RIS, para lo que se requiere conformar una RIS con diversos nodos con sensores. Para la implementación de dichos nodos se utilizó tecnología ZigBee, dadas las necesidades requeridas como son: distancia, alimentación, conectividad de gran cantidad de sensores, por mencionar algunas. Una vez obtenidos los datos de los sensores se requiere realizar el envío a un nodo concentrador, el cual registra todos los datos entregados por cada nodo.

Por otro lado, para llevar la información obtenida por la RIS a otro nivel de comunicación se utiliza una red cableada 802.3 (*LAN*) encargada del envío de los datos del nodo concentrador a un servidor, mediante el uso de un *router*. El servidor se instala en una tarjeta de desarrollo Raspberry Pi 2, el cual cuenta con LAMP: Linux (Sistema Operativo), Apache (Servidor Web), MySQL (Gestor de Base de Datos) y PHP (Lenguaje de programación); así como una herramienta escrita en PHP, llamada phpMyAdmin, para la administración de la base de datos de MySQL. Todo lo anterior para que el servidor cuente con un ambiente amigable para los usuarios con entorno gráfico sencillo.

Si únicamente una persona es la encargada de supervisar la zona remota, entonces, sólo se necesitaría un servidor web estándar que responda a la solicitud de página web del ordenador con que se cuente; por lo que, se podría definir una interfaz gráfica sencilla para el usuario. En caso contrario, si se trata de múltiples usuarios, solicitando información, se debería considerar un servidor de base de datos.

Aunado a lo anterior, se incorpora tecnología celular para avisar de la existencia de datos de la RIS fuera del umbral deseado (por ejemplo, lectura de temperatura excesiva), mediante un mensaje de texto SMS, o en caso necesario se realiza una llamada al supervisor de la red.

De esta manera, el sistema implementado obtiene los datos recabados por la RIS, y la almacenada en una base de datos, la cual puede consultarse de manera directa tanto en el monitor del servidor como vía remota. Con ello, se ahorra el

desplazamiento físico de las personas a zonas remotas o de difícil acceso y se tiene la información rápidamente.

El sistema de recolección de datos queda integrado por: una RIS con un nodo coordinador, cuatro *routers* y dos ED (nodos sensor). Para implementar esta red se usaron diferentes dispositivos disponibles en el laboratorio de “Desarrollo de proyectos tecnológicos B-207” del plantel San Lorenzo Tezonco. De esta manera, el nodo coordinador está adaptado a una tarjeta de desarrollo Arduino Ethernet, los *router* de la RIS se implementaron con tarjetas de desarrollo Arduino Uno y otros en tarjetas de desarrollo de marca Digi. Uno de estos *router* incluye una tarjeta con comunicación celular. Los Dispositivos finales se incorporaron con Arduino Uno y sensores analógicos de temperatura y humedad. Un *router* de la familia Cisco se utilizó para implementar la LAN y una Raspberry Pi 2 se usó como servidor web (véase figura 12).

En los siguientes párrafos se hace una descripción detallada de cada uno de los componentes del sistema de recolección diseñado.

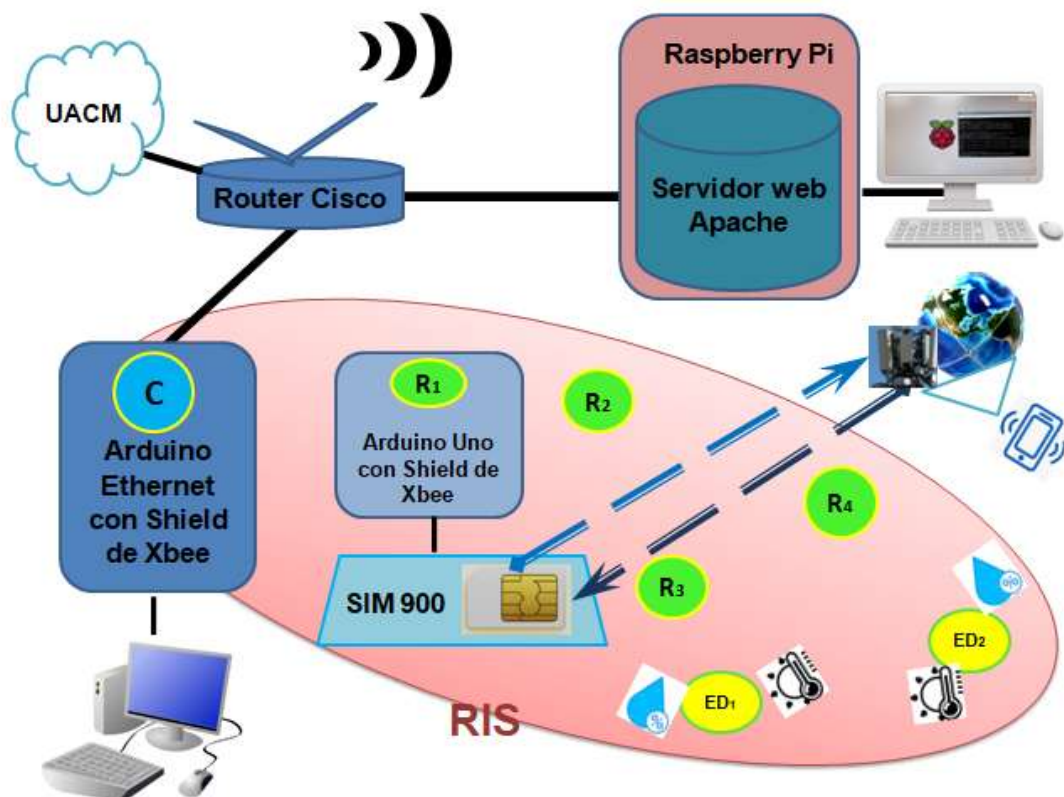


Figura 12. Sistema de recolección.

3.1. Red de sensores

La RIS implementada realiza la recolección de todos los datos entregados por cada nodo sensor en el nodo concentrador, así como el envío de datos al servidor web, obteniendo así, la fecha, hora, nodo, temperatura y humedad. Dicha recolección se lleva a cabo con los protocolos de comunicación antes mencionados, pero se detallan más adelante. En la figura 13 se muestra la estructura general de la RIS y en la tabla IV se especifica la dirección única de cada elemento de la RIS, resaltando en negritas el dato que estaremos trabajando como identificador de cada Xbee.

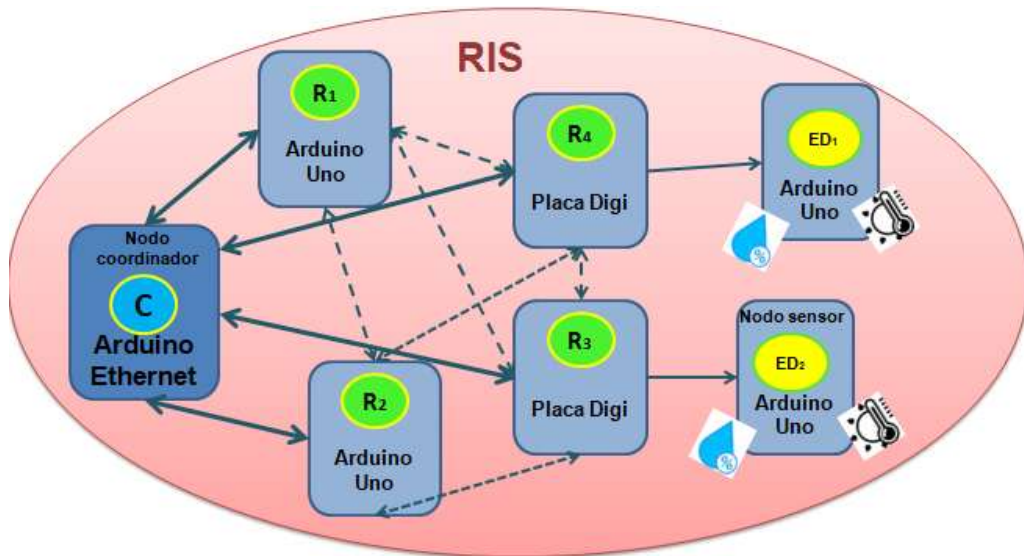


Figura 13. RIS.

TABLA IV. Dirección de cada elemento de la RIS

Elemento de red	Dirección
Coordinador	0013A200414E50 21
Router 1	0013A20040CC06 30
Router 2	0013A200414E50 1C
Router 3	0013A20040CC06 24
Router 4	0013A20040B584 8B
Nodo sensor 1 (ED1)	0013A200414E4E 37
Nodo sensor 2 (ED2)	0013A200414E4E 43

3.1.1 Nodo sensor

El nodo sensor está formado por el siguiente hardware: Xbee S2 configurado como End Device (ED₁), un *shield*² para conectar con Arduino Uno, que es alimentado con una batería de 5 V y cuenta con dos sensores analógicos, uno de temperatura (LM35) y uno de humedad (MD0021). En la figura 14 se muestra una representación gráfica del nodo sensor, mientras que en las figuras 15 y 16 se ilustran los dos nodos sensor reales.



Figura 14. Representación gráfica del nodo sensor.

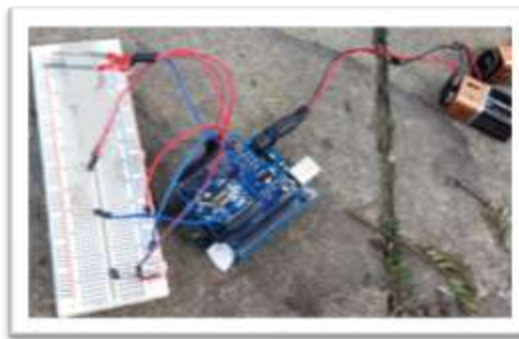


Figura 15. Nodo sensor ED1 real.

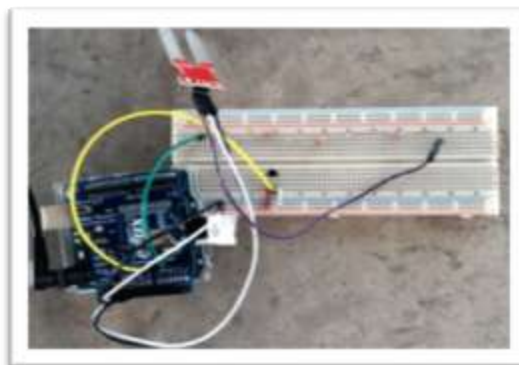


Figura 16. Nodo sensor ED2 real.

² El término *shield* se refiere a una tarjeta para adaptar módulos a otras tarjetas de desarrollo.

El sensor de temperatura utilizado en el presente trabajo se calibró de tal manera que el sistema quedó con 10 mV en presencia de 1 °C, referenciado a un voltaje máximo de 1.1 V (igual a 110 °C). Por ello, es necesario habilitar el convertidor analógico digital, para que la salida del sensor analógico resulte sencilla de interpretar. En nuestro caso se utiliza la siguiente fórmula:

$$\text{Temperatura} = 110(\text{adc_data1} / 1024)$$

En la fórmula anterior el valor 110 representa la temperatura máxima de respuesta a la salida del sensor estipulada previamente, el valor adc_data1 es el dato analógico entregado por el sensor LM35 y el valor 1024, son todos los valores posibles entregados por el sensor.

Si hubiésemos optado por el voltaje de referencia de 5 V, se debe emplear otra fórmula, con ello, podríamos obtener temperaturas de hasta 500 °C. Lo anterior no sería de mucha utilidad ya que, como se mencionó anteriormente, el LM35 soporta la temperatura máxima de 150 °C.

En el caso del sensor de humedad se hizo un proceso de calibración que consistió en probar diferentes niveles de humedad. Dado que la salida para un ambiente seco podría variar entre un rango de valores, se estableció un umbral en el cual el ambiente podía considerarse seco. Así mismo, para el caso de ambientes muy húmedos, se sumergió el sensor en un vaso con agua y a partir de allí se eligió un umbral en el que el sensor enviaría un valor 1 que indica exceso de humedad en el ambiente, ver figura 17.

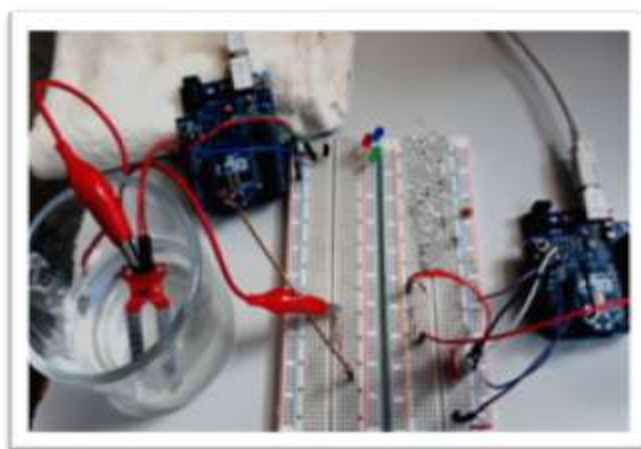


Figura 17. Calibración del sensor de humedad.

Después de realizar la calibración del sensor de humedad se obtuvieron los siguientes valores:

- De cero a veinte ($0 < 20 \%$) se encuentra seco, en el programa arroja el valor de cero (0).
- De veintiuno a 110 ($21 < 110 \%$) húmedo estándar, en el programa arroja el valor de 0.5.

Mayor a 111 % mojado - exceso de humedad, en el programa arroja el valor de uno (1).

La configuración de los radios Xbee S2 End Device (ED₁ y ED₂) en el software XCTU, se muestra en la tabla V y en la figura 18.

TABLA V. Configuración de los módulos End Device

Parámetro para configurar	Valor asignado	Descripción
CH	B	Se refiere al canal de comunicación. Debe ser el mismo para todos los módulos Xbee.
PAN ID	881C	Indica el identificador de red. Debe ser el mismo para todos los módulos Xbee. Si se establece en 0 (valor predeterminado), se seleccionará una ID PAN aleatoria de 64 bits.
SC		Determina la máscara de bits de los canales de exploración (hasta 16 canales) que utiliza el coordinador al formar una red. El coordinador realiza un escaneo de energía en todos los canales SC habilitados. Después genera un PAN ID y luego forma la red en uno de los canales SC disponibles.
DH	0	Se refiere a la dirección alta de destino
DL	FFFF	Se refiere a la dirección baja de destino, FFFF es en modo broadcast
SH	0013A200	Es el valor que viene por defecto, número alto de serie
SL	414E4E37	Es el valor que viene por defecto, número bajo de serie
Node Identifier	End Device 1	Indica el nombre identificador del nodo
CE	End Device	Indica el tipo de dispositivo: Coordinador, Router o End Device
BD	9600	Indica la velocidad de transmisión en baudios para la comunicación entre Xbee. Debe tener el mismo valor para todos los módulos Xbee
AP API Enable	1 (Habilitado)	Se indica la habilitación en modo API
AO API OUTPUT MODE	Explicit [1]	Indica un modo de entrada salida API explícito
Entradas/Salidas ADC	AD0 (humedad) y AD3 (temperatura) habilitados	Habilita las entradas y salidas ADC
IR IO Sampling Rate	DA23 (cada 55.84 segundos para ED1) A002 (cada 40.96 segundos para ED2)	Indica la frecuencia de muestreo de los módulos Xbee

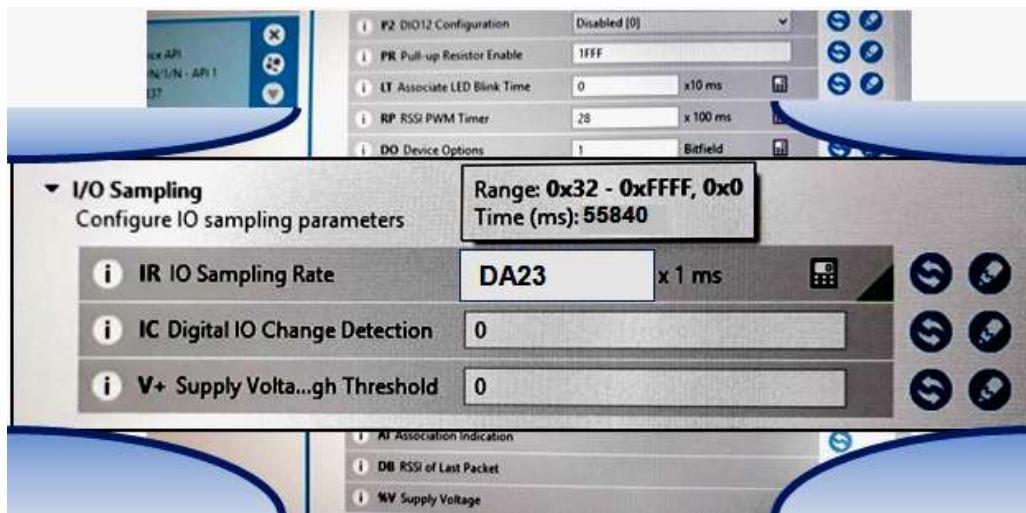


Figura 18. Configuración de la velocidad de muestreo del módulo Xbee ED1.

En este proyecto las tramas empleadas por el nodo sensor para responder a solicitudes son:

- 0x97 Respuesta de comando remoto
- 0x88 Respuesta de comando AT
- 0x10 Solicitud de transmisión Zigbee

El nodo sensor puede responder solicitudes que se hagan usando las tramas:

- 0x17 Solicitud de comando remoto
- 0x08 Comando AT

3.1.2 Nodo concentrador

El nodo concentrador está formado por el siguiente hardware: placa de desarrollo Arduino Ethernet con *shield* para conectar el Xbee S2 en modo Coordinador, alimentado vía USB a una computadora, para poder ver vía serial tanto el estado de la conexión como los valores arrojados de los nodos sensor (véase figura 19).



Figura 19. Nodo concentrador.

La configuración del módulo Xbee coordinador se muestra en la tabla VI.

Tabla VI. Parámetros del nodo Xbee Coordinador

Parámetro para configurar	Valores
CH	B
PAN ID	881C
DH	0
DL	FFFF o el valor del dispositivo al que se desea comunicar
SH	0013A200
SL	414E5021
Node Identifier	Coordinador
CE	Coordinador
BD	9600
AP API Enable	1 (Habilitado)
AO API OUTPUT MODE	Explicit [1]

En este proyecto las tramas empleadas por el nodo concentrador para responder a solicitudes son:

- 0x90 Paquete recibido de Zigbee
- 0x8B Estado de transmisión Zigbee
- 0x88 Respuesta de comando AT

La trama empleada por el nodo coordinador para dar una orden es:

- 0x17 Solicitud de comando remoto

El nodo concentrador puede responder solicitudes que se hagan usando la trama:

- 0x92 Zigbee IO indicador de recepción de muestra de datos
- 0x10 Solicitud de transmisión Zigbee
- 0x08 Comando AT

3.1.3 Nodo router en modo alertador

El nodo alertador queda compuesto por un módulo Xbee Router con *shield* para Arduino Uno, alimentado con batería de 5 V como se muestra en la figura 21. Así mismo, se realiza la conexión de dicho Router con la SIM 900, la cual cuenta con su propia fuente de alimentación. Se muestra el esquema gráfico en la figura 20 y el nodo real en la figura 21.



Figura 20. Esquema gráfico del nodo alertador.

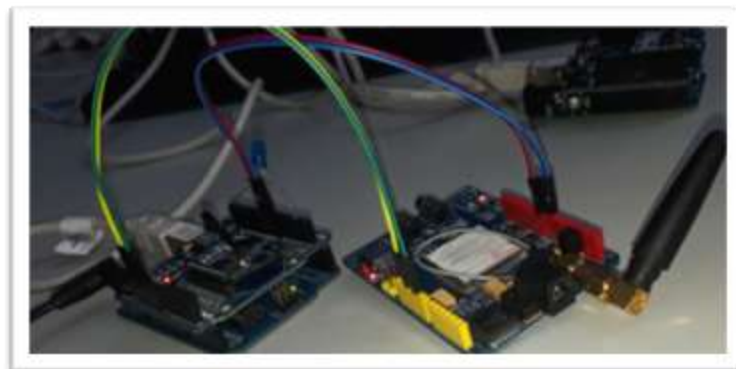


Figura 21. Nodo alertador real.

La configuración en XCTU del módulo Xbee Router 1 se muestra en la TABLA VII.

TABLA VII. Configuración del Router 1

Parámetro para configurar	Valores
CH	B
PAN ID	881C
DH	0
DL	FFFF o el valor del dispositivo al que se desea comunicar
SH	0013A200
SL	40CC0630
Node Identifier	Router 1
CE	Router
BD	9600
AP API Enable	1 (Habilitado)
AO API OUTPUT MODE	Explicit [1]

En este proyecto las tramas empleadas por el nodo *router* alertador para responder a solicitudes son:

- 0x97 Respuesta de comando remoto
- 0x88 Respuesta de comando AT

El nodo *router* alertador puede responder solicitudes que se hagan usando las tramas:

- 0x17 Solicitud de comando remoto
- 0x08 Comando AT

Se estipuló que en caso de que el valor de temperatura recabado por la RIS sea mayor a 27 °C, será momento de enviar una trama al *router* alertador para que realice primero el envío de un mensaje SMS avisando de lo sucedido y posteriormente realice una llamada. Cabe mencionar que dicho umbral de temperatura puede modificarse según las necesidades del sistema a monitorear.

Para ello, se agregó un programa extra en el Arduino Uno para que se llevara a cabo el envío de un SMS y la llamada. En las próximas dos figuras se visualizan ambas acciones. En la figura 22 se muestran dos imágenes de salida del puerto serie, el COM8 es del nodo coordinador (lado izquierdo de la figura 22) y el COM3 es del nodo alertador (lado derecho de la figura 22). En el COM8 se observa que al momento en el que el nodo coordinador detecta una elevación en la temperatura, muestra el mensaje de “se está enviando un mensaje por exceso de temperatura”. Sin embargo, lo que realiza el nodo coordinador al momento de detectar una temperatura mayor a 27 °C es el envío de la trama solicitud de comando remoto (0x17) con la orden de encendido de LED para el nodo alertador, luego el nodo alertador contesta con la trama respuesta de comando remoto (0x97) y efectúa el encendido de LED. Continúan con el intercambio de mensajes y en el momento que se enciende el LED azul por parte del nodo alertador, éste realiza el envío del mensaje SMS (figura 22) y después una llamada (figura 23). Del lado izquierdo de la figura 22 se muestra el registro de exceso de temperatura en el nodo coordinador, mientras que en lado derecho se muestra el envío de mensaje por el nodo alertador.

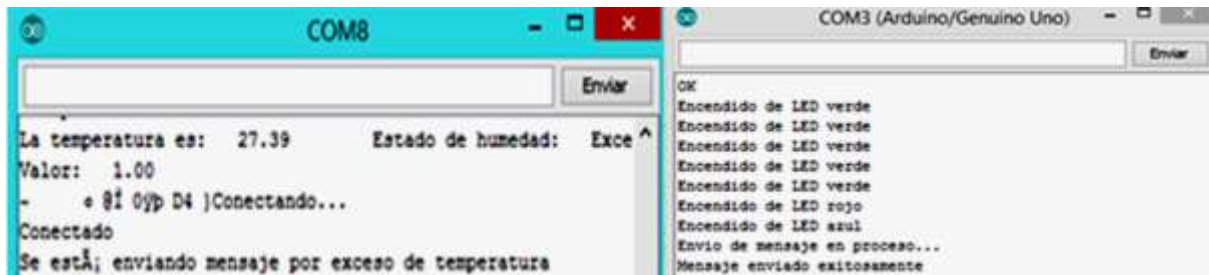


Figura 22. Envío de mensaje exitoso por parte del *router* alertador.

En la figura 23 se observa a la salida del puerto serie un “OK”, el cual forma parte de la programación, este mensaje representa la inicialización y conexión correcta de la tarjeta con tecnología celular SIM 900 integrada al *router* alertador. El siguiente mensaje en el monitor serie es el número al que se llamará y finalmente se realiza la llamada que, por cada tono de llamada, en el monitor serie se imprime la palabra “RING”.

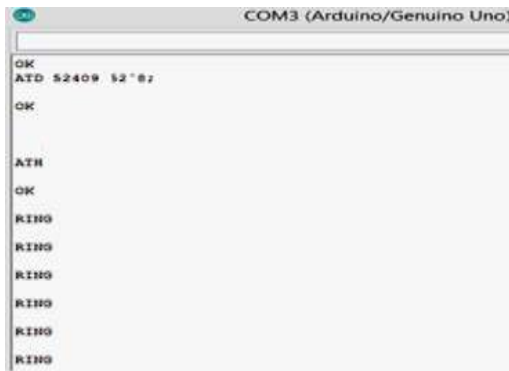


Figura 23. Llamada realizada por el *router* alertador.

Para el resto de los dispositivos Xbee Router se utilizan los mismos valores mostrados en la tabla VII, sólo cambian los valores SH, SL además de que en los Arduino Uno no se requiere ejecutar programa alguno.

3.2. Transmisión de datos en la red de sensores

Con base en AODV, Zigbee hace una adecuación para que sus nodos establezcan una comunicación tipo Ad Hoc. Las tramas para la recolección de datos con el protocolo de comunicación API son utilizadas en múltiples aplicaciones, en nuestro caso se emplean para establecer el envío de datos como: humedad, temperatura, realizar llamadas y envío de mensajes SMS.

Las primeras pruebas realizadas para la creación de la RIS fueron en modo AT. Posteriormente se realizaron envíos en modo API utilizando software XCTU para enviar mensajes, donde el nodo coordinador envía una orden de encendido o apagado de LED a varios nodos y éstos realizan el encendido o apagado de un LED. En la figura 24 se muestran las tramas usadas por el nodo coordinador para dar la orden de encendido y apagado de un LED por parte de los *routers* o ED. Dicha figura muestra el modo consola activado del software XCTU, se observa al nodo coordinador ubicado en la parte superior izquierda de la figura, la trama; puede verse en la parte central de la imagen con nombre: registro de trama – *frame log* -. Justo debajo de este último se ubica el encendido de LED con letra H, corresponde al alto - *High*- y el apagado con letra L que corresponde a un bajo - *Low*-.



Figura 24. Envío de instrucción para encendido de LED con trama solicitud de comando remoto 0x17.

El encendido y apagado de LED puede verse en la figura 25, la sección (a) muestra la orden de encendido de LED por parte del coordinador que se encuentra al centro de la imagen marcado con un círculo color naranja, mientras que los nodos *router* se encuentran a extremos del coordinador apagados marcados por un círculo color blanco. La sección (b) muestra la respuesta de los *router*, allí se encuentran todos con el LED encendido. La sección (c) ilustra la orden de apagado de LED por parte del coordinador. Finalmente, la sección (d) muestra la respuesta de los nodos *router* con sus LEDes apagados.

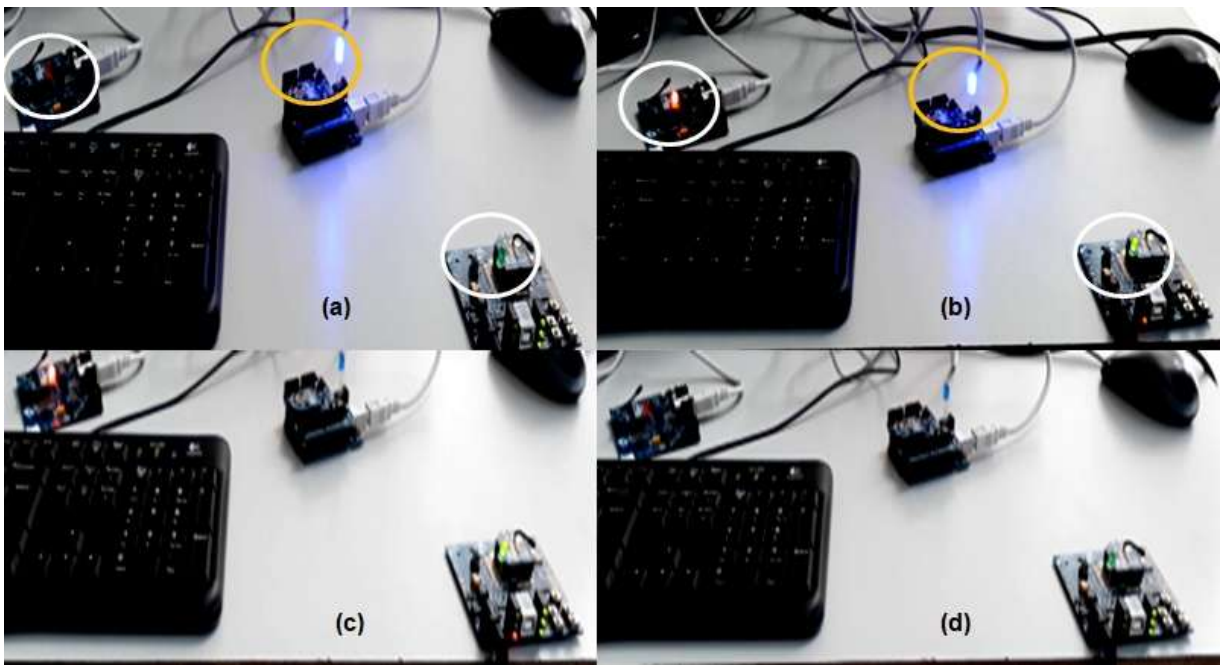


Figura 25. Orden de encendido de un LED por parte del coordinador. (a) Orden de encendido de LED, (b) respuesta de encendido de LED, (c) orden de apagado de LED, (d) respuesta de apagado de LED.

La trama que envía el nodo coordinador para el encendido del LED es de solicitud de comando remoto (0x17) en ella se envía un alto (H, *High* - 05) y por respuesta del nodo destino, el coordinador recibe una trama de tipo respuesta de comando remoto (0x97). La figura 26 muestra el intercambio de tramas entre nodo coordinador y un End Device.

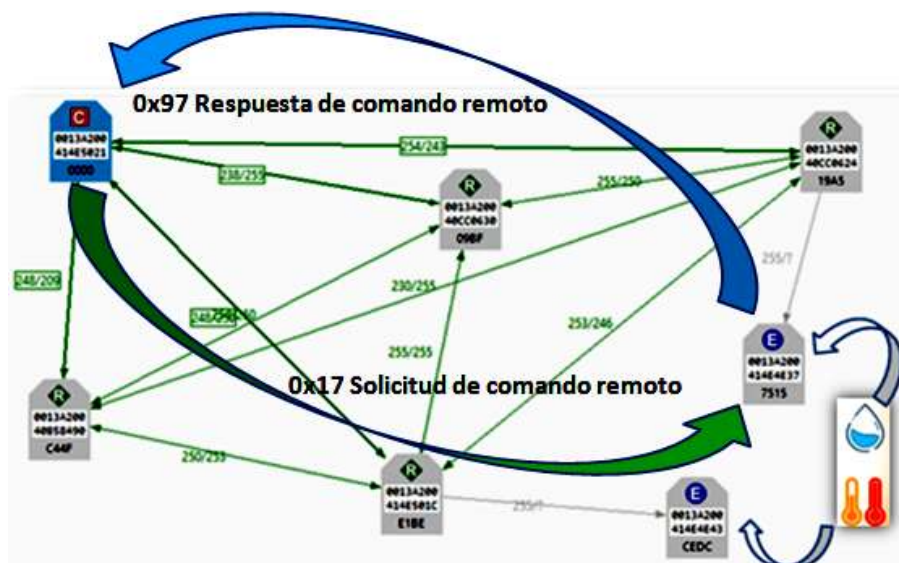


Figura 26. Ejemplo ilustrativo de intercambio de tramas

En la tabla VII el último valor de “Datos de la trama” indica el encendido o apagado de LED. Para el apagado del LED se envía la misma trama 0x17 sólo que modificada con un bajo (L – *low* - 04). De la misma manera que en el encendido, el nodo receptor envía una trama de respuesta 0x97. Las estructuras de las tramas mencionadas se detallan en la tabla VIII.

TABLA VIII. Estructura de tramas para encendido y apagado de LED

Delimitador de inicio 1 byte	Longitud 2 bytes		Datos de la trama 4-n bytes Hasta 100 bytes por paquete	Suma de comprobación n+1 1 byte
	MSB	LSB		
0x7E			API - Estructura específica	Byte único
7E 00 10 17 01	00 13 A2 00	40 CC 06 30	FF FE 02 44 34 05	74
7E 00 10 17 01	00 13 A2 00	40 CC 06 30	FF FE 02 44 34 04	75

3.2.1 Escenarios de prueba para la transmisión de datos

Una vez que se estableció la comunicación, optamos por conocer los alcances de los radios en escenarios con línea de vista y sin línea de vista. Para probar la topología se utilizó el encendido y apagado de un LED. Los escenarios que se utilizaron son los siguientes:

- Escenario sin línea de vista en el interior de un edificio
- Escenario con línea de vista
- Escenario sin línea de vista en exteriores

El primer escenario se experimentó en el segundo piso del edificio B de la UACM plantel San Lorenzo Tezonco (SLT) con un nodo coordinador y dos *router*. Cabe señalar que el edificio cuenta con muros de concreto. En el salón B-207 se colocó el nodo coordinador, el primer *router* (R₁) se colocó en dirección a la zona de sanitarios, el otro *router* (R₂) en dirección a los salones finales del pasillo.

En este escenario se obtuvo una distancia máxima para el R₁ de 12.8 m, mientras que para el R₂ fue de 11.5 m, cabe mencionar que en el experimento se mantuvieron cerradas las puertas involucradas (véase figura 27).



Figura 27. Escenario uno: sin línea de vista en el interior de un edificio.

El segundo escenario fue con un nodo coordinador y un *router*, se llevó a cabo en el patio aledaño a los edificios A, B y C del mismo plantel SLT. El escenario se situó al final de los salones y escaleras de emergencia de los edificios. Se colocó el nodo coordinador en la planta baja del edificio A y, poco a poco, se fue alejando al *router* (R1) hacía el edificio C. Se obtuvo que con línea de vista (sin obstáculos) existía comunicación a una distancia máxima de 94 m (véase figura 28).



Figura 28. Escenario dos con línea de vista.

Otro escenario con la mayor cantidad de nodos con línea de vista fue tomar el escenario uno sin línea de vista por parte del coordinador colocado en el laboratorio B-207 y el *router* 1 a la distancia máxima establecida de 11.5 m y el resto de los nodos R2, R3, R4, ED1 y ED2 con línea de vista como se muestra en la figura 29. En relación a los nodos con línea de vista de la figura 29, el R2 se colocó al final del pasillo del edificio B, mientras que el R3, se dejó en la planta baja igual a final de pasillo (justo en las escaleras de emergencia) del mismo edificio, el ED1 se colocó a 32 m del R3, el R4 se colocó al final del edificio C, a unos metros de las escaleras de emergencia y finalmente se colocó al ED2 frente a los primeros salones de la planta baja del edificio C, entonces, el R4 y el ED2, quedaron a distancia máxima entre sí.

En este escenario se activó el envío de datos por parte de los dos ED, esto es, enviaron los valores de humedad y temperatura con la trama 0x92 “IO indicador de recepción de muestra de datos” al nodo coordinador.



Figura 29. Escenario dos con línea de vista en la mayoría de los nodos.

El tercer escenario fue con un nodo coordinador, cuatro *router* y dos End Device (ED), se ubicó en el jardín que está entre el edificio de profesores y el edificio A de alumnos. Se colocó al nodo coordinador en el cubículo D - 107, posteriormente los *router* a distancia máxima entre sí y a los ED como se muestra en la figura 30. En este escenario se obtuvieron distancias desde 20 m hasta 60 m sin línea de vista. Este escenario es el que se asemeja considerablemente a un ambiente real o a una zona de monitoreo, puesto que en él se incluyen árboles y obstáculos parecidos a las zonas remotas.

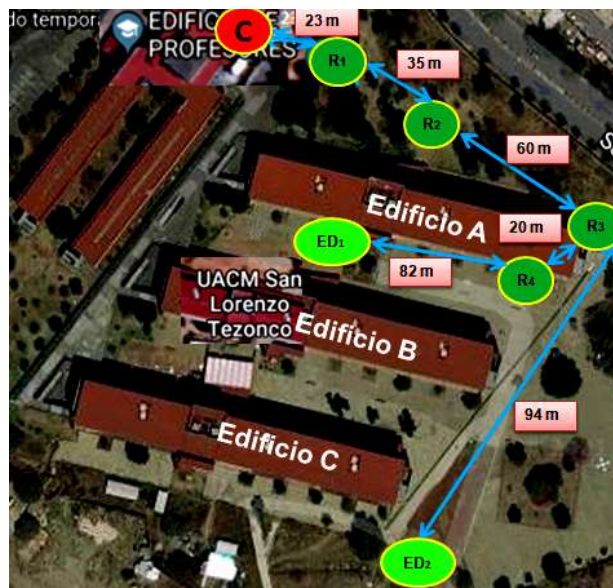


Figura 30. Escenario tres: sin línea de vista en exteriores.

Cuando el envío de datos fue exitoso entre nodos, se dejaron establecidas las distancias máximas. Lo anterior para aprovechar la capacidad de los nodos *router* que permiten reenviar la información hasta los ED utilizando el protocolo de encaminamiento que provee Zigbee. Como se explicó anteriormente, esto permite alcanzar a nodos que no están en el radio de cobertura del nodo coordinador. La representación del escenario de prueba generado en el software XCTU se muestra en la figura 31. En dicho escenario se puede observar al nodo coordinador, los cuatro *router* y los dos nodos sensor.

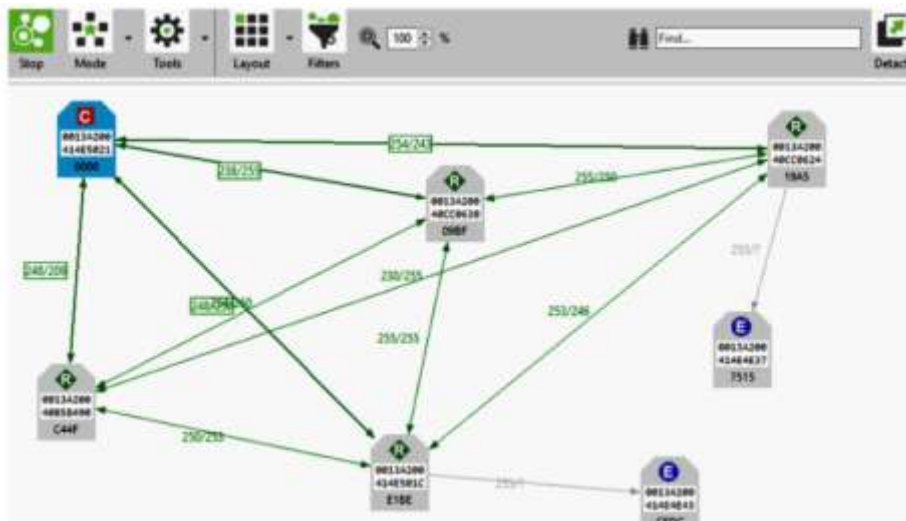


Figura 31. RIS óptima con 1 coordinador, 4 *router* y 2 ED.

Una vez creada la RIS con nodos a distancias máximas entre sí, como “el escenario dos con línea de vista en la mayoría de sus nodos”, se fijó la trama 0x92 para que el nodo coordinador recibiera los datos de temperatura y humedad de los nodos sensor. En caso de que el nodo coordinador detectara una temperatura fuera del umbral establecido, debía enviar la trama 0x17 al nodo *router* alertador con la orden de encendido de LED.

Así pues, se prosigió con el procesamiento de los datos obtenidos. Para este sistema se estableció que al recibir el nodo coordinador una trama de tipo Zigbee “IO indicador de recepción de muestra de datos” (0x92) se debían extraer 3 valores para guardarse en la base de datos, estos son: la temperatura, el valor de humedad y el nombre del nodo emisor, indicado por los 2 caracteres menos significativos de la dirección del Xbee. El dato de interés del elemento de red quedaría regido por los

dos últimos valores de su dirección completa, por ejemplo, los valores 1C y 37 para los nodos sensor, ver la tabla IX.

TABLA IX. ID de interés para los nodos sensor

Elemento de red	Dirección completa	Dato de interés
Router 2	0013A200414E501C	1C
ED1	0013A200414E4E37	37

Cuando se tuvieron en el nodo coordinador todos los datos de la RIS, de manera ordenada y como se pretendía, se procedió a almacenar dicha información en un servidor web.

3.3 Subsistema de almacenamiento de datos

El objetivo principal de guardar los datos obtenidos de la RIS en una base de datos incorporada en el servidor web es para tener un respaldo de información y poder consultarlo de manera sencilla en cualquier momento. Para ello, se implementó un servidor web y una red LAN.

3.3.1 Servidor web

El servidor web se implementó en una tarjeta de desarrollo Raspberry Pi 2, como ya se mencionó, es un servidor de tipo LAMP, puesto que, el sistema operativo utilizado es de Raspbian, el servidor web es Apache con un gestor de base de datos de MySQL con lenguaje de programación de php y la herramienta phpMyAdmin.

Se inició con la actualización de repositorios, programas de la Raspberry Pi 2 y la instalación de apache 2. Las instrucciones de instalación se muestran en la tabla X.

Tabla X. Instalación de servidor tipo LAMP

Comando	Acción
sudo apt-get update sudo apt-get upgrade	Actualización a partir de repositorios
sudo apt-get install apache2	Instalación de Apache
sudo apt-get install php7.0 libapache2-mod-php7.0	Instalación de php
sudo reboot	Reinicio de Raspberry Pi 2
sudo apt-get install mysql-server mysql-client php5-mysql	Instalación de MySQL
sudo apt-get install php5-mysql phpMyAdmin	Instalación de la herramienta phpMyAdmin
sudo nano /etc/apache2/apache2.conf	Modifica el archivo de configuración con el editor nano

Durante el proceso de instalación se deben establecer las contraseñas tanto de MySQL como de phpMyAdmin, lo que garantiza un acceso restringido tanto al servidor web como a la base de datos. Una vez ejecutados los comandos de la tabla X, se puede ingresar a la herramienta phpMyAdmin colocando el usuario y contraseña preestablecidos, con ello está listo el servidor web para crear la base de datos.

3.3.2 Almacenamiento en la base de datos

Para almacenar la información en la base de datos del servidor web creado, se envían los datos desde el nodo coordinador hasta el servidor web. Fue entonces que se agregaron al código de la tarjeta de desarrollo Arduino Ethernet instrucciones en modo GET, además de las condiciones necesarias para establecer la conexión vía cable Ethernet usando direcciones IP. En la figura 32 se muestra el esquema de conexión entre los elementos mencionados.

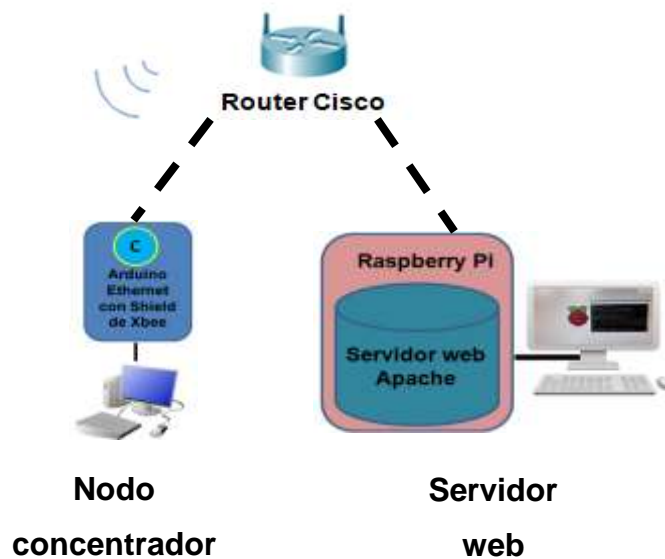


Figura 32. LAN para el envío de datos al servidor web.

En un inicio se pensó en realizar la conexión usando una WLAN con tarjetas de desarrollo Arduino Yún para obtener los datos de la RIS. Sin embargo, no fue posible puesto que no encontré la manera de comunicar a los radios de Digi con la tarjeta

Arduino Yún. Lo que logré fue establecer una red inalámbrica con 2 Arduino Yún y el *router* inalámbrico del laboratorio de ISET. La topología empleada para lo anterior se muestra en la figura 33, Cabe precisar que una vez establecida la red, ésta presentaba inestabilidad recurrente [31].

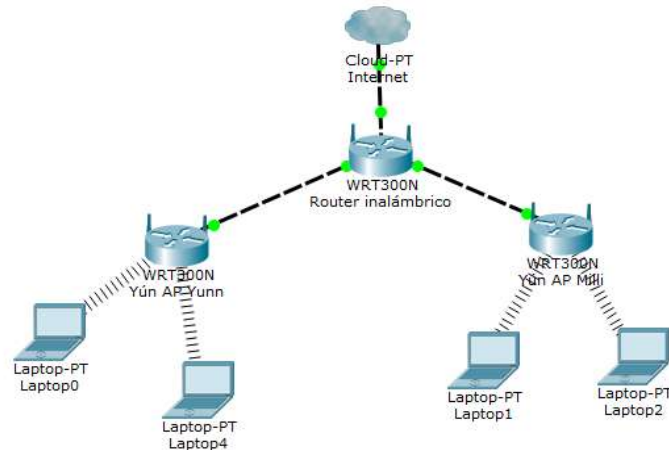


Figura 33. Creación de WLAN con Arduino Yún.

Los elementos mencionados anteriormente se conectan por una red LAN usando un *router* Cisco, que da acceso a la Internet de la UACM. Para la conexión se utilizaron los puertos Ethernet tanto de la tarjeta de desarrollo Arduino Ethernet como del Router Cisco, en el caso de la Raspberry Pi 2 se podía realizar la conexión vía cable o vía Wifi, en nuestro caso se realizó de ambos modos y en ninguno hubo fallas, pero para evitar ello, la mayoría de las veces fue vía cable Ethernet. En la figura 34 se muestra la interfaz inalámbrica wlan0 de Raspberry Pi 2, asociada a un punto de acceso, mientras que la interfaz de Ethernet eth0 está deshabilitada.



Figura 34. Arquitectura de la red LAN, Conexión del servidor web vía Wifi.

La configuración de las direcciones IP se muestra en la tabla XI

Tabla XI. Direcciones IP de los elementos de la LAN

Elemento de red	Dirección IP
Router Cisco	192.168.125.254
Servidor web / Raspberry Pi 2	192.168.125.105
Nodo Coordinador / Arduino Ethernet	192.168.125.103

Una vez establecida la conexión del servidor web con el *router*, éste proporciona una dirección IP, la cual se introduce al código de la tarjeta de desarrollo Arduino Ethernet. Así se logra la comunicación entre el nodo coordinador y el servidor web.

Ahora bien, para el envío de datos de la RIS, proporcionados por el nodo coordinador, hacia el servidor web se requiere crear una base de datos y dos archivos, uno para la conexión con la base de datos y otro para subir los datos recibidos del nodo coordinador a dicha base.

Para crear la base de datos en el servidor web se utiliza la herramienta phpMyAdmin, en el menú principal se selecciona el botón *new* y se asigna el nombre que queremos asignar a la base de datos, en mi caso fue “sensor”. Al seleccionar del menú desplegable el nombre “sensor”, aparece la opción crear tabla, dicha tabla será llamada en mi caso “VALORES” y tendrá un tamaño de 4 registros; para el ID, la temperatura, la humedad y la fecha (ver figura 35).

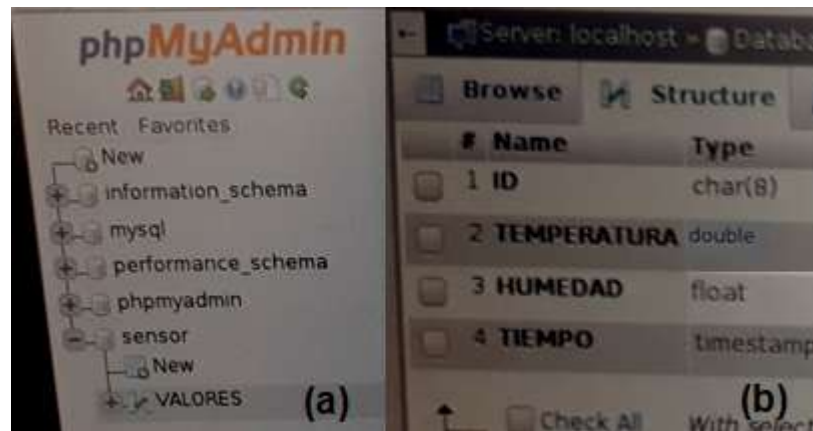


Figura 35. Creación de la base de datos. (a) Nombre de la base de datos: sensor, (b) Tabla de tamaño 4 con nombre: VALORES.

Los datos que almacena la base de datos son: el valor ID, que puede ser de 8 caracteres de tipo *char*, pero que en el nodo coordinador limitamos a sólo dos valores (2 bytes); el valor de la temperatura es de tipo doble precisión *double* (8 bytes), el valor de humedad de tipo precisión sencilla de tipo *float* (4 bytes), y el valor del tiempo de tipo *timestamp* (4 bytes). En total, cada registro de datos requiere de 18 bytes para su almacenamiento. Es conveniente destacar que en una base de datos de Mysql pueden generarse hasta 50 millones de registros, 60,000 tablas, 64 índices por tabla (1-16 columnas), sin embargo, aún no se tiene la certeza de la capacidad total, puesto que debe considerarse el sistema operativo y la memoria del equipo donde se implementa el servidor web. En particular, para el caso de la información almacenada, se estima un total de 2.7 KB por hora.

Ahora bien, los archivos creados en el servidor web son dos como ya habíamos mencionado, el primero es para obtener la conexión con la base de datos llamado archivo `config.php` y el otro es para subir los datos recibidos del nodo coordinador a dicha base nombrado archivo `iot.php` (ver figura 36).

```

<?php
//iot.php
//importamos la configuración
require("config.php");
//Leemos los valores que llegan por GET
$TEMPERATURA = mysqli_real_escape_string($con, $_GET['TEMPERATURA']);
$ID = mysqli_real_escape_string($con, $_GET['ID']);
$HUMEDAD = mysqli_real_escape_string($con, $_GET['HUMEDAD']);

//Orden para insertar valores
$query = "INSERT INTO VALORES(ID, TEMPERATURA, HUMEDAD) VALUES('$ID','$TEMPERATURA', '$HUMEDAD' )";
//Se ejecuta la instrucción
mysqli_query($con, $query);
mysqli_close($con);
echo "Pagina para subir los datos<br />";
echo "<br />ID = $ID <br />";
echo "<br />TEMPERATURA = $TEMPERATURA °C<br />";
echo "<br />HUMEDAD = $HUMEDAD <br />";

```

Figura 36. Archivo iot.php.

Ambos archivos se agregaron a una carpeta creada en la Raspberry Pi 2 llamada mimo como se puede ver en la figura 37.

```

pi@raspberrypi ~ $ cd mimo
pi@raspberrypi ~/mimo $ ls
config.php index.php iot.php
pi@raspberrypi ~/mimo $ █

```

Figura 37. Carpeta con dos archivos: iot.php y config.php.

Los datos de ID, TEMPERATURA y HUMEDAD se envían al servidor web a través del método GET como se muestran en la figura 38, en ella se muestra parte del programa ejecutado en el nodo coordinador.

```

if (client.connect(server, 80)>0) { // Conexion con el servid

    client.print("GET /iot.php?ID=¢TEMPERATURA=¢HUMEDAD"); //
    client.print("¢ID=");
    client.print(Lsb, HEX);
    client.print("¢TEMPERATURA=");
    client.print(temperatura);
    client.print("¢HUMEDAD=");
    client.print(humedad);
    client.println(" HTTP/1.0");
    client.println("User-Agent: Arduino 1.0");
    client.println();
    Serial.println("Conectado");
}
else {
    Serial.println("Falla en la conexion");
}

if (!client.connected()) {
    Serial.println("Desconectado!");
}

```

Figura 38. Envío de datos del nodo coordinador al servidor web con método GET

Para probar que ambos archivos funcionan correctamente se incrustan vía URL un dato de temperatura y el nombre de un nodo, y se verifica que se hayan guardado correctamente en la base de datos (véanse figura 39 y 40). En la figura 39 se muestra que, al insertar una muestra de datos de temperatura y nombre del nodo en la base de datos, éstos se registran correctamente. En este ejemplo se simula que el valor de 20.3 °C es el valor de la temperatura entregada por el nodo sensor llamado E5T9.

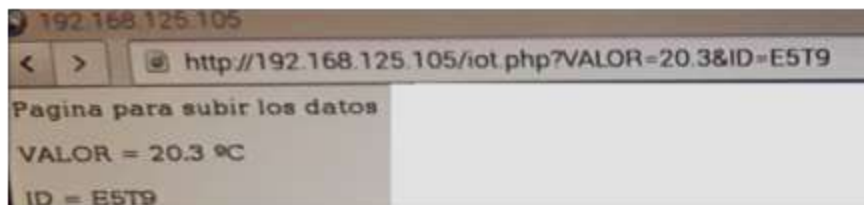


Figura 39. Prueba de funcionamiento de la base de datos: incrustación de datos vía URL.

En la figura 40 se muestra cómo se guardan los valores insertados vía URL de manera correcta en la base de datos, el registro proporciona el nombre del nodo, la temperatura y la fecha de registro.

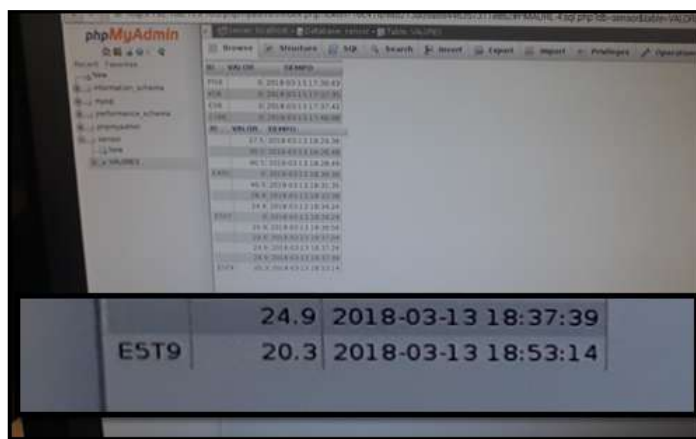


Figura 40. Datos guardados en la base de datos.

Posteriormente se hicieron más pruebas agregando el valor de humedad (ver figura 41) y se modificaron los datos que se registran en la base de datos. En esta figura para el caso de la humedad se escribió la palabra “Buena”, sin embargo, en el registro real sólo será el dato registrado por el nodo coordinador de la RIS, restringido a los valores 0, 0.5 o 1.

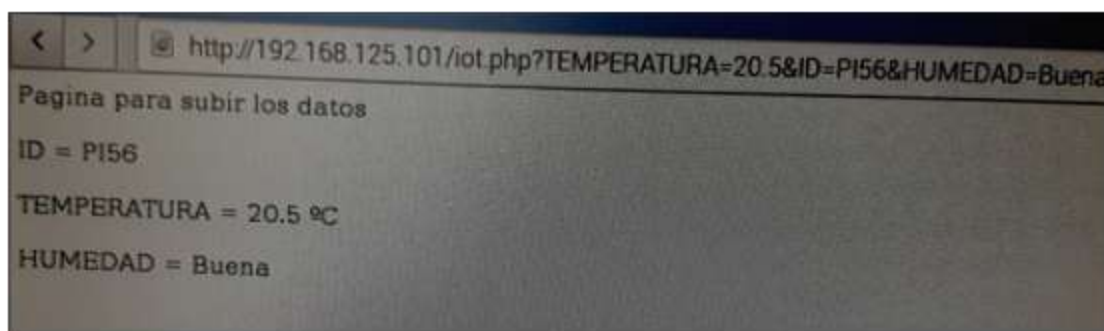


Figura 41. Prueba de inserción de datos vía URL.

3.4 Costo del sistema de recolección

El costo del material utilizado en el presente proyecto se muestra en la tabla XII. Cabe destacar que no se considera el costo del CPU, monitor, *mouse*, teclado, utilizado en el servidor web con la tarjeta de desarrollo Raspberry Pi 2, así como el módulo adaptador USB y cable para Xbee con el que se configuran los radios Xbee.

Tabla XII. Costo del material utilizado en este proyecto

Cantidad	Elemento de red	Hardware	Costo por unidad	Costo
1	Coordinador	Tarjeta de desarrollo Arduino Ethernet	\$860	1,575
		Xbee S2	\$631	
		<i>shield</i> de Xbee para Arduino	\$84	
5	Router	Arduino Uno	\$626	4,023
		Xbee S2	\$631	
		<i>shield</i> de Xbee para Arduino	\$84	
1	Router alertador	Arduino Uno	\$626	2,040
		Xbee S2	\$631	
		<i>Shield</i> de Xbee para Arduino	\$84	
		SIM 900	\$699	
2	End Device	Arduino Uno	\$626	1,428
		Xbee S2	\$631	
		<i>Shield</i> de Xbee para Arduino	\$84	
		Sensor de humedad	\$13	
		Sensor de temperatura	\$74	
	End Device	Placa Digi	\$800	1,518
		Xbee S2	\$631	
		Sensor de humedad	\$13	
1	Servidor web	Sensor de temperatura	\$74	\$2,500
		Raspberry Pi 2	\$2,500	
			Subtotal	\$13,084.00

Capítulo IV. Pruebas de funcionamiento del sistema completo

En este apartado se muestra el funcionamiento del sistema completo, en él se utiliza la RIS del escenario tres, sin línea de vista en exteriores, descrito en el capítulo anterior.

4.1 Pruebas realizadas

Recordemos que se distribuyeron los nodos a distancias máximas entre sí (figura 42).



Figura 42. RIS utilizada para probar el funcionamiento del sistema completo.

A los nodos sensor se les colocaron los dos sensores, al sensor de temperatura se le dejó registrar la temperatura ambiente, mientras que al sensor de humedad con identificador 43 se sumergió en agua por un rato para detectar un nivel máximo de humedad. Así el sistema debía registrar en la base de datos valores estándar ya conocidos, por ejemplo, al haber agua en el sensor de humedad arrojaría el valor de 1, mientras que la temperatura estaría en el rango de 18 °C a 25 °C. Lo anterior se corroboró en la base de datos y se muestra en la figura 43. Posteriormente, para incrementar la temperatura se tocó con la mano al sensor de temperatura del nodo sensor con identificador 37 y se sacó del agua al sensor de humedad con

identificador 43. Se percibió el aumento en la magnitud de los datos registrados en la base de datos, lo anterior puede verse en la figura 44.

ID	TEMPERATURA	HUMEDAD	TIEMPO
43	19.66	1	2018-08-25 13:40:55
43	19.55	1	2018-08-25 13:45:32
43	19.55	1	2018-08-25 13:45:37
43	19.66	1	2018-08-25 13:45:57
43	19.55	1	2018-08-25 13:46:48
43	19.55	1	2018-08-25 13:47:48
43	19.55	1	2018-08-25 13:48:35
37	19.87	0	2018-08-25 14:03:06
37	19.87	0	2018-08-25 14:04:36
37	19.87	0	2018-08-25 14:04:51
37	19.87	0	2018-08-25 14:06:21
37	19.87	0	2018-08-25 14:07:51

Figura 43. Registro en la base de datos de temperatura y humedad.

ID	TEMPERATURA	HUMEDAD	TIEMPO
37	19.98	0	2018-08-25 14:24:21
37	22.67	0	2018-08-25 15:05:17
37	22.67	0	2018-08-25 15:07:17
37	22.67	0	2018-08-25 15:09:17
37	24.06	0	2018-08-25 15:13:36
37	24.06	0	2018-08-25 15:15:41
37	24.81	0	2018-08-25 15:17:46
37	24.71	0	2018-08-25 15:19:51
43	17.4	0	2018-08-25 15:36:55
43	17.51	0	2018-08-25 15:38:55
43	17.62	0	2018-08-25 15:40:55
37	32.33	0	2018-08-25 15:53:05

Figura 44. Registro de elevación de temperatura por ID 37 y cambio en valor de humedad de ID 43.

Otro experimento realizado con todo el sistema fue manteniendo la misma RIS descrita en este último apartado, sólo que se colocó un sensor de temperatura al *router* con ID 30, se indujo elevación en la temperatura, para así, realizar el envío de mensaje SMS y la llamada. Se pudo corroborar tanto la elevación de temperatura en la base de datos (figura 45) como el envío de mensaje y la realización de llamada en el teléfono inteligente (figura 46).

ID	TEMPERATURA	HUMEDAD	TIEMPO	✓ 1
	27		0 2018-06-28 13:59:23	
30	23.31		1 2018-06-28 13:47:20	
30	23.63		1 2018-06-28 13:46:06	
1C	22.67		0 2018-06-28 13:44:52	
1C	22.56		0 2018-06-28 13:43:41	
30	56.29		1 2018-06-28 13:35:34	
43	20.84		1 2018-04-25 15:00:51	
1C	20.84		0 2018-04-25 14:59:41	

Figura 45. Registro en la base de datos de temperatura elevada.



Figura 46. Teléfono inteligente con registro. (A) Mensaje SMS (B) Llamada.

Para poder consultar los registros de la base de datos, también podía realizarse mediante un teléfono inteligente (celular), siempre y cuando se esté conectado a la red LAN o WLAN del punto de acceso. En nuestro caso se hizo mediante la red inalámbrica y se pudo consultar la base de datos como se muestra en la figura 47. Cabe señalar que sólo los usuarios que cuentan con la contraseña de phpMyAdmin pueden ingresar a la base de datos.

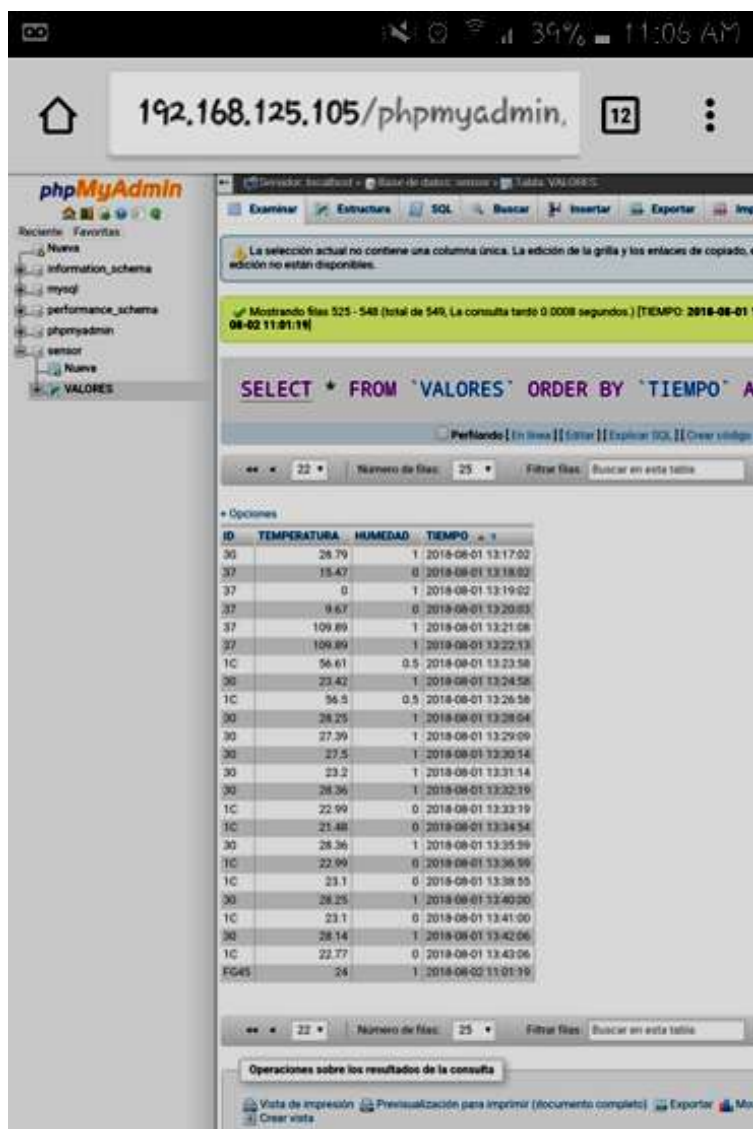


Figura 47. Consulta de la base de datos desde un teléfono inteligente.

4.2 Análisis de resultados

Mediante la serie de experimentos antes expuestos se determina que el sistema puede recabar los datos de temperatura y humedad. En el caso de la temperatura el sensor se ajustó para operar en un rango de 0 a 110 °C, sin embargo, las temperaturas probadas fueron de 17 a 59 °C. En el caso del sensor de humedad se observa que cumple con la función de detectar presencia o ausencia de agua en el suelo y dicho umbral puede calibrarse según las necesidades.

4.2.1 Conectividad

En el caso de la conectividad de la red, ésta se puso a prueba en escenarios con y sin limitación de energía. Los datos que envía la red son las tramas 0X17 y la 0X97, las cuales son enviadas cada 41 y 56 segundos, con ello, se pudo analizar el comportamiento del sistema al tener constantemente envío y recepción de datos, tanto por la RIS como en el almacenamiento y la consulta de la información.

Para esta prueba se usó el software XCTU, en el cual se muestra de manera gráfica la calidad de los enlaces entre los nodos de la red. En el caso limitado en energía, se usaron baterías para alimentar a los nodos, específicamente pilas recicladas. En este escenario los enlaces sólo duraron un máximo de 2 horas, mientras que, en el escenario con energía ilimitada con conexión al suministro eléctrico de la Universidad, estuvieron activos las 5 horas continuas, que es el tiempo que duró el experimento, sin falla alguna. El experimento consistió en enviar y recibir distintas tramas a diferentes intervalos. Las tramas usadas en la red se muestran en la tabla XIII, se presentan de menor a mayor según la velocidad de transmisión. Las tramas empleadas en el sistema de recolección de datos final fueron 4: 0x17, 0x97, 0x90 y 0x92, lo que indica un total de 28.8 bps.

Tabla XIII. Velocidad de transmisión de las tramas utilizadas

Trama	Bytes	Bits	Velocidad de transmisión con frecuencia de muestreo ³ de 55.84 segundos	Velocidad de transmisión con frecuencia de muestreo de 40.96 segundos	Total de bits por segundo	Velocidad de transmisión con frecuencia máxima de muestreo 1,048.57 segundos
0x08 Comando AT	7	56	1.00 bps	1.37 bps	2.37 bps	0.05 bps
0x88 Respuesta de comando AT	8	64	1.15 bps	1.56 bps	2.71 bps	0.06 bps
0x8B Estado de transmisión Zigbee	10	80	1.43 bps	1.95 bps	3.38 bps	0.07 bps
0x17 Solicitud de comando remoto	19	152	2.72 bps	3.71 bps	6.43 bps	0.14 bps
0x90 Paquete recibido de Zigbee	21	168	3.01 bps	4.10 bps	7.11 bps	0.16 bps
0x97 Respuesta de comando remoto	22	176	3.15 bps	4.30 bps	7.45 bps	0.17 bps
0x92 Zigbee IO Indicador de recepción	23	184	3.29 bps	4.50 bps	7.79 bps	0.17 bps
0x10 Solicitud de transmisión Zigbee	25	200	3.58 bps	4.89 bps	8.46 bps	0.19 bps

Cabe destacar que se debe considerar el tipo de trama con la que se trabaja para cada necesidad, dado que se usan en pares, es decir, una para solicitar y otra para responder, deben ser correspondientes. Por ejemplo, la trama 0x17 es para solicitar y la trama 0x97 contesta la solicitud. Por ello, evitar utilizar la misma trama para

³ La frecuencia de muestreo mínima permitida para los radios Xbee es de 1 ms

distintos objetivos, dado que se pueden producir errores en la entrega de datos de la RIS.

En mi caso, tuve en algún momento un error al intentar leer los datos de la RIS y querer realizar una llamada con la misma trama.

4.2.2 Subsistema de almacenamiento

En cuanto al servidor web, no tuve dificultad para registrar los datos de la RIS: todas las veces que consulté, los valores estuvieron activos en la base de datos del servidor. Esto indica que la información se mantiene guardada y puede accederse a ella, a menos que haya una falla en el suministro eléctrico. La cantidad de datos que se almacenan en el servidor web por día es de 65 KB. Lo anterior, indica que es muy bajo lo almacenado en la base de datos considerando las capacidades actuales de la tarjeta de desarrollo utilizada, por lo que no es necesario realizar un respaldo de información.

El sistema planteado puede utilizarse en tierras de cultivo como viñedos, hortalizas, bosques, entre otras zonas. En el caso de las tierras de cultivo se piensa en poder monitorear la humedad para evitar plagas y muerte de las plantas, por exceso o escases de agua. Para el caso de las zonas boscosas se pensó en la detección de incendios y la detección de plagas por exceso de humedad, considerando los sensores con los que ya cuenta el sistema.

4.3 Factibilidad para su implementación para su uso en plantaciones o cultivos

El sistema diseñado puede implementarse en zonas remotas que cuenten con al menos un inmueble con suministro de corriente eléctrica y de ser posible un modem o router. Esto es, zonas aledañas a la Ciudad de México. En la figura 48 se muestra un ejemplo de este tipo de zonas, el Parque Ecoturístico de San Bernabé Ocotepéc en la Alcaldía La Magdalena Contreras.

Para la implementación del sistema se podría considerar el uso de equipo con el que ya se cuente, como un monitor, CPU, teclado y ratón, para no incrementar el costo y aprovechar dichos equipos. Específicamente, dichos periféricos o hardware se

utilizarían para la creación del servidor web para almacenar y consultar los datos deseados de la tierra de cultivo en sitio, así se estaría evitando el desplazamiento de la persona al área de cultivo, pero no al centro de control. Posteriormente, se podría generar una LAN con un *Access Point* o *router* que se tenga en existencia y que cuente con acceso a Internet, para realizar la consulta a la base de datos desde cualquier lugar usando un dispositivo con acceso a Internet, por ejemplo, un teléfono inteligente, evitando así el desplazamiento físico al sitio.

En caso de que no contar con los recursos necesarios para implementar el sistema, se puede hacer uso de recursos locales o federales, como el programa de Presupuesto participativo de la Ciudad de México. Con ello, se realizaría la implementación del sistema con una RIS con los sensores necesarios para el monitoreo de variables específicas, según las necesidades del área de cultivo. La alerta con tecnología celular es opcional, dado que se requiere de un módulo de identificación de abonado (SIM, *Subscriber Identity Module*) con saldo disponible y desde luego cobertura de un sistema de 3G de tecnología celular.

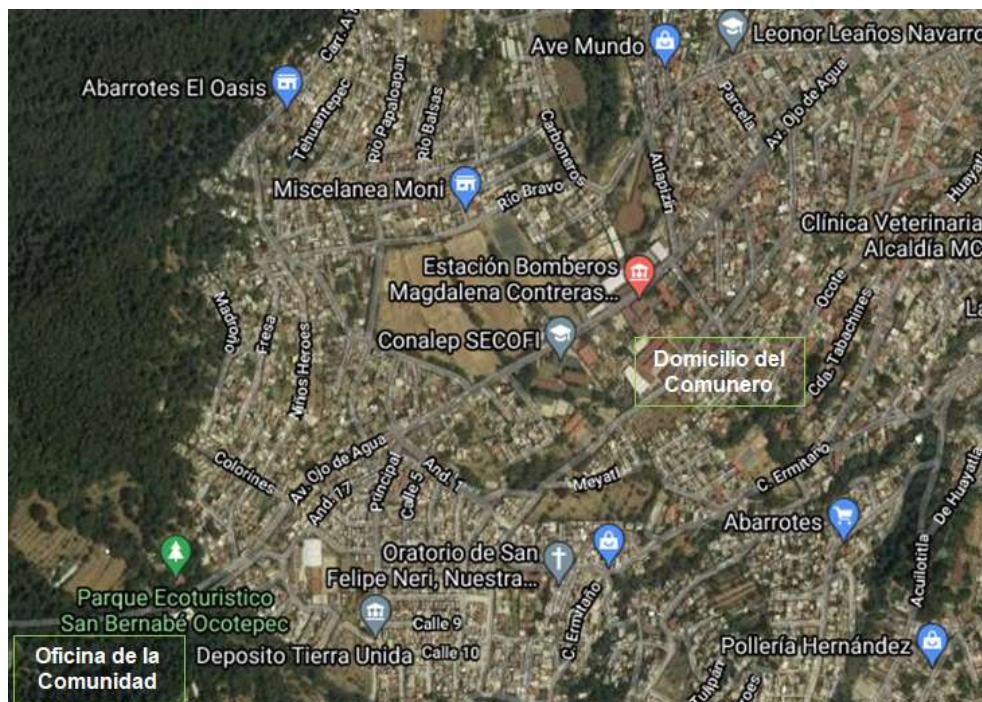


Figura 48. Ejemplo de zona aledaña a la Ciudad de México.

Retomando el ejemplo anterior, se considera un núcleo agrario, que es una zona comunal donde hay una oficina de servicio. De la oficina a la zona de cultivo se tienen distancias considerables, como, por ejemplo, en la figura 49 se muestra una distancia de 118 metros al área de cultivo, con senderos que conducen a las tierras de cultivo que se recorren con dificultad por tratarse de terrenos escabrosos y zonas boscosas. Entonces, la idea sería colocar en la oficina el servidor web y el nodo coordinador, para generar la RIS de manera que sea posible cubrir el área de cultivo deseada con diversos sensores. De esta forma, se podría supervisar el área de cultivo desde la oficina, evitando el desplazamiento al área de cultivo.



Figura 49. Zona comunal con oficina y área de cultivo.

El costo total del sistema dependerá de la aplicación que se le dé, a manera de ejemplo, se proporcionan dos casos de aplicación.

4.3.1 Caso de ejemplo 1: tierra de cultivo

Para un terreno de cultivo de 500 metros cuadrados, localizado en una zona boscosa, se consideran 5 radios en modo nodo sensor, ubicados en los extremos de interés o dónde muy posiblemente puede haber exceso o déficit de humedad. Se utilizarían de 8 a 10 radios en modo *router* y 1 nodo coordinador. La ubicación del nodo coordinador debe tener acceso a la tecnología celular y a Internet para que pueda emplearse mi propuesta tal como está descrita en este trabajo. Los detalles estarían en modificar la frecuencia de entrega de datos por parte de los nodos sensor puesto que, al tratarse de una tierra de cultivo, es suficiente la entrega de datos una vez por día, así se prolonga la vida de las baterías. El costo sería de \$25,324.00 m/n con IVA incluido, el presupuesto se muestra en la tabla XIV.

Tabla XIV. Costo del sistema de monitoreo para caso de ejemplo 1⁴

Cantidad	Elemento de red	Hardware	Costo por unidad	Costo
1	Coordinador	Tarjeta de desarrollo Arduino Ethernet	\$860	\$1,575.00
		Xbee S2	\$631	
		<i>shield</i> de Xbee para Arduino	\$84	
9	Router	Arduino Uno	\$607	\$11,898.00
		Xbee S2	\$631	
		<i>shield</i> de Xbee para Arduino	\$84	
1	Router alertador	Arduino Uno	\$607	\$2,021.00
		Xbee S2	\$631	
		<i>Shield</i> de Xbee para Arduino	\$84	
		SIM 900	\$699	
5	End Device	Arduino Uno	\$607	\$7,045.00
		Xbee S2	\$631	
		<i>Shield</i> de Xbee para Arduino	\$84	
		Sensor de humedad	\$13	
		Sensor de temperatura	\$74	
1	Servidor web	Raspberry Pi 2	\$2,500	\$2,500
			Total	\$25,039.00

⁴ AG Electrónica <https://www.agelectronica.com/>

4.2.4 Caso de ejemplo 2: incendio en un bosque

Al tratarse de un bosque se requiere de una mayor cobertura, lo que implicaría una mayor cantidad de nodos, en este caso, se podría pensar en una superficie de 2 Km², para ello, se necesitarían 22 nodos sensor aproximadamente, 44 *router* y 1 coordinador. En este caso, el uso de la red sería por estación del año, puesto que sólo se usaría en temporada de calor y los nodos sensor estarían ubicados en puntos específicos donde se propician los incendios o donde ya se tienen localizados.

En algunos casos, el acceso a Internet estaría limitado, puesto que se necesitarían muchos más nodos *router* para dar la cobertura y llegar a un punto donde se pueda tener acceso, eso elevaría el costo de la red, más aún si se agregan nodos para tener redundancia en el sistema.

El costo estimado para un sistema como el caso de ejemplo 2 sería de \$93,280.00 M/N con IVA incluido, véase la tabla XV para los detalles del presupuesto.

Tabla XV. Costo del sistema de monitoreo para caso de ejemplo 2

Cantidad	Elemento de red	Hardware	Costo por unidad	Costo
1	Coordinador	Tarjeta de desarrollo Arduino Ethernet	\$860	\$1,575.00
		Xbee S2	\$631	
		<i>shield</i> de Xbee para Arduino	\$84	
43	Router	Arduino Uno	\$597	\$56,416.00
		Xbee S2	\$631	
		<i>shield</i> de Xbee para Arduino	\$84	
1	Router alertador	Arduino Uno	\$597	\$2,011.00
		Xbee S2	\$631	
		<i>Shield</i> de Xbee para Arduino	\$84	
		SIM 900	\$699	
22	End Device	Arduino Uno	\$597	\$30,778.00
		Xbee S2	\$631	
		<i>Shield</i> de Xbee para Arduino	\$84	
		Sensor de humedad	\$13	
		Sensor de temperatura	\$74	
1	Servidor web	Raspberry Pi 2	\$2,500	\$2,500
			Total	\$93,280.00

En ambos casos planteados, el mantenimiento es necesario y se piensa que sería para el cambio de baterías y sensores dos veces por año.

Como trabajo futuro se podría llevar a cabo el experimento de monitorear una tierra de cultivo que cumpla con las características descritas en el caso ejemplo 1, así se podría ver el desempeño del sistema en una zona remota real y considerar más variables para obtener un sistema más preciso.

En el caso del proyecto, como se mencionó anteriormente, se utilizó el material con que se contaba en la UACM-SLT, pero podrían adecuarse equipos con la misma tecnología, pero de costo menor, tal es el caso de las tarjetas de desarrollo Nodemcu, éstas tienen un costo de \$150.00 M/N, lo cual disminuye considerablemente la inversión. El cambio entre el proyecto implementado y el que sería más económico, consistiría en reemplazar los nodos sensor con la tarjeta Nodemcu, mientras que el nodo coordinador y el *router* alertador tendrían que mantenerse igual, éste último puesto que la SIM 900 utilizada es compatible solo con Arduino.

Además, si se contara con un *shield* de Xbee para Nodemcu, se garantizaría la conexión, el detalle estaría en el manejo o creación de bibliotecas en la tarjeta Nodemcu para habilitar el envío de tramas entre radios Xbee y después al servidor web.

Conclusiones

El sistema diseñado podría facilitar la monitorización de cultivos en zonas aledañas de la Ciudad de México, al considerar sus resultados de desempeño, entre los cuales se puntualizan los siguientes.

- El sistema puede ampliar su cobertura.

Al implementar y probar la red diseñada en este proyecto usando tecnología Zigbee se obtuvo una distancia máxima entre nodos con línea de vista de 94 m, lo que posibilita una cobertura aproximada de 2,500 m² de manera parcial, puesto que sólo se realiza el monitoreo de 2 puntos con 4 sensores con una red de 8 nodos. Debe destacarse que al haber presencia de obstáculos hay menor potencia de señal recibida por parte de los radios receptores y, por tanto, menor distancia entre radios, de lo contrario habría pérdida en la calidad del enlace. Sin embargo, el sistema diseñado puede incrementar su cobertura dado que es posible agregar un máximo teórico de hasta 255 nodos en esta red, además de agregar más subredes, lo que permite una mayor cobertura por la agregación de nodos.

- El sistema tiene la capacidad de medir humedad y temperatura, sin embargo, se pueden adaptar diversos sensores, según el tipo de cultivo.

Aunque se realizaron pruebas con máximo 2 sensores por nodo sensor, Xbee cuenta con 9 entradas o salidas, 7 de ellas como convertidor ADC, por lo que se podrían incorporar hasta 9 sensores o actuadores, según las necesidades.

- El sistema de almacenamiento, implementado en un servidor propio, guarda y permite la consulta de la base de datos en cualquier momento que se desee, de manera fácil y rápida.

En cuanto a la capacidad de almacenamiento en la base de datos, lo implementado requiere almacenar 65 KB por día. Lo que indica que no es necesario realizar respaldo alguno por posible sobre escritura a falta de memoria. Debe hacerse notar que al tratarse de un servidor web implementado en una tarjeta Raspberry Pi 2 con software libre y códigos abiertos, el servidor queda en dicha tarjeta, por lo tanto, es un servidor web propio, lo cual no genera gasto extra por *hosting*. Así mismo, la

interfaz gráfica utilizada para la consulta de la base de datos resulta fácil de interpretar, puesto que se presenta de forma ordenada y amigable al usuario.

- El costo del sistema diseñado es accesible, pero depende de la extensión del territorio que se desee monitorear.

La utilidad del sistema es evitar el traslado físico por parte de los dueños de las zonas remotas y en caso de tener algún riesgo en las tierras de cultivo o bosque, el sistema da aviso por mensaje y llamada a cualquier lugar donde se encuentre el responsable de dicha área. Para el caso de los adultos mayores, a los que se les dificulta el desplazamiento físico a zonas remotas, el sistema representa una opción de costo-beneficio, ya que, se puede invertir en este proyecto que les evita un agotador desplazamiento o que físicamente implica mayor riesgo en adultos mayores.

La solución propuesta resuelve el problema de recolección de datos planteado, sin embargo, aún falta hacer pruebas en zonas boscosas.

Anexo A

Códigos en Arduino del nodo concentrador y nodo alertador

Código del nodo concentrador

```
#include <Ethernet.h>
#include <SPI.h>
#include <SoftwareSerial.h>
#include <XBee.h>
uint8_t recv = 0;
uint8_t trans = 1;
uint8_t Msb = 0;
uint8_t Lsb = 0;
int LED1=1;
int LED2=4;
int LED3=7;
SoftwareSerial soft_serial(recv, trans);
XBee xbee = XBee();
ZBRxIoSampleResponse ioSample = ZBRxIoSampleResponse();
// Configuracion del Ethernet Shield
byte mac[] = {0x90, 0xA2, 0xDA, 0x0F, 0x94, 0x7E}; // Direccion MAC
byte ip[] = { 192,168,125,103 };// Direccion IP del Arduino
byte server[] = { 192,168,125,105 }; // Direccion IP del servidor
EthernetClient client;
float temperatura;
float humedad;
void setup(void) {
  Serial.begin(9600);
  SPI.begin();
  Ethernet.begin(mac,ip);
  delay(1000);
  while (!Serial);
  xbee.setSerial(soft_serial);
  soft_serial.begin(9600);
  pinMode (LED1,OUTPUT);
  digitalWrite(LED1,HIGH);
  pinMode (LED2, OUTPUT);
```

```

}
void get_temperatura(ZBRxIoSampleResponse *ioSample) {
float adc_data = ioSample->getAnalog(3);
Serial.print("La temperatura es: ");
Serial.print(Msb, HEX);
Serial.print(Lsb, HEX);
float temperatura = 110*(adc_data/1024);
Serial.print(temperatura);
Serial.print("c, ");
}
void get_address(ZBRxIoSampleResponse *ioSample) {
  Msb=ioSample->getRemoteAddress64().getMsb();
  Lsb=ioSample->getRemoteAddress64().getLsb();
Serial.print("Recepcion de datos del radio: ");
//Serial.print(Msb, HEX);
Serial.print(Lsb, HEX);
Serial.println("");
}
void get_humedad(ZBRxIoSampleResponse *ioSample) {
float adc_data1= ioSample->getAnalog(0);
Serial.print("      Estado de humedad: ");
if(adc_data1>=111){
  humedad=1;
  Serial.println("Exceso de humedad");}
else if(adc_data1<=20){
  humedad=0;
  Serial.println("Deficiente");
}
else if(adc_data1>=21 && adc_data1<=110){
  humedad=0.5;
  Serial.println("Optima");
}
Serial.println("Valor: ");
Serial.println(humedad);
Serial.println("");
}
void setRemoteState(char value) {
  Serial.write((byte)0x7E);
}

```

```
Serial.write((byte)0x00);  
Serial.write((byte)0x10);  
Serial.write((byte)0x17);
```

```
Serial.write((byte)0x01);  
Serial.write((byte)0x00);  
Serial.write((byte)0x13);  
Serial.write((byte)0xA2);  
Serial.write((byte)0x00);
```

```
Serial.write((byte)0x40);  
Serial.write((byte)0xCC);  
Serial.write((byte)0x06);  
Serial.write((byte)0x30);
```

```
Serial.write((byte)0xFF);  
Serial.write((byte)0xFE);
```

```
Serial.write(0x02);
```

```
Serial.write(0x44);
```

```
Serial.write(0x34);  
Serial.write(value);
```

```
long sum = 0x17 + 0x01 + 0x13 + 0xA2 + 0x40 + 0xCC + 0x06 + 0x30 + 0xFF + 0xFE + 0x02 + 0x44  
+ 0x34 + value;
```

```
Serial.write( 0xFF - (sum & 0xFF) );
```

```
}
```

```
void loop(void) {
```

```
  xbee.readPacket();
```

```
  if (xbee.getResponse().isAvailable()) {
```

```
    if (xbee.getResponse().getApiId() == ZB_IO_SAMPLE_RESPONSE) {
```

```
      xbee.getResponse().getZBRxIoSampleResponse(ioSample);
```

```
      get_address(&ioSample);
```

```
      get_temperatura(&ioSample);
```

```
      get_humedad(&ioSample);
```

```
      if(temperatura>=27){
```

```

digitalWrite(LED2,HIGH);
setRemoteState(0x5);
delay(5000);
digitalWrite(LED2,HIGH);
setRemoteState(0x4);
delay(5000);
Serial.println("Se esta enviando mensaje por exceso de temperatura");
}
Serial.println("Conectando...");
if (client.connect(server, 80)>0) { // Conexion con el servidor
  client.print("GET /mimo/iot.php?ID=&TEMPERATURA=&HUMEDAD"); // Enviamos los datos por GET
  client.print("&ID=");
  client.print(Lsb, HEX);
  client.print("&TEMPERATURA=");
  client.print(temperatura);
  client.print("&HUMEDAD=");
  client.print(humedad);
  client.println(" HTTP/1.0");
  client.println("User-Agent: Arduino 1.0");
  client.println();
  Serial.println("Conectado");
} else {
  Serial.println("Falla en la conexion");
}
if (!client.connected()) {
  Serial.println("Desconectado!");
}
client.stop();
client.flush();
delay(60000); // Espero un minuto antes de tomar otra muestra
}
else {
Serial.println("Se esperaba muestra I/O, pero se obtuvo");
Serial.print(xbee.getResponse().getApild(), HEX);
}
} else if(xbee.getResponse().isError()) {
Serial.print("Error al leer el paquete, codigo de error: ");

```

```

Serial.println(xbee.getResponse().getErrorCode());
}
}

```

Código del nodo alertador para realizar el envío de un SMS y una llamada

```

#include <SoftwareSerial.h>
#include <String.h>
#include <XBee.h>
int LED1=2;
int LED2=4;
int LED3=12;
SoftwareSerial SIM900(7,8);
XBee xbee = XBee();
char incoming_char=0;
int salir=0;
void setup(){
  SIM900.begin(19200);
  delay(25000);
  Serial.begin(19200);
  Serial.println("OK");
}
void mensaje_sms(){
  SIM900.print("AT+CMGF=1\r");
  delay(100);
  SIM900.println("AT + CMGS = \"+52554090xxxx\"");
  delay(100);
  SIM900.println("RIS en riesgo, temperatura elevada");
  delay(100);
  Serial.println("Envio de mensaje en proceso...");
  SIM900.println((char)26);
  delay(100);
  SIM900.println();
  delay(5000);
  Serial.println("Mensaje enviado exitosamente");
}
void llamar(){
  SIM900.println("ATD 52044554090XXXX;");
}

```

```
delay(30000);
SIM900.println("ATH");
delay(1000);
}
void loop(){
  xbee.readPacket();
  digitalWrite(LED1,HIGH);
  delay(3000);
  digitalWrite(LED1,LOW);
  delay(3000);
  if (xbee.getResponse().isAvailable()){
    digitalWrite(LED2,HIGH);
    delay(3000);
    if(xbee.getResponse().getApiId()==REMOTE_AT_COMMAND_RESPONSE){
      digitalWrite(LED3,HIGH);
      delay(3000);
    }
    mensaje_sms();
    delay(3000);
    llamar();
    delay(3000);
  }
}
```

Referencias

- [1] J. A. Ramírez, J. A. Buitrago y J. I. Marín, «Red de sensores de larga distancia usando zigbee para el monitoreo y la gestión del riesgo en el departamento del Quindío- Colombia,» *Revista De Investigaciones Universidad Del Quindío*, vol. 25, nº 1, pp. 63-72, 2014.
- [2] ENDUTIH, «Encuesta Nacional sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares 2019,» INEGI, 15 Mayo 2019. [En línea]. Available: <https://www.inegi.org.mx/rnm/index.php/catalog/556>. [Último acceso: 27 Agosto 2020].
- [3] D. D. Constain Prado, «Cobertura y capacidad en redes 2G, 3G y 4G,» 7 Febrero 2019. [En línea]. Available: <http://hdl.handle.net/11349/14691>. [Último acceso: 22 Abril 2020].
- [4] S. Medina, F. Romero, A. E. De Giusti y F. G. Tinetti, «Análisis para despliegue de una red de sensores heterogénea,» de *XXIV Congreso Argentino de Ciencias de la Computación (La Plata)*, Buenos Aires, 2018.
- [5] Á. Villalba Macías, *Aplicación de una red inalámbrica de sensores para detección de presencia y conteo*, Sevilla, 2015.
- [6] N. Aakvaag y J. E. Frey, «Redes de sensores inalámbricos,» *Revista ABB*, pp. 39-42, 2006.
- [7] N. D. Castro, L. E. Chamorro y C. A. Viteri, «Una red de sensores inalámbricos para la automatización y control del riego localizado,» *Ciencias Agrícolas*, vol. 33, nº 2, pp. 106-116, 2016.
- [8] L. Iacono, P. Godoy, O. Marianetti y C. García Garino, «Estudio de plataformas de hardware empleadas en redes de sensores inalámbricas,» de *XVI Congreso*

Argentino de Ciencias de la Computación, La Plata, 2010.

- [9] C. Bell, *Beginning sensor networks with Arduino and Raspberry Pi*, Apress, 2014.
- [1 J. Prieto Blázquez, *Introducción a los sistemas de comunicación inalámbrica*,
0] Catalunya: Universitat Oberta de Catalunya, 2011.
- [1 J. P. Dignani, *Análisis del protocolo ZigBee*, La Plata, 2012.
1]
- [1 J. Salazar Soler, *Redes Inalámbricas*, 2016.
2]
- [1 K. C. Sahoo y U. C. Pati, «Sistema de detección de intrusos basado en IoT
3] mediante sensor PIR,» de *2da Conferencia Internacional IEEE sobre Tendencias
Recientes en Electrónica, Tecnología de la Información y la Comunicación
(RTEICT)*, Bangalore, India, 2017.
- [1 A. M. Ortiz Torres, *Técnicas de enrutamiento inteligente para redes de sensores
4] inalámbricas*, Albacete: Tesis Doctoral, 2011.
- [1 M. A. Vallejo Velásquez, P. A. Gómez Puerta y G. A. Posada Pérez, «Evaluación
5] del desempeño del protocolo de enrutamiento AODV para diferentes escenarios
de redes de sensores inalámbricos,» *Científica Ingeniería y Desarrollo*, vol. 32, nº
1, pp. 80-101, 2014.
- [1 «Digi.com,» [En línea]. Available:
6] https://www.digi.com/resources/documentation/Digidocs/90002002/Content/Reference/r_api_frame_format_900hp.htm. [Último acceso: 8 Marzo 2018].
- [1 J. A. Ruiz Rivas y F. Reina Toranzo, «Redes de área local,» *Redes Locales*,
7] 2017.
- [1 R. Carrasquero, «Utilización de una L.A.N (red de área local) para propiciar una

- 8] comunicación efectiva en la gestión del programa nacional de formación en informática en el IUTET,» *Educare* , vol. 16, nº 2, pp. 50-70, 2015.
- [1 K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi y J. A. 9] Maestro, «IEEE 802.3az: el camino hacia ethernet energéticamente eficiente,» *IEEE Communications Magazine*, vol. 48, nº 11, pp. 50-56, 2010.
- [2 A. Abad, «La instalación física de una red,» Recuperado de 0] <http://assets.mheducation.es/bcv/guide/capitulo/8448180828.pdf>, 2015.
- [2 F. J. Mereci Apolo, «Diseño e implementación de una celda celular con 1] tecnologías 2G, 3G y LTE para la ciudad de San Jose de Chimbo en la provincia de Bolivar,» EPN, Quito, 2020.
- [2 F. W. Villao Quezada, L. A. Naula Yungán y J. J. Ruiz Daker, Diseño de la 2] conectividad para dotar del servicio universal para la Isla Puná, Cantón Guayaquil, Provincia del Guayas, utilizando servicio móvil avanzado, tecnología 3G, Guayaquil: Espol, 2017.
- [2 «Flylib.com,» [En línea]. Available: <https://flylib.com/books/en/4.215.1.33/1/>. 3] [Último acceso: 25 agosto 2020].
- [2 M. Ballesteros García, «Evidencia de la eficiencia espectral entre UMTS y LTE,» 4] Valencia, 2019.
- [2 E. J. Pérez Solórzano, «Propuesta de utilización de metro celdas para la 5] descongestión de redes 3G,» Guatemala, 2015.
- [2 E. Huancollo Aracca, *Desarrollo de una aplicación Web para el control y Registro 6] de Pacientes del Centro Odontológico Dental Plus de la Ciudad de Juliaca*, Juliaca, 2017.
- [2 J. C. Saavedra Chapilliquen, «Diseño de un servidor web utilizando hardware

7] libre,» Perú, 2019.

[2 J. Márquez Díaz, L. Sampedro y F. Vargas, «Instalación y configuración de
8] Apache, un servidor Web,» *Revista Científica Ingeniería y Desarrollo*, vol. 12, nº
12, pp. 10-23, 2011.

[2 Á. Cobo, PHP y MySQL: Tecnología para el desarrollo de aplicaciones web,
9] Ediciones Díaz de Santos, 2005.

[3 J. M. Lozano Banqueri, Creación y gestión de una base de datos con MySQL y
0] phpMyAdmin, Jaén: Universidad de Jaén, 2018.

[3 M. Lloret Bachs, *Nodo IOT de captación de datos contaminantes atmosféricos.*
1] *Licenciatura. Facultat d'Informàtica de Barcelona - Grau en Enginyeria*
Informàtica., Barcelona, 2019.