



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

IMPLEMENTACIÓN DEL MODO API EN UNA RED DE SENSORES WSN
PARA LA MEDICIÓN DE CONTAMINACIÓN AUDITIVA



AUTOR

Milton Vinicio Cacuango Cacuango

AÑO

2017



FACULTAD DE INGENIERÍAS Y CIENCIAS AGROPECUARIAS

IMPLEMENTACIÓN DEL MODO API EN UNA RED DE SENSORES WSN
PARA LA MEDICIÓN DE CONTAMINACIÓN AUDITIVA.

Trabajo de titulación presentado en conformidad con los requisitos establecidos
para optar por el título de Ingeniero en Redes y Telecomunicaciones

Profesor Guía

MSc. Jorge Wilson Granda Cantuña

Autor

Milton Vinicio Cacuango Cacuango

Año

2017

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante orientando sus conocimientos y competencias para un eficiente desarrollo del tema elegido y cumpliendo con todas las disposiciones vigentes que regulan los Trabajos de titulación”

Jorge Wilson Granda Cantuña
Máster en Ingeniería Eléctrica
C.I.: 1708594187

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante orientando sus conocimientos y competencias para un eficiente desarrollo del tema elegido y cumpliendo con todas las disposiciones vigentes que regulan los Trabajos de titulación”

Milton Neptali Román Cañizares
Magister en Gerencia de Redes y Telecomunicaciones
C.I.: 0502163447

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”

Milton Vinicio Cacuango Cacuango

C.I.: 1723255756

AGRADECIMIENTO

Agradezco a mis padres, mis hermanos y a toda mi familia por su amor y comprensión. Además, por su apoyo económico y moral para culminar mis estudios. También agradezco a la empresa LA COMPETENCIA S.A. y a sus directivos por su apoyo.

De manera especial y sincera expreso mi agradecimiento al Ing. Jorge Granda mi director de tesis por su valiosa orientación y apoyo, por el tiempo invertido en la dirección y corrección de este proyecto para la culminación del mismo.

DEDICATORIA

A mi padre Agustín Cacuango, mi cuñada Magdalena Lema mi hermano Dario Cacuango. También a un ser querido que ya no se encuentra conmigo, mi madre Antonia Cacuango Tito pero que ha sido mi mejor motivación para seguir adelante. A toda mi familia, a mis hermanos y amigos quienes a lo largo de toda mi vida me han apoyado y motivado en mi formación académica, creyeron en mí en todo momento.

RESUMEN

La contaminación auditiva en el Ecuador es un fenómeno que llega a afectar a miles de personas, provocando molestias de diferentes índoles en la salud. Estos problemas son causados por diferentes factores como: el tráfico automotriz, aviones al aterrizar y despegar, equipos de construcción, fábricas, eventos musicales entre otros. Según la Organización mundial de la salud considera que la medida de 65 dB es el límite superior tolerable del ruido, al sobrepasar este límite se considera perjudicial para la salud. Llegando a definir a la contaminación auditiva como ruido desagradable que alteran las condiciones normales del ambiente creado por los factores ya mencionados.

El ruido excesivo es un factor que llega a provocar varias consecuencias negativas en la salud, tales como: problemas de estrés mental, pérdida de audición, falta de sueño (insomnio), variación en la presión arterial, problemas en el sistema nervioso, falta de productividad y afectación en la salud mental. Por estos motivos en el presente proyecto se realizó un sistema de medición de contaminación auditiva, implementando el modo API en una red de sensores WSN (Wireless Sensor Networks). Además, se mejoró un trabajo de tesis similar denominado "SISTEMA DE MEDICIÓN DE CONTAMINACIÓN AUDITIVA, EMPLEANDO UNA RED DE SENSORES INALÁMBRICOS Y SENSORES BAN", realizado en el 2016, por Santiago Solórzano estudiante de la Universidad de las Américas. El proyecto consta de tres etapas: En la primera fase se realizó la construcción y diseño de un prototipo de medición de contaminación auditiva, el mismo que está compuesto de un sonómetro que se encarga de medir el nivel de ruido en un determinado lugar, un GPS para determinar la posición del nodo y un dispositivo inalámbrico XBee para la transmisión de los datos aproximadamente a 1.2Km de longitud desde el Gateway al nodo.

En la segunda etapa se realizó el diseño de la trama API, para la transmisión de datos desde el nodo al coordinador y viceversa. La tercera etapa se basó del diseño de una interfaz gráfica, realizada en el software Visual Studio, en cual se puede visualizar los datos obtenidos del nivel de ruido, posición de GPS y distancia aproximada del nodo con respecto al coordinador.

ABSTRACT

Auditory pollution in Ecuador, is a phenomenon that affects thousands of people, causing discomfort of different kinds in health. These problems are caused by different factors such as: automotive traffic, aircraft landing and taking off, construction equipment, factories, musical events, among others. According to the World Health Organization, the 65 dB measure is the tolerable upper limit of noise, exceeding this limit is considered to be harmful to health. The definition of auditory pollution as an unpleasant noise that alter the normal conditions of the environment created by the factors already mentioned.

Excessive noise is a factor that leads to a number of negative health consequences, such as: mental stress problems, hearing loss, lack of sleep (insomnia), variation in blood pressure, nervous system problems, lack of Productivity and affectation in mental health. For these reasons in the present project a system of measurement of auditory contamination was realized, implementing the mode API in a network of sensors WSN (Wireless Sensor Networks). In addition, a similar thesis work called "AUDITIVE CONTAMINATION MEASUREMENT SYSTEM, EMPLOYING A NETWORK OF WIRELESS SENSORS AND BAN SENSORS", made in 2016, by Santiago Solórzano, student at the University of the Americas, was improved. The project consists of three stages: on the first phase was carried out the construction and design of a prototype of measurement of auditory pollution, which is composed of a sound level meter that is responsible for measuring the noise level in a certain place, a GPS To determine the position of the node and an XBee wireless device for transmitting data about 1.2Km in length from the Gateway to the node.

In the second stage the design of the API frame was carried out, for the transmission of data from the node to the coordinator and vice versa. The third stage was based on the design of a graphical interface, made in Visual Studio software, in which the data obtained from the noise level, GPS position and approximate distance of the node with respect to the coordinator can be visualized.

ÍNDICE

Introducción.....	1
Alcance.....	2
Justificación	2
Objetivos.....	3
Objetivo General.....	3
Objetivos específicos.....	3
1. CAPÍTULO I. MARCO TEÓRICO.....	4
1.1 Contaminación Acústica	4
1.1.1 La contaminación acústica y su legislación	8
1.1.2 Sonido.....	11
1.1.3 Ruido	15
1.1.4 Afectación de Ruido.....	22
1.2 Redes Inalámbricas	23
1.2.1 Red WSN.....	23
1.2.2 Componentes de un nodo WSN	25
1.2.3 Protocolos de Enrutamiento.....	26
1.2.4 Tipos de Topología	27
1.3 Dispositivo XBee.....	30
1.3.1 Modos de funcionamiento del XBee	32
1.3.2 Modo Serie	34
1.3.3 Modo Transparente (AT).....	34
1.3.4 Modo API (Application Programming Interface).....	36
1.3.5 Estructura de la Trama API.....	38
1.3.6 Modo API 1 y API 2.....	40
1.3.7 Estructura de la trama API 1	40
1.3.8 Estructura de la trama API 2	42
1.3.9 Comandos AT en la trama API	43
1.4 RSSI en XBee.....	47

1.5 Software XCTU	48
1.6 Arduino IDE	49
1.7 Visual Studio.....	50
2. CAPÍTULO II. DISEÑO E IMPLEMENTACIÓN	
DEL SISTEMA	50
2.1 Diseño en bloques del sistema de medición de contaminación auditiva	51
2.2. Estudio comparativo de hardware y software.....	52
2.2.1 Elementos de Hardware	52
2.2.2 Elementos de software	60
2.3 Base de datos.....	61
2.3.1 SQL Server	62
2.4 Diseño del sistema de medición de contaminación auditiva	63
2.4.1 Diseño del nodo sensor	65
2.5 Diseño de la red Zig Bee en modo mesh	72
2.5.1 Enrutamiento de la red Mesh.....	73
2.5.2 Configuración de la red WSN en modo mesh.....	74
2.6 Estructura de la programación en Arduino del nodo Principal.....	80
2.7 Protocolo para el diseño de la trama API.....	85
2.8 Diseño de las tramas API	88
2.8.1 Trama para encender remotamente el sonómetro.....	88
2.8.2 Trama para apagar remotamente el sonómetro.....	90
2.8.3 Trama para enviar los datos del sonómetro y del GPS a la interfaz visual desde Arduino	92
2.8.4 Trama para obtener los datos del nivel de RSSI.....	94
2.10 Diseño e implementación de la interfaz gráfica	97
2.10.1 Capa Presentación	98

2.10.2 Capa Negocio	99
2.10.3 Capa Conexión	100
2.10.4 Capa de origen de Datos	100
2.10.5 Diagrama de flujo de la interfaz gráfica.....	101
2.11 Presentación del prototipo final.....	108
3. CAPÍTULO III. PRESENTACIÓN DE	
RESULTADOS	111
3.1 Calibración de Equipos	111
3.2 Protocolos para realizar la prueba Mesh.....	112
3.2.1 Pruebas de la red Mesh.....	114
3.3 Protocolos para las pruebas de RSSI	121
3.3.1 Pruebas de RSSI	124
3.4 Pruebas del prototipo de Ruido.....	128
4. CAPÍTULO IV. ANÁLISIS DEL COSTO DE	
IMPLEMENTACIÓN DEL SISTEMA DE	
CONTAMINACIÓN AUDITIVA	132
4.1 Costos de materiales	132
4.2 Costos de Programación	133
4.3 Costos de diseño	133
4.4 Costo total del proyecto	134
5. CONCLUSIONES Y RECOMENDACIONES.....	135
5.1 Conclusiones.....	135
5.2 Recomendaciones	136
REFERENCIAS.....	138
ANEXOS	142

Introducción

El mundo de las comunicaciones inalámbricas se encuentra en constante evolución, generando cada día nuevas tecnologías y mejorando los sistemas actuales. Uno de los principales cambios es la evolución desde tiempos antiguos, empezando desde la comunicación punto a punto de muy corto alcance por medio de la tecnología infrarroja, luego pasando por la tecnología multipunto por medio del Bluetooth, estos sistemas permitían muy poca capacidad de transmisión de datos. En la actualidad existen redes que tienen un mayor alcance, más cobertura y una alta tasa de transmisión de datos con grandes velocidades. Así tenemos la tecnología WI-FI, WSN, telefonía celular, Wimax entre otras.

Entre las tecnologías que se ha desarrollado a gran escala se encuentran las redes de sensores WSN; estos dispositivos se interconectan a través de las ondas de radio. Una de las principales funciones es la captación y procesamiento de variables análogas y digitales como: niveles de ruido, temperatura, humedad, velocidad. Actualmente son muy utilizados en la industria, la agricultura, la agro industria, la robótica, la electrónica y para proyectos de monitoreo de señales. Su bajo costo y su fácil implementación los hace prácticos de utilizar. Lombardero, L (2015)

Las redes WSN (redes de sensores inalámbricos), pueden trabajar en diferentes topologías, su estructura de funcionamiento está basado en nodos remotos administrados por un coordinador, además de routers que administran y enrutan la información. Los nodos finales son los encargados de recolectar la información de su entorno, después estos datos son enviados directamente al nodo coordinador en el caso de estar enlazados directamente, si no lo están hacen uso de los otros nodos para llegar a su destino.

La contaminación auditiva en la actualidad es un fenómeno que se da a nivel mundial sobre todo en las grandes ciudades. Donde existen altísimos índices

de contaminación auditiva causados por: el tráfico de la ciudad, aviones al despegar y aterrizar, la construcción, locales de ocio, eventos musicales, sector automotriz. Los mismos que puede llegar afectar de forma negativa el sistema central nervioso a través del oído. Sobre todo, al sobrepasar los límites establecidos, pudiendo producir problemas de salud irreversibles como: sordera y en varios casos efectos patológicos, los mismo que pueden ser instantáneos o a futuro.

En el presente proyecto se realizó un prototipo de medición de contaminación auditiva en modo API utilizando una red mesh que proporciona redundancia para la comunicación entre los nodos.

Alcance

La investigación del presente proyecto se efectuó con información de tesis, libros, páginas web y artículos científicos; el alcance del proyecto de titulación consiste en el desarrollo de una red WSN (Wireless Sensor Networks) tipo Mesh, con el protocolo de comunicación más idóneo, transmitiendo datos empaquetados utilizando modo API.

Además, se va realizar el diseño e integración de un software de escritorio para PC, en donde se muestra gráficamente los datos obtenidos de la medición de ruido, desde los diferentes nodos remotos.

Para finalizar se realizó el análisis de resultados obtenidos en las zonas con mayor contaminación auditiva, y emitir las recomendaciones y conclusiones del proyecto.

Justificación

La contaminación auditiva es un fenómeno que actualmente afecta a miles de personas sobre todo en las grandes ciudades. Exponerse a altos índices de ruido puede llegar a dañar de manera crítica la salud de los seres humanos y perjudicar el rendimiento físico y mental; tanto en el área laboral como estudiantil. Por lo tanto, surge la necesidad de realizar una medición constante

del nivel de ruido en lugares específicos como: guarderías, colegios, universidades, hospitales, empresas y sitios con un alto índice de concurrencia de personas. Entre las zonas con mayor índice de ruido están las siguientes:

- Aeropuertos
- Hospitales
- Instituciones educativas
- Principales carretas
- Empresas de construcción
- Empresas del sector industrial
- Discotecas y bares.

Por este motivo se necesita de un sistema de medición de contaminación auditiva, que se encuentre equipado con sonómetros, los mismos que se encargan de medir el nivel de ruido. Además, que se pueda monitorear los resultados desde una PC.

Objetivos

Objetivo General

- Diseñar un prototipo de red WSN para la medición de ruido en zonas afectadas por contaminación auditiva.

Objetivos específicos

- Diseñar e implementar una red de sensores inalámbricos tipo Mesh con la utilización de: un coordinador, nodos enrutadores, dispositivos finales, y protocolos de enrutamiento.

- Implementar una interfaz gráfica en un computador, que permita mostrar los niveles de ruido de las zonas monitoreadas.
- Analizar y diseñar la trama API que permitirá enviar la información empaquetada, desde los sensores hacia el Gateway y posteriormente mostrar en un software los resultados.
- Validar el funcionamiento de la red WSN, midiendo el nivel de contaminación auditiva existente en un sitio determinado.
- Mostrar los resultados y costos del proyecto.

1. CAPÍTULO I. MARCO TEÓRICO

1.1 Contaminación Acústica

Según Escuntar, “el sonido es un elemento importante del sentido y la convivencia de las diferentes ciudades, países en que las personas han desarrollado su vida, considerado como un factor indiferente tanto en las políticas de planificación como en el diseño urbanístico. El incesante crecimiento de las naciones sobre todo de los más países más desarrollados como: EEUU, China ha sido paralelo con la contaminación acústica. Permitiendo entender la situación actual del ambiente sonoro al que asimilamos como ruido, que incluso en otras décadas ha llegado a tener una estimación positiva y características de las sociedades modernas y dinámicas”. (Escuntar, 2017)

En los tiempos modernos esta percepción se está superando, debido a que los gobiernos están implementando estudios y políticas en la legislación para combatir la contaminación auditiva.

Según la Organización mundial de la Salud (OMS), los efectos nocivos tanto psicológicos como fisiológicos, son resultados de la exposición prolongada a

ruidos en lugares que exigen un alto nivel de concentración de personas como: lugares de diversión (discotecas, bares), eventos deportivos, eventos musicales, empresas constructoras, tráfico de la ciudad etc. En la figura 1 se muestra los diferentes niveles de ruido en el ser humano.

Sin embargo, si la exposición al ruido es ocasional, sus efectos pueden quedar reducidos a una molestia transitoria, pero si es de forma continua y mayor a 85dB; se deberá asumir sus diferentes efectos en la salud. Dentro de los cuales están los problemas de carácter psicológico como: cansancio, tensión, ansiedad, alteración del sueño, conductas agresivas etc. También problema de aspecto fisiológico: insomnio, afecciones cardiovasculares, pérdida de capacidad auditiva, daños irreversibles en el oído entre otros.

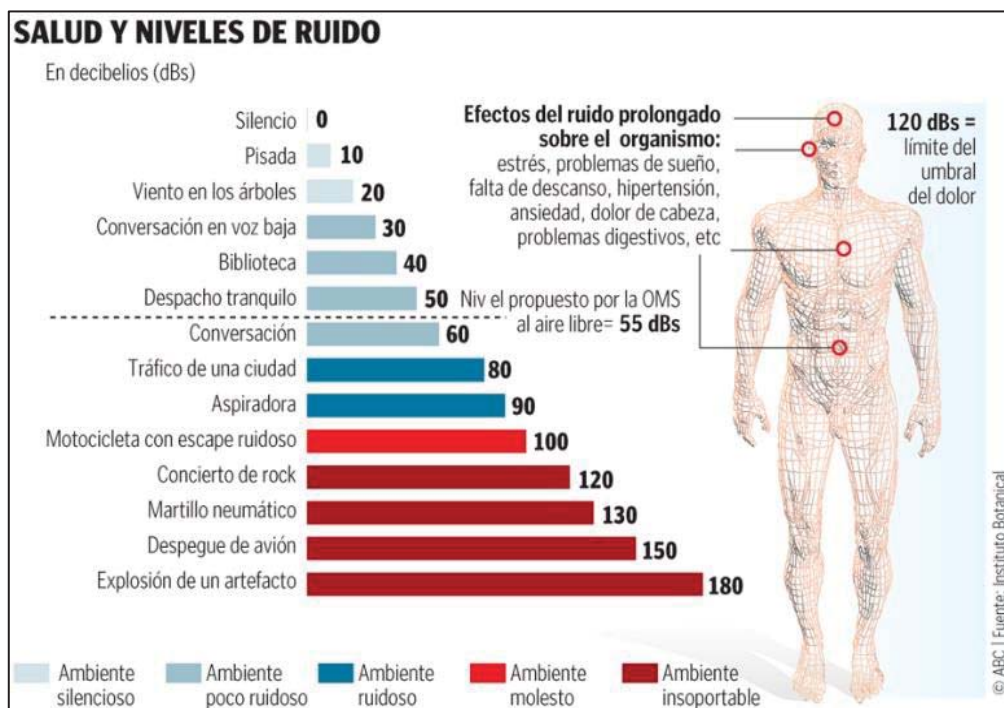


Figura 1. Niveles de ruido en el ser humano.

Tomado de (Hisco, s.f.)

En pocas palabras el ruido tiene efectos perjudiciales en el confort y en la salud de las personas. También se debe destacar la gran capacidad del ruido para desplazarse en el espacio y grandes áreas, debido a sus propiedades físicas

que le caracterizan. Por lo tanto, la combinación de las dos características hace que el ruido deba ser considerado como agente de contaminación medio ambiental. Y que sea necesario aplicar un control adecuado del mismo, para no exceder los límites aceptables. A continuación, en la tabla 1 se muestra un resumen de los valores críticos de ruido.

- **Leq**: Nivel equivalente durante la medición.
- **Leq24**: Nivel equivalente durante 24 horas
- **LeqA**: Nivel equivalente con compensación de frecuencia A.
- **LeqC**: Nivel equivalente con compensación de frecuencia C.
- **dBA**: Decibel compensación A.
- **Lmax**: Máximo nivel con una dada respuesta (rápida, lenta o impulsiva).
- **h**: Horas.
- **X¹**: Nivel de pico L peak, no LAfmax, medido a 100 mm del oído.
- **X²**: Utilizando auriculares, valores adaptado a campo libre.
- **X³**: Deben preservarse las áreas exteriores existentes, y mantener una baja relación entre ruidos intrusivos y ruido de fondo natural.

Tabla 1.

Efectos del Sonido.

Ambiente Especifico	Efectos Críticos sobre la salud	Leq [dBA]	Base de tiempo [h]	Lmax [dBA]
Exteriores de zonas de vivienda	Seria molestia de día y al atardecer	55	16	-
	Molestia moderada de día y al atardecer	50	16	-
Interior de vivienda	Inteligibilidad de la palabra y molestia moderada, de día y al atardecer	35	16	-
	Perturbación del			

Interior dormitorios	sueño de noche	30	8	45
Exterior dormitorios	Perturbación del sueño, ventana abierta (valores exteriores)	45	8	60
Aulas escolares y preescolares, interior	Inteligibilidad de la palabra, perturbación de la extracción de la información, y la comunicación de mensajes	35	Durante las clases	-
Dormitorios preescolares, interior	Perturbación del sueño	30	En horas de sueño	45
Patio de recreo escolar, exterior	Molestia (fuentes externas)	55	Durante los juegos	-
Hospital, dormitorio de guardia, interior	Perturbación del sueño de noche	30	8	40
	Perturbación del sueño, de día y atardecer	30	16	-
Hospitales, habitaciones interiores	Interferencia con el descanso y recuperación	Lo mejor posible	-	-
Áreas industriales, comerciales y de tránsito, interior y exterior	Daño auditivo	70	24	110

Ceremonias, festivales, actos de entrenamiento	Daño auditivo (concurrentes: <5 veces por año)	100	4	110
Sistemas públicos de sistemas de refuerzo sonoro, exteriores e interiores	Daño auditivo	85	1	110
Música y otros sonidos a través de auriculares	Daño auditivo (valor de campo libre)	85 ⁽²⁾	1	110
Sonidos impulsivos de juguetes, pirotecnia, armas de fuego	Daño auditivo adultos	-	-	140 ⁽¹⁾
	Daño auditivo niños	-	-	120 ⁽¹⁾
Exteriores en parques y reservas naturales	Perturbación de la tranquilidad	(3)	-	-

Adaptado de (World Health Organization, 2017)

1.1.1 La contaminación acústica y su legislación

La creciente sensibilidad social, hacia la contaminación ambiental ha llevado a la creación de normativas relacionadas con los niveles máximos de ruido permitido. En el Ecuador se está empezando a tomar medidas legales para mejorar esta situación.

Según la Ordenanza 213 expedida en el año 2007, del Distrito Metropolitano de Quito Ordenanza Sustitutiva del Título V “Del Medio Ambiente” determina:

Art. 11.358.- **Ámbito.** - Las normas de este capítulo se aplicarán a las personas naturales y jurídicas, públicas y privadas, cuyas actividades produzcan u originen emisiones contaminantes de ruido y vibraciones, que no estén contemplados en capítulo V de la Ordenanza, provenientes de fuente emisoras de ruido, móviles y fijas.

En el segundo capítulo de la Ordenanza municipal indica que; todo los que generen sonidos graves deberán estar sujetos a emitir informes y ser sancionados, en el caso de que un automotor sobrepase los niveles permisibles de ruido y deben realizar los cambios adecuados en el plazo máximo de 30 días. También están censurado realizar eventos automovilísticos en predios urbanos y lugares ecológicos. En el caso de no acatar estarán sancionados con el artículo 371 del mismo capítulo en el que dice: “Art. II.371.- Las infracciones a lo dispuesto en los artículos II.365.a., II.365.d., II.365.e., II.365.f., II.370 y II.370.a., se sancionarán con multa de 0,20 a 1,00 RBUM., para los casos previstos en el artículo II.365.f, y en aquellos lugares donde mantengan equipos que superen los límites permitidos de ruido, la Comisaría de Salud y Ambiente y la Unidad de Protección Ambiental de la Policía Nacional, procederán al retiro inmediato de los mismos.”

La normativa específica que existirá sanciones fuertes para las personas que no acaten y no apoyen al buen vivir, además se menciona en este mismo documento que las recaudaciones por este concepto se destinaran en un 50% para un fondo ambiental, con el objetivo de realizar programas de estudio de concientización de la contaminación acústica. Mientras que la parte restante se transferirá a la Unidad de Protección Ambiental de la Policía Nacional previo a la presentación de un proyecto. (Consejo Metropolitano de Quito, 2013). En la tabla número 2, se muestra los niveles máximos permitidos y en la tabla número 3 se indica los niveles de presión sonora.

Tabla 2.

Niveles máximos permitidos para vehículos automotores.

Categoría de vehículo	Descripción	Velocidad del motor en la prueba (rpm)	Máximo dBA
Motocicletas o similares	Motocicletas, tricars, cuadrones y vehículos de transmisión de cadena con motores de 2 o 4 tiempos.	De 4.000 a 5.000	90
Vehículos livianos	Automotores de 4 o más ruedas con un peso neto vehicular inferior a 3500 kg.	De 2500 a 3500	88
Vehículos pesados para carga	Automotores de 4 o más ruedas destinados al transporte de carga, con peso superior o igual a 3500 kg.	De 1500 a 2500	90
Buses, Busetas	Automotores pesados destinados al transporte de personas, con un peso neto vehicular superior o igual a 3500 kg.	De 1500 a 2500	90

Adaptado de (Consejo Metropolitano de Quito, 2013)

Tabla 3.

Nivel de Presión Sonora (NPS).

Tipos de zona según el uso del suelo	Nivel de presión sonora equivalente NPS (dBA)	
	De 06h00 a 20h00	De 20h00 a 06h00
Zona Equipamientos y Protección (1)	45	40

Zona Residencial	50	35
Zona Residencial Múltiple (2)	55	45
Zona Industrial (1)	60	50
Zona Industrial (2) (3)	65	55
Zona Industrial (3) (4)	70	60

Adaptado de (Consejo Metropolitano de Quito, 2013)

De acuerdo con el artículo 86, Sección segunda del medio Ambiente de la Carta Magna del Ecuador indica que el “El Estado protegerá el derecho de la población a vivir en un medio ambiente sano y ecológicamente equilibrado, que garantice un desarrollo sustentable. Es decir, realizará las leyes que sean necesarias para que este derecho no sea afectado y garantizará la conservación de la naturaleza.” (Asamblea Nacional, 2008).

Por lo tanto, en la ciudad de Quito, existe la justificación legal por medio de la ordenanza 213, para realizar estudios sobre la contaminación acústica y sancionar a las fuentes que lo provoquen.

1.1.2 Sonido

El sonido puro se describe como las ondas de presión que se transmiten a través de un medio: casi siempre el aire o a través de elementos sólidos y líquidos que permitan su propagación. Estas ondas se caracterizan por estar compuestas de amplitud y frecuencia. En la figura 2, se puede ver una onda de sonido puro, es de forma sinusoidal, la misma que tiene presiones positivas y negativas dentro de un ciclo. Según Vesilind, Morgan, Heine (2013, p. 514) el número de estos de estos ciclos por unidad de tiempo se llaman frecuencia del sonido, y en general se expresa como ciclos por segundo, o Hertz.

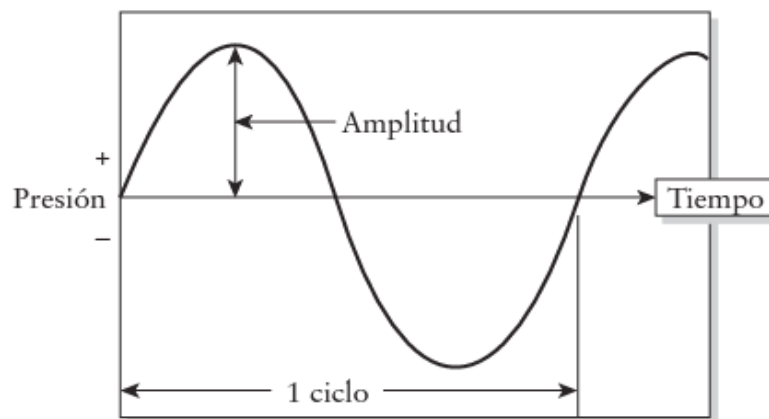


Figura 2. El sonido puro viaja como una onda sinusoidal perfecta.
Tomado de (Sonido Digita, 2017)

La edad de las personas es un aspecto crítico al momento de captar los sonidos, a medida que los seres humanos envejecen también disminuye la probabilidad de escuchar ciertas frecuencias. El oído de las personas sanas percibe en un rango desde 20 Hz hasta 20,000 Hz. Las fuentes de ruido más perjudiciales son las ocupacionales y la música fuerte: en particular los conciertos de rock y los audífonos con alto volumen. La unidad de medida del sonido es el decibelio (dBA) y cuando mayor sea el sonido mayor será el valor en decibelios. (Escuntar, 2016)

Para realizar mediciones en decibelios se utiliza la siguiente fórmula.

$$dB = 20 \log \frac{P_x}{P_0} \text{ (Nivel de presión del sonido } x) \quad \text{(Ecuación 1)}$$

P_x = es la presión sonora eficaz (RMS), para cualquier sonido.

$P_0 = 2 * 10^5 \text{ Pa}$ (**P₀** es presión de referencia y **P_a** es Pascal) la intensidad se calcula comparando con un nivel arbitrario de presión sonora.

Frecuencia (Hz): Según Leonardo, R. (2007, p 1.3) se define como el número de ondas que existen en la unidad de tiempo (segundos), se mide en Hertzios. También es un factor que permite determinar la agudeza del sonido, ejemplo frecuencias altas corresponden a tonos graves y las bajas corresponden a tonos agudos.

Decibel: es denominada la medida de presión que posee el sonido, también es utilizada como unidad logarítmica y no lineal, la misma que es utilizada en diferentes áreas como unidad de medición. Leonardo, R. (2007, p 1.11)

- **SPL:** Nivel de presión sonora.

En la tabla 4 se muestra el nivel típico de la presión sonora del sonido.

Tabla 4.

Niveles típicos de presión sonora del sonido.

Sonido típico	SPL (Nivel de Presión sonora) dB
Umbral de la audición	0
Interior de una cámara de examen de audición	10
Área remota de Yellowstone	20
Biblioteca	40
Subdivisión suburbana no cercana a ninguna fuente importante de ruido	45
Salón de clases típico	50
Habla normal	60
Oficina ocupada	65
Alarma cerca del despertador cerca de la cabeza	80
Trafico promedio de la calle	85
Podadora	90
Camión individual que pasa a 50 pies de distancia	90
Concierto de Rock	110
Llanto promedio de un recién nacido	110
Paso elevado de un avión caza F-16 al despegar	120
Cohete Saturn al despegue	134
Umbral de dolor	140
SPL máximo en el aire	194

Tomado de (Sistemas de radio y televisión, niveles de presión sonora, 2001)

La forma de constatar la presión procedente del aire es mediante el decibel, para ello se debe tener una presión de referencia, que puede ser el valor mínimo que el oído es capaz de captar. Además, el oído del ser humano se encuentra cerca de una función logarítmica más que de una lineal y por lo tanto se utilizan logaritmos para realizar su medida. Solórzano, S. (2017)

Nivel de Referencia para la Presión Sonora (en el aire) = $2E^{-5}$ Pa(ms) **(Ecuación 2)**

Nivel de Referencia para la Intensidad Sonora (en el aire) = $1E^{-12}$ w/m² **(Ecuación 3)**

Nivel de Referencia para la Potencia Sonora (en el aire) = $1E^{-12}$ w **(Ecuación 4)**

$$L_p = 20\log P/P_r$$

L_p = Presión Sonora.

P = La presión Medida.

P_r = Presión de referencia ($2E^{-5}$ Pa).

Configuración de la relación de decibeles y voltaje:

$$dB = 20\log(V_1/V_2)$$

$$dB = 20\log(100/1) \quad \textbf{(Ecuación 5)}$$

$$dB = 40$$

Llegando a la conclusión que 40dB es mayor que un voltio.

- Si el voltaje es idéntico el uno (V_1) con el otro (V_2) la diferencia es 0dB
- Cuando el valor del voltaje (V_1) llega ser el doble que el otro (V_2) la diferencia es de 6dB
- Cuando el valor del voltaje (V_1) es diez veces más que el otro (V_2) la desigualdad es de 20dB

dB: Es un valor sin ninguna adaptación a los sonidos ya sea para valores bajos o altos en relación al oído humano.

dBA: son valores en decibeles adaptados a la capacidad de captar del oído humano. En la figura 3 se muestra el sonido en la estructura del oído.

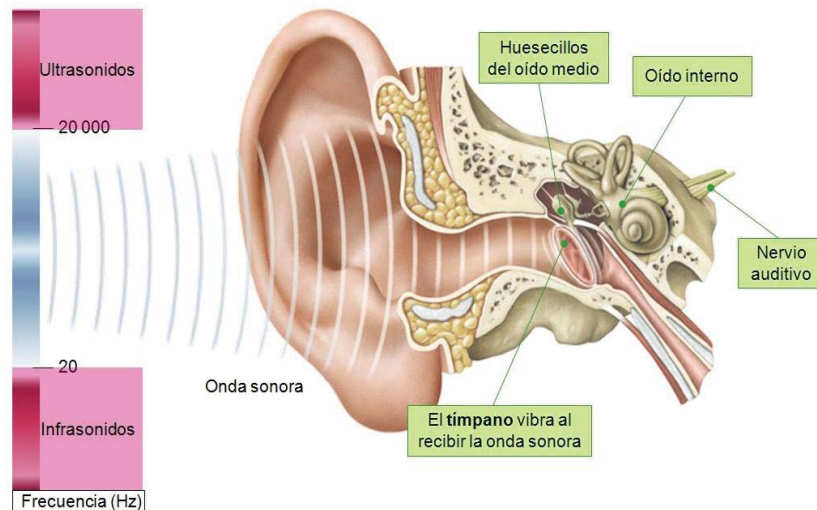


Figura 3. El sonido en la estructura del oído.

Tomado de (Sonido del oído, 2017)

1.1.3 Ruido

Según Henao, (2013, p.5). El ruido desde el punto de vista físico es la unión de ondas sonoras presentes en el campo de frecuencias audibles y de intensidades variables. Se llega a considerar ruido a todo sonido indeseable que provoca molestias o que puede causar problemas a la salud y el bienestar de las personas.

El ruido urbano es diferente en cada país, ciudad y zona, siendo de situación dinámica, ya que cada año varía. Cualquier cambio urbanístico influye sobre los niveles que soportan los ciudadanos. Entre las principales fuentes de ruido urbano tenemos: el tráfico automotriz, la industria, la aviación, equipos de construcción, fábricas, ruidos personales entre otros. En figura 4 se muestra una gráfica de la afectación del ruido.

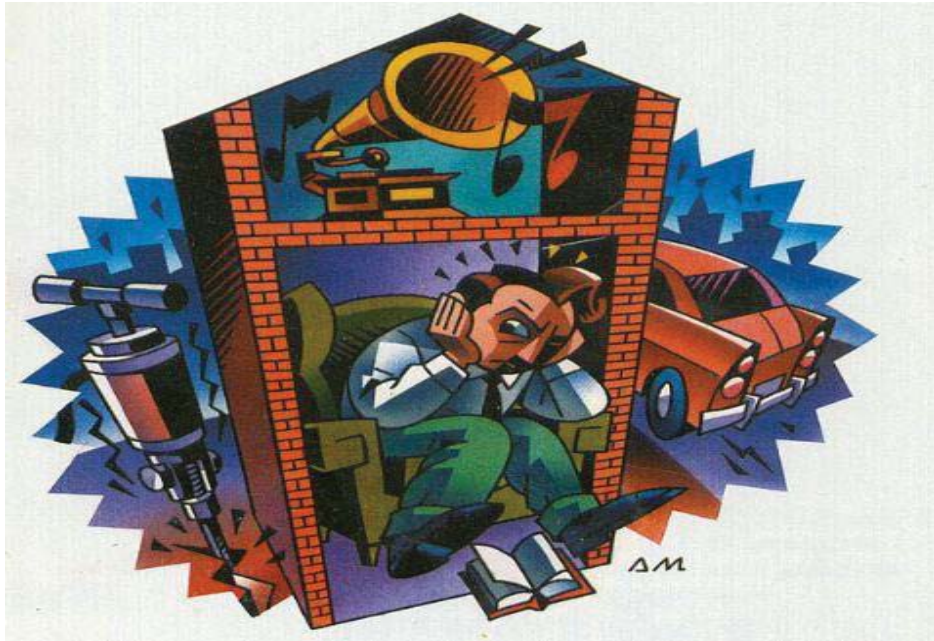


Figura 4. Afectación del Ruido en el ser humano.

Tomado de (Escuela de Organización industrial, 2015)

El ruido llega a ser una variación del sonido, por los cambios y la variación que sufre, se torna irritante y molesto, con el tiempo llega a ser causante de varias enfermedades como: la hipoacusia que es una de las enfermedades actuales, mencionada por el centro American College of Occupational and Environmental Medicine, la misma que ataca de forma silenciosa y lentamente va disminuyendo la audición hasta llegar a la pérdida total. También están las enfermedades que causan malestar, pérdida de atención, deficiencia del sueño, dolores de cabeza, estrés, trastornos psicológicos. Además, existen varios estudios que indican que un ser humano expuesto al ruido por más de 48 horas llega a desarrollar comportamientos agresivos, bajo rendimiento escolar en el caso de los niños y pudiendo llegar hasta provocar infartos. (American College of Occupational and Environmental Medicine, 2003)

Se llega a considerar que es ruido cuando sobrepasa niveles mayores a los 70dBA, dependiendo de las zonas en las cuales se realice la medición, actualmente existe la norma técnica Resolución N°0002- DMA-2008 de la Dirección Metropolitana Ambiental del Distrito Metropolitano de Quito

(actualmente Secretaría de Ambiente), que determina la zonificación y niveles del ruido para cada área.

A continuación, en la figura 5 se muestra un mapa de niveles de ruido realizado por un estudiante en el año 2009, previo a la obtención del título de ingeniera ambiental, de la universidad Internacional SEK.

Nivel Sonoro (dBA)	Nombre del Color	Color	Trama	Grado de Contaminación
>35	Verde Claro		Puntos pequeños, densidad baja	No contaminado
35 – 40	Verde		Puntos medianos, densidad media	
40 – 45	Verde Oscuro		Puntos grandes, densidad alta	
45 – 50	Amarillo		Líneas verticales, densidad baja	
50 – 55	Ocre		Líneas verticales, densidad media	
55 – 60	Naranja		Líneas verticales, densidad alta	
60 – 65	Cinabrio		Entramado de cruces, densidad baja	
65 – 70	Carmín		Entramado de cruces, densidad media	Contaminado
70 – 75	Rojo Lila		Entramado de cruces, densidad alta	Levemente Saturado
75 – 80	Azul		Rayas verticales anchas	Saturado
< 80	Azul Oscuro		Totalmente negro	Muy Saturado

Tabla 9.- Definición de tramas de contaminación acústica establecidas en la Norma ISO 1996-2

Figura 5. Definición de tramas de contaminación acústica establecida.

Tomado de Francisco, J (2009)

En la figura 6, 7, 8 y 9 se muestra un mapa del ruido promedio en el distrito metropolitano de Quito. Las mediciones se pueden ver al final de cada figura, las mismas que se encuentran divididas en colores, tomando como protocolo la normativa de la figura 5.

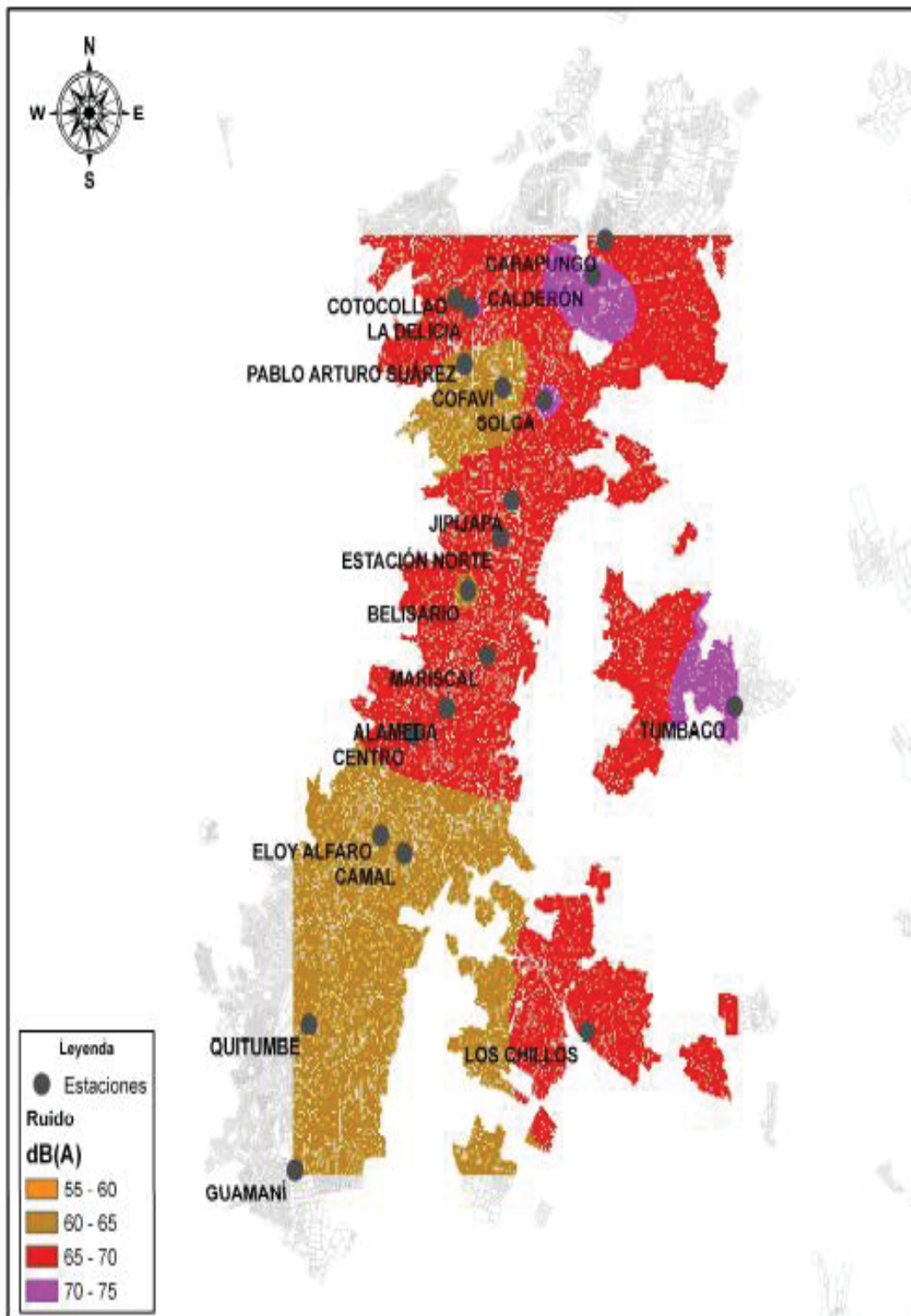


Figura 6. Mapa de ruido, promedios diarios en el DMQ.

Tomado de Francisco, J (2009)

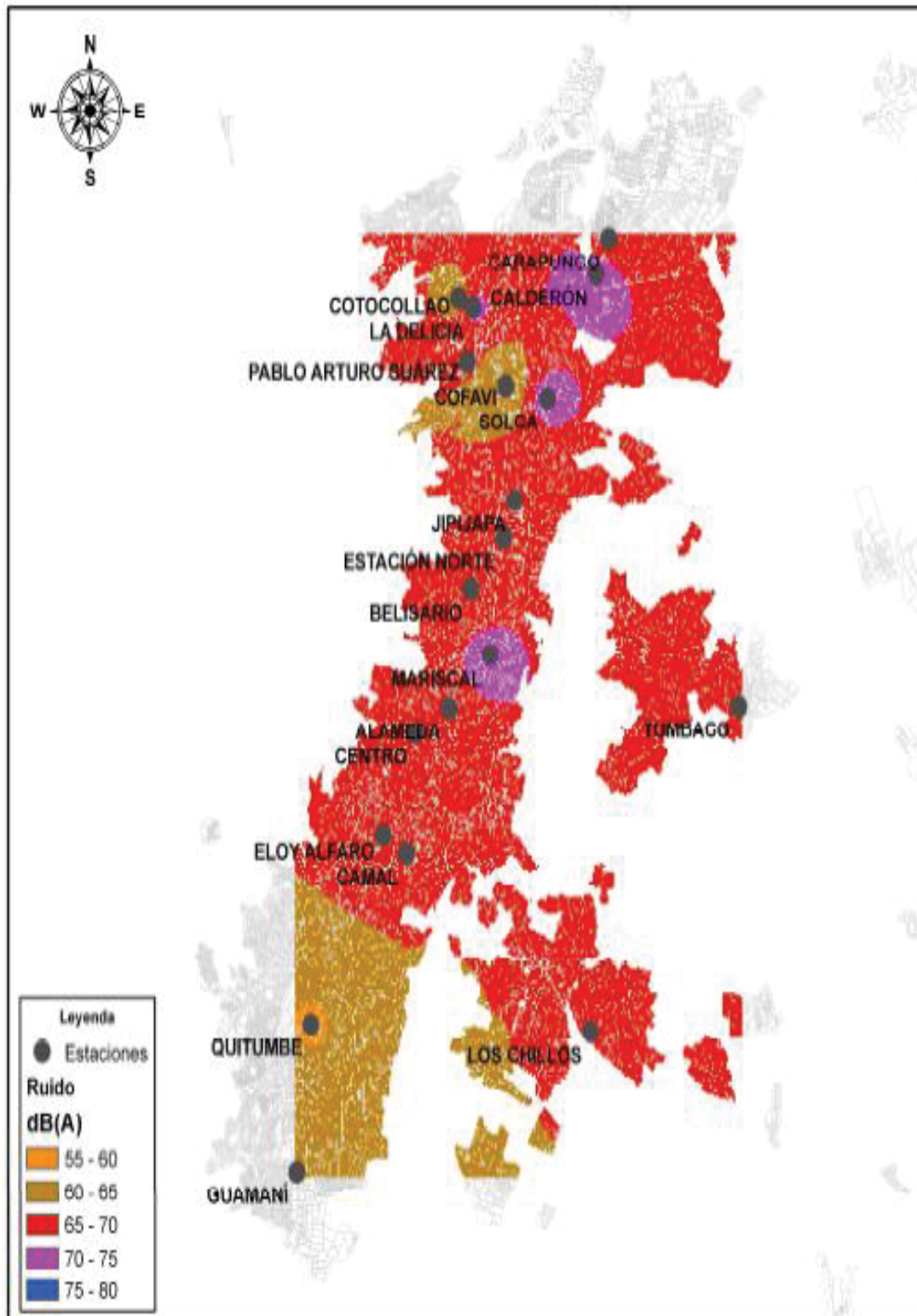


Figura 7. Mapa de ruido, promedio en las mañanas.

Tomado de Francisco, J (2009)

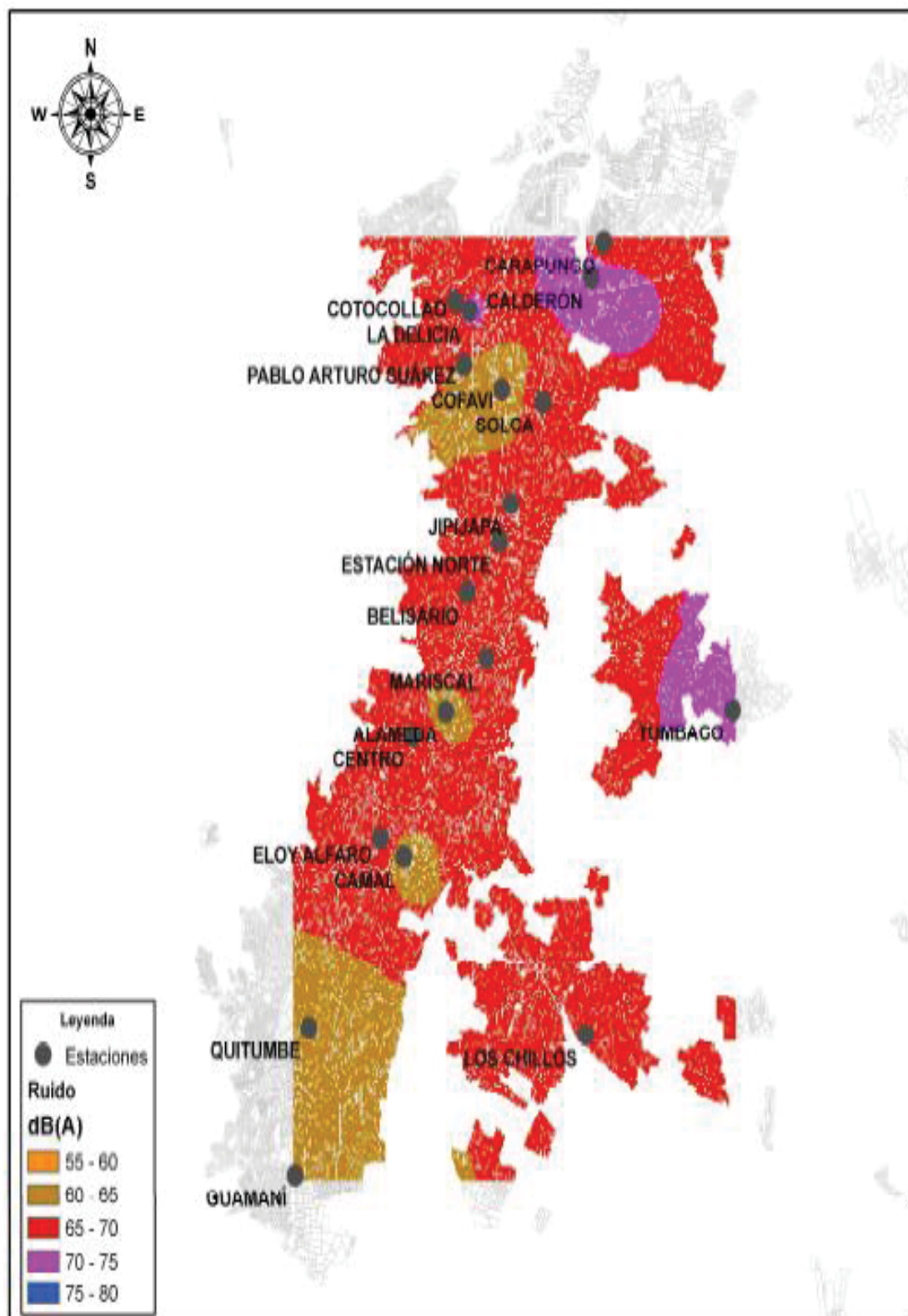


Figura 8. Mapa de ruido, promedio en las tardes.

Tomado de Francisco, J (2009)

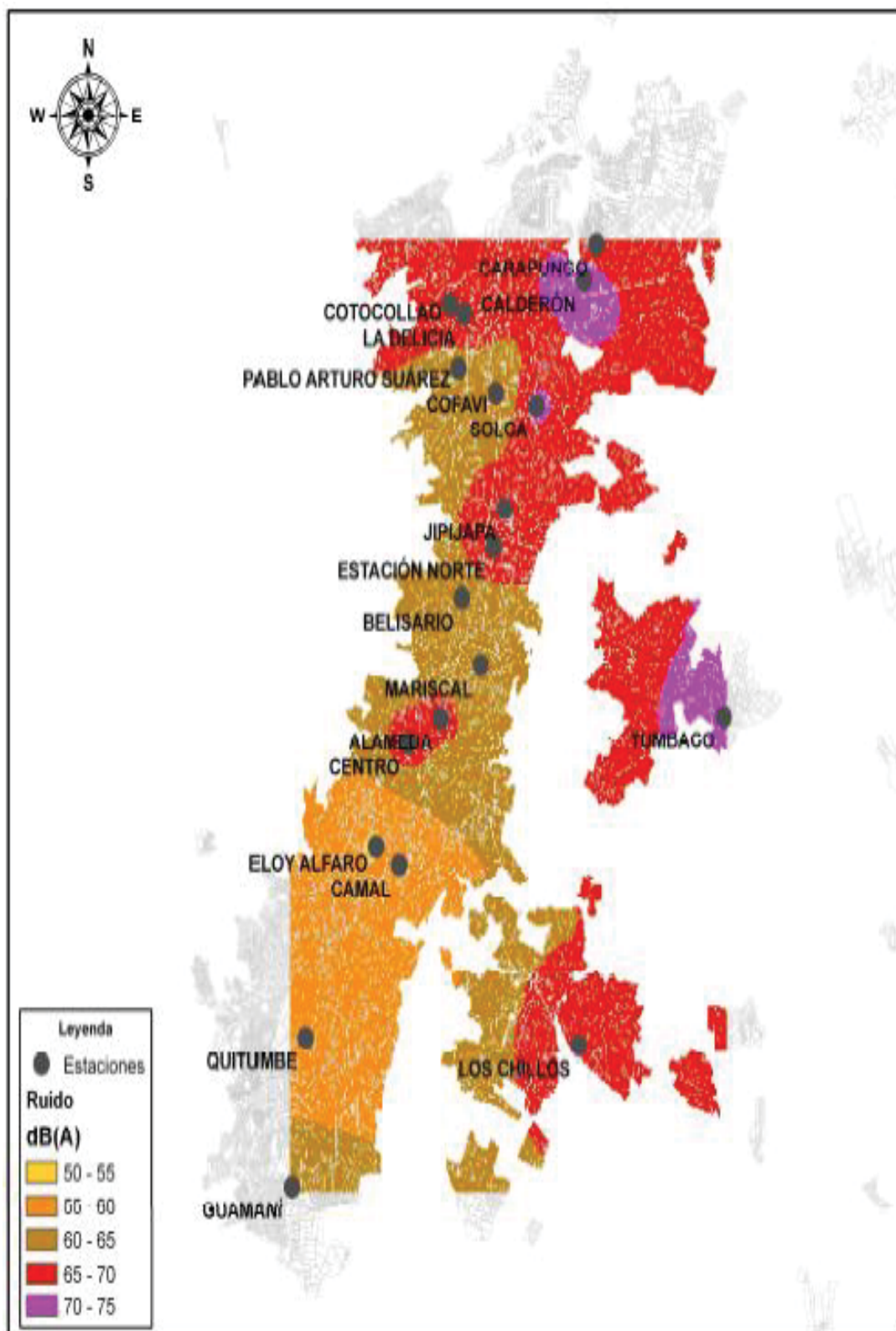


Figura 9. Mapa de ruido, promedio en las noches.
Tomado de Francisco, J (2009)

1.1.4 Afectación de Ruido

El exceso de ruido puede llegar a afectar de forma negativa el sistema central nervioso a través del oído. Sobre todo, al sobrepasar los límites establecidos, pudiendo producir sordera y en varios casos efectos patológicos, los mismo que pueden ser instantáneos o a futuro. Además, a niveles menores llega a provocar problemas en diferentes formas como: problema de concentración, problemas para dormir, problemas para descansar entre otros.

Si la exposición al ruido es de forma reiterada llega a producir problemas crónicos de nerviosismo y estrés, llevando a cambios psicológicos, enfermedades cardiovasculares y alteraciones en el sistema inmunitario. En la figura 10 se indica un mapa de los posibles problemas de salud provocados por el ruido.

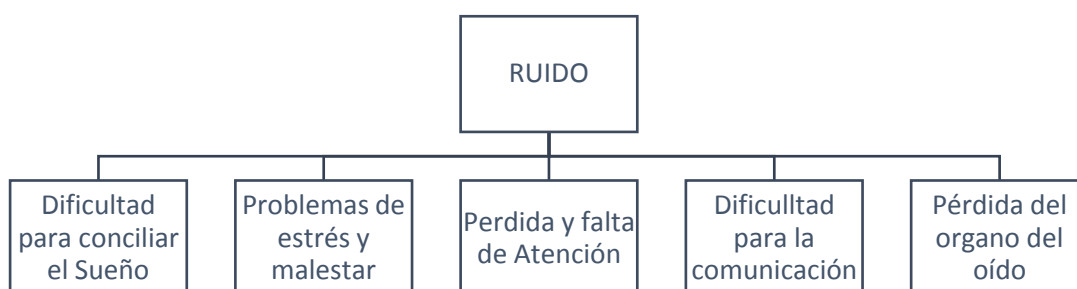


Figura 10. Problemas del Ruido.

Varios problemas como el bajo rendimiento escolar, laboral, accidentes de trabajo como de tránsito, muchas veces son ocasionadas por trastornos en el ser humano producidos por el sonido, debido a que al ser intensos y estar expuestos a estos sonidos graves afectan de manera significativa la salud humana. Santiago, S (2016)

1.2 Redes Inalámbricas

Una LAN inalámbrica está basada en la utilización de la radiofrecuencia para establecer conexiones por varios puertos sin la necesidad de tener cables o estar sujetos al mismo. Utilizan nodos que transmiten la información por medio de ondas electromagnéticas las mismas que pueden ser de forma: infrarrojos, laser, microondas y bluetooth. Una de las principales ventajas de estas redes es la de permitir el acceso a lugares inaccesibles y grandes distancias. Algunas de estas redes son: WMAN, WPAN, WLAN, WBAN y RIS. (Aguirre, 2013)

Estas redes se mantienen en constante evolución, mejorando el área de cobertura como la velocidad de transmisión. En la figura 11, se muestra las principales redes inalámbricas.

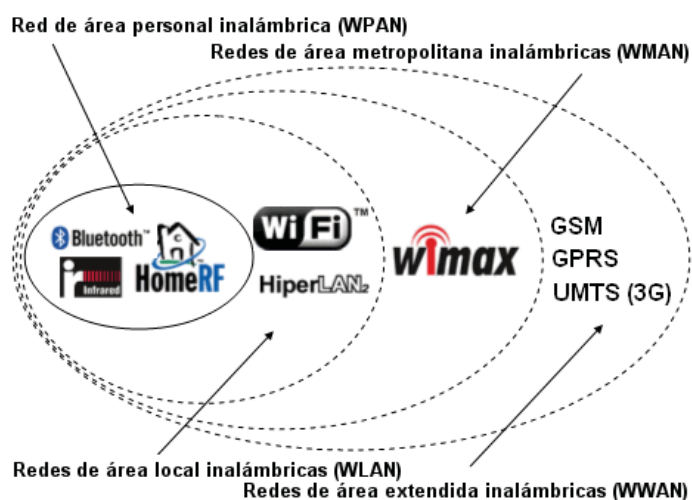


Figura 11. Clasificación de las redes inalámbricas.

Tomado de (CCM, 2017)

1.2.1 Red WSN

Según National Instruments, una red inalámbrica WSN consiste en dispositivos distribuidos, ubicados en forma espaciada y autónoma que utilizan sensores para monitorear condiciones físicas o ambientales. Esta tecnología trabaja en

la banda libre ISM de 2.4 GHz, con una modulación en espectro ensanchado por secuencia directa (DSSS). Además, tiene una velocidad de transmisión de hasta 250kb/s y soporta hasta 255 nodos por red.

Está compuesto de tres componentes principales: nodos, gateways y software. Los nodos de medida distribuidos de forma espacial establecen comunicación con sensores para monitorear el equipo. Los datos adquiridos son transmitidos de manera inalámbrica al Gateway. Los ruteadores son un tipo especial de nodo de medida que se usa para ampliar la distancia y la fiabilidad de una red WSN.

La mayor parte de los sensores inalámbricos utilizan estándares abiertos, permitiendo que se realice modificaciones en su estructura y lo más importante no se necesita permisos, están enfocados para diversas tareas entre las cuales se encuentran: la agricultura, seguridad, monitoreo, control ambiental, automatización, domótica entre otros. En la figura 12 se muestra la arquitectura de una red WSN.

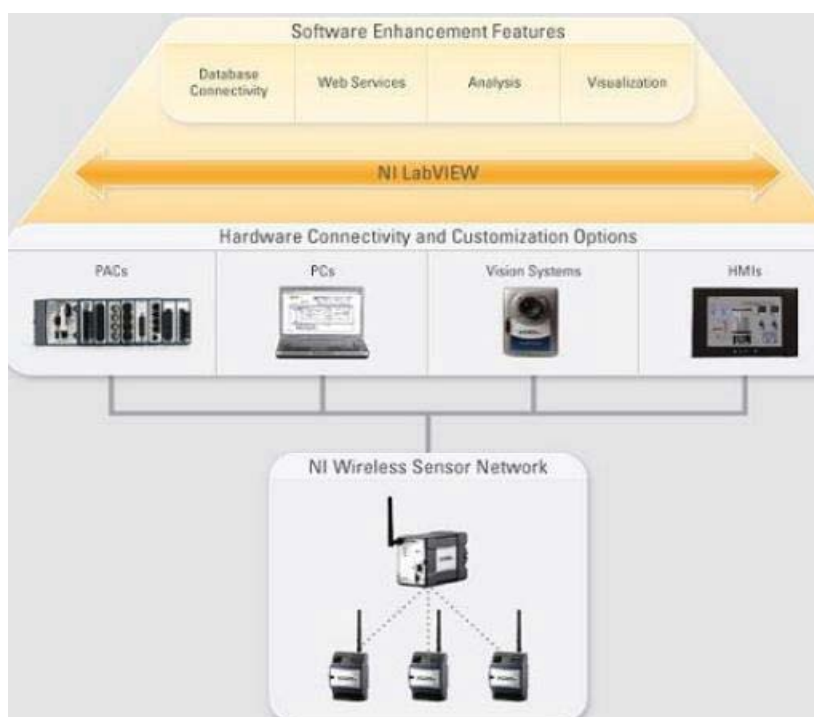


Figura 12. Arquitectura de las redes WSN.

Tomado de (National Instruments, 2009)

1.2.2 Componentes de un nodo WSN

Un nodo WSN está compuesto por varios componentes técnicos como se muestra en la figura 13. Entre las principales se encuentran: microcontrolador, el radio, circuito analógico, una interfaz al sensor y una batería. Al momento de hacer uso de la tecnología de radio WSN, se deberá realizar compensaciones importantes, en los sistemas alimentados con batería, ya que al utilizar altas tasas de datos y usar frecuencia la radio para la comunicación, existe un mayor consumo de energía. Se predice un aproximado de tres años de vida para la batería, por este motivo en la actualidad varios de los sistemas WSN están basados en ZigBee debido a su bajo consumo de potencia. (National Instruments, 2009)

Una de las formas para extender la vida de las baterías, consiste en configurar que periódicamente un nodo WSN se enciende y transmite datos alimentándose del radio y posteriormente apagándose para conservar energía. (National Instruments, 2009)

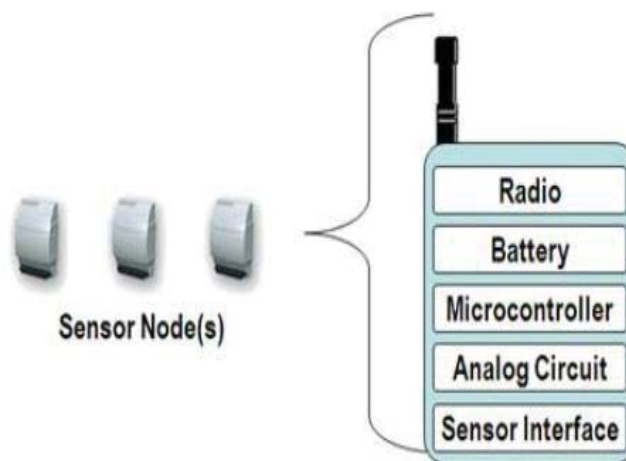


Figura 13. Componentes de un sensor de nodo WSN.

Tomado de (National Instruments, 2009)

1.2.3 Protocolos de Enrutamiento

Los protocolos de enrutamiento se encargan de mantener la comunicación de una manera óptima, tratan de garantizar la seguridad al usuario con el objetivo de obtener un menor riesgo de intrusos, manejar una transmisión óptima de los paquetes y resolver problemas en la transmisión de datos.

Los sensores inalámbricos posibilitan sostener una comunicación permanente, debido a que se los puede configurar a través de estos protocolos. En el caso de sufrir alguna desconexión o si llegara a detectar alguna falla en medio de la red, podrían buscar otra ruta hacia su receptor y de esta forma no se pierda la conexión y por ende los datos. En la figura 14 se indica la comparación entre el modelo WSN y el modelo OSI.



Figura 14. Comparación entre el modelo OSI y WSN.

Tomado de (Guaña y Escuntar, 2016)

Al inicio los nodos desconocen la topología que está utilizando la red, y son los mismos dispositivos quienes deben descubrirla. Una vez descubierto la topología y cuando un nuevo nodo trata de ingresar a la red, debe anunciar su presencia a los vecinos y escuchar los broadcast y anuncios de los otros equipos. El nuevo nodo después de aprender la manera de enrutarse por medio de los nodos antiguos debe informar que cuentan con su presencia y

que podrán acceder a él. Con el transcurso del tiempo estos dispositivos llegan a ser capaces de conocerse entre ellos y acceder de diferentes formas a todos los nodos.

Uno de los criterios para determinar el diseño de los protocolos de enrutamiento depende de la base de la red y el encaminamiento que se pretende realizar. Los protocolos según su estructura de red se dividen en los siguientes: (Guaña y Escuntar, 2016)

- Enrutamiento Plano: denominados también Flat- Based, es donde los dispositivos piensan que todos son iguales y realizan las mismas funciones dentro de la red.
- Enrutamiento Jerárquico: conocido también como cluster-based, en este protocolo todos los dispositivos tienen diferentes papeles dentro de la red. Una de las características es la de crear grupos de nodos que puede ser coordinadores, otros nodos pueden ser recolectores de la información y otro grupo de transmitir y procesar los datos, facilitando la captación y la transmisión de información.
- Enrutamiento basado en la localización (Location-based): como su nombre indica utilizan la localización por medio de la posición de los nodos que están formando la red.

1.2.4 Tipos de Topología

La topología es la forma física o lógica de la red para intercambiar datos. Los dispositivos al iniciar se hallan conectados de forma lógica y física, teniendo un coordinador que administra los demás nodos, su función principal es la de buscar la mejor ruta de comunicación. Dependiendo de la forma jerárquica, se encuentran las siguientes topologías: (National Instruments, 2009)

a) Topología tipo Estrella

En este modo de conexión todos los nodos se encuentran conectados directamente al Gateway o coordinador como muestra la figura 15. Se caracteriza porque su topología es muy resistente, tiene poca latencia, es fácil de elaborar, su costo es bajo y mantiene un gasto de energía igual. Una de las principales desventajas es la que pueden sufrir colisiones y en caso de dañarse el nodo central se pierde toda la conexión. (Digi, s.f.)

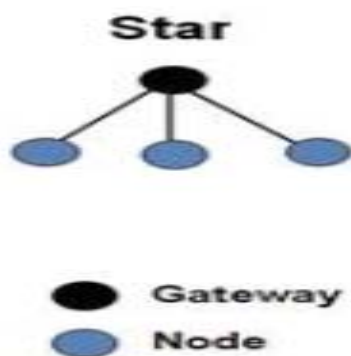


Figura 15. Topología Estrella.

Tomado de (National Instruments, 2009)

b) Topología tipo Malla o Mesh

Este tipo de red ofrece una mayor confiabilidad, su funcionamiento se basa en que sus nodos se pueden conectar a múltiples nodos remotos y transmitir la información por la ruta que presente disponibilidad y una mayor confiabilidad. Una de las mayores ventajas es cuando existe la caída de un nodo ya sea principal o secundario, la misma red se encarga de buscar otra ruta de comunicación y por lo tanto no existirá una caída total de la red como se muestra en la figura 16. Llegando a ser un sistema más avanzado, muy confiable y permite la reducción de mantenimiento de nodos. (National Instruments, 2009)

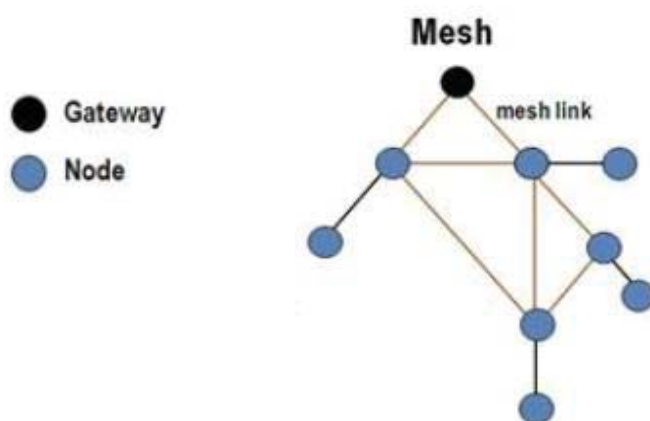


Figura 16. Topología de Malla.

Tomado de (National Instruments, 2009)

c) Topología árbol

La red tipo árbol se caracteriza por la conexión de cada uno de sus nodos a un nodo de mayor jerarquía en el sistema árbol y después se conecta al Gateway principal, como se muestra en la figura 17. Generalmente la información es encaminada desde el nodo de menor jerarquía de la red hacia el Gateway. Esta configuración posee una semejanza a la red tipo estrella: ya que permite la reducción de colisiones en la transferencia de datos. La configuración del sistema es crítica ya que, si no se lo realiza correctamente, los algoritmos de enrutamiento pueden llegar a ser inseguros. (National Instruments, 2009)

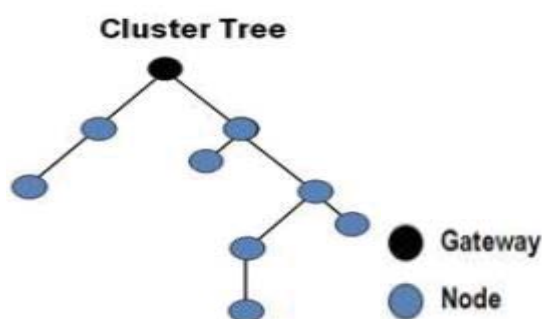


Figura 17. Topología de Árbol.

Adaptado de (National Instruments, 2009)

e) Modelo Multi-hop

En este modelo los datos son enviados a través de un nodo vecino hasta la estación base, buscando así el nodo más cercano para saltar al próximo nodo, hasta llegar a encontrar al coordinador. (Digi, s.f.)

Es decir, la información es enviando de salto en salto hasta llegar a su destino. Algunos protocolos utilizan este modelo, entre las principales se encuentran las siguientes:






- MultiHop de Tmote Sky.
- Telos, MultiHop LQI.
- MintRoute, Router.

1.3 Dispositivo XBee

Es un dispositivo electrónico (Módulo de radio frecuencia RF) que se utiliza para enviar y recibir señales de radio en diferentes frecuencias. La empresa Digi es el mayor fabricante de estos dispositivos. Existe una amplia gama de módulos de RF para ser usados en soluciones inalámbricas como: bajo costo, largo alcance y bajo consumo de energía entre los más comunes están los dispositivos xBee. El protocolo de red a utilizar es el IEEE802.15.4 que es utilizado para crear redes con varias topologías, los mismos que se utilizan ampliamente en redes que requieren: un bajo consumo de potencia, poca latencia y un alto tráfico de datos. En la tabla 5 se muestra la comparación de los dispositivos XBee. (Digi, s.f.)

Tabla 5.

Comparativa de dispositivos para redes WSN.

Características	Lora	Tymote Telos B	XbeeS1	XbeeS2	XBee Pro S2B
					
Distribuido	Libelium	Tymote	Digi International	Digi International	Digi International
Compatibilidad	-	-	Serie1	Serie2	Serie2
Antena	Wire	Wire	Wire	Wire	Wire on PCB, RPSMA or Conector U.FL.
Rango de Datos	-	-	250 kbps/up to 115 kbps (1)	250 kbps/up to 1 Mbps (1)	250 kbps/up to 1 Mbps (1)
Rango interior	100 m	40m	100 ft (30 m)	133 ft (40m)	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant
Línea de Visión	300 ft (100 m)=	-	300 ft (100 m)=	400 ft (120 m)=	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant
Configuración	API Y AT	-	API Y AT	API Y AT	API, AT Y MESH
Frecuencia de	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz

Banda					
Batería	Solar/ Pila	Pila	Pila	Pila	Pila

Adaptado de (Escuntar, 2016) Y (Digi, s.f.)

En el presente proyecto se utilizó el módulo XBee S2B el mismo que tiene algunas mejoras en el hardware y en el manejo de la potencia. En la figura 18 se muestra el módulo XBee PRO S2B. Ver hoja de datos, en Anexo A.

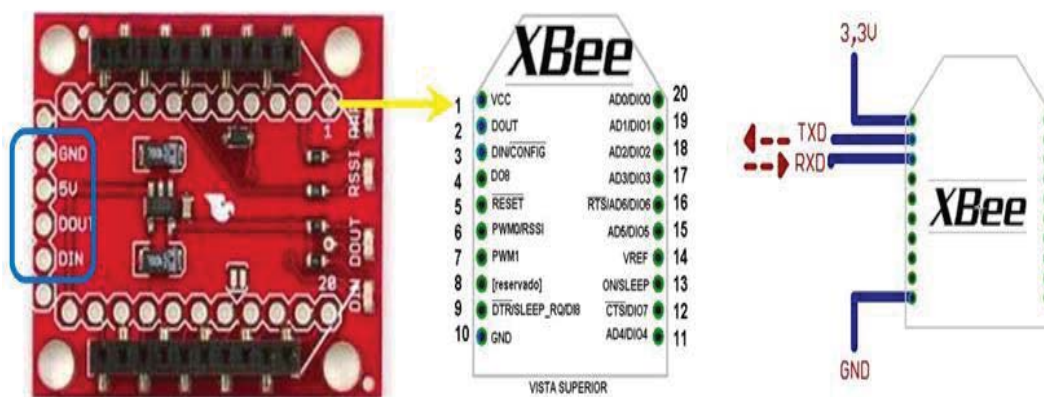


Figura 18. Dispositivo XBee más tarjeta Explorer 5V.

Tomado de (Digi, s.f.)

1.3.1 Modos de funcionamiento del XBee

Los dispositivos XBee se comunican de forma inalámbrica entre ellos, enviando y recibiendo mensajes como se puede ver en la figura 19. Estos dispositivos no pueden gestionar los datos enviados o recibidos, sin embargo, pueden comunicarse con otros dispositivos a través del interfaz serie. También transmiten al aire los datos que llegan del puerto y transmiten al puerto serie cualquier dato que llega por el aire. Los microcontroladores o los PCs pueden controlar que envíen los XBee y gestionan los mensajes inalámbricos entrantes. (Aprendiendo Arduino, s.f)

Por lo tanto, tenemos dos tipos de comunicación en los dispositivos XBee:

- Comunicación inalámbrica: es la comunicación entre los módulos XBee que no utilizan un medio alámbrico, estos módulos deben ser parte de la misma red y usar la misma frecuencia de radio.
- Comunicación serie: es la comunicación entre el módulo XBee y el microcontrolador o el PC a través de un puerto serie.

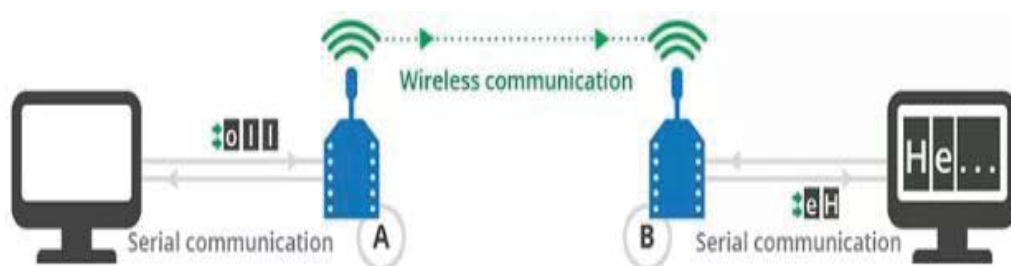


Figura 19. Modos de comunicación del XBee.

Adaptado de (Digi, 2016)

En la comunicación inalámbrica los módulos transmiten y reciben información a través de la modulación de las ondas electromagnéticas. Para realizar la transmisión ambos módulos deben estar en la misma frecuencia y en la misma red. Los parámetros que deben ser iguales son:

- Channel (CH).- es la frecuencia usada para comunicar, es decir, el canal dentro de la red.
- Personal Area Network Identifier (ID).- es un identificador único que establece que los módulos están en la misma red.

Un módulo XBee solo recibirá y transmitirá datos a otros XBee dentro de las mismas redes (mismo ID) y usando el mismo canal (mismo CH), como se muestra en la figura 20.

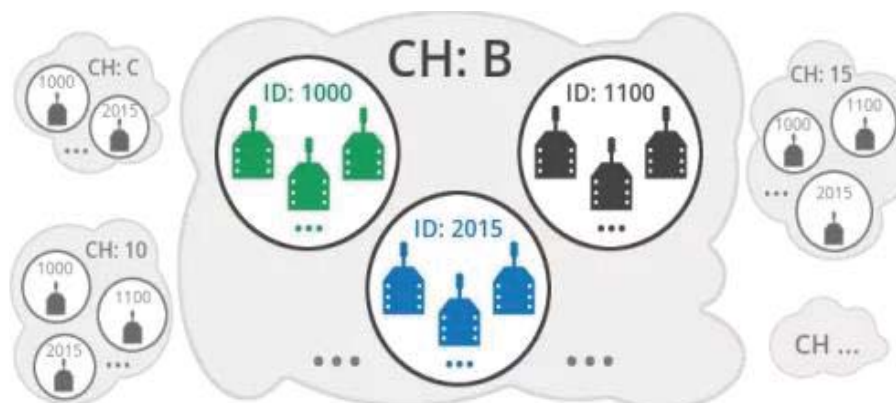


Figura 20. Canales e IDs para la comunicación.

Adaptado de (Digi, 2016)

1.3.2 Modo Serie

Un dispositivo XBee puede funcionar como un módulo independiente o conectado a un microcontrolador o PC. Cuando opera como módulo independiente simplemente manda datos al nodo central de los sensores o dispositivos conectados a los puertos del módulo. Y cuando está conectado a un microcontrolador o un PC, el módulo XBee usa la comunicación serie. En el caso de Arduino, se comunican a través del puerto serie y envía la información, para que sea transmitida por el aire. El módulo XBee hace de interface con el microcontrolador a través de la UART (puerto serie asíncrono).

Un arduino o el dispositivo externo conectado a un módulo XBee mediante puerto serie puede tener varios modos de operación en función de cómo se comunican por el puerto serie. Los módulos XBee soportan dos modos de operación: Modo API y modo transparente.

1.3.3 Modo Transparente (AT)

En modo transparente el módulo XBee funciona de forma que todo lo recibido por el puerto serie es enviado inmediatamente al XBee receptor, y todo lo

recibido del receptor se manda tal cual al puerto serie, como se muestra en la figura 21. Para comunicar dos módulos, es necesario configurar la dirección del destino en el módulo que envía datos. Esta dirección se programa en los parámetros: Destination Address High (DH) y Destination Address Low (DL).

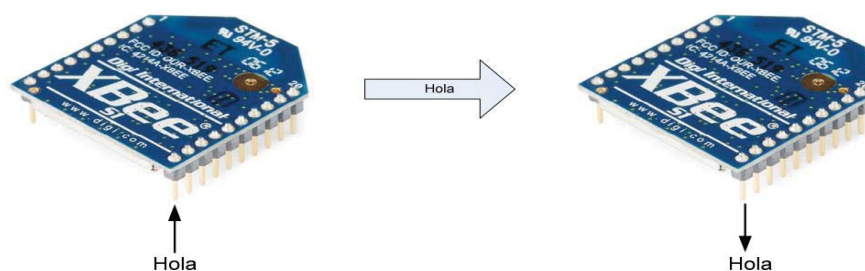


Figura 21. Comunicación AT.

Tomado de (Xbee, 2013)

Este modo tiene algunas limitaciones, por ejemplo, al trabajar con varios módulos es necesario configurar la dirección de destino antes de mandar un mensaje. Sin embargo, el modo transparente es perfecto cuando hay comunicación punto a punto entre dos elementos. En la tabla 6 se muestra los principales comandos del modo AT.

Tabla 6.

Principales comandos del modo AT.

PRINCIPALES COMANDOS DEL MODO AT	
ID	Fija identificación de la Red
CH	Muestra el canal usado
DH/DL	Fija dirección del remoto a transmitir
SH/SL	Muestra la dirección del módulo
JV	En el encendido busca al coordinador
ND	Busca los módulos conectados
CI	12, modo loopback
PL	Nivel de potencia
PM	Opción de modo de potencia

EE	Habilita la encriptación de datos
EO	Opción de encriptación
NB	Paridad
BD	Baudrate
SM	Modo de bajo consumo
SN	Extiende el tiempo en modo SLEEP
SO	Opciones de modo SLEEP
SP	Tiempo en modo SLEEP
ST	Tiempo de modo activo luego de modo SLEEP
VR	Versión de firmware
HV	Versión de hardware
DB	Nivel de señal de la última recepción
%V	Valor de la tensión de alimentación
IR	Tiempo de muestreo

Adaptado de (Soluciones Inalámbricas, s.f)

1.3.4 Modo API (Application Programming Interface)

El modo API ofrece un interfaz estructurado, donde los datos son comunicados a través del interfaz serie en paquetes organizados y en un determinado orden. Esto permite establecer una comunicación compleja entre módulos sin tener que definir un protocolo propio.

Por defectos los módulos XBee trabajan en modo transparente, pero esto tiene limitaciones:

- Si un módulo necesita transmitir mensajes a módulos diferentes debe cambiar la configuración para establecer el nuevo destino.
- Para leer y escribir la configuración del módulo hay que entrar primero en modo comando.
- En modo transparente un módulo XBee no puede distinguir el origen de la comunicación que recibe.

Para resolver estas limitaciones XBee da la alternativa del modo API (Application Programming Interface). En la figura 22 se muestra la comunicación en modo API.

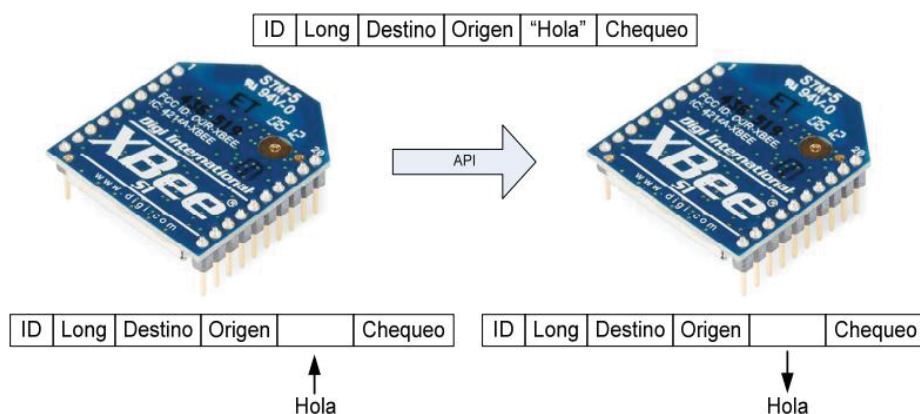


Figura 22. Transmisión en Modo API.

Adaptado de (Soluciones inalámbricas, 2016)

Ventajas del modo API:

- Configurar módulos locales y remotos en la red y sin necesidad de entrar en modo comando.
- Comunicar con uno o varios destinatarios.
- Identificar el módulo que ha mandado la comunicación.
- Recibir el estado de la transmisión de los paquetes.
- Obtener la fuerza de la señal de los paquetes recibidos.
- Hacer gestión y diagnóstico de la red
- Los paquetes pueden contener datos o comandos AT.
- Los datos siempre son retransmitidos a los módulos remotos.
- Los paquetes que contienen comandos AT pueden ser enviados al módulo local o pueden estar dirigidos a los módulos remotos para la reprogramación.
- Hacer funciones avanzadas como actualización de firmware remota.

1.3.5 Estructura de la Trama API

En modo API los datos enviados están estructurados en una trama. Se envían a través de interfaz serie de XBee y contiene el mensaje inalámbrico e información adicional como la calidad de la señal. En la figura 23 se muestra la estructura de la trama API.

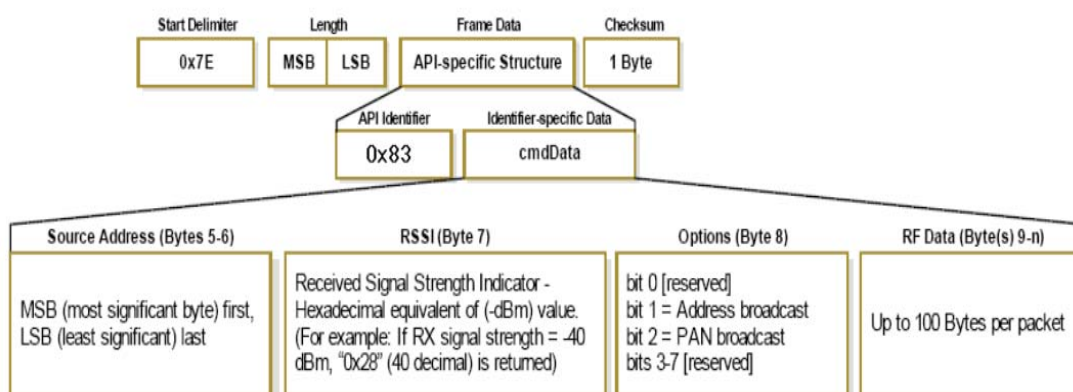


Figura 23. Trama API.

Adaptado de (Digi, 2016)

Descripción del formato de la trama API, de acuerdo con la figura 23:

- El dato 0x7E es el primer byte, que indica el inicio del frame, en un valor hexadecimal.
- Luego van dos bytes indican la longitud de la trama. MSB byte más significativo y LSB byte menos significativo.
- Frame Data.- Es la estructura específica API, en esta parte se envía todo el mensaje, direccionamiento o configuración del módulo.
- Checksum.- ayuda a comprobar la integridad de los datos mandados. Las tramas enviadas por puerto serie con checksum incorrecto nunca serán procesadas por el módulo y serán ignoradas.

Descripción de los datos dentro del FRAME DATA, como se muestra en la figura 23:

- El identificador API 0x83 (cuarto byte del frame) muestra que los datos se están recibiendo en un direccionamiento de 16 bits.
- Los bytes 5 y 6, indican el bit más significativo y el menos significado. Además indican la dirección de origen.
- El byte 7 muestra el valor del RSSI (Received Signal Strength Indicator) que permite determinar la potencia de la señal.
- El byte 8, se divide en más bits, de los cuales el bit 1 muestra si es un broadcast de direccionamiento (broadcast con DL=0xFFFF).
- El bit 2 muestra un broadcast de todas las redes PAN (broadcast con ID=0xFFFF).
- Después del byte 9, hasta el número de bytes dado por el **byte de Length**, corresponden a los datos obtenidos del modo Cable Virtual provenientes de otro módulo.

Para configurar el modo API, se modifica el parámetro del mismo, que permite seleccionar entre los dos modos soportados y el modo por defecto que es el transparente. En la tabla 7 se muestra los parámetros para la configuración en modo AT y API.

Tabla 7.

Parámetros para configurar API o AT.

Mode	AP Value	Description
Transparent	0	API modes are disabled and the module operates in transparent mode
API 1	1	API mode without escaped characters
API 2	2	API mode with escaped characters

Adaptado de (Digi, 2016)

1.3.6 Modo API 1 y API 2

La diferencia entre API 1 y API 2 es que la trama de API 2 sirve para utilizar caracteres especiales. La estructura de la trama es exactamente la misma, solamente va a cambiar en que dentro del mensaje va a haber unos bytes que pueden generar inconvenientes dentro de la trama. Ejemplo 0x7E es un byte delimitador de inicio y el mismo no puede aparecer en ningún otro campo ya que se confundiría y pensaría que es una nueva trama que está llegando. En API 2 esto se corrige.

El modo API non-escaped (API 1) se basa únicamente en el delimitador de inicio y la longitud de los bytes para diferenciar las tramas, si los bytes en un paquete se pierden, la cuenta de bytes será incorrecta y el siguiente paquete (trama) también se perderá. API escaped (API 2) implica secuencias de caracteres de escape en una trama para mejorar la fiabilidad especialmente en entornos ruidosos.

1.3.7 Estructura de la trama API 1

Se desea enviar el mensaje: 0x23 0x11

Se crea la trama:

- 0x7E: Delimitador.
- 0x00 0x02: Longitud del mensaje a enviar, ya que se va a enviar dos bytes.
- 0x23 0x11: Datos a enviar.
- Checksum.

Cálculo del Checksum

El checksum es utilizado para verificar la integridad de los datos. Tomado de Modo API 1.

1. Se empieza sumando todos los bytes excepto los delimitadores y la longitud. ($0x23 + 0x11 = 0x34$). Si la suma es mayor a FF

(Transformando a decimal 255) únicamente se toma los 8 bits menos significativos. Ejemplo si la suma sale 01 7E, únicamente se cogerá 7E. En este caso la suma salió 34, por lo tanto, se procederá a tomar ese valor ya que esta menos que FF.

2. Se toma los 8 bits menos significativos del resultado (0x34).
3. Al valor 0xFF se le resta el valor de estos 8 bits menos significativos. (0xFF – 0x34 = 0xCB).
4. Si el cálculo esta realizado correctamente la trama se enviara correctamente, caso contrario no se envía la trama o es descartada.

Trama Final: 0x7E 0x00 0x02 0x23 0x11 0xCB

En algunos casos se pueden presentar algunos inconvenientes en el envío de la trama API 1, los valores que se muestran en la tabla 8, pueden afectar la estructura de la trama. Para corregir estos problemas se utiliza la trama API 2.

Tabla 8.

Valores que afectan a la trama API 1, en la información.

Bits	Indicativo de los bits
0x7E	Byte de inicio de trama
0x7 D	Byte de escape
0x11	XON Encendido
0x13	XOFF Apagado

Adaptado de (Modo API 1, 2017)

El valor 0x7E es un byte de inicio y no puede estar dentro del mensaje, ya que el módulo receptor puede llegar a considerar que es el inicio de una nueva trama. Por lo tanto, se perdería los mensajes, para corregir este error se procede a configurar en modo API 2.

1. En la posición donde va el byte a escapar se inserta el valor en hexadecimal 0x7D.

- Después realizar una operación XOR, entre el byte a ser escapado con 0x20. La misma que se indica en la tabla de verdad número 9.

Tabla 9.

Operación A **XOR** B para la trama API 2.

A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	0

La tabla indica que cuando los bytes son iguales es 0 y cuando los bytes son diferentes es 1.

- Después de realizar la operación el valor obtenido, se debe agregarlo después del byte 0x7D.
- La longitud de la trama y el checksum siguen siendo el mismo.

1.3.8 Estructura de la trama API 2

Trama a enviar: **0x23 11**

- Lo primero que se debe ver es el byte que puede interferir en la trama, en este caso el byte 0x11, puede causar problemas y se tiene que realizar el escape.
- Agregar el byte 0x7D (Byte de escape) en la posición donde está el 0x11.
- Luego realizar la operación XOR (0x11 XOR 0x20 = 0x31).
- A continuación, poner el valor obtenido después del byte 0x7D.

5. La longitud de la trama sigue siendo la misma.

0x00 0x02

6. El checksum sigue siendo el mismo.

0xCB

7. Trama final.

0x7E 0x00 0x02 0x23 0x7D 0x31 0xCB

- 0x7E Byte de inicio.
- 0x00 0x02 Numero de bytes a enviar.
- 0x23 Información
- 0x7D 0x31 byte identificador no es información, aunque este en el campo de la información.
- 0xCB Checksum.

1.3.9 Comandos AT en la trama API

Dentro de la trama API se puede enviar comandos AT con el objetivo de modificar remotamente a los módulos arduino. En la figura 24 y 25 se muestra los comandos AT más importantes.

	Byte	Ejemplo	Descripción
Trama API para envío de comandos AT remotos	0	0x7E	Byte de inicio – indica el comienzo del paquete de datos (frame)
	1	0x00	Largo – Número de bytes (ChecksumByte# – 1 – 2)
	2	0x10	
	3	0x17	Tipo de mensaje - 0x17 significa que es solicitud de comando AT
	4	0x01	Frame ID – secuencia del paquete
	5	0x00	Dirección de destino de 64-bit (número de serie) MSB es el byte 5, LSB es el byte 12
	6	0x13	
	7	0xA2	0x0000000000000000 = Coordinador 0x000000000000FFFF = Broadcast
	8	0x00	
	9	0x40	
	10	0x8B	
	11	0x78	Dirección de la red de destino (configúralo como 0xFFFE para enviar un bodcast)
	12	0x4E	
	13	0xFF	
	14	0xFE	Opción del comando remoto (configuralo como 0x02 para aplicar los cambios)
	15	0x02	
	16	0x44 (D)	Nombre del comando AT (Dos caracteres ASCII)
	17	0x34 (2)	
	18	0x05	Parámetro del comando
19	0x25	Checksum	

Figura 24. Comandos AT para transmisión en la trama API.

Tomado de (XBee & XBee – PRO, 2017)

Byte	Ejemplo	Descripción
0	0x7E	Byte de inicio - indica el comienzo del paquete de datos (frame)
1	0x00	
2	0x14	
3	0x92	Tipo de frame 0x92 indica que es un muestreo de las entradas del XBee
4	0x00	Dirección de origen de 64-bit (número de serie)
5	0x13	MSB es el byte 4, LSB es el byte 11
6	0xA2	
7	0x00	
8	0x40	
9	0x8B	
10	0x78	
11	0x4E	
12	0xA4	Dirección de 16-bit de la red de origen
13	0x02	
14	0x01	Opciones de recepción: 01 = Packet acknowledged 02 = Broadcast packet
15	0x01	Número de muestras. Siempre debe ser 1 dadas las limitaciones de XBee
16	0x00	Máscara para el canal digital, indica que pines están configurados como DIO
17	0x30	
18	0x01	Máscara para el canal analógico, indica cuales pines están configurados como ADC
19	0x00	Lectura de los canales digitales. Estos dos bytes contienen los estados de los pines configurados como DIO
20	0x20	
21	0x02	Lectura del canal analógico.
22	0x0C	Cada canal entrega 2 bytes con el resultado de la lectura del ADC
23	0x20	Checksum (0xFF - la suma de todos los bytes desde el byte 3 a hasta el último)

Modo Sleep

End Device puede dormir para ahorrar energía. Un End Device que solo despierta cada 5 minutos para enviar datos puede solo estar despierto por 6 segundos en un día.

SM - 4 = Cyclic Sleep
 SP - Sleep time (hasta 28 segundos)
 SN - Número de ciclos sleep
 ST - Tiempo que permanecerá despierto

Pin I/O Opciones

0 -Disabled
 1 -N/A
 2 -ADC
 3 -Digital IN
 4 -Digital OUT, LOW
 5 -Digital OUT, HIGH

Máscara para canal digital

Primer Byte
 n/a n/a n/a D12 D11 D10 n/a n/a
 Segundo Byte
 D7 D6 D5 D4 D3 D2 D1 D0
 Ejemplo:
 0x00 0x13 = 0000 0000 0000 1101
 Pins D3, D2 y D0

Máscara para canal analógico

(volt) n/a n/a n/a A3 A2 A1 A0
 Ejemplo:
 0x05 = 0000 0101 = Pin A2 and A0

Figura 25. Comandos AT para recepción en la trama API.

Tomado de (XBee & XBee – PRO, 2017)

Estos comandos son utilizados dentro de la trama para realizar cambios en los nodos remotos, existen dos comandos que permiten realizar esta función:

Comandos AT para consultar o activar los parámetros en el nodo.

a) Comando AT (0x008)

Permite leer o activar los parámetros del módulo, en línea, es decir una vez que el nodo recibe el mensaje, ejecuta inmediatamente la acción.

b) AT (0x09)

Permite leer o activar los parámetros del módulo, en contraste al comando anterior, los nuevos valores quedan en cola y no son aplicados hasta que el tipo de comando AT (0x08) ha sido modificado o hasta que el comando AC no haya sido aplicado.

Comando AT de respuesta del nodo remoto.

a) AT (0x88)

En respuesta a un comando AT, el módulo enviará un comando de confirmación, donde informara si el parámetro ha cambiado satisfactoriamente.

Comando AT para requerimientos remotos

a) AT (0x17)

Permite modificar parámetros de un módulo remoto. En la figura 26 se indica la estructura de la trama.

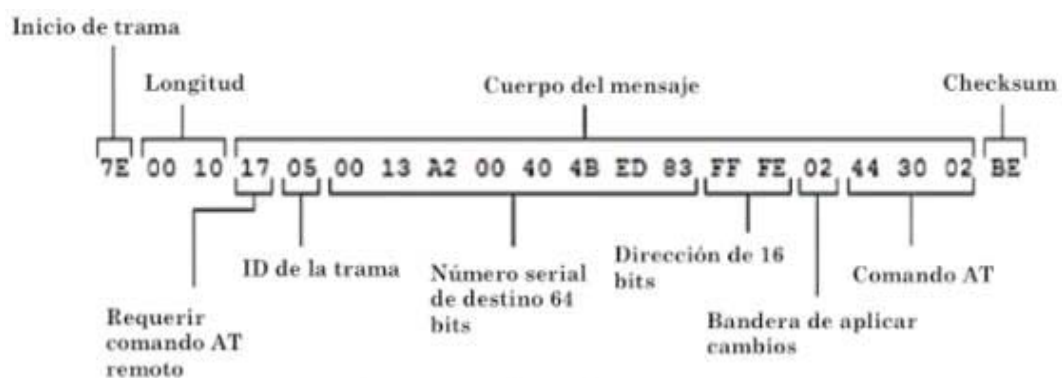


Figura 26. Uso de la trama API con comandos AT.

Tomado de (Modo API 2, 2017)

Este tipo de trama se utiliza para enviar de manera remota comandos a otros módulos que se encuentran dentro de la red. En la tabla 10, se describe las diferentes partes de la trama:

Tabla 10.

Estructura de la trama.

BITS	Significado
7E	Indica el inicio de la trama.
00 10	Indica la longitud de la trama, se encuentra en hexadecimal

	(decimal =16).
17	Bit que indica el requerimiento remoto
05	ID de la trama de envió al nodo remoto, el mismo que debe responder con la misma ID dentro de su trama.
0013A200404BED83	Dirección serial de 64 bits del nodo remoto.
FF FE	Dirección de 16 bits, en caso de no saberlo se ingresa se ingresa los valores FF FE. (Se puede transmitir a 64 y 16 bits).
02	Bit que indica que se realice los cambios inmediatamente.
44 30 02	Comando AT, en esta trama indica que la entrada D0 la vuelva entrada análoga.
BE	Checksum, indica que la trama se encuentre bien diseñada.

Adaptado de (Comandos AT, 2016)

Comando AT para respuesta de los requerimientos remotos

a) AT (0x97)

El módulo remoto al recibir un comando, en respuesta enviará un mensaje de confirmación. El módulo puede enviar varias tramas de respuesta, dependiendo el tipo de comando que reciba. A continuación, se muestra una trama de respuesta. La misma que se puede ver en la tabla 11.

Trama de respuesta.

7E - 00 0F - 97 - 01 - 00 13 A2 00 40 70 53 2E - 00 00 - 64 30 - 00 -ED

Tabla 11.

Trama de respuesta de un equipo remoto.

Bits de la trama	Significado de los bits
7E	Indica el inicio de la trama.
00 0F	Indica la longitud de la trama, se encuentra en hexadecimal (decimal = 16).

97	Indica que es una respuesta en comando AT
01	Identificación de la trama
00 13 A2 00 40 70 53 2E	Dirección serial de 64 bits del nodo principal.
00 00	Bits de la dirección de 16 bits, no se conoce cual es del nodo principal. (Se puede transmitir a 64 y 16 bits).
64 30	Parámetro AT de respuesta.
00	Comando AT de estado: Que indica okey o error. En este caso es okey.
ED	Checksum, indica que la trama se encuentre bien diseñada.

Adaptado de (Comandos AT, 2016)

1.4 RSSI en XBee

El RSSI en los dispositivos inalámbricos proporciona un resultado aproximado de la intensidad de señal recibida. Los módems de radio Digi contienen receptores que son capaces de recibir y amplificar señales muy débiles de un transmisor distante. La señal más débil que un receptor puede capturar y desmodular con éxito a una tasa de error de bits aceptable se denomina sensibilidad del receptor. (Tomado de DIGI, 2017)

Si el transmisor distante se mueve más cerca de un receptor, la intensidad de la señal transmitida en la antena receptora aumenta. El valor de la intensidad de la señal en la antena receptora permite determinar la calidad del enlace de comunicación. Si la aplicación está tratando de medir la fiabilidad del enlace y no sólo la intensidad de la señal puede ser útil tener en cuenta el% de paquetes recibidos o datos similares.

Uso del comando **DB**

RSSI está disponible como salida en el pin 6 de la XBee. Para determinar el RSSI, utilice XCTU para enviar el comando DB al módulo XBee. El mismo que lee el nivel de señal en decibelios (dB) del último paquete recibido (RSSI).

El XBee informa el valor absoluto de RSSI, por ejemplo, 0x58 = -88 Decibel-miliwatts (dBm). El valor que el XBee informa es exacto entre -40 dBm y la sensibilidad del receptor (RX). No considere valores de DB más fuertes que -40 dBm como precisos, porque el sensor está saturado en ese rango y su comportamiento es no lineal.

1.5 Software XCTU

Es un software multi-plataforma gratuito que se lo puede descargar de la red, es utilizado por los programadores para la administración y configuración de los módulos XBee. También cuenta con una interfaz visual que permite realizar las pruebas de conexión y de representación gráfica de los elementos de la red. Entre las principales características están las siguientes:

- Permite diseñar las tramas API y manejo de comandos AT.
- Se utiliza para configurar los módulos XBee y su topología de trabajo.
- Permite visualizar el nivel de la intensidad de la señal de cada conexión.
- Faculta la posibilidad de configuración remota de los nodos.
- Actualiza el firmware de los XBee.
- Posee dos tipos de consolas: para el modelo API y la otra para el modo AT. Las mismas que permiten realizar pruebas de las diferentes tramas y comandos AT.
- Permite la recuperación de módulos que se encuentran dañados su firmware, los restaura a modo fábrica.
- Se puede realizar pruebas de rango entre dos módulos de la misma red.
- Toda la documentación se encuentra en la página de DIGI, la empresa creadora de los dispositivos XBee.

En la figura 27 se puede ver la pantalla principal del software.

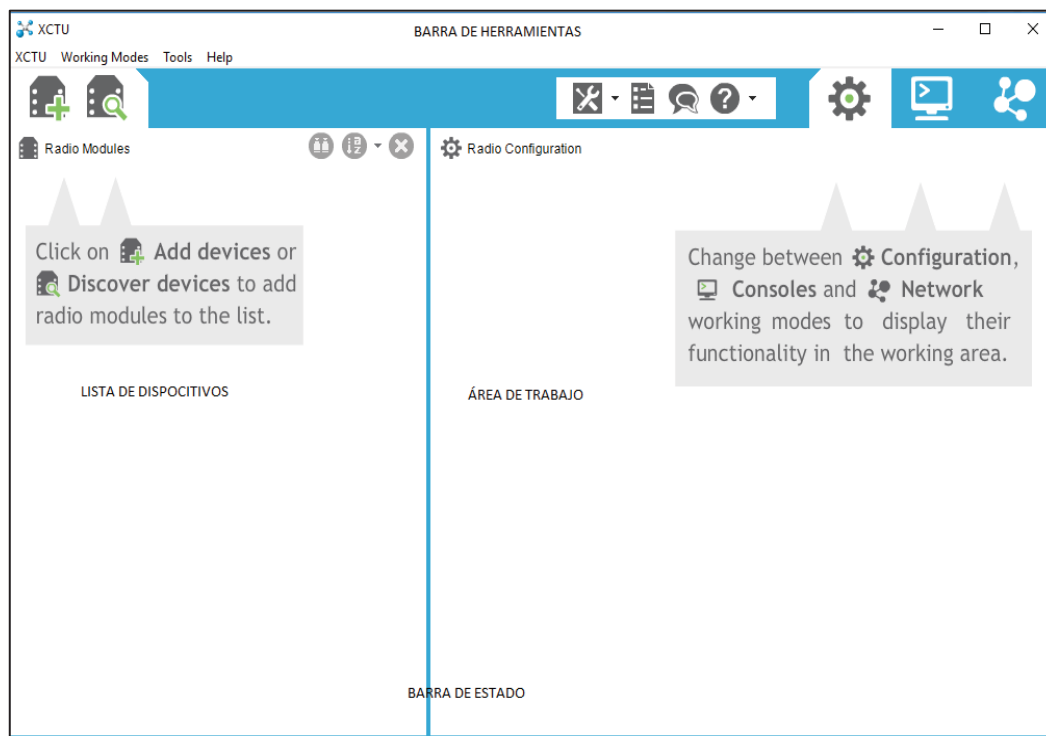


Figura 27. Interfaz del programa XCTU.

1.6 Arduino IDE

El software Arduino es un programa abierto (IDE) que facilita la escritura de código y su posterior montaje en tarjetas Arduino. Además, el software es multiplataforma, está escrito en Java y basado en Processing y otros programas de código libre. En la figura 28 se muestra el símbolo de Arduino.



Figura 28. Símbolo de Arduino IDE.

Tomado de (Arduino UNO, 2017)

1.7 Visual Studio

Es un programa que posee un conjunto de herramientas para el diseño y desarrollo de aplicaciones de escritorio, móviles, Servicios Web XML y aplicaciones WEB. Además, soporta múltiples lenguajes de programación tales como: Visual C#, Visual Basic y Visual C++, Python, Ruby y PHP. Todos estos lenguajes utilizan las funciones de .NET Framework, que permite acceder a tecnologías importantes para el desarrollo de las aplicaciones. (Microsoft Introducción a Visual Studio, 2017)



Figura 29. Logo Visual Studio.

Tomado de (Visual Studio, 2017)

El programa es muy interactivo ya que permite arrastrar los labels, cajas de texto, botones, para el desarrollo de la aplicación. Permite realizar el diseño gráfico y modificarlo cuantas veces se desee, se programa lo que se requiere que pase en los diferentes botones cuando el usuario interactúe con el programa. Además, posee un corrector de errores dentro de la interfaz de programación, que permite realizar los cambios requeridos. En la figura 29 se muestra el logo de Visual Studio.

2. CAPÍTULO II. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se realizó el diseño, desarrollo e implementación del proyecto de medición de contaminación auditiva. Primero se empieza realizando el estudio de los elementos y el software a utilizar en la red WSN, en la construcción del nodo sensor y en la programación del software de monitoreo.

Luego se continúa con el diseño la red WSN, haciendo uso de la topología MESH y construcción del nodo sensor. Además, se indica los parámetros utilizados para el diseño de la trama API. En este capítulo también se incluye la programación que se realizó en los módulos Arduino, y en el software de monitoreo. Por último, se indica el nodo terminado de construir, y el funcionamiento del mismo con la adquisición de la información desde los nodos remotos y la visualización de la información en el software.

2.1 Diseño en bloques del sistema de medición de contaminación auditiva

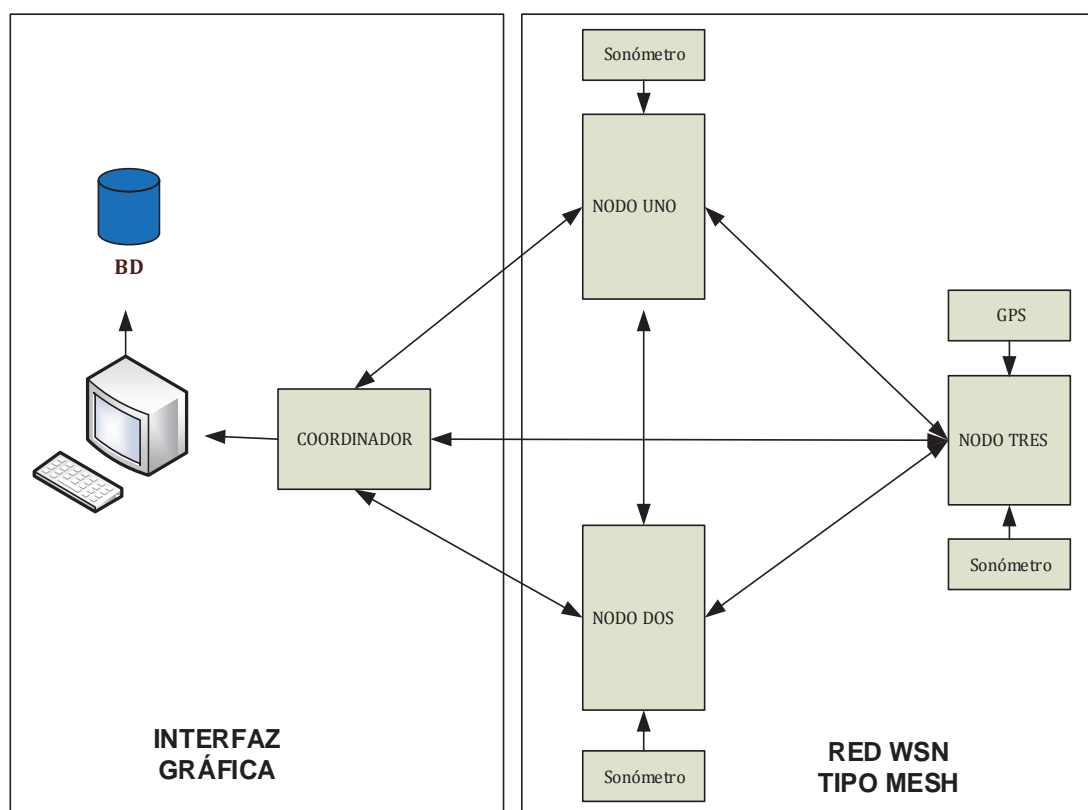


Figura 30. Diagrama en bloques del sistema WSN Mesh.

En la figura 30 se muestra un diagrama en bloques del diseño del medidor de contaminación auditiva. La misma que se divide en dos bloques: red WSN tipo mesh e interfaz gráfica. A continuación, se describe cada bloque.

a) Bloque de la Red WSN tipo Mesh

Este bloque consta de tres nodos configurados en conexión tipo MESH que monitorean y captan la señal análoga, luego la información se lo empaqueta en una trama API, junto a los datos del GPS y son enviados de forma inalámbrica hacia el coordinador. Además, los nodos son los encargados de encender y apagar los sensores de ruido.

b) Bloque de la interfaz gráfica

Para la parte de la interfaz gráfica se utilizó un software que permite realizar, la programación de lo requerido de una manera eficiente. El mismo que permite visualizar la información, del nivel de ruido de los nodos remotos. Además, esta entrelazada a una base de datos, en la cual se guarda la información monitoreada. También consta de un dispositivo coordinador, que es el encargado de la comunicación entre el software de monitoreo y los nodos remotos.

2.2. Estudio comparativo de hardware y software

Para seleccionar los dispositivos que se utilizaron en el proyecto se realizó un estudio comparativo de los elementos. Entre las principales características que se tomó, fueron las siguientes: costos, tipo de conexión, voltaje y corriente requeridos para su funcionamiento, alcance de transmisión y modos de funcionamiento.

2.2.1 Elementos de Hardware

A continuación, se detalla los elementos comparados para determinar el mejor dispositivo para el proyecto.

2.2.1.1 Arduino Mini Pro versión de 5V

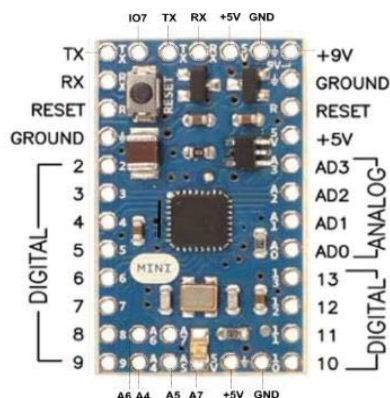


Figura 31. Arduino MiniPro.

Tomado de (Arduino Mini Pro, 2017)

El arduino mini pro es un dispositivo electrónico que opera con el microcontrolador ATmega328, debido a su diseño permite fácilmente realizar la soldadura directa de cables o el montaje de los pines según lo requiera. El dispositivo es pequeño permitiendo utilizarlo en circuitos que poseen poco espacio. Además, se los puede encontrar en dos versiones: de 5v y de 3.3vDC. En la figura 31 se observa el arduino Mini Pro y en la tabla 12 se muestra las principales características del dispositivo. Ver hoja de datos en Anexo B.

Tabla 12.

Parámetros técnicos del Arduino Mini Pro

Parámetros	Descripción
Micro controlador	ATmega328
Voltaje de Operación	5-12V
E/S Digitales	14
Corriente para E/S	40mA
Pines Analógicos	6
PWM	6
UART	1
Memoria Flash	32K
Velocidad de Reloj	16MHz

Adaptado de (Arduino Mini Pro, 2017)

2.2.1.2 Módulo ZigBee XBee PRO S2B



Figura 32. Módulo XBee PRO S2B.

Tomado de (ZigBee RF Modules, 2017)

Entre sus principales características están las siguientes: bajo consumo de corriente en las comunicaciones inalámbricas, permite tener autonomía extendida a 63mW y maneja velocidades de hasta 1Mbps en la banda de 2,4GHz, como se muestra en la tabla 13. Estos dispositivos son bastante utilizados en redes WSN debido a su tamaño reducido, como se puede ver en la figura 32. Para el montaje del módulo XBee se utiliza la tarjeta Xbee Arduino Compatible Shield Module V3.0. Su protocolo de comunicación es el modo AT y API versión uno o dos. Ver hoja de datos en Anexo C.

Tabla 13.

Características técnicas del módulo XBee.

Parámetros	Descripción
Alcance Línea de vista	1500m
Alcance Interiores	90m
Frecuencia	2,4GHz
Velocidad de Datos	1Mbps
Voltaje de Operación	5V

Adaptado de (ZigBee RF Modules, 2017)

2.2.1.3 XBee Arduino Compatible Shield Module V3.0

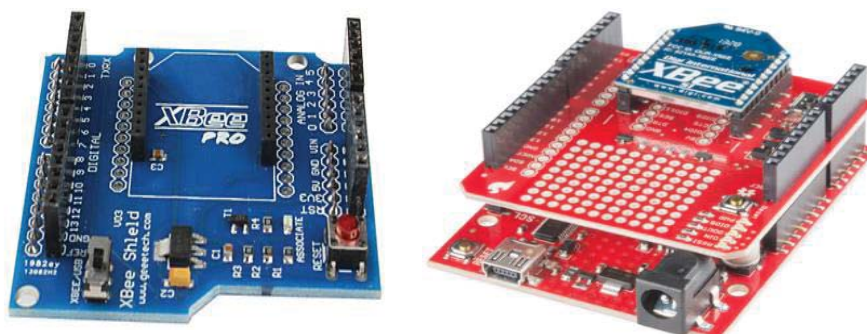


Figura 33. Módulo Xbee Arduino Compatible Shield Module V3.0.
Tomado de Módulo (Xbee Arduino Compatible Shield, 2017)

Dispositivo utilizado para el acople de los módulos XBee en la placa arduino. Básicamente va montado sobre el Arduino Uno y sobre el módulo Xbee Arduino Compatible Shield Module V3.0 va montado el Módulo XBee. Posee un regulador de voltaje de 5V y cuatro pines en la parte posterior los mismos que son: Dout, 5V, Din y GND. Además, es compatible con todos los módulos XBee, de las versiones: Pro, Serie 1 y Serie 2.5 estándar, como se muestra en la figura 33.

2.2.1.4 Módulo GPS modelo GY-GPS6MV2.

El dispositivo GPS contiene una memoria EEPROM configurada de fábrica, la misma que mantiene sus datos gracias a que en el módulo viene incorporado una pequeña pila, como se puede ver en la figura 34. También consta de un led indicador de transmisión y recepción, una pequeña antena de cerámica y cuatro pines de RX, TX, VCC, GND, para la comunicación serial a la placa principal. Se recomienda hacer las pruebas en ambientes abiertos o cercanos a un lugar con vista al exterior, para obtener una correcta recepción de la señal. En la tabla 14, se muestra las principales características del módulo GPS.



Figura 34. Módulo GPS.

Tomado de (Módulo GPS Arduino, 2017)

Tabla 14.

Características técnicas del módulo GPS.

Parámetros	Descripción
Voltaje de Operación	5V
Canales	50
Frecuencia de refresco	5Hz
Soporta	SBAS,WAAS,EGNOS
Baud Rate	9600bps
Batería de Respaldo	MS621FE

Adaptado de Módulo GPS Arduino (2017)

2.2.1.5 Sonómetro Modelo SL-1361



Figura 35. Sonómetro modelo SL-1361.

Tomado de (Maiten, 2017)

Dispositivo compuesto de un receptor de sonido que permite captar un rango de frecuencias variada desde 20Hz-8KHz, +/- 0,5 dB, permitiendo cumplir la normativa IEC651 Tipo 2 ANSI S1.42. También dispone de una pantalla a color y permite escoger la ponderación A y C, como se puede ver en la figura 35. En el presente proyecto se utilizará la salida analógica DC. En la tabla 15 se muestra las principales características del sensor de ruido. Ver hoja de datos en Anexo D.

Tabla 15.

Características técnicas del sonómetro SL-1361.

Parámetros	Descripción
Voltaje de Operación	6V
Rango	30dB-130dB
Resolución	0,1dB
Tiempo de Respuesta	2-5 veces por segundo
Peso	300g
Pines de Salida	AC/DC
Precisión	+/- 1,5dB (94dB a 1Khz)

Adaptado de (Maiten Chile, 2017)

2.2.1.6 Arduino Uno modelo R3.

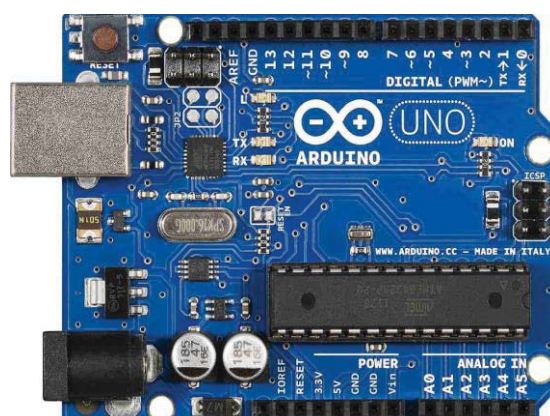


Figura 36. Arduino Uno R3.

Tomado de (Arduino Uno, 2017)

Arduino Uno es una placa electrónica open-source basada en el microcontrolador ATmega328P, dispone de catorce pines entre entradas / salidas, un conector de alimentación, un conector USB, un cristal de cuarzo de 16 MHz, una cabecera ICSP y un botón de reinicio, la misma que se puede ver en la figura 36. También cuenta con ATmega16U2 que maneja el USB, utilizado para la transferencia de datos a la PC. Su programación se realiza haciendo uso del software Arduino C.C, que también es open-source. (Arduino Uno, 2017). En la tabla 16 se muestra las principales características.

Tabla 16.

Características técnicas del Arduino Uno R3.

Parámetros	Descripción
Voltaje de Operación	5V
E / S digitales	14 (de los cuales 6 proporcionan salida PWM)
Memoria flash	32 KB (ATmega328P) de los cuales 0,5 KB utilizado por el gestor de arranque
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
PWM digital pines I / O	6
Pines de entrada analógica	6

Adaptado de (Arduino Uno, 2017)

2.2.1.7 Arduino MEGA

Es una placa electrónica basada en un microcontrolador ATmega 1280. Entre sus principales características estas las siguientes:

- Posee con 54 pines digitales de entrada / salida.
- Tiene 14 pines para ser utilizados como salidas PWM y 16 entradas analógicas.

- Un oscilador de cristal de 16 MHz.
- Conexión USB, un conector de alimentación de energía.
- Memoria Flash 128 KB de los cuales 4 KB utilizados por bootloader.
- Una SRAM 8 KB.

En la figura 37, se puede ver la placa Arduino MeGA.

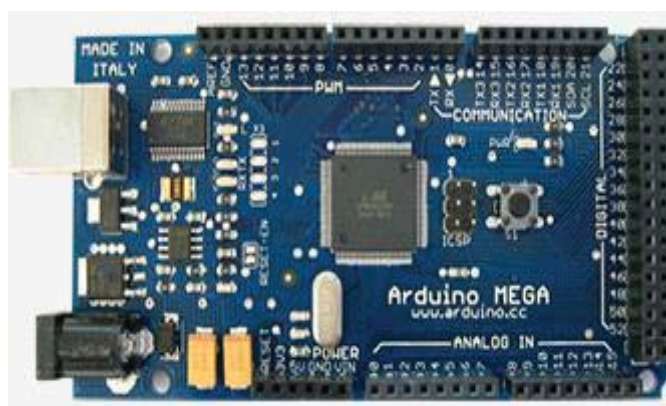


Figura 37. Arduino MEGA.
Tomado de (Arduino Mega, 2017)

2.2.1.8 Batería Z ZTDM 2Pcs 18650



Figura 38. Batería 3,7V.
Tomado de (Amazon s.f)

La batería Vamery 2Pcs modelos 18650 de 3.7V-4.2V 5000MAH Li-ion Rechargeable Efficient Battery Blue, tiene una descarga lenta y dura un

promedio de 6 a 8 horas, antes de ser cargadas nuevamente. Ideal para el proyecto que se está realizando. En la figura 38 se muestra la batería de 3.7VDC.

2.2.1.9 Resumen de los elementos a utilizar

Después de haber realizado la descripción de las especificaciones de cada dispositivo, se determina que los elementos antes mencionados cumplen con las características técnicas y óptimas para el proyecto.

Se descarta la utilización de un módulo XBee que no sea ya del modelo PRO S2B, ya que las demás versiones no soportan, la utilización de la trama API. También no se va utilizar el arduino mega ya que no se necesita muchos pines de salida y entrada, en su lugar se utilizó el arduino UNO.

2.2.2 Elementos de software

Para el diseño de la interfaz gráfica se determinó, el software a utilizarse después de realizar un estudio previo de los posibles softwares. A continuación, se muestra los softwares comparados.

2.2.2.1 Lenguaje C

Lenguaje de programación creado en los años 80, su programación es de una forma estructurada, de alto nivel. No es orientada a objetos, y es más difícil su programación ya que no tiene interfaz gráfica de pruebas.

2.2.2.2 Lenguaje C++

Software C++ es la evolución del lenguaje C, permite programar tanto en bajo como en alto nivel, y es orientado a objetos. Su principal desventaja es que todo se debe realizar de forma manual, haciendo difícil su programación.

2.2.2.3 Java

Es un software libre orientado a objetos, muy flexible, funciona en cualquier plataforma que lo vayan a ejecutar, no representa costo ya que no es licenciado y es un lenguaje expandible. Su principal desventaja es que no tiene herramientas visuales para realizar las pruebas de funcionamiento.

2.2.2.4 Visual Studio

Es un programa que posee un conjunto de herramientas para el diseño y desarrollo de aplicaciones de escritorio, móviles, Servicios Web XML y aplicaciones WEB. Además, soporta múltiples lenguajes de programación tales como: Visual C#, Visual Basic y Visual C++, Python, Ruby y PHP. Todos estos lenguajes utilizan las funciones de .NET Framework, que permite acceder a tecnologías importantes para el desarrollo de las aplicaciones. (Microsoft Introducción a Visual Studio, 2017). También permite la conexión a la base de datos. Su principal ventaja en relación a los demás programas es su conjunto de herramientas visuales que permite realizar las pruebas en tiempo real, de una manera rápida.

2.2.2.5 Resumen del software a utilizar

Después de revisar los diferentes softwares, se determina que la mejor opción es Visual Studio. Ya que la gran ventaja que tiene es la compatibilidad con la base de datos SQL server. Además, tiene una consola para las pruebas en tiempo real, su alta gama de herramientas visuales permite fácilmente crear los diseños del programa de monitoreo.

2.3 Base de datos

Para determinar la base de datos a utilizarse, primero se determinó en que lenguaje de programación se iba a desarrollar la interfaz gráfica. Una vez que

se escogió el software Visual Studio con lenguaje C sharp, para el desarrollo de la interfaz, también se determinó que la base de datos seria SQL server 2014, ya que visual studio es compatible solo con la mayoría de versiones de SQL.

2.3.1 SQL Server

Según Microsoft, SQL server es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos. En la tabla 17, se muestra las versiones de SQL.

Tabla 17.

Versiones de SQL Server.

Año	Nombre de Versión
1989	SQL Server 1-0
1993	SQL Server 4.21
1995	SQL Server 6.0
1996	SQL Server 6.5
1998	SQL Server 7.0
1999	SQL Server 7.0 OLAP Tools
2000	SQL Server 2000
2003	SQL Server 2000 64-bit Edition
2005	SQL Server 2005
2008	SQL Server 2008
2010	SQL Azure DB
2010	SQL Server 2008 R2
2012	SQL Server 2012
2014	SQL Server 2014
2016	SQL server 2016

Se decidió utilizar la base de datos SQL SERVER 2014, ya que es compatible con Visual Studio y es más robusta en comparación con las demás bases de datos.

2.4 Diseño del sistema de medición de contaminación auditiva

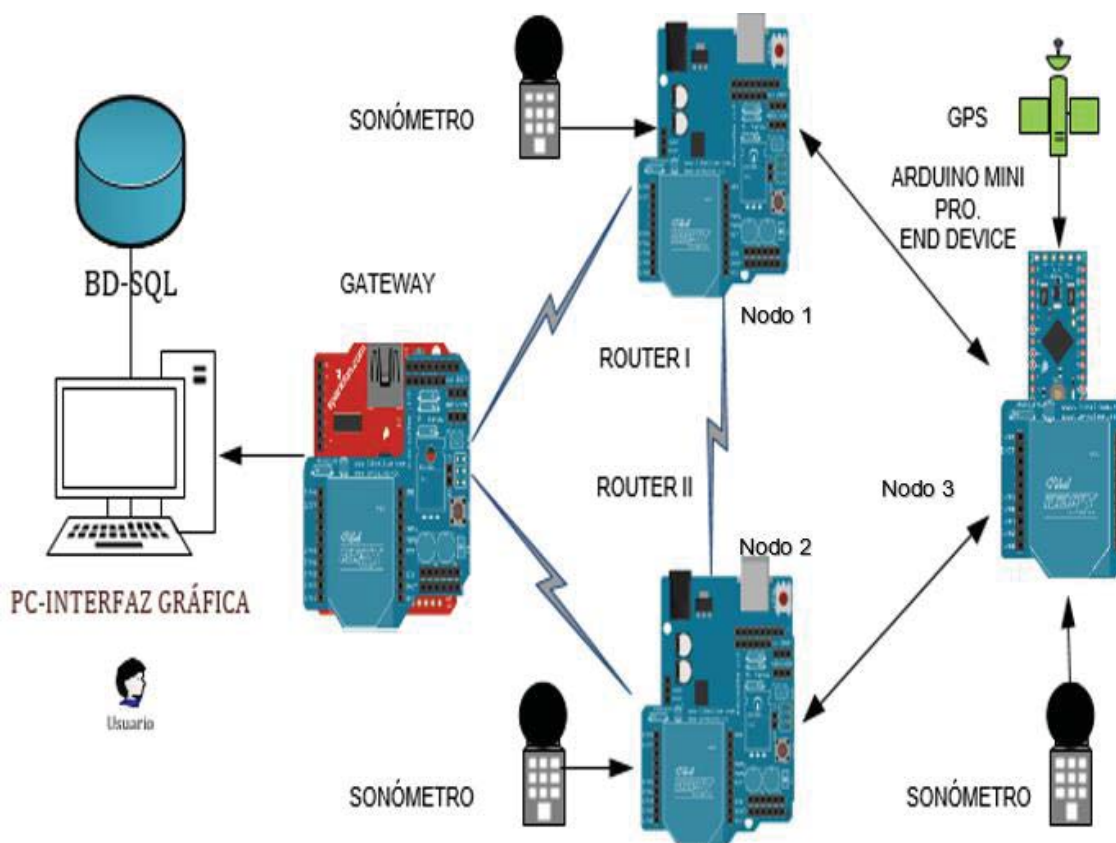


Figura 39. Diseño del sistema de medición de contaminación auditiva.

El diseño del sistema y la interconexión de los dispositivos se muestran en la figura 39. La alimentación es de 5VDC constantes mediante el circuito de baterías o fuente de poder externa. En la placa principal se conecta, el módulo GPS, módulo Xbee Pro S2B, el arduino Mini Pro, baterías y el Sonómetro. Este último encargado de censar el nivel de ruido. En el arduino mini Pro se realizó la programación, que permite enviar los datos encapsulados en una trama API, hacia el coordinador.

Los nodos enrutadores son los dispositivos que permiten tener dos caminos para llegar al Gateway. El nodo coordinador es quien recibe la información y transmite en forma serial al software de monitoreo. También se encarga de determinar la mejor ruta para la comunicación de todo el sistema.

En el proyecto se **usó** dos nodos tipo routers, los mismos que se **utilizaron** para armar una red tipo mesh. En la figura 40 se muestra un diagrama en bloques que indica la lógica de funcionamiento del sistema de medición de contaminación auditiva en modo API.

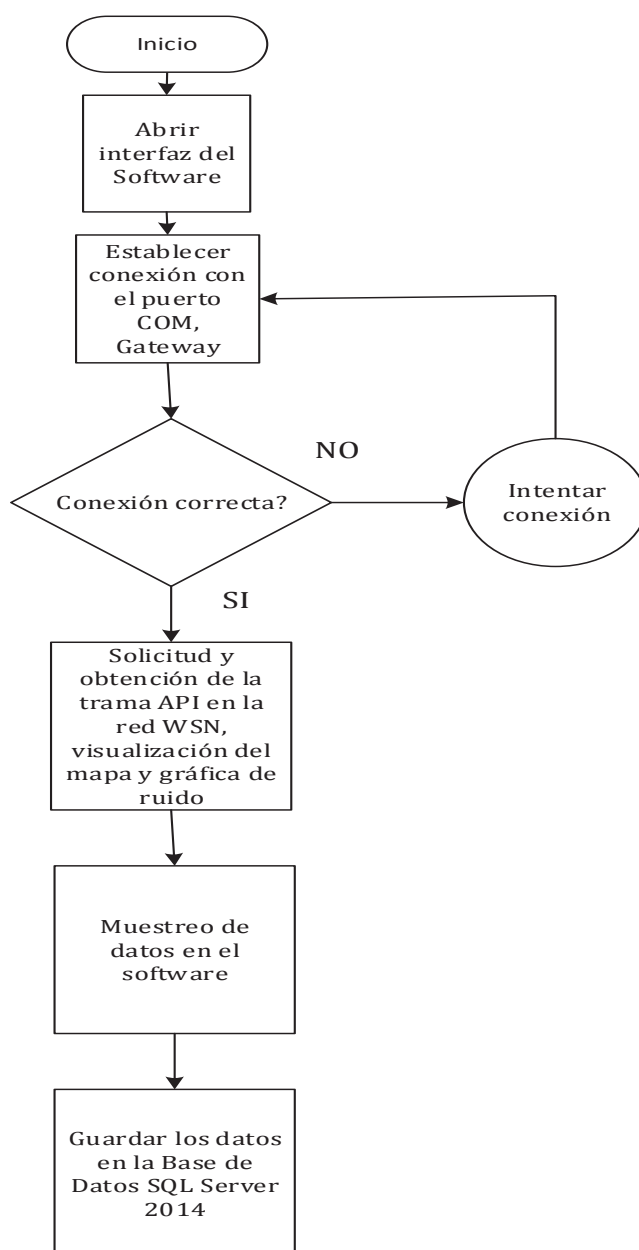


Figura 40. Funcionamiento del sistema WSN.

Al iniciar el sistema lo primero que se debe realizar es la búsqueda del puerto COM, que se encuentre disponible para la conexión del GATEWAY con el software. Una vez encontrado el puerto, desde el software Visual Studio presionar conectar y se establecerá la comunicación con el GATEWAY, si la conexión no es establecida se debe reiniciar el programa o se selecciona otro puerto COM.

Luego escoger que nodo se va a monitorear y enviar a encenderlo remotamente, para certificar que se ha encendido el programa muestra un mensaje de ENCENDIDO en la pantalla. Una vez encendido presionar iniciar y en la pantalla mostrara los valores de ruido del nodo remoto. Además, se puede determinar en qué parte está ubicado el nodo (Solo el nodo principal), simplemente escoger el nodo a ubicar y presionar UBICACIÓN en el software, automáticamente mostrara la posición haciendo uso del GPS. Y para finalizar se guardará los datos en la base de datos SQL SERVER 2014.

2.4.1 Diseño del nodo sensor

Par la construcción del nodo sensor, primero se realizó el diseño en el software PROTEUS 8 y EAGLE 7.6.0, para luego pasarla a una baquelita, una vez realizado esto, se conectan los elementos como: GPS, SONÓMETRO, XBEE, reguladores, y baterías pequeñas para la alimentación de energía. Para realizar la programación del nodo se utiliza el software ARDUINO 1.8.1, y para pasar la programación al arduino MINI PRO, se utilizará un Arduino Uno una vez realizado las respectivas conexiones en los pines programadores.

Esta forma de conexión permite enlazarse al computador y realizar pruebas en tiempo real, con el nodo. El GPS está conectado a los pines 2 TX y pin 3 RX. La señal del sonómetro se encuentra ingresando a través del pin análogo A1.

El transistor BT151, permite controlar el encendido y apagar el sonómetro, esto se logra utilizando el ping 10 del arduino. En la figura 41, se puede ver el diagrama de conexión, de los diferentes dispositivos.

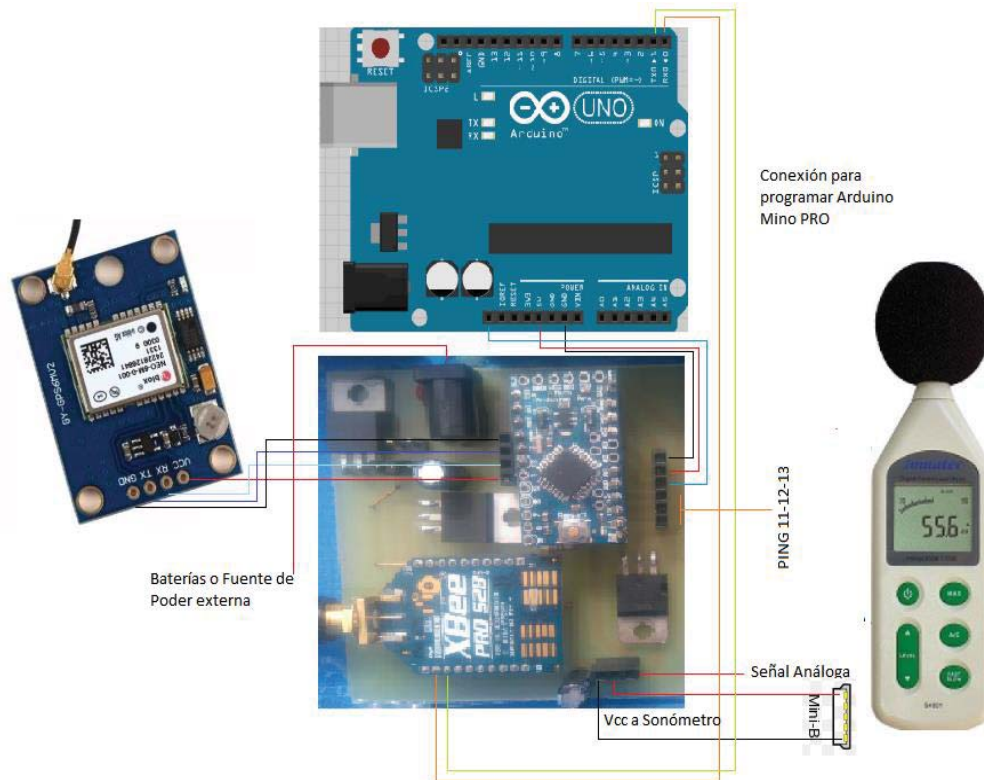


Figura 41. Diseño del nodo sensor.

En la figura 42, se observa el diseño de la placa principal realizado en el programa Arduino. La placa principal consta de dos etapas: la primera denominada etapa de energía. La misma que está compuesta por un Plug de alimentación, que permite el uso de baterías o fuente de energía externa. A continuación se encuentra un regulador de voltaje 7805, que regula el voltaje para alimentar a los demás elementos con un máximo de salida de 5v a 1A.

En la parte de control se encuentra un Arduino Mini pro, sobre el cual se encuentran conectadas varias salidas: para el módulo XBee, GPS y sonómetro. Para el encendido y apagado del sonómetro se está utilizando un transistor BT151, el mismo que es activado por el PIN número diez del arduino. También se utiliza un regulador LM1117DT-3.3 que permite regular a 3.3V, requeridos por el Xbee. Adicionalmente posee un puerto programador que permitirá cargar la programación a la placa desde la PC.

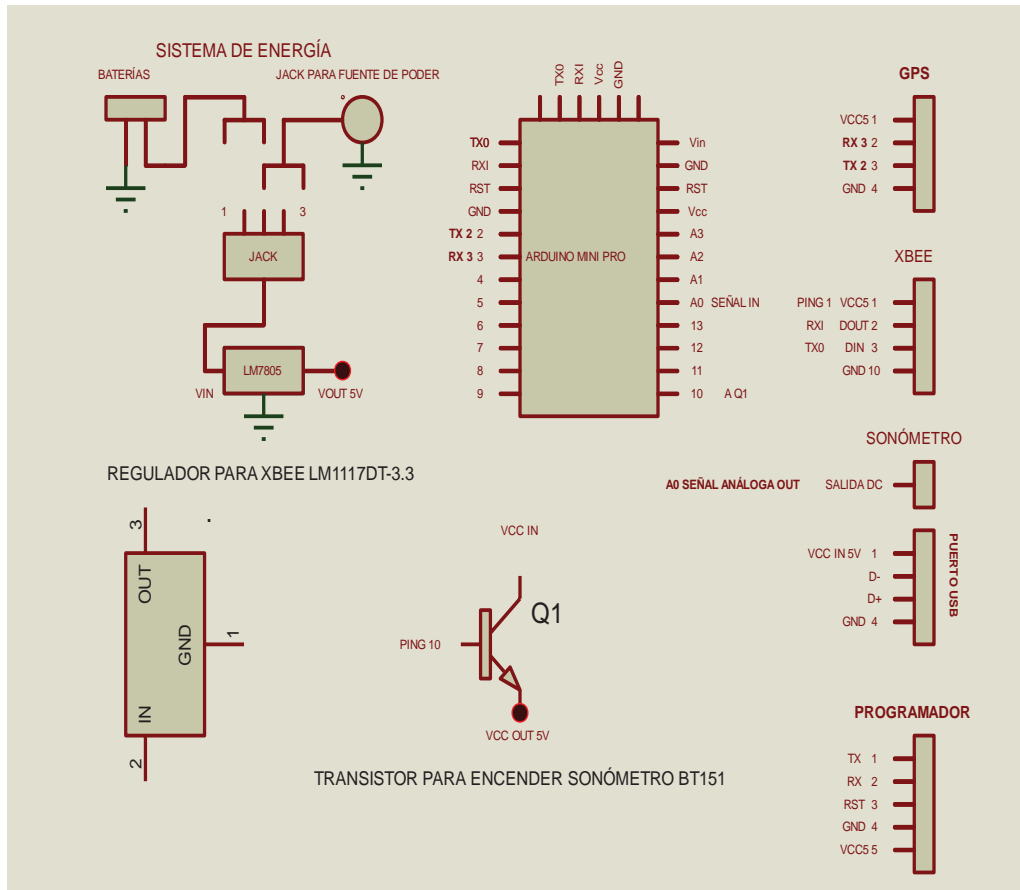


Figura 42. Circuito electrónico de la placa principal.

Para los nodos routers se utiliza una fuente de poder externa de 9VDC 1A y un transistor BT151 para controlar el encendido y apagado el sonómetro. De la misma manera que en la placa principal se utiliza el pin número diez del Arduino UNO para activar el transistor. El diseño de los nodos routers se lo puede ver en la figura 43.

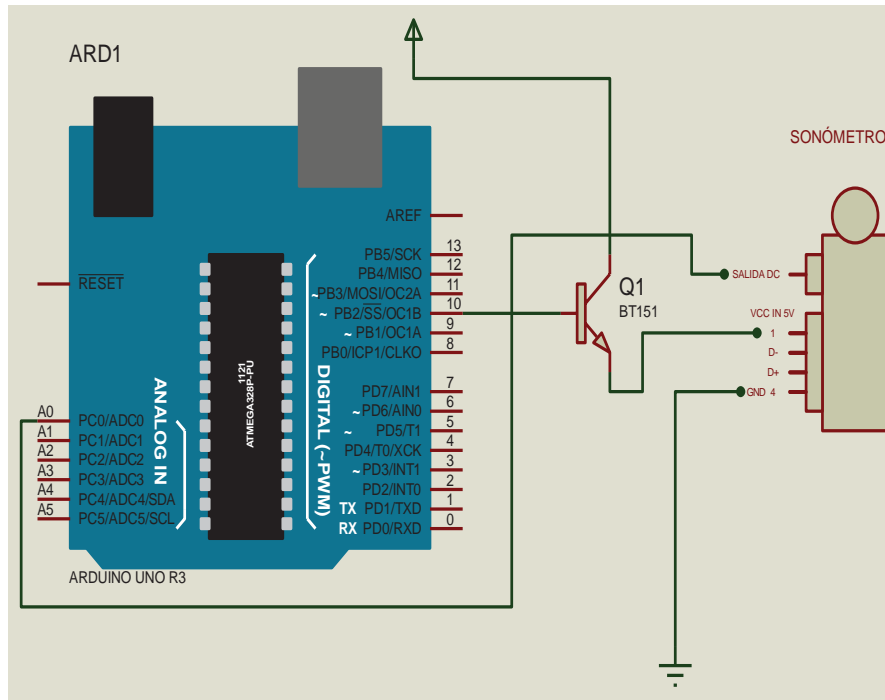


Figura 43. Circuito electrónico de nodos routers.

Mediante la utilización del software EAGLE se realiza el esquema del circuito impreso, como se muestra en la figura 44. El software permite realizar diseños de circuitos electrónicos y a su vez, tiene la función imprimir en formato PDF, para a su vez quemarlo en una baquelita y tener el diseño impreso.

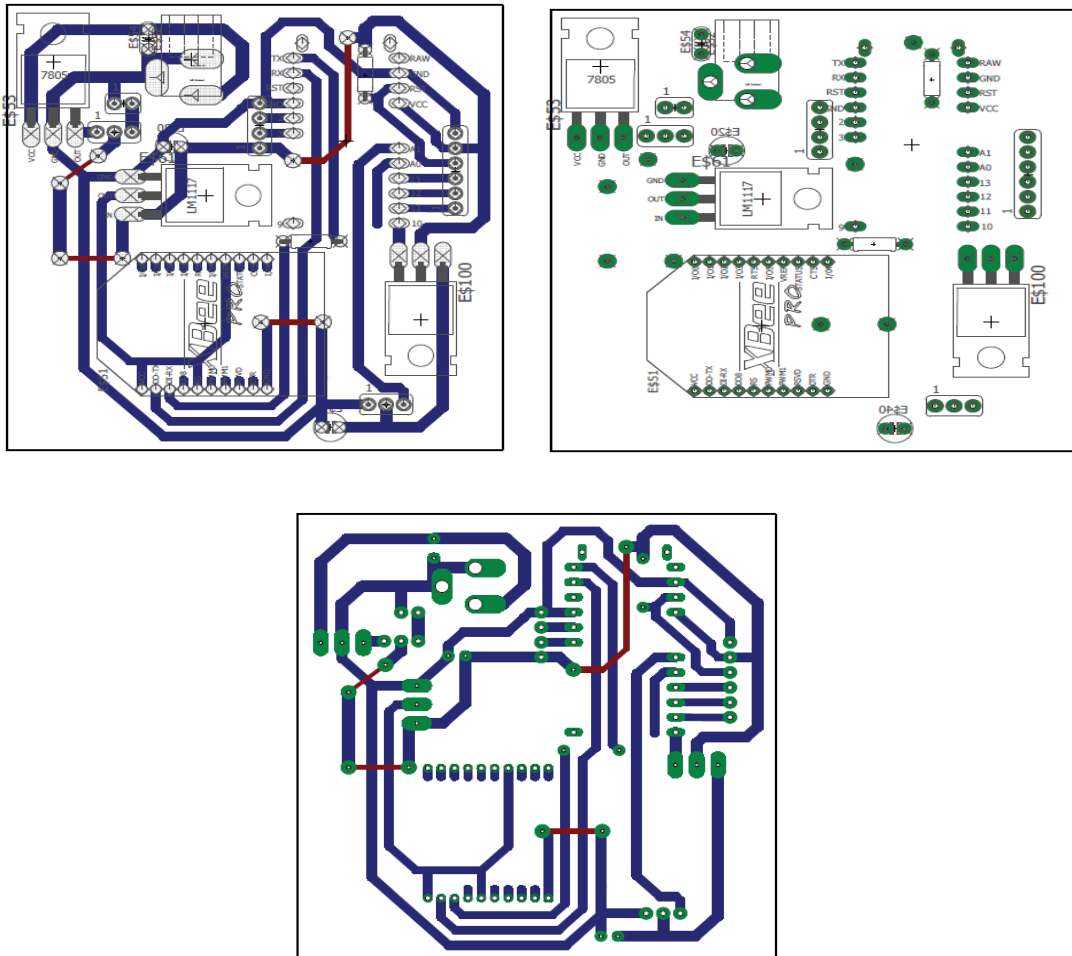


Figura 44. Circuito de la placa principal en software EAGLE.

Haciendo uso del software AutoCad se realiza el diseño de una caja, donde se ubicarán la placa electrónica y todos sus elementos, como se puede ver en la figura 45. Se utiliza este software ya permite obtener las medidas exactas del diseño y también se puede visualizar en formato 3D.

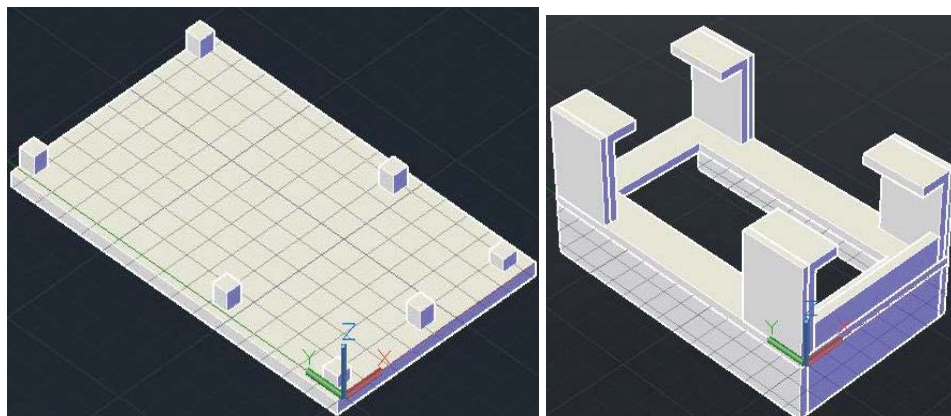


Figura 45. Caja realizada en software Autocad 2018.

Después de realizar el diseño, se procede a imprimir en una impresora en 3D, como muestra la figura 46.

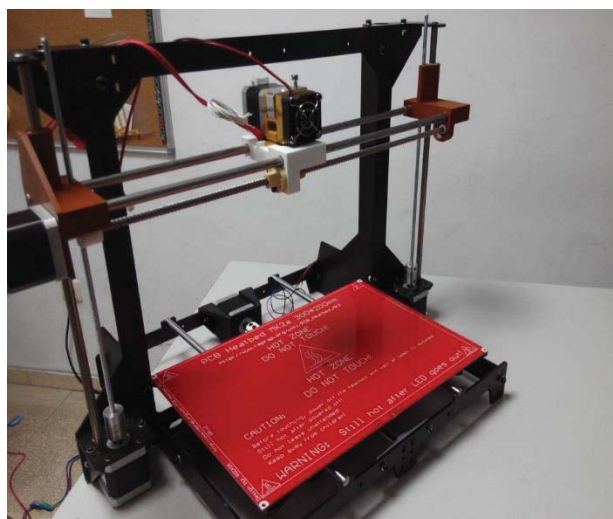


Figura 46. Impresión en 3D.

Y finalmente en la figura 47, 48 y 49, se puede ver el nodo sensor terminado de construir.



Figura 47. Placa del prototipo de medición de ruido.

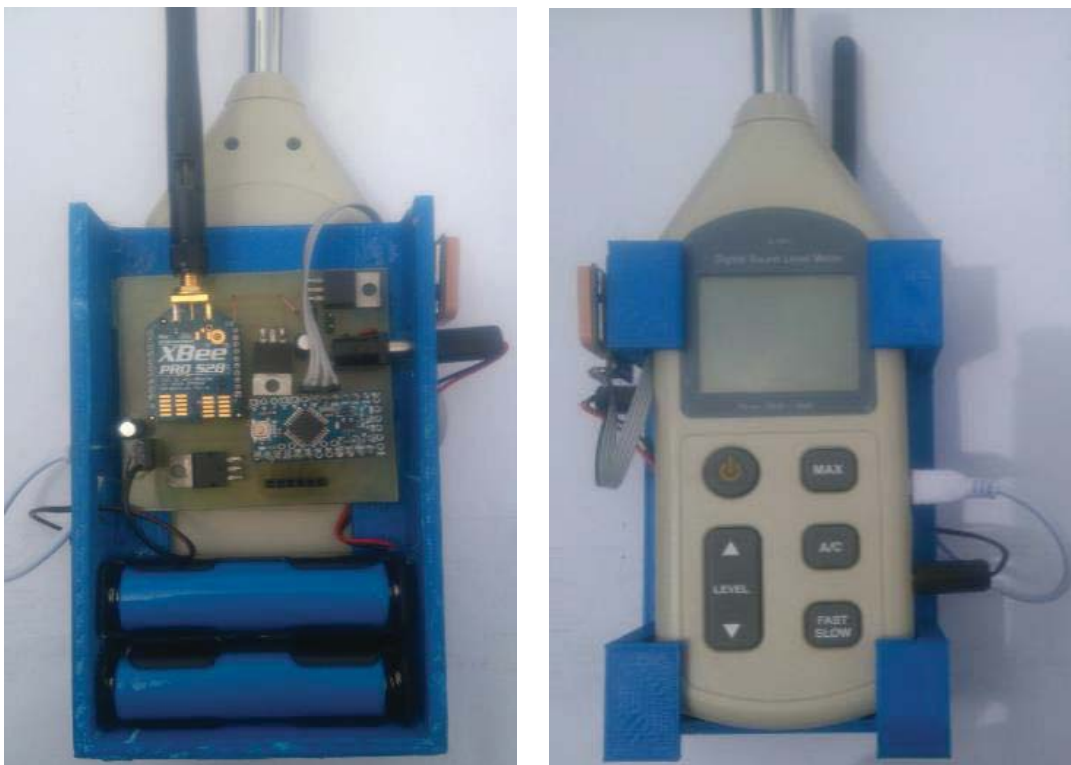


Figura 48. Prototipo terminado de construir.



Figura 49. Prototipo router terminado de construir.

2.5 Diseño de la red Zig Bee en modo mesh

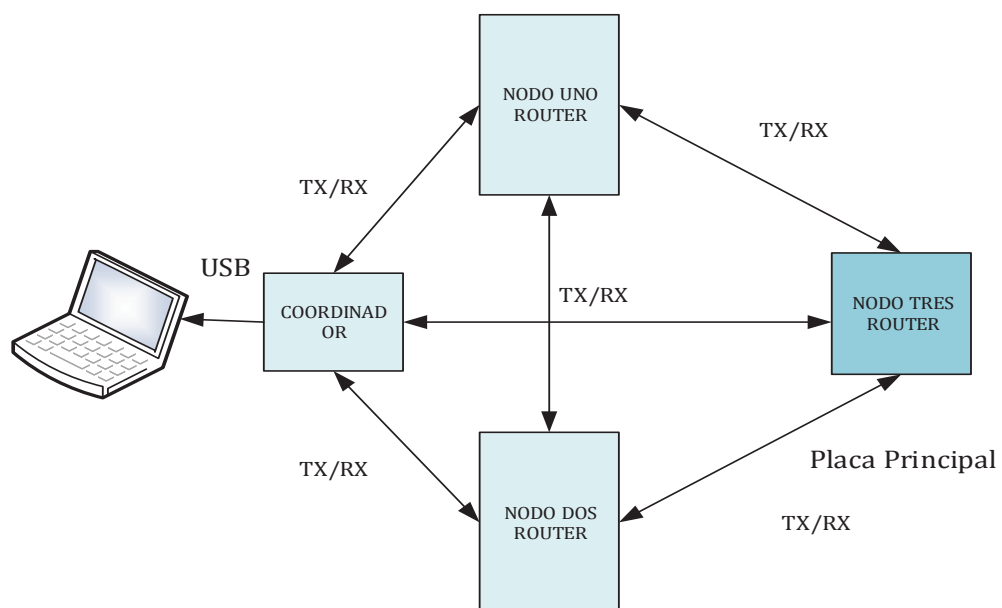


Figura 50. Diseño de la red MESH.

El diseño de la red fue basado en principios de la escalabilidad y ahorro de energía (figura 50), para el crecimiento de la red se tuvo que cambiar de

configuración el nodo dispositivo final, que se encontraba en el prototipo y se lo convirtió en nodo router. Por características el nodo configurado como dispositivo final tiene dos diferencias esenciales que son: a) pueden entrar en modo de sueño b) no pueden enrutar paquetes, por lo que no permite escalabilidad, de ahí el principal motivo de convertirlo en nodo router. Con la finalidad de ahorrar energía al nodo router, también se lo puede configurar en modo sueño. En la figura 51, se puede ver los 5 modos permitidos.

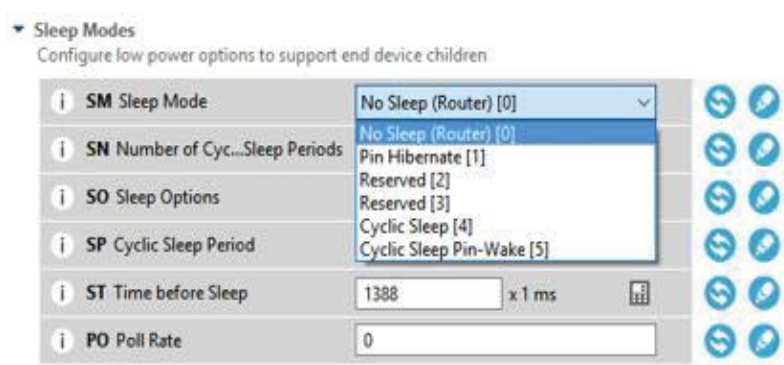


Figura 51. Tipos de configuraciones en modo sueño.

Considerando lo expuesto en el proyecto se utiliza dos arduinos UNO como routers, un Xbee explorer USB como coordinador y un arduino mini PRO como prototipo que también está configurado como router.

2.5.1 Enrutamiento de la red Mesh

El coordinador es el dispositivo que se encarga de realizar el descubrimiento de todos los nodos remotos y armar la red. También se encarga de verificar que se encuentre operativos todos los enlaces. El dispositivo configurado como enrutador se encarga de enrutar los paquetes, desde los nodo lejanos hacia el coordinador, para lo cual toma como referencia los niveles de enlace (LQI) más estables (varían desde un rango de 255 nivel óptimo a 190 valor mínimo para un correcto enlace), tomado este valor como referencia determinan cuales son los enlaces con mejor niveles de LQI y transmiten a través de esa vía.

También los nodos enrutadores se encuentran censando que los demás dispositivos estén operativos, si un nodo enrutador se cae, los otros equipos buscan los caminos alternos, para nuevamente llegar al coordinador. Como conclusión el enrutamiento en la red MESH, lo realizan los propios dispositivos, no se necesita configurar otro parámetro en el software XCTU, que no sea la de nodos enrutadores. En la figura 55 se puede ver la red MESH funcionando, con nodos enrutadores y coordinador.

A diferencia de una red punto a punto o estrella en la que solo se configura como, coordinador y dispositivo final, por lo tanto, no existe enrutamiento.

2.5.2 Configuración de la red WSN en modo mesh

Para la configuración de los XBee se utiliza el software XCTU.exe versión 6.3.5, lo primero que se realizó es añadir los módulos para empezar con la configuración. Una vez agregados, aparecerán en la lista de dispositivos, como se ve en la figura 52, se encuentran añadidos el coordinador, nodo 1 y nodo 2.

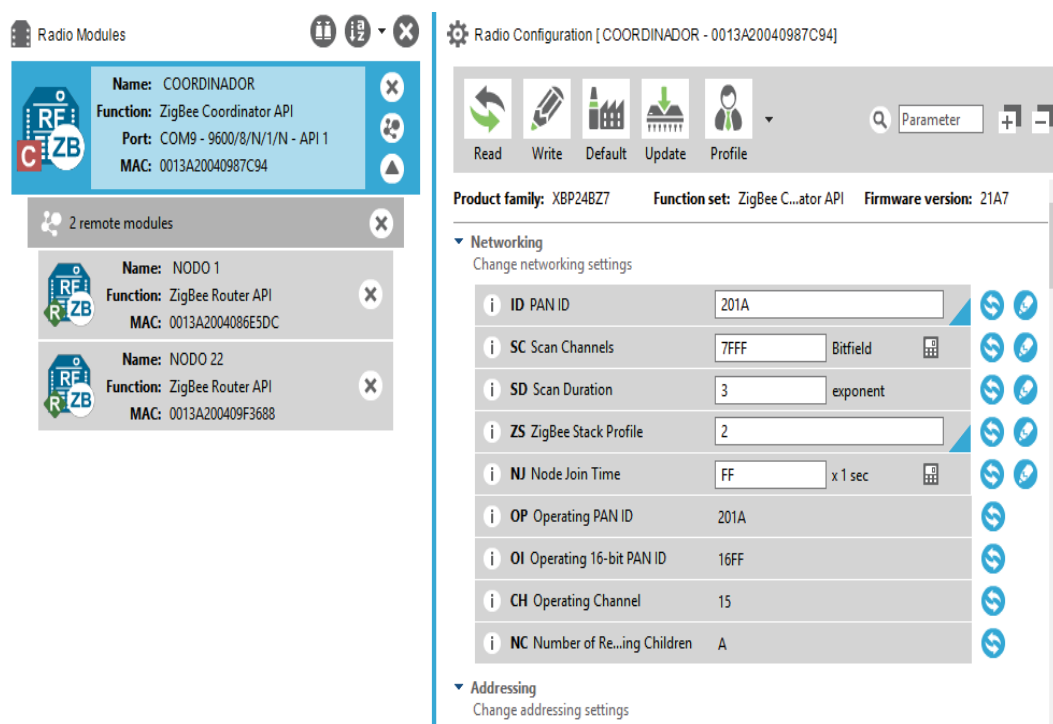


Figura 52. Interfaz XCTU, configuración de parámetros de la red.

Después se debe configurar la forma de trabajo del dispositivo ya sea como router, coordinador o dispositivo final. Seleccionar el dispositivo y dar clic en **Update**, inmediatamente aparecerá la siguiente figura (53).

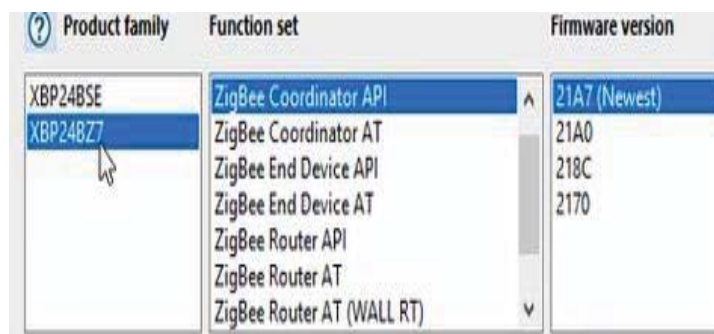


Figura 53. Selección del modo de trabajo de los dispositivo Xbee.

En la parte de **Product family** escoger el número que pertenece a la familia del dispositivo en este caso es XBP24BZ7, ese detalle se lo puede encontrar en el datashet. Luego en la parte de **Function set** escoger el modo a trabajar del dispositivo ya sea como router, coordinador o dispositivo final y por último en la parte de **Firmware versión**, es utilizada para realizar actualizaciones.

A continuación, en la tabla 18, se muestra los parámetros de configuración de la red mesh, en el área de trabajo del software XCTU.

Tabla 18.

Parámetros de configuración de la red WSN en modo MESH.

Campo	Nodo/Router	Coordinador	Característica
ID (PAN ID)	201A	201A	Es ID de la red
SC (Scan Channels)	7FFF	7FFF	Canal de comunicación. Cuenta con 16 canales.
ZS (Zigbee Stack Profile)	2	2	0=Network Specific, 1=ZigBee-2006, 2=ZigBee-PRO
JV (Channel Verification)	Enabled 1	N/A	Establecer / leer la configuración de verificación del canal. Si está activado, un enrutador verificará que

			existe un coordinador, para asegurarse de que está operando en un canal válido. Si está deshabilitado, el enrutador permanecerá en el mismo canal a través de ciclos de alimentación. Nota: Solo en los routers.
DH (Destination Address High)	0	0	Establece / lee los 32 bits superiores de la dirección extendida de destino de 64 bits. 0x0000000000000000FFFF es la dirección de difusión para el PAN. 0x0000000000000000 se puede utilizar para dirigirse al coordinador de pan.
DL (Destination Address Low)	FFFF	FFFF	Establece / lee los 32 bits inferiores de la dirección extendida de destino de 64 bits.
NI (Nodo Identifier)	Nodo 1	Nodo 1	Utilizado para poner el nombre, de los dispositivos.
NH (Maximun Hops)	1E	1E	Máximo número de saltos hasta encontrar los nodos dentro de la red. El fabricante recomienda no exceder de los 20 saltos.
EE (Encryption Enable)	Enable (1)	Enable (1)	Enable or disable ZigBee encryption.
BD (Baud Rate)	9600	9600	Establecer / leer la velocidad en baudios de la interfaz serie para la comunicación entre el puerto serie del módem y el host
AP (API Enable)	1	1	1= API 1 2= API 2. En el Proyecto se está realizando en API 1

Adaptado de (XCTU, 2017)

En la tabla 18, se muestran los valores que se necesitan configurar para armar la red mesh. La diferencia de la red en modo AT, es la no utilización de SH en DH y SL en DL. Debido a que en la red mesh todos los nodos tienen la dirección broadcast configurada lo que implica que los datos son recibidos

hacia y por todos los nodos de la red. La dirección broadcast es $DH=0x00000000$ $DL=0x0000FFFF$. (Xbee serie 2 configuración, 2017)

Estos pasos se deben realizar para la configuración de los demás nodos modificando únicamente el campo NI y JV. Es el presente proyecto se trabaja en modo API 1, ya que también existe API 2. En la siguiente figura se muestra los valores de SH y SL, que es definido como la dirección MAC. El mismo que será utilizado para en las tramas de comunicación. En la figura 54, se puede ver los valores de SH y SL.

El Gateway o coordinador recibe toda la información de la red, por ende, conoce todos los PAN ID de los Router y dispositivos finales. El pan ID es un número de cero a cuatro bits en la red, que permite conocer la pertenencia a esa red, el Gateway siempre se quedara libre para recibir el PAN ID de cualquier dispositivo.



Figura 54. XBee valores SH/SL.


Adaptado de (Xbee, 2017)

En la 19, tabla se muestra la configuración de los dispositivos del proyecto. Solo cambiaran los valores de NI y JV.

Tabla 19.

Configuración de los dispositivos del proyecto en modo Mesh.

Campo	Coordinador	Router 1	Router 2	Router 2
ID (PAN ID)	201A	201A	201A	201A
SC (Scan Channels)	7FFF	7FFF	7FFF	7FFF
ZS (Zigbee Stack Profile)	2	2	2	2
JV (Channel Verification)	N/A	Enabled 1	Enabled 1	Enabled 1
DH (Destination Address High)	0	0	0	0
DL (Destination Address Low)	FFFF	FFFF	FFFF	FFFF
NI (Nodo Identifier)	Coordinador	Nodo 1	Nodo 2	Nodo 3
NH (Maximum Hops)	1E	1E	1E	1E
EE (Encryption Enable)	Enable (1)	Enable (1)	Enable (1)	Enable (1)
BD (Baud Rate)	9600	9600	9600	9600
AP (API Enable)	1	1	1	1

En el software XCTU específicamente en la pestaña Switch to Network Working Mode  se puede ver la topología de trabajo de los XBEE, en este caso se encuentra trabajando en modo MESH. La misma se observa en la figura 55, en donde se ve, que todos los nodos se encuentran interconectados, generando caminos de redundancia ante posibles fallos de comunicación.

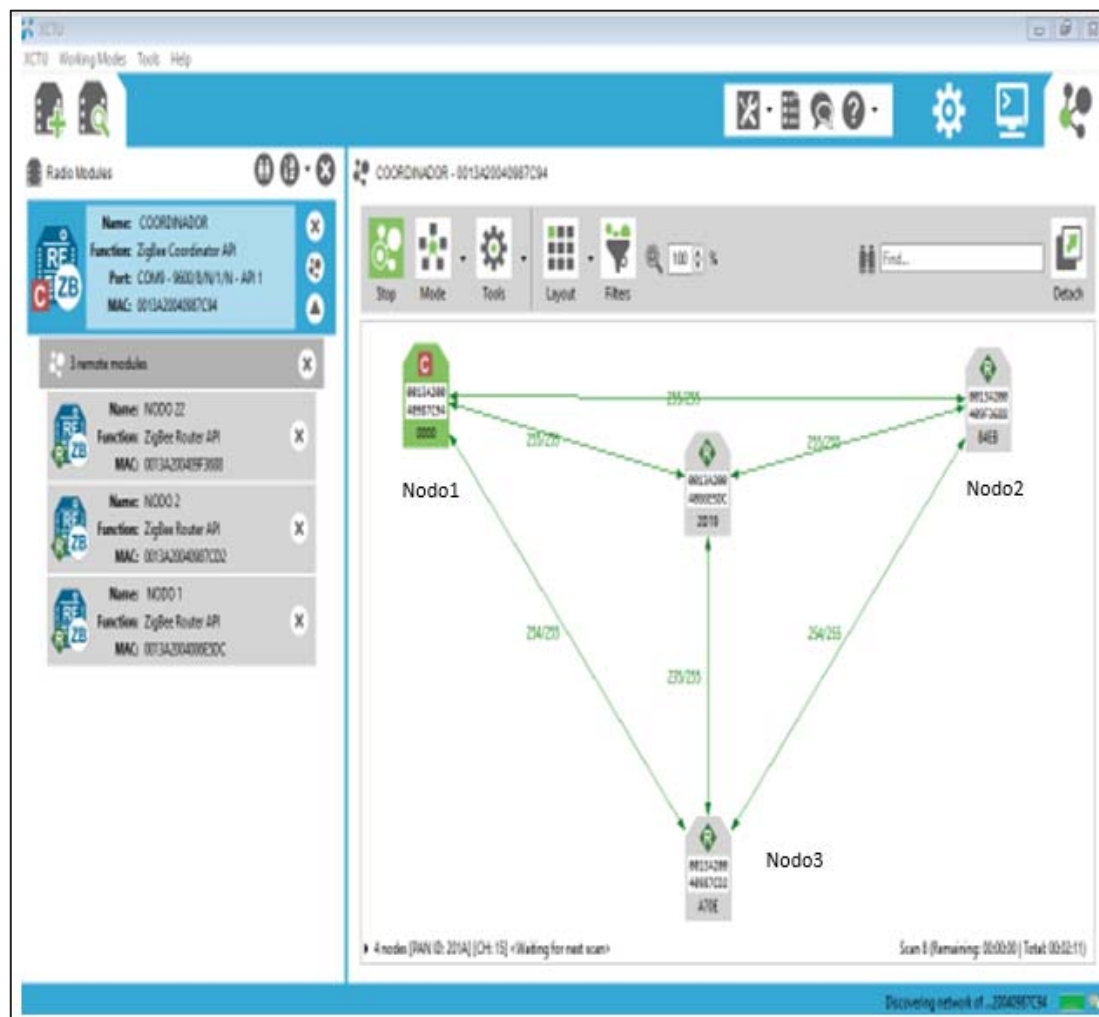


Figura 55. Esquema de la red MESH.

Para la configuración el dispositivo cuenta con una única dirección de red y dirección MAC. En la tabla 20, se encuentra los parámetros utilizados para la configuración de los nodos en el software XCTU.

Tabla 20.

Configuración de los equipos del proyecto.

Rol del dispositivo	MAC	Dirección de Red	Posición
Coordinador	0013A20040987C94	0000	Estación base
Router 1	0013A2004086E5DC	201A	Nodo desplegable - prototipo

Router 2	0013A20040987CD2	201A	Nodo desplegable
Router 22	00134200409F3688	201A	Nodo desplegable

La estación base se encuentra conectado al computador y los nodos son desplegables y se los puede cambiar de lugar según la configuración de la topología.

2.6 Estructura de la programación en Arduino del nodo Principal

Para realizar la programación del nodo principal, primero se debe analizar las variables a utilizar. En este caso se debe transmitir los siguientes datos: GPS, señal análoga medida por el sensor, RSSI y los datos para el control del sonómetro.

La placa Arduino Mini Pro contiene 8 canales análogos digitales, estos trabajan con un convertidor de 10bits. Esto significa que puede medir tensiones de entrada entre 0 y 5 voltios en valores entre 0 y 1023. La resolución se muestra en la siguiente formula: Santiago, S (2016)

$$\text{resolucion} = \frac{5V}{1024} \quad \text{(Ecuación 6)}$$

$$\text{resolucion} = 4,9\text{mV}$$

Llegando a obtener el valor de 4,9mV por unidad.

El datashet del sonómetro indica que 10mV/dB en la salida DC con una resolución de 0,1dB, y el rango de medición es desde 30dB hasta 130dB, por lo tanto, para una lectura en voltios se tomara la siguiente medida. Santiago, S (2016)

$$\text{voltios} = \left(\frac{5}{1024}\right) \times \text{Lectura} \quad \text{(Ecuación 7)}$$

Como cada voltio equivale a un dB se tiene:

$$\left(\frac{1\text{V}}{10\text{mV}}\right) = 100 \quad \text{(Ecuación 8)}$$

Si se multiplica el valor en voltios por 100 se obtendrá los dB del sonómetro:

$$\text{on} = \left(\frac{5 \times \text{Lectura} \times 100}{1023}\right) \quad \text{(Ecuación 9)}$$

A continuación, se muestra la figura 56, que contiene un diagrama de flujo de la programación en Arduino.

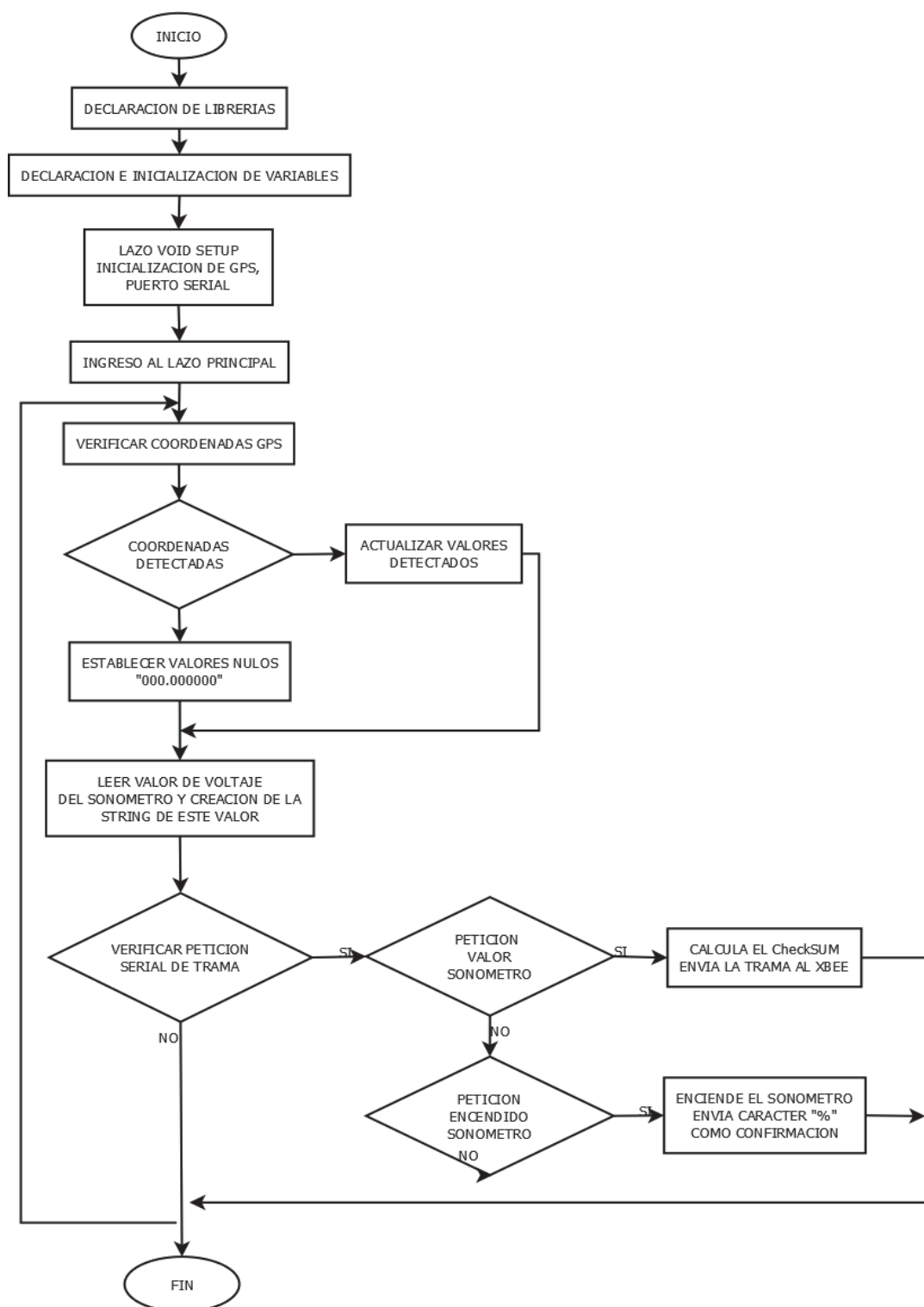


Figura 56. Diagrama de flujo de la programación en Arduino.

El diagrama se divide en varias etapas que se redactan a continuación:

- a) Primera etapa

En la primera parte se inicia con la declaración de las librerías Serial.h, TinyGPS.h y se declara las variables a utilizar en el microcontrolador. Luego se declara un lazo void setup, y se inicializa la parte del GPS y el puerto serial.

b) Segunda etapa

En la etapa dos, se ingresa a un lazo denominado Lazo principal que permitirá realizar varias tareas antes de salir del método. Lo primero que realiza es la verificación de las coordenadas del GPS, para determinar si existen datos a ser llenados, ya que cuando está iniciando o no existe línea de vista la información es nula por lo tanto se llenara de ceros ("000.000000"), hasta obtener datos reales. La latitud se mide entre 0 y 90° la longitud se mide entre 0 y 180° . }

c) Tercera etapa

La tercera etapa está basada en la lectura del valor de voltaje del sonómetro, el mismo que va desde 0 a 1.3V, y la creación del string de este valor. En la figura 57 y tabla 21, muestra el string realizado en Arduino, que está compuesto de los siguientes valores: valor de la latitud, longitud del GPS y del valor análogo del sonómetro. Además, se agrega signos en ASCII que se utilizan en C# para la identificación del string.

```
StrFailed = String(failed);
StrGpsAnlg="";
StrGpsAnlg = "%x" + StrLAT_Sd + "x" + StrLON_Sd + "x" +StrFailed + "x" + "x" +StrVan1 + "x#";
```

Figura 57. String de datos del GPS y sonómetro.

Tabla 21.

Significado de los signos utilizados en la trama.

Signo	Nombre	Función en el string
&	Ampersand	Indica en inicio
#	Numeral	Indica el final
X	Equis	Para separar los datos
StrLaT_Sd	Señal análoga	Latitud del GPS

StrLON_Sd	Señal análoga	Longitud del GPS
StrVan1	Señal análoga	Nivel de ruido del Sonómetro

d) Cuarta etapa

En la cuarta etapa se verifica si existe una petición de la trama en el puerto serial. Si se determina que existe petición de la trama, se realiza la petición del valor del sonómetro y se calcula el **Checksum** y se envía la trama. Si no se tiene el valor del sonómetro, se envía un valor UNO lógico al ping 10 del arduino para encender el sonómetro. Además, se envía un signo de “%” a C# que indica que esta encendido el equipo.

Si no existe una petición de la trama en la comunicación serial, se termina el programa, hasta que nuevamente se realice alguna solicitud. A continuación, se muestra las figuras 58 y 59, de la transmisión de datos crudos sin procesar.

```

ValAnlg: 122 ValPorc: 11
ValAnlg11: 24
ValAnlg: 118 ValPorc: 11
ValAnlg11: 4
ValAnlg: 143 ValPorc: 13
ValAnlg11: 45
ValAnlg: 144 ValPorc: 14
ValAnlg11: 38
ValAnlg: 130 ValPorc: 12
ValAnlg11: 11
ValAnlg: 136 ValPorc: 13
ValAnlg11: 27
ValAnlg: 127 ValPorc: 12
ValAnlg11: 11
ValAnlg: 124 ValPorc: 12
ValAnlg11: 12
ValAnlg: 128 ValPorc: 12
ValAnlg11: 23
ValAnlg: 148 ValPorc: 14
ValAnlg11: 62
ValAnlg: 119 ValPorc: 11
ValAnlg11: 15
ValAnlg: 124 ValPorc: 12
ValAnlg11: 24
ValAnlg: 119 ValPorc: 11
ValAnlg11: 14
ValAnlg: 148 ValPorc: 14
ValAnlg11: 66
ValAnlg: 114 ValPorc: 11
ValAnlg11: 7

```

Figura 58. Captura de datos sin sonómetro y GPS.


```

COM7 (Arduino/Genuino Uno)
LAT=-0.147852 LON=-78.450477 SAT=5 PREC=358 CHARS=6603 SENTENCES=16 CSUM ERR=0
LAT=-0.147853 LON=-78.450477 SAT=5 PREC=358 CHARS=6990 SENTENCES=18 CSUM ERR=0
LAT=-0.147853 LON=-78.450477 SAT=5 PREC=358 CHARS=7374 SENTENCES=20 CSUM ERR=0
LAT=-0.147852 LON=-78.450477 SAT=5 PREC=358 CHARS=7758 SENTENCES=22 CSUM ERR=0
LAT=-0.147853 LON=-78.450477 SAT=5 PREC=358 CHARS=8142 SENTENCES=24 CSUM ERR=0
LAT=-0.147854 LON=-78.450477 SAT=5 PREC=358 CHARS=8526 SENTENCES=26 CSUM ERR=0
LAT=-0.147855 LON=-78.450477 SAT=5 PREC=358 CHARS=8910 SENTENCES=28 CSUM ERR=0
LAT=-0.147856 LON=-78.450477 SAT=5 PREC=358 CHARS=9294 SENTENCES=30 CSUM ERR=0
LAT=-0.147856 LON=-78.450477 SAT=5 PREC=358 CHARS=9678 SENTENCES=32 CSUM ERR=0
LAT=-0.147857 LON=-78.450477 SAT=5 PREC=358 CHARS=10062 SENTENCES=34 CSUM ERR=0
LAT=-0.147858 LON=-78.450469 SAT=5 PREC=358 CHARS=10446 SENTENCES=36 CSUM ERR=0
LAT=-0.147859 LON=-78.450469 SAT=5 PREC=358 CHARS=10830 SENTENCES=38 CSUM ERR=0
LAT=-0.147859 LON=-78.450469 SAT=5 PREC=358 CHARS=11214 SENTENCES=40 CSUM ERR=0
LAT=-0.147858 LON=-78.450477 SAT=5 PREC=358 CHARS=11598 SENTENCES=42 CSUM ERR=0
LAT=-0.147858 LON=-78.450477 SAT=5 PREC=358 CHARS=11982 SENTENCES=44 CSUM ERR=0
LAT=-0.147857 LON=-78.450477 SAT=5 PREC=358 CHARS=12366 SENTENCES=46 CSUM ERR=0
LAT=-0.147857 LON=-78.450477 SAT=5 PREC=358 CHARS=12750 SENTENCES=48 CSUM ERR=0
LAT=-0.147856 LON=-78.450477 SAT=5 PREC=358 CHARS=13134 SENTENCES=50 CSUM ERR=0
LAT=-0.147855 LON=-78.450477 SAT=5 PREC=358 CHARS=13518 SENTENCES=52 CSUM ERR=0
LAT=-0.147855 LON=-78.450477 SAT=5 PREC=358 CHARS=13902 SENTENCES=54 CSUM ERR=0
LAT=-0.147855 LON=-78.450477 SAT=5 PREC=358 CHARS=14286 SENTENCES=56 CSUM ERR=0
LAT=-0.147855 LON=-78.450477 SAT=5 PREC=358 CHARS=14670 SENTENCES=58 CSUM ERR=0
LAT=-0.147856 LON=-78.450477 SAT=5 PREC=358 CHARS=15054 SENTENCES=60 CSUM ERR=0
LAT=-0.147856 LON=-78.450477 SAT=5 PREC=358 CHARS=15438 SENTENCES=62 CSUM ERR=0
LAT=-0.147855 LON=-78.450477 SAT=5 PREC=358 CHARS=15822 SENTENCES=64 CSUM ERR=0
LAT=-0.147855 LON=-78.450477 SAT=5 PREC=358 CHARS=16206 SENTENCES=66 CSUM ERR=0
Autoscroll Sin ajuste de línea 9600 baudio

```

Figura 59. Captura de datos del sonometro y GPS.

2.7 Protocolo para el diseño de la trama API

Para la transmisión de datos en el proyecto se utilizó la implementación de la trama API 1 y no se utiliza el modo AT. La principal ventaja que tiene API es poder encapsular la información en una trama y poder transmitirla a diferentes nodos, mientras que en modo AT, la transmisión es similar a una transmisión serial y no permite enviar la información a varios nodos a la vez. A continuación, se muestra los componentes de una trama.

- **0x7E**: Inicio de trama.
- **0x00 – 0x11**: Longitud de la trama.
- **0x10**: Indica Tx y Rx en la trama.
- **0x01**: ID de la trama.
- **0x00, 0x13, 0xA2, 0x00, 0x40, 0x98, 0x7C, 0xD2**: Mac del nodo remoto.
- **0xFF, 0xFE**: Dirección de 16 bits.
- **0x00, 0x00**: Option default.
- **0x25, 0x25, 0x25**: Información a procesar en el nodo remoto, que indica que debe realizar alguna función.

- **0xA7**: Checksum.

En el programa XCTU se diseña la trama para lo cual se realiza los siguientes pasos, en la figura 60, se muestra la consola de pruebas para el envío y recepción de información.

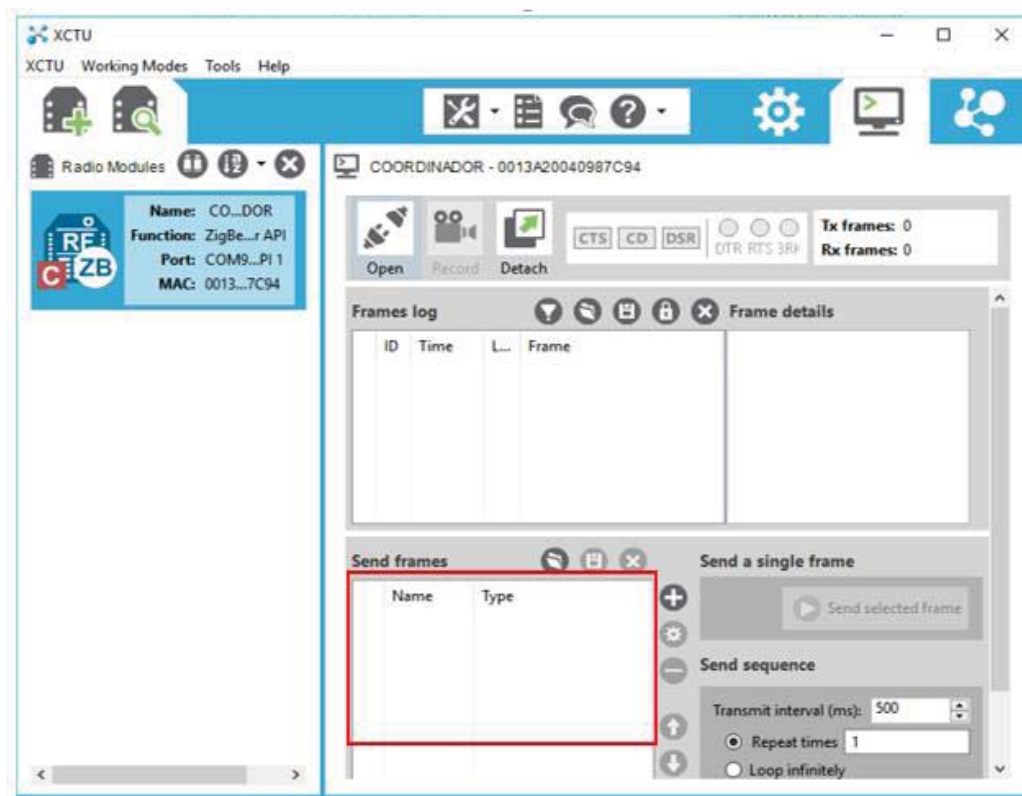


Figura 60. Interfaz XCTU, para generar la trama.

Descripción de las diferentes partes del programa XCTU:

1. En el programa XCTU en la barra de herramientas dar clic en la consola (Switch consoles).
2. En el área de trabajo de **Send Frames** dar clic en la pestaña con el signo “+” que significa añadir nuevo frame.
3. En la pantalla que se despliega agregar el nombre de la trama y dar clic en **Create frame using Frames Generator Tool**.
4. Se despliega la pantalla de configuración, la misma que se puede ver en la figura 61.

XBee API Frames Generator
This tool allows you to generate any kind of API frame and copy its value. Just fill in the required fields.

Protocol: ZigBee Mode: API 1 - API Mode Without Escapes
Frame type: 0x10 - Transmit Request

Frame parameters:

Start delimiter	7E
Length	00 0E
Frame type	10
Frame ID	01
64-bit dest. address	00 00 00 00 00 00 00 00
16-bit dest. address	FF FE
Broadcast radius	00
Options	00
RF data	ASCII HEX

Generated frame:
7E 00 0E 10 01 00 00 00 00 00 00 00 00 FF FE 00 00 F1

Byte count: 18

Copy frame Close OK

Figura 61. Interfaz XCTU, generación de la trama.

5. En la cual muestra los siguientes pasos de configuración:

- **Protocol:** El tipo de protocolo a usar, ya sea Digi Mesh, WIFI entre otros. Para el proyecto escoger ZigBee.
- **En Frame Type:** En este campo escogeremos el tipo de trama: AT y comandos. Para lo cual escogemos **0x10**, para la transmisión y recepción de paquetes.
- **Start delimiter:** Determina el inicio de la trama.
- **Length:** Indica la longitud del mensaje, se calculara automáticamente una vez ingresado los datos a enviar.
- **Frame Type:** Indica el tipo de la trama. Para ver las diferentes opciones Frame Type en la parte superior de la misma.
- **Frame ID:** Proporciona el ID de la trama, utilizado por los dispositivos para identificar el envío y recepción de la trama.
- **64-bit dest, address:** Ingresar la dirección MAC del nodo remoto.

- **16-bit dest, address:** Dirección de 16 bits. El coordinador siempre asigna la dirección **0xFFFFE** de 64-bits para transmitir, y utiliza direcciones de 16-bits para la recepción.
- **Broadcast radius:** Determina el número de saltos que puede ocurrir durante una transmisión. Si se deja configurado con “0” indica que va realizar el número máximo de saltos. El fabricante recomienda no exceder de los 20 saltos.
- **Options:** Campo que permite agregar opciones de soporte para la transmisión de datos. Ejemplo: 00 ninguno, 01 Disable AC entre otros. En el proyecto se deja en “00” ya que no se agrega ninguna opción.
- **RF Data:** En este campo agregar la información a enviar. Se puede generar en ASCII y HEXADECIMAL.
- **Generated Frame:** Muestra la trama creada.

2.8 Diseño de las tramas API

En el proyecto se utilizó cuatro tipos de tramas para diferentes procesos, a continuación, se muestra las tramas creadas:

2.8.1 Trama para encender remotamente el sonómetro

Tabla 22.

Datos para controlar el encendido del sonómetro.

Dato	Nombre	Función	ASCII	Hexadecimal
%	Porcentaje	Encender remotamente el sonómetro	%	25

En la figura 62, se muestra el diseño de la trama. Entre los datos más importantes a ingresar, están la dirección MAC del nodo remoto y el dato a enviar. En el programa Arduino una vez recibido la trama, mediante una instrucción cambia de estado el ping 10, a un valor alto y activa un transistor

que enciende el sonómetro. En la tabla 22, se muestra los datos utilizados para esa función.

XBee API Frames Generator
This tool allows you to generate any kind of API frame and copy its value. Just fill in the required fields.

Protocol: All Mode: API 1 - API Mode Without Escapes

Frame type: 0x10 - Transmit Request

Frame parameters:

Start delimiter	7E
Length	00 11
Frame type	10
Frame ID	01
64-bit dest. address	00 13 A2 00 40 98 7C D2
16-bit dest. address	FF FE
Broadcast radius	00
Options	00
RF data	ASCII HEX %%%

Generated frame:

```
7E 00 11 10 01 00 13 A2 00 40 98 7C D2 FF FE 00 00 25 25
25 A7
```

Byte count: 21

Copy frame Close OK

Figura 62. Trama de encendido remoto del sonómetro.

Una vez creado la trama en XCTU, se la utiliza en el programa C#. A continuación, se muestra la línea de programación. El dato se lo ingresa en modo ASCII. Figura 63, línea de programación en C#.

```
if (SelecSonom.Text == "S2_ON")
{
    this.BtnIniciar.Enabled = true;

    byte[] Nodo2_on = {
        0x7E ,0x00 ,0x11 ,0x10 ,0x01 ,0x00 ,0x13 ,0xA2 ,0x00 ,0x40 ,
        0x98 ,0x7C ,0xD2 ,0xFF ,0xFE ,0x00 ,0x00 ,0x25 ,0x25 ,0x25 ,0xA7};

    for (int i = 0; i < Nodo2_on.Length; i++)
    {
        write[i] = Nodo2_on[i];
    }
}
```

Figura 63. Trama de encendido remoto en C#.

El significado de las partes que conforman la trama API para el encendido del sonómetro es la siguiente.

- **x7E**: Inicio de trama.
- **0x00 – 0x11**: Longitud de la trama.
- **0x10**: Indica Tx y Rx en la trama.
- **0x01**: ID de la trama.
- **0x00, 0x13, 0xA2, 0x00, 0x40, 0x98, 0x7C, 0xD2**: Mac del nodo remoto.
- **0xFF, 0xFE**: Dirección de 16 bits.
- **0x00, 0x00**: Option default.
- **0x25, 0x25, 0x25**: Información a procesar en el nodo remoto, que indica que se encienda el sonómetro.
- **0xA7**: Checksum.

2.8.2 Trama para apagar remotamente el sonómetro

Tabla 23.

Datos para controlar el apagado del sonómetro.

Dato	Nombre	Función	ASCII	Hexadecimal
@	Arroba	Apagar remotamente el sonómetro	@	25

A continuación, en la figura 64 y tabla 23, se muestra la trama que permite apagar remotamente el sonómetro. En la cual se utiliza el signo “@”, en Arduino se encuentra programado que una vez que recibe la trama, mediante una instrucción cambie de estado el ping 10, a un nivel bajo. Inmediatamente se corta voltaje que circula por el transistor y apaga el sonómetro.

Protocol: All Mode: API 1 - API Mode Without Escapes

Frame type: 0x10 - Transmit Request

Frame parameters:

Start delimiter	7E
Length	00 11
Frame type	10
Frame ID	01
64-bit dest. address	00 13 A2 00 40 98 7C D2
16-bit dest. address	FF FE
Broadcast radius	00
Options	00
RF data	ASCII HEX @@@

Generated frame:

```
7E 00 11 10 01 00 13 A2 00 40 98 7C D2 FF FE 00 00 40 40 40 56
```

Byte count: 21

Copy frame Close OK

Figura 64. Trama para el apagado remoto en XCTU.

En la figura 65, se muestra la trama API programada en el software C#.

```

if (SelecSonom.Text == "S2_OFF")
{
    byte[] Nodo2_off = {
        0x7E, 0x00, 0x11, 0x10, 0x01, 0x00, 0x13, 0xA2, 0x00, 0x40, 0x98,
        0x7C, 0xD2, 0xFF, 0xFE, 0x00, 0x00, 0x40, 0x40, 0x40, 0x56};

    for (int i = 0; i < Nodo2_off.Length; i++)
    {
        write[i] = Nodo2_off[i];
    }
}

```

Figura 65. Trama de apagado remoto en C#.

El significado de las partes que conforman la trama API para el apagador del sonómetro es la siguiente.

- **0x7E**: Inicio de trama.
- **0x00 – 0x11**: Longitud de la trama.
- **0x10**: Indica Tx y Rx en la trama.
- **0x01**: ID de la trama.
- **0x00, 0x13, 0xA2, 0x00, 0x40, 0x98, 0x7C, 0xD2**: Mac del nodo remoto.
- **0xFF, 0xFE**: Dirección de 16 bits.
- **0x00, 0x00**: Option default.
- **0x40, 0x40, 0x40**: Información a procesar en el nodo remoto, que indica que se apague el sonómetro.
- **0x56**: Checksum.

2.8.3 Trama para enviar los datos del sonómetro y del GPS a la interfaz visual desde Arduino

Para el diseño de la trama en Arduino que permita enviar los datos del sonómetro y del GPS al coordinador, se tomó de referencia el proceso de diseño del programa XCTU. En la figura 66, se muestra la declaración de variables para crear la trama. La estructura es la misma que presenta XCTU.

```
//GENERACIÓN DE TRAMA
const byte Dltr_init = 0x7E;//DELIMITADOR DE INICIO
const byte Long_Msb = 0x00; //Largo de la trama MSB
const byte Long_Lsb = 0x34; //Largo de la trama LSB
const byte Frame_Ty = 0x10; //Largo de la trama MSB
const byte Frame_ID = 0x01; //Largo de la trama LSB
//00 13 A2 00 40 98 7C 94
const byte Address64[8] = {0x00, 0x13, 0xA2, 0x00, 0x40, 0x98, 0x7C,0x94};
const byte Address16[2] = {0xFF, 0xFE};
const byte BroadCast = 0x00; //Largo de la trama LSB
const byte Option = 0x00; //Largo de la trama LSB
//TRAMA TAMAÑO ESTA CONFORMADA POR => "%-00.342179-078.487503001661023#" INFORMACION DEL GPS
//MAS DATOS ANALOGOS.
byte ByteChecksum=0;
```

Figura 66. Generación de trama en Arduino.

En el software Arduino se programa, que los datos del GPS y del sonómetro se encuentren en forma de string. Una vez que recibe una petición por parte de la interfaz de virtualización, se empieza él envío de la trama.

A continuación, en la figura 67, se muestra la transmisión de los primeros datos de la trama API, a través del puerto serial hacia el coordinador.

```

void TrmApiTransmit (void)
{
  // XBee remoto: API
  Serial.write((byte)Dltr_init); // byte de inicio
  Serial.write((byte)Long_Msb); // Largo MSB (siempre 0)
  Serial.write((byte)Long_Lsb); // byte de inicio
  Serial.write((byte)Frame_Ty); // es el tipo de mensaje para enviar comandos AT
  Serial.write((byte)Frame_ID); // Frame ID (no solicitamos repuesta)
  Serial.write((byte)Address64[0]); // Envía los 64 bit de la dirección de destino
  Serial.write((byte)Address64[1]);
  Serial.write((byte)Address64[2]);
  Serial.write((byte)Address64[3]);
  Serial.write((byte)Address64[4]);
  Serial.write((byte)Address64[5]);
  Serial.write((byte)Address64[6]);
  Serial.write((byte)Address64[7]);
  Serial.write((byte)Address16[0]);
  Serial.write((byte)Address16[1]);
  Serial.write((byte)BroadCast);
  Serial.write((byte)Option);
}

```

Dirección de
64bits del
coordinador

Figura 67. Transmisión de los primeros datos de la Trama API desde el Arduino a C#

Luego se envía la información del sonómetro y GPS en forma de string, como se muestra en la figura 68.

```

//Convierto "StrGpsAnlg" a una Array para poder trabajar con la misma y determinar el Check SUM
StrGpsAnlg.toCharArray(Array_Trm_dat, Trm_dat+1);

//ENVIA LA INFORMACIÓN GPS Y ANÁLOGA
for (byte i = 0; i < Trm_dat ; i ++ )
{
  Serial.write((byte)Array_Trm_dat[i]);
}
Serial.write((byte)ByteChecksum); // Checksum
}

```

Figura 68. Envío de la información del GPS y del sonómetro.

Y por último se envía valor del Checksum, una vez realizado el cálculo. El programa lo calcula automáticamente, como se muestra en la figura 69.

```
//ENVIA LA INFORMACIÓN GPS Y ANÁLOGA
for (byte i = 0; i < Trm_dat ; i ++ )
{
  Serial.write((byte)Array_Trm_dat[i]);
}
Serial.write((byte)ByteChecksum); // Checksum
}
void ValorCheckSUM (void)
{
  int ChkSUM = 0;
  for (byte i = 0; i < Trm_dat ; i ++ )
  {
    ChkSUM += Array_Trm_dat[i];
    //Serial.write(Array_Trm_dat[i]);
  }
  ChkSUM = ChkSUM + Frame_Ty + Frame_ID +
  Address64[0] + Address64[1]+ Address64[2] + Address64[3] + Address64[4]+ Address64[5] + Address64[6] + Address64[7] +
  Address16[0] + Address16[1] + BroadCast + Option;
  byte Min_Lsb = (byte)ChkSUM;
  ByteChecksum = 0xFF - Min_Lsb;
}
```

Figura 69. Cálculo del Checksum en Arduino.

2.8.4 Trama para obtener los datos del nivel de RSSI

En nivel del RSSI nos permite obtener un valor de potencia recibida, para calcular la distancia aproximada del nodo remoto hacia el coordinador. En la tabla 23, se indica el dato que se utiliza en la trama para la solicitud del nivel de RSSI.

Tabla 23.

Datos para solicitar el nivel de RSSI del dispositivo XBee.

Dato	Nombre	Función	ASCII	Hexadecimal
DB	AT Command Name (Two ASCII characters)	Obtener los datos del RSSI	DB	44,42

En la figura 70, se muestra el diseño de la trama API en XCTU. El principal cambio está en la parte de **Frame Type**: en la misma se escoge la opción de **Remote AT Command**. El dato en **Remote cmd. options** es el número dos que hace referencia a una solicitud de respuesta por parte del XBee. Los datos

DB es un estándar para realizar ese tipo de solicitud, se lo confirma en la página de DIGI.

Protocol: ZigBee Mode: API 1 - API Mode Without Escapes

Frame type: 0x17 - Remote AT Command

Frame parameters:

Start delimiter	7E
Length	00 0F
Frame type	17
Frame ID	01
64-bit dest. address	00 13 A2 00 40 86 E5 DC
16-bit dest. address	FF FE
Remote cmd. options	02
AT command	ASCII HEX DB
Parameter value	ASCII HEX
Checksum	26

Generated frame:

7E 00 0F 17 01 00 13 A2 00 40 86 E5 DC FF FE 02 44 42 26

Byte count: 19

Figura 70. Trama para verificar el nivel de RSSI, hacia el nodo Router 1.

A continuación, en la figura 71 se muestra la trama API de RSSI de los nodos programados en C#, la única diferencia son las diferentes direcciones MAC de cada dispositivo.

```

//NODO 1 DB
byte[] Nodo1 = {0x7E ,0x00 ,0x0F ,0x17 ,0x01 ,0x00 ,0x13 ,0xA2 ,0x00 ,0x40 ,0x86
,0xE5 ,0xDC ,0xFF ,0xFE ,0x02 ,0x44 ,0x42 ,0x26};

this.SpPuerto.DiscardOutBuffer();
SpPuerto.Write(Nodo1, 0, Nodo1.Length);
}

if (Cont_RSSI == 2)
{
//NODO 2
byte[] Nodo2 = {0x7E ,0x00 ,0x0F ,0x17 ,0x01 ,0x00 ,0x13 ,0xA2 ,0x00 ,0x40 ,0x98
,0x7C ,0xD2 ,0xFF ,0xFE ,0x02 ,0x44 ,0x42 ,0x87};

this.SpPuerto.DiscardOutBuffer();
SpPuerto.Write(Nodo2, 0, Nodo2.Length);
}

if (Cont_RSSI == 3)
{
//NODO 3
byte[] Nodo3 = {0x7E ,0x00 ,0x0F ,0x17 ,0x01 ,0x00 ,0x13 ,0xA2 ,0x00 ,0x40 ,0x9F
,0x36 ,0x88 ,0xFF ,0xFE ,0x02 ,0x44 ,0x42 ,0x10};

this.SpPuerto.DiscardOutBuffer();
SpPuerto.Write(Nodo3, 0, Nodo3.Length);
}

```

Figura 71. Trama de los tres nodos en C#.

El significado de las partes que conforman la trama API para obtener el nivel de RSSI es la siguiente.

- **0x7E:** Inicio de trama.
- **0x00 – 0x11:** Longitud de la trama.
- **0x17:** Indica que se está utilizando comando remoto AT.
- **0x01:** ID de la trama.
- **0x00, 0x13, 0xA2, 0x00, 0x40, 0x98, 0x7C, 0xD2:** Mac del XBee remoto.
- **0xFF, 0xFE:** Dirección de 16 bits.
- **0x02:** Remote cmd. Options. Utiliza el número dos para solicitar respuesta al nodo.
- **0x44, 0x42:** Información a procesar en el XBee remoto, que indica que le devuelva el nivel de RSSI. En ASCII es DB.
- **0x56:** Checksum.

Para obtener el RSSI de los nodos se utilizó el siguiente método:

- Primero se coloca el nodo 1, a una x distancia. Ejemplo a dos metros del Gateway. Y se solicita el nivel de RSSI mediante el software.

- Luego de coloca nuevamente al Nodo 1, a otra x distancia. Ejemplo a 4 metros del Gateway. Y se solicita el nivel de RSSI mediante el software.
- Se realiza una tercera medición con el Nodo 1, a otra x distancia. Ejemplo a 6 metros del Gateway. Y se solicita nuevamente el nivel de RSSI mediante el software.
- Con estos valores se calcula un valor denominado constante que permite calcular el nivel de RSSI de los demás nodos.
- Calcular el valor de la constante.
- Por último, se solicita el valor de RSSI de todos los nodos, los mismos que serán transformados a una distancia aproximada en metros, con respecto al Gateway.

Ecuación (10)

$$\text{Constante} = \left(\frac{\text{Medida 1}}{-\text{dB1}} \right) + \left(\frac{\text{Medida 2}}{-\text{dB2}} \right) + \left(\frac{\text{Medida 3}}{-\text{dB3}} \right) / 3$$

Ecuación (11)

$$\text{Distancia Aproximada en metros} = (\text{Constante} * -\text{dB medido de cada nodo})$$

2.10 Diseño e implementación de la interfaz gráfica

Para diseñar la interfaz gráfica primero se definió los requerimientos que se necesitaba, para el monitoreo de los nodos. A continuación, se muestra las características requeridas:

- Permita visualizar mediante una gráfica el nivel de ruido censado de los nodos.
- Se interconecte de forma serial al nodo coordinador.
- Tenga una pantalla que permita visualizar el posicionamiento del nodo, haciendo uso del GPS.
- Posea iconos que permitan encender y apagar remotamente los sensores de ruido.

- Se conecte a una base de datos SQL SERVER, y guarde la información censada en los nodos.
- Permita hacer mediciones del nivel de RSSI, de los dispositivos XBee.

Luego de analizar los requerimientos y una vez decidido el software a utilizar, el mismo que será VISUAL STUDIO en lenguaje C sharp (C#), se diseña el programa en un diagrama en bloques, el mismo que se muestra en la figura 72.

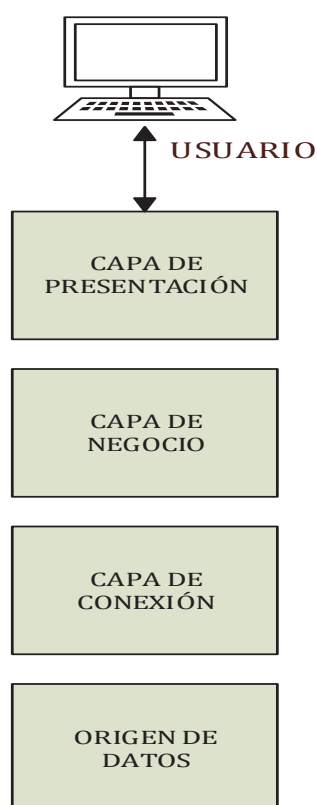


Figura 72. Diagrama en bloques de la interfaz gráfica.

2.10.1 Capa Presentación

La capa de presentación consta de las interfaces visuales y de interactividad entre el software y el usuario, pudiendo realizar todas las características para las que fue creado el software, la misma que se muestra en la figura 73.

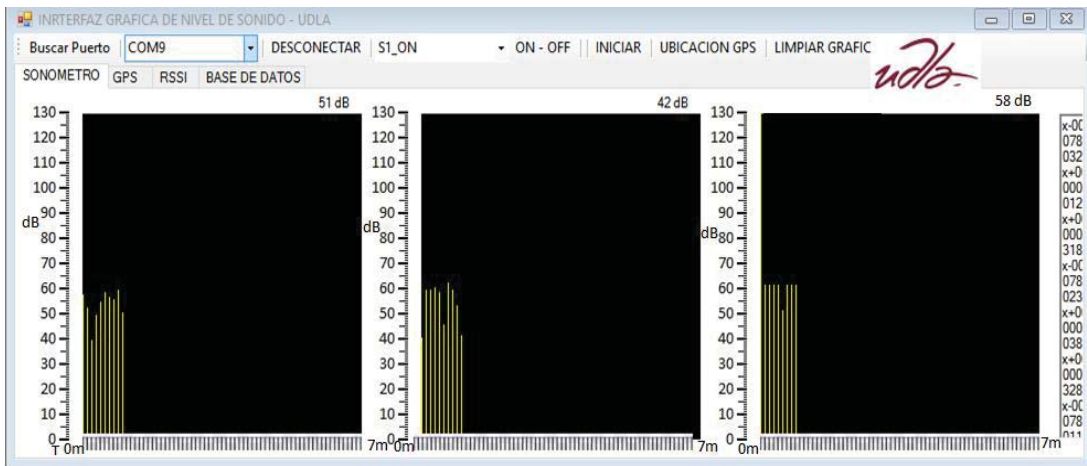


Figura 73. Capa de presentación de la red WSN en modo API.

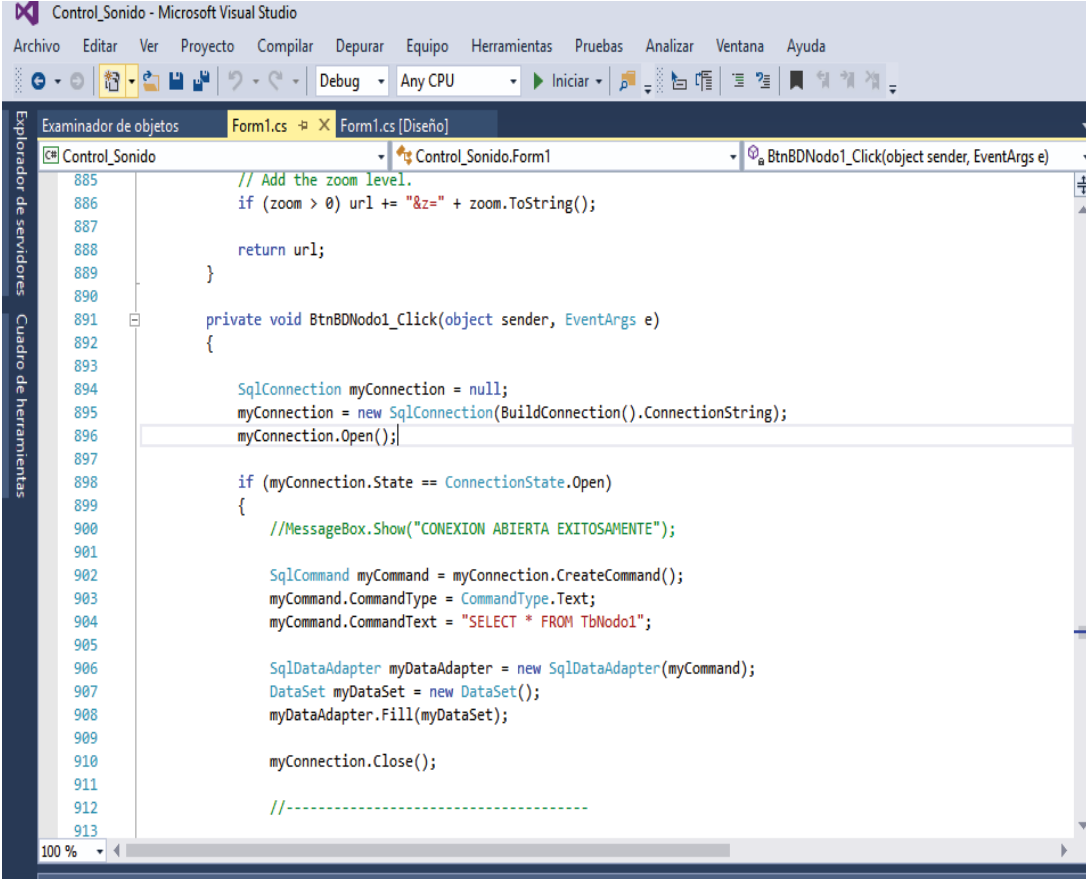
2.10.2 Capa Negocio

Es la capa lógica del diseño del software, ya que permite las transacciones desde la interfaz hacia el ingreso de los datos, además está encargada de evaluar y encapsular los datos ingresados en la Unidad de Interfaz Gráfica (GUI). La misma se puede ver en la figura 74.

Figura 74. Capa negocio del sistema WSN en modo API.

2.10.3 Capa Conexión

La capa de conexión se encarga de adquirir los datos y encapsularlos para la transmisión a la Base de datos. En el presente proyecto se utiliza el motor SQLServer, como se muestra en la figura 75.



```
Control_Sonido - Microsoft Visual Studio
Archivo  Editar  Ver  Proyecto  Compilar  Depurar  Equipo  Herramientas  Pruebas  Analizar  Ventana  Ayuda
Debug  Any CPU  Iniciar
Examinador de objetos  Form1.cs  Form1.cs [Diseño]
Control_Sonido  Control_Sonido.Form1  BtnBDNodo1_Click(object sender, EventArgs e)
885 // Add the zoom level.
886 if (zoom > 0) url += "&z=" + zoom.ToString();
887
888 return url;
889 }
890
891 private void BtnBDNodo1_Click(object sender, EventArgs e)
892 {
893
894     SqlConnection myConnection = null;
895     myConnection = new SqlConnection(BuildConnection().ConnectionString);
896     myConnection.Open();
897
898     if (myConnection.State == ConnectionState.Open)
899     {
900         //MessageBox.Show("CONEXION ABIERTA EXITOSAMENTE");
901
902         SqlCommand myCommand = myConnection.CreateCommand();
903         myCommand.CommandType = CommandType.Text;
904         myCommand.CommandText = "SELECT * FROM TbNodo1";
905
906         SqlDataAdapter myDataAdapter = new SqlDataAdapter(myCommand);
907         DataSet myDataSet = new DataSet();
908         myDataAdapter.Fill(myDataSet);
909
910         myConnection.Close();
911
912         //-----
913
```

Figura 75. Capa de datos del sistema WSN en modo API.

2.10.4 Capa de origen de Datos

En esta capa se refiere a los datos que llegan desde los nodos remotos, básicamente a los datos del GPS, niveles de ruido y RSSI. Los mismos que son procesados, mediante programación y mostrados en la interfaz gráfica.

2.10.5 Diagrama de flujo de la interfaz gráfica

Para el desarrollo de la interfaz gráfica se utilizó el lenguaje C# con el software Visual Studio, el diseño de la interfaz se presenta en el siguiente diagrama de flujo. Se ha realizado dos diagramas para la explicación, en la figura 76, se muestra el primer diagrama de flujo.

Para la programación del software se utilizó las siguientes librerías:

- using System;
- using System.Collections.Generic;
- using System.ComponentModel;
- using System.Data;
- using System.Drawing;
- using System.Linq;
- using System.Text;
- using System.Threading.Tasks;
- using System.Windows.Forms;
- //-----
- using System.Device.Location;
- using System.Web;
- using System.IO.Ports;
- using System.Data.SqlClient;// PARA BASE DE DATOS

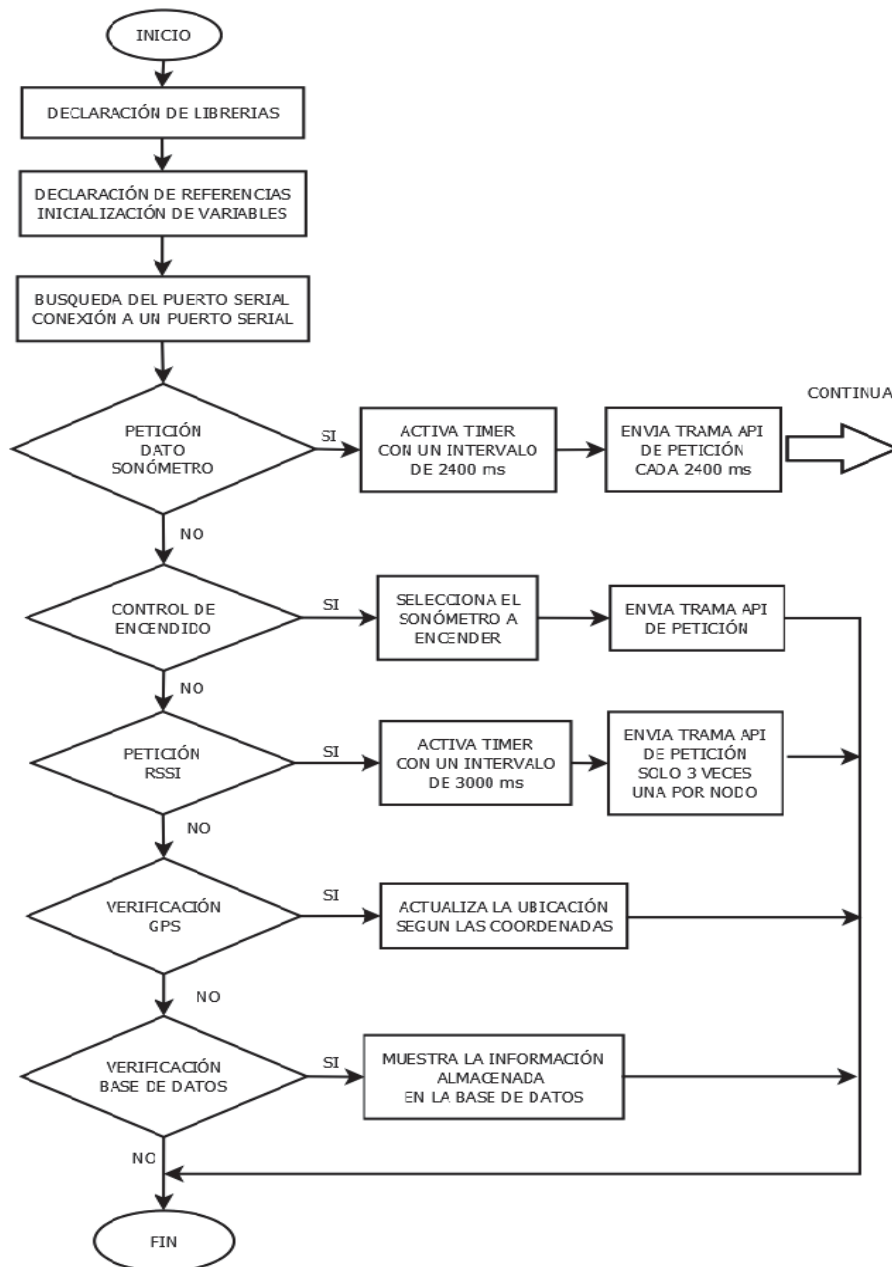


Figura 76. Diagrama de flujo del programa realizado en C# parte uno.

La interfaz de visualización en C# está diseñada en una sola capa, que contempla las siguientes opciones: Conexión con el puerto serial, RSSI, conexión a la base de datos, transmisión y recepción de tramas y la adquisición de datos del GPS y sonómetro.

En el diagrama de flujo número uno, se muestra la estructura del programa. La misma que consta de los siguientes pasos:

- a) El programa inicia con la declaración de librerías, referencias e inicialización de variables.
- b) La siguiente instrucción es la búsqueda del puerto serial para la conexión de la interfaz con el coordinador.
- c) Una vez realizado la conexión el programa se encuentra bloqueado, hasta que se realice una petición de encendido de los nodos remotos e inicie la solicitud de datos del sonómetro. En ese momento automáticamente se activa un timer con un tiempo de intervalo de 2400ms para la petición a cada nodo.
- d) Al momento de solicitar la información del RSSI a los nodos, se activa un timer con una velocidad de 3000ms para el envío y recepción de la información.
- e) La opción de GPS permite realizar una petición de posicionamiento al nodo y actualiza los datos recibidos en el mapa.
- f) Además, la interfaz cuenta con una conexión a la base de datos que permite el almacenamiento de los datos registrados, permitiendo realizar reportes.

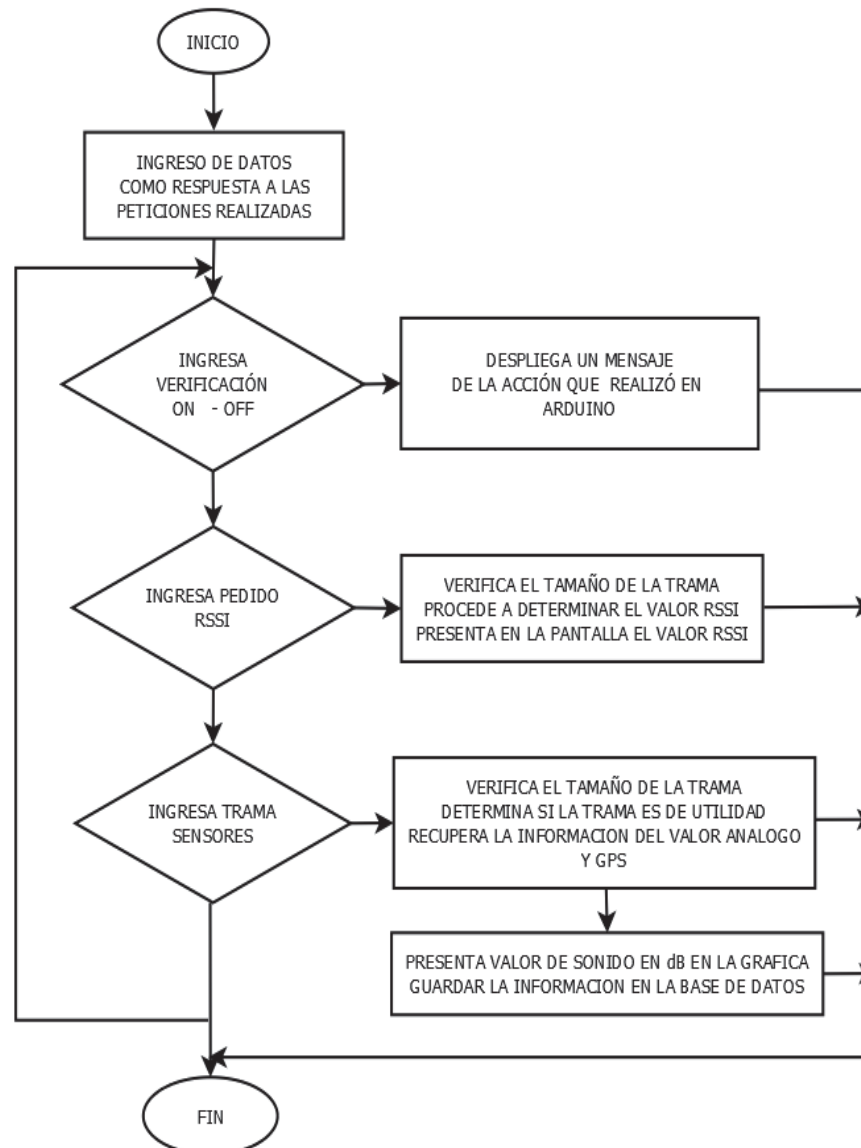


Figura 77. Diagrama de flujo del programa realizado en C# parte dos.

Una vez realizado la petición de datos a los nodos, se realiza los siguientes procesos para la recepción de información: Descripción de la figura 77, en la cual se encuentra el diagrama de flujo número dos.

- La información que ingresa de la solicitud de encendido o apagado del sonómetro, está programada para mostrar un mensaje de la acción realizada.
- Al momento de ingresar la información del RSSI, el programa verifica el tamaño de la trama y procede a determinar el valor del RSSI, para luego mostrar el valor en la interfaz.

- c) En la trama de los sensores y GPS, se recupera la información de utilidad, se descarta los demás datos de la que no se necesita y se gráfica lo valores obtenidos en la pantalla de la interfaz.

Una vez revisado los requerimientos y realizado el programa se tiene el software en funcionamiento, como se ve en la figura 78.

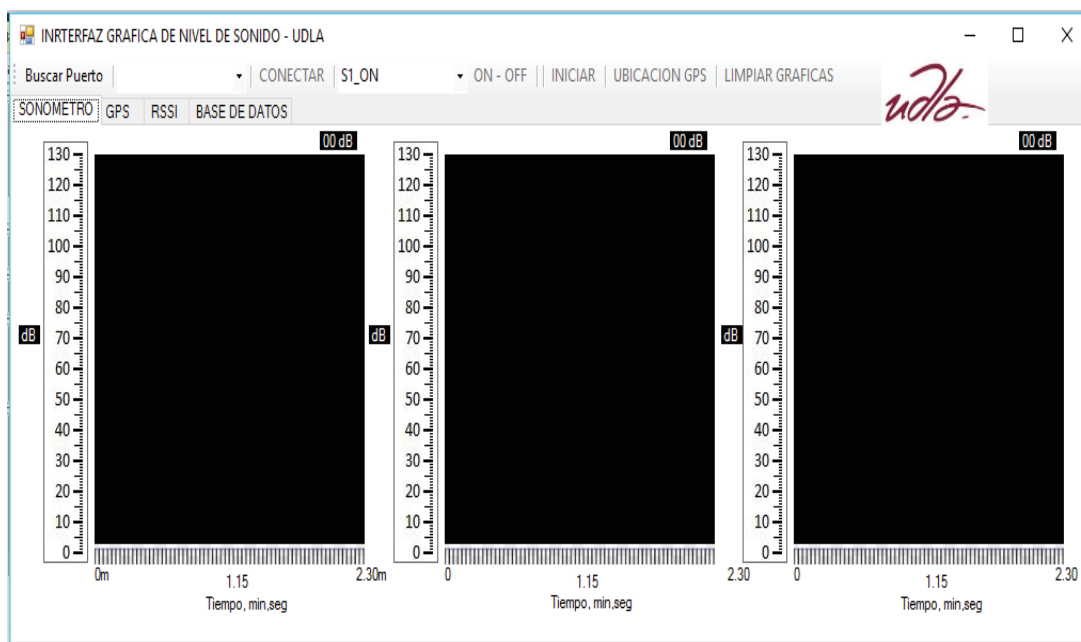


Figura 78. Pantalla principal de la interfaz gráfica.

La pantalla principal de la interfaz tiene varias pestañas, las mismas que se describen a continuación:

- **Buscar puerto:** Permite seleccionar el puerto serial a trabajar, para conectarse con el Gateway o coordinador.
- **Conectar:** Una vez seleccionado el puerto serial, permite conectarse o desconectarse del Gateway.
- **S1_ON-OFF:** Permite seleccionar los nodos a encender o apagar remotamente.
- **Iniciar:** Utilizado para iniciar con el monitoreo del nivel de ruido de los nodos remotos.

- **Ubicación GPS:** Junto con la pestaña GPS, permite ubicar el posicionamiento del nodo principal.
- **Limpiar graficas:** Utilizado para eliminar la gráfica, que se encuentra mostrando el nivel de ruido en ese momento.
- **Sonómetro:** Muestra las tres pantallas de color negro, en donde se indicara la gráfica de ruido.
- **GPS:** Junto con ubicar GPS, permite ubicar el posicionamiento del nodo principal, como se muestra en la figura 79.

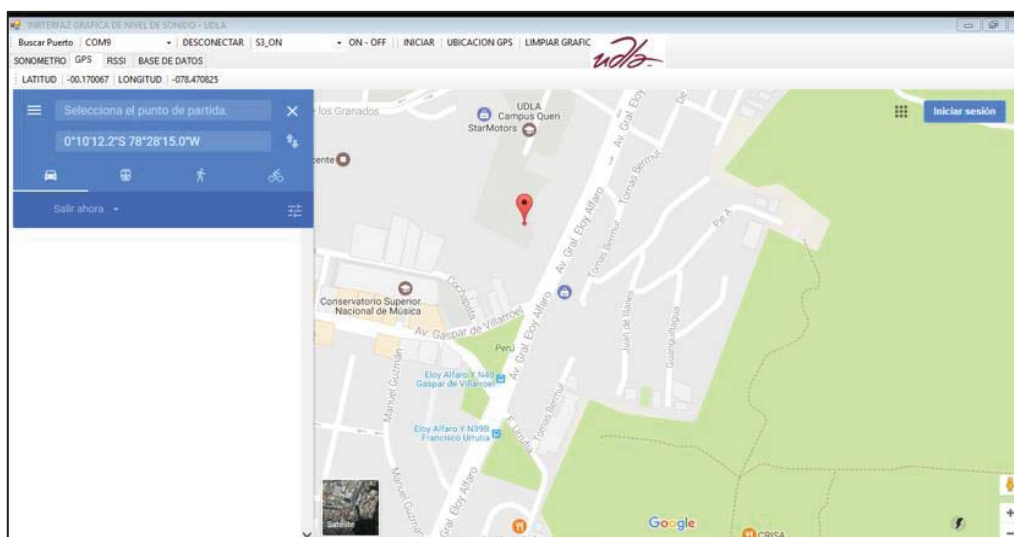


Figura 79. Interfaz del posicionamiento del GPS.

Adaptado de (Google maps, s.f)

- **RSSI:** Utilizado para realizar las mediciones del nivel de RSSI, de los dispositivos XBee. En la figura 80, se puede ver la interfaz.



Figura 80. Interfaz para medir el nivel de RSSI.

- **Base de datos:** Permite guardar la información del nivel de ruido y posicionamiento del GPS, en la base de datos SQL SERVER 2014. En la figura 81 se puede ver la interfaz.

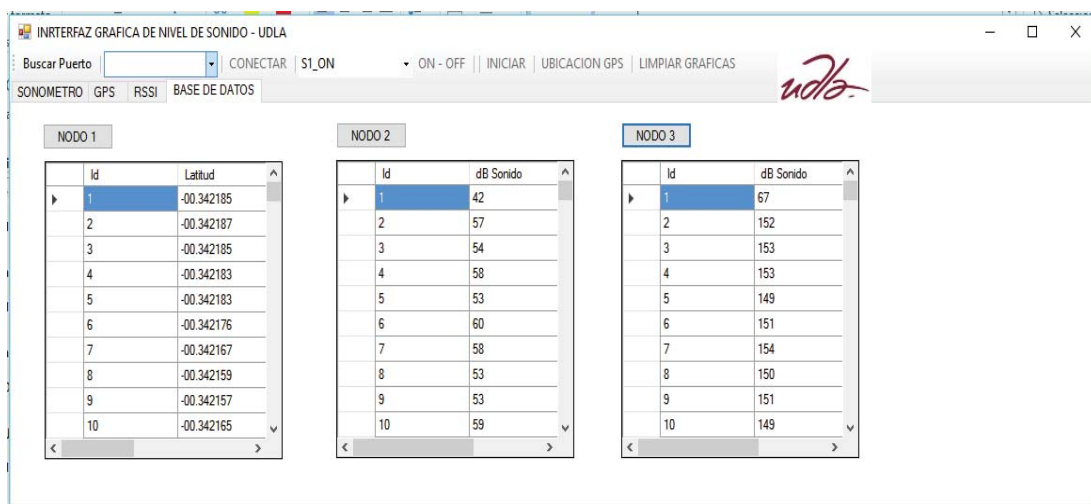


Figura 81. Interfaz de los parámetros que se guardan en la base de datos.

Los datos que guardarse en la base de datos son los siguientes: ID del nodo, Latitud, Longitud, Valor del sonido, la fecha y hora. Como se muestra en la figura 82.

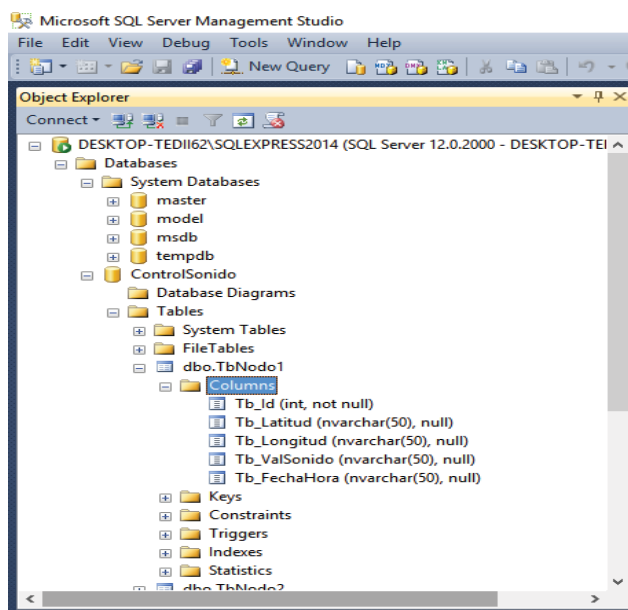


Figura 82. Tabla creada en la base de datos SQL SERVER 2014.

2.11 Presentación del prototipo final

Finalmente se tiene el proyecto funcionando. En la figura 83, se puede ver el sistema funcionando en el laboratorio de la UITEC de la universidad de las Américas. En la misma se observa los tres nodos monitoreados desde el software en la PC. Para lo cual al iniciar el programa se debe buscar el puerto que se encuentre conectado al Gateway. Una vez conectado se debe encender los nodos, luego dar un click en INICIAR, y automáticamente empezara a mostrar los valores de los nodos remotos. Además, se hizo la presentación del prototipo en el día de las Telecomunicaciones UDLA 2017.

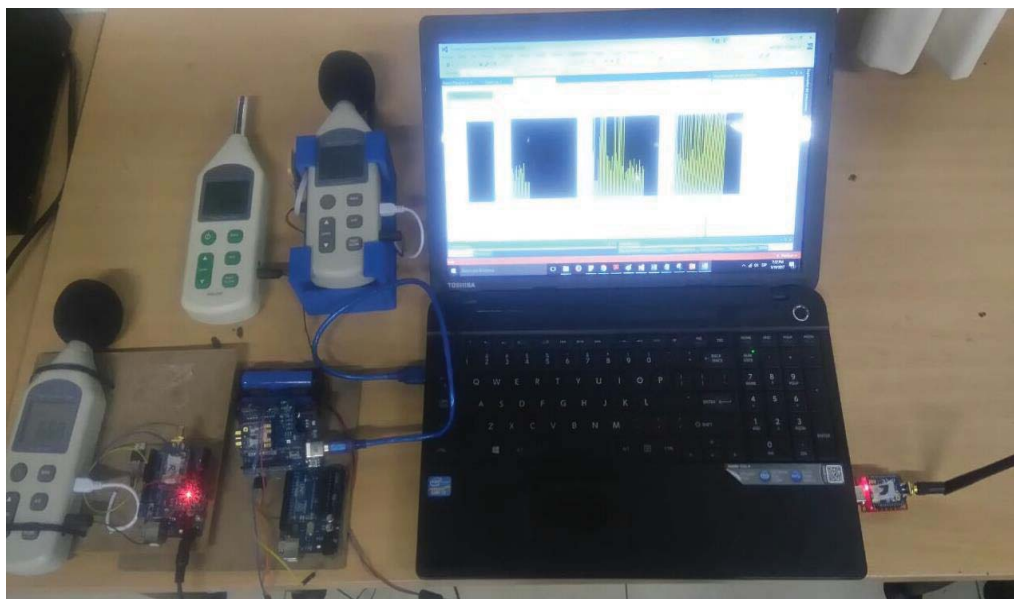


Figura 83. Sistema WSN en modo API funcionando.

En la figura 84, se muestra el funcionamiento de la parte de posicionamiento, mediante el uso del GPS. En la misma se indica la posición desde donde nos encontramos monitoreando los nodos. En este caso en las instalaciones de la UDLA. Para esto se debe dar un click en GPS y luego en Ubicación GPS.

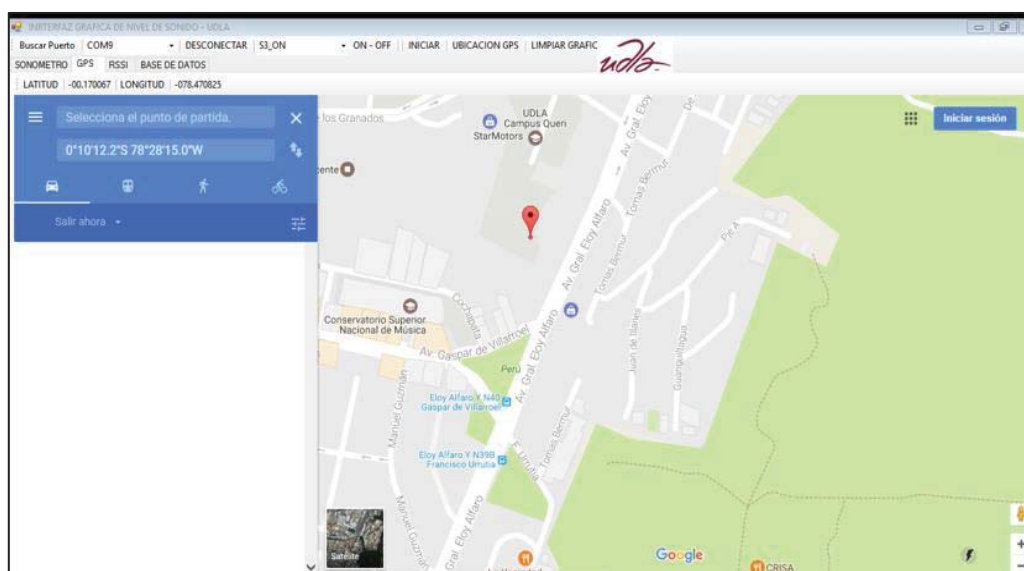


Figura 84. Sistema GPS de la red WSN en modo API.

Adaptado de (Google maps, s.f)

A continuación, en la figura 85 se muestra la interfaz que permite obtener el nivel de RSSI, para determinar la distancia aproximada de los nodos con respecto al Gateway. En la parte de determinar constante; es en la cual se realiza las mediciones con respecto al nodo uno para determinar la constante. Para cada medición se debe activar las casillas de Medidas. Una vez realizado las mediciones se debe dar un click en calcular constante y aparecerá un valor de constante. Luego se desactivan todas las casillas y se debe dar un click en DISTANCIA APROXIMADA, automáticamente mostrara el valor de RSSI y el valor en metros.

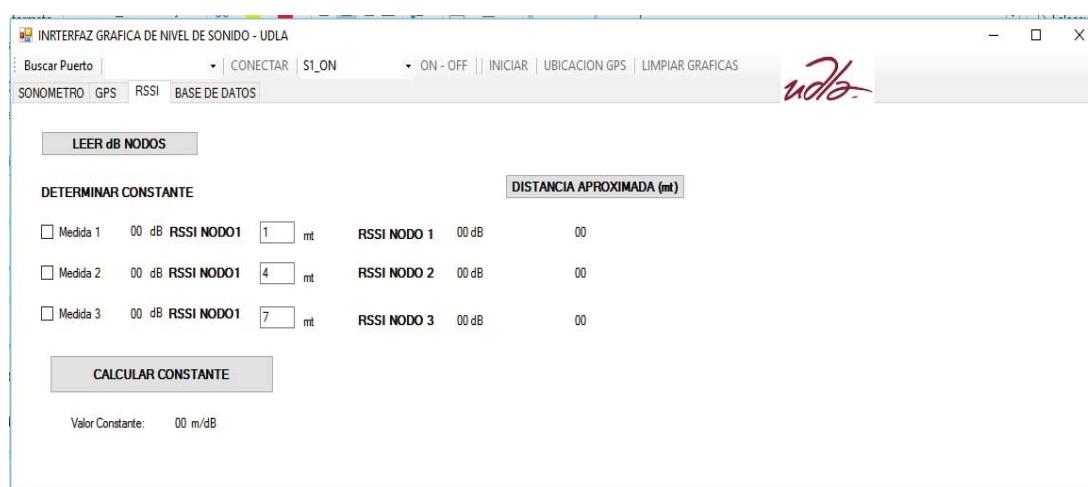


Figura 85. Sistema RSSI de la red WSN en modo API.

En la figura 86, se muestra los datos que se van a guardar en la base de datos SQL SERVER 2014. En el momento que inicia el monitoreo automáticamente se empieza a guardar los datos censados.

NODO 1		NODO 2		NODO 3	
Id	Latitud	Id	dB Sonido	Id	dB Sonido
1	-00.342185	1	42	1	67
2	-00.342187	2	57	2	152
3	-00.342185	3	54	3	153
4	-00.342183	4	58	4	153
5	-00.342183	5	53	5	149
6	-00.342176	6	60	6	151
7	-00.342167	7	58	7	154
8	-00.342159	8	53	8	150
9	-00.342157	9	53	9	151
10	-00.342165	10	59	10	149

Figura 86. Sistema base de datos de la red WSN en modo API.

3. CAPÍTULO III. PRESENTACIÓN DE RESULTADOS

3.1 Calibración de Equipos

Para obtener una óptima medición se debe calibrar el sonómetro, el mismo que se realiza con la colaboración del personal del laboratorio de Ingeniería en Sonido y Acústica de la Universidad de las Américas.

Con la ayuda de un calibrador acústico clase UNO, que se utiliza para la verificación de sonómetros y dosímetros, se realiza la calibración. El proceso para realizar este trabajo es inyectar una sola frecuencia de 1KHz y un solo nivel acústico de 94dB. En la figura 87, se muestra el equipo de calibración.



Figura 87. Calibrador acústico clase Uno.

Con el equipo CESVA se inyecta una señal constante de frecuencia al sonómetro (SL-1361), y se verifica los 94dB que debe mostrar el equipo en su pantalla, como se observa en la figura 88. En la cual el sonómetro del nodo uno, tiene un error de 0,1dB con respecto a 94 dB y el equipo dos posee un error de 0,3dB comparado a los 94dB.



Figura 88. Calibración de los dos sonómetros SL1361.

3.2 Protocolos para realizar la prueba Mesh

Las variables que tiene un gran impacto en el desempeño de la red de sensores de ruido son el alcance que prestan los módulos XBee con su red

ZigBee y el tiempo de duración de las baterías que alimentan al circuito. La empresa DIGI fabricante de los modules XBee, ha diseñado un parámetro que permite clasificar una señal como muy fuerte, fuerte, moderada y débil; basándose en un parámetro llamado LQI (Link Quality Indication), que es un numero entero entre 256 y 0. En la tabla 24, se muestra los valores fijados por la empresa Digi.

Tabla 24.

Niveles de enlace - LQI (Link Quality Indication).

Calidad	V. Máximo	V. Mínimo	Unidades
Muy fuerte /Excelente	256	195	LQI
Fuerte	195	130	LQI
Moderado	130	65	LQI
Débil, perdida de comunicación	65	0	LQI

Estos valores serán tomados en cuenta para realizar las pruebas de la red mesh.

Pasos para realizar las pruebas:

- Determinar el lugar a realizar las pruebas.
- Realizar el diseño del sitio en los cuales se va poner los nodos.
- Revisar en el software XCTU que exista enlace entre todos los nodos y verificar que se encuentre operativo la red Mesh.
- Revisar los niveles de LQI, que indiquen un óptimo funcionamiento de los enlaces.
- Mediante el software XCTU verificar, que si un nodo se desconecta del enlace directo que mantiene con el Gateway, se debe conectar a otro nodo y desde ahí debe llegar al Gateway, con esto se verificara que funcione la redundancia en la red MESH.





En la figura 89, se puede ver los niveles de LQI, de una red DigiMesh y una red ZigBee.

Connection default color:

Show colored connections based on their quality

DigiMesh network

Modify the minimum values of the quality ranges and their colors:

Quality	Maximum	Minimum	Units	Color
Very strong	0	-70	dBm	 (0,125,0)
Strong	-70	-80	dBm	 (12,150,159)
Moderate	-80	-90	dBm	 (212,105,0)
Weak	-90	-100	dBm	 (231,0,0)

ZigBee network

Modify the minimum values of the quality ranges and their colors:





Quality	Maximum	Minimum	Units	Color
Very strong	256	195	LQI	 (0,125,0)
Strong	195	130	LQI	 (12,150,159)
Moderate	130	65	LQI	 (212,105,0)
Weak	65	0	LQI	 (231,0,0)

Figura 89. Valores de los parámetros de enlace (LQI).

Para evaluar las baterías se realizó un monitoreo constante en el software XCTU, al momento que las baterías se descargan ya no se tiene conexión con el nodo remoto. Llegando a la conclusión del tiempo de duración de la autonomía del nodo, no pasa más de una hora en monitoreo constante.

3.2.1 Pruebas de la red Mesh

A continuación, se muestra las pruebas de la red Mesh. Para lo cual se sigue los pasos descritos en los protocolos de pruebas.

3.2.1.1 Determinar el lugar a realizar las pruebas

La prueba se realizó en el bosque de la parroquia el INCA específicamente en el barrio BALCONES DEL INCA de la ciudad de Quito. Para seleccionar el sitio se consideró que todos los nodos tengan línea de vista y el clima soleado que permita no tener pérdidas por fenómenos atmosféricos como la lluvia y neblina. Para las pruebas se consideró el valor de LQI, el mismo que es crítico para

mantener una comunicación estable. El valor debe mantenerse entre 256 y 195, por lo que se ha tomado distancias entre 20 y 40 metros entre los nodos.

A continuación, en la figura 90, se muestra el mapa del sitio de pruebas.

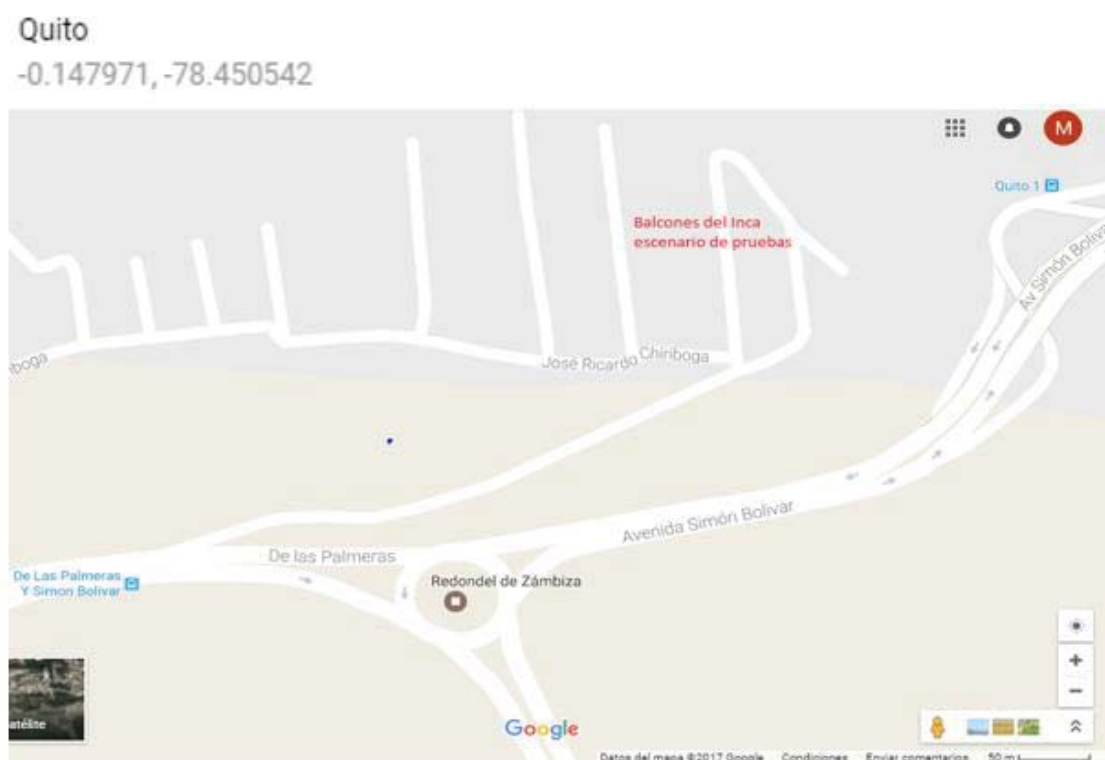


Figura 90. Vista en el mapa del sector a realizar pruebas.

Adaptado de (Google maps, s.f)

En la figura 91, se muestra el sitio de las pruebas visto desde el satélite. En la misma se ha colocado iconos de color amarillo, que indican los puntos que se utilizó para instalar los nodos.



Figura 91. Distribución de nodos en el sitio de pruebas.

Adaptado de (Google earth, s.f)

3.2.1.2 Diseño de los nodos en el sitio, en el cual se va instalar

Mediante la utilización del software Google Earth se realizó un recorte del sitio de las pruebas, vistas desde el satélite y se instaló los nodos de acuerdo a las distancias recomendadas de LQI, para mantener un enlace óptimo. En la figura 92, se muestra los nodos instalados, los mismos que son: El nodo coordinador (C), nodo Uno (R1), nodo dos (R2) y nodo tres (R3).



Figura 92. Distancias en metros entre los nodos. R1) router 1 R2) router 2 R3) router 3.

Adaptado de (Google earth, s.f)

3.2.1.3 Pruebas de funcionamiento de la red MESH con redundancia

Una vez instalado todos los nodos y haciendo uso de una PC, con el software XCTU, se empezó a realizar las pruebas. Para lo cual primero se enciende todos los nodos y se los agrega al software XCTU, desde ahí se visualiza si los enlaces se encuentran operativos.

En la figura 93, se puede ver funcionando todos los enlaces de la red Mesh. El software muestra líneas y números de color verde y azul, que indican que los enlaces se encuentran funcionando correctamente, con redundancia ante posibles fallas en los enlaces. En la misma se observa que todos los nodos se pueden comunicar entre ellos, además tienen conexión directa con el coordinador.

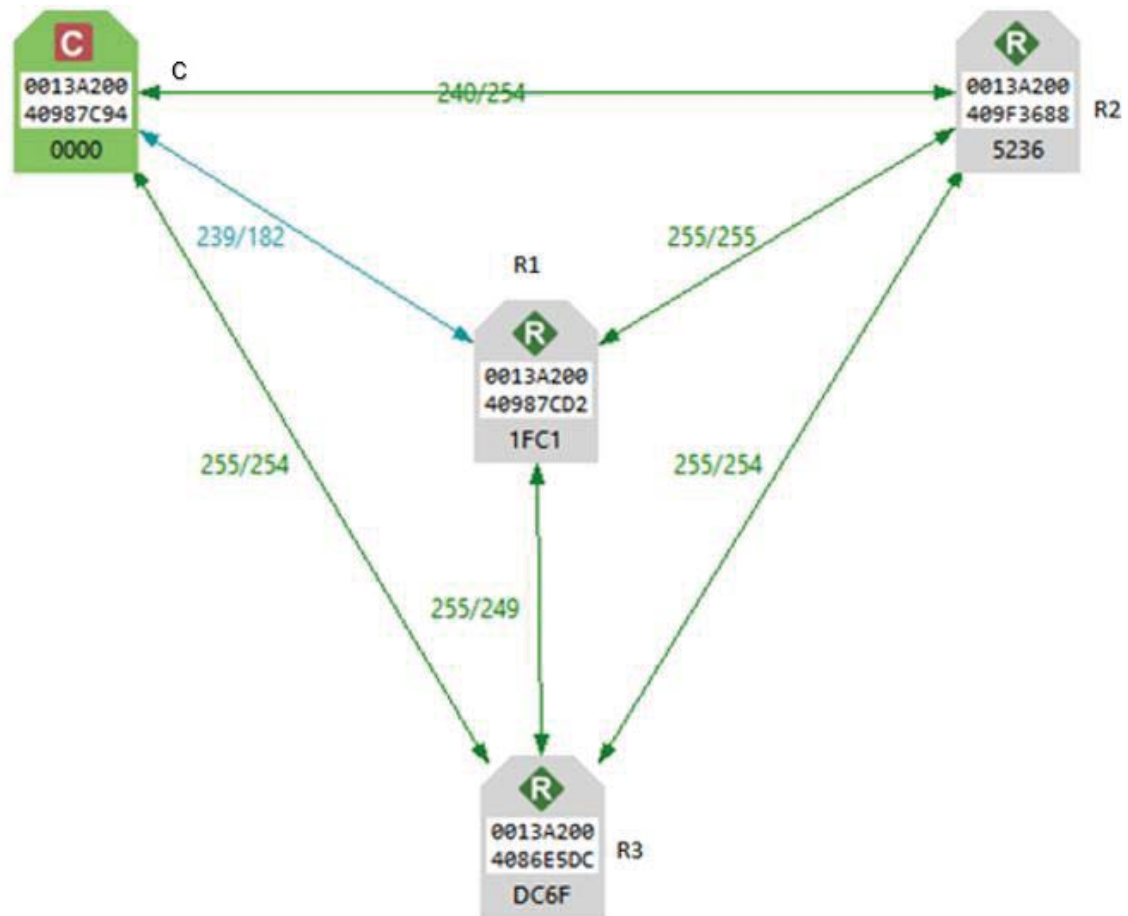


Figura 93. Pruebas de redundancia con conexión correcta. C: coordinador, R1: nodo 1, R2: nodo 2 y R3: nodo 3.

En la tabla 25, se muestra los niveles de enlace (LQI) que presentan los diferentes enlaces de la red MESH. Como resultado se tiene un óptimo nivel de transmisión ya que todos los enlaces presentan niveles superiores a 190, que es el límite aceptable para tener una buena transmisión y recepción.

Tabla 25.

Niveles de LQI, en los enlaces. Coordinador (C), R1: nodo 1, R2: nodo 2 y R3: nodo 3.

	Coordinador	Nodo Uno (R1)	Nodo dos (R2)	Nodo tres (R3)
Coordinador (C)	-	TX de C a R1 289	TX de C a R2 240	TX de C a R3 255

R1	TX de R1 a C 182	-	TX de R1 a R2 255	TX de R1 a R3 255
R2	TX de R2 a C 254	TX de R2 a R1 255	-	TX de R2 a R3 254
R3	TX de R3 a C 254	TX de R3 a R1 249	TX de R3 a R2 255	-

3.2.1.4 Demostración de la red Mesh con nodos remotos fuera de la línea de vista

Para realizara esta prueba se realizó un movimiento de distancia del nodo R1, a 55 metros con respecto al coordinador, como se muestra en la figura 94. Se realizó ese movimiento con el objetivo de perder comunicación directa entre el Coordinador y el Nodo R1.



Figura 94. Distancia en metros de los nodos para las pruebas de Mesh.
Adaptado de (Google earth, s.f)

En la figura 95 se muestra el funcionamiento de la red mesh, cuando el nodo remoto pierde la línea de vista hacia el coordinador, en este caso R1. En la misma se observa que a pesar de haber perdido la conexión directa con el coordinador, el nodo no deja de transmitir datos, para lo cual se interconecta con los otros nodos R2 Y R3.

Además, se observa que su valor de LQI se encuentra entre **147 y 112**, quedando por muy debajo de lo establecido que es 190. Para obtener estos resultados: se mueve el nodo router 1, hacia adelante, haciendo que su valor LQI, se encuentre fuera del rango establecido para una buena conexión. Como se puede observar R1 se encuentra desconecta en relación al Gateway de forma directa ya que los números y la línea muestra un color rosado, indicativo de desconexión. Pero no se ha desconectado de toda la red ya que se ha conectado a los otros nodos que si están operativos. Demostrado el funcionamiento de la red MESH con redundancia.

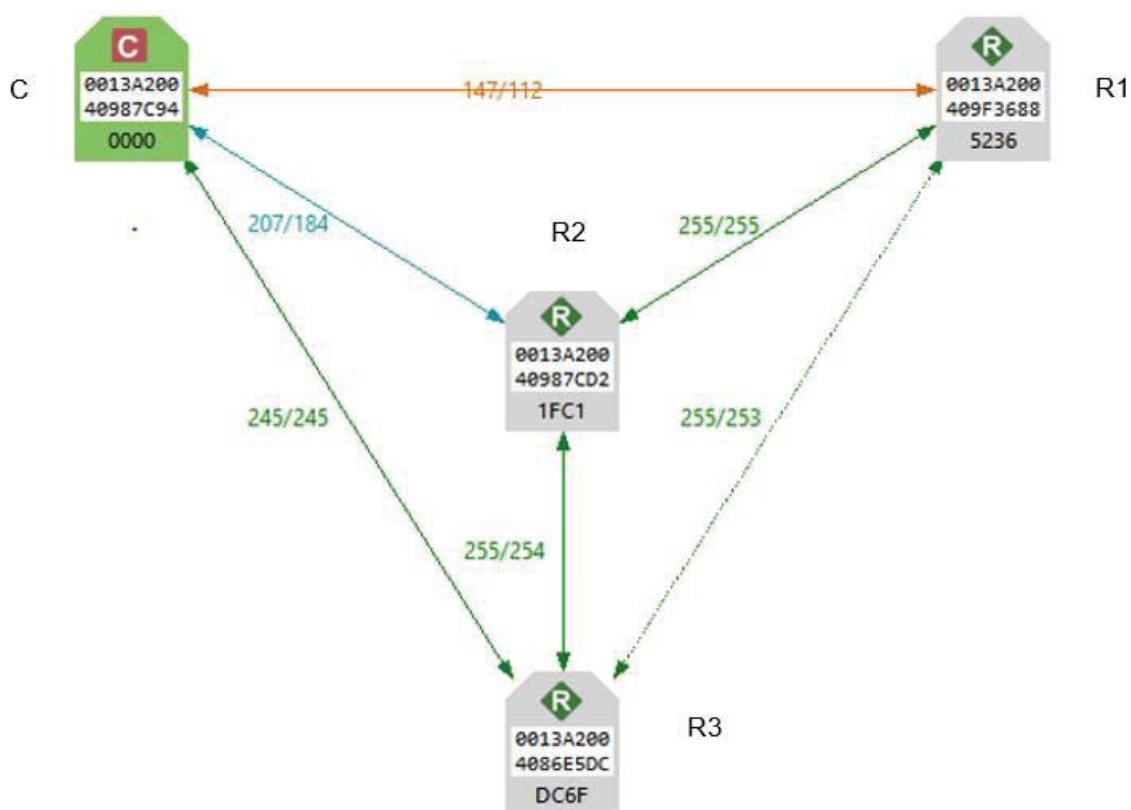


Figura 95. Prueba número uno de la red mesh. C: coordinador, R1: nodo 1, R2: nodo 2 y R3: nodo 3.

En la figura 96, se muestra la desconexión del R1, tanto hacia el R3 como al coordinador. Pero se enlaza al nodo R2 y se conecta a la red. Para lo cual se tuvo que mover al nodo fuera del rango de conexión. Demostrando nuevamente el funcionamiento de la red mesh.

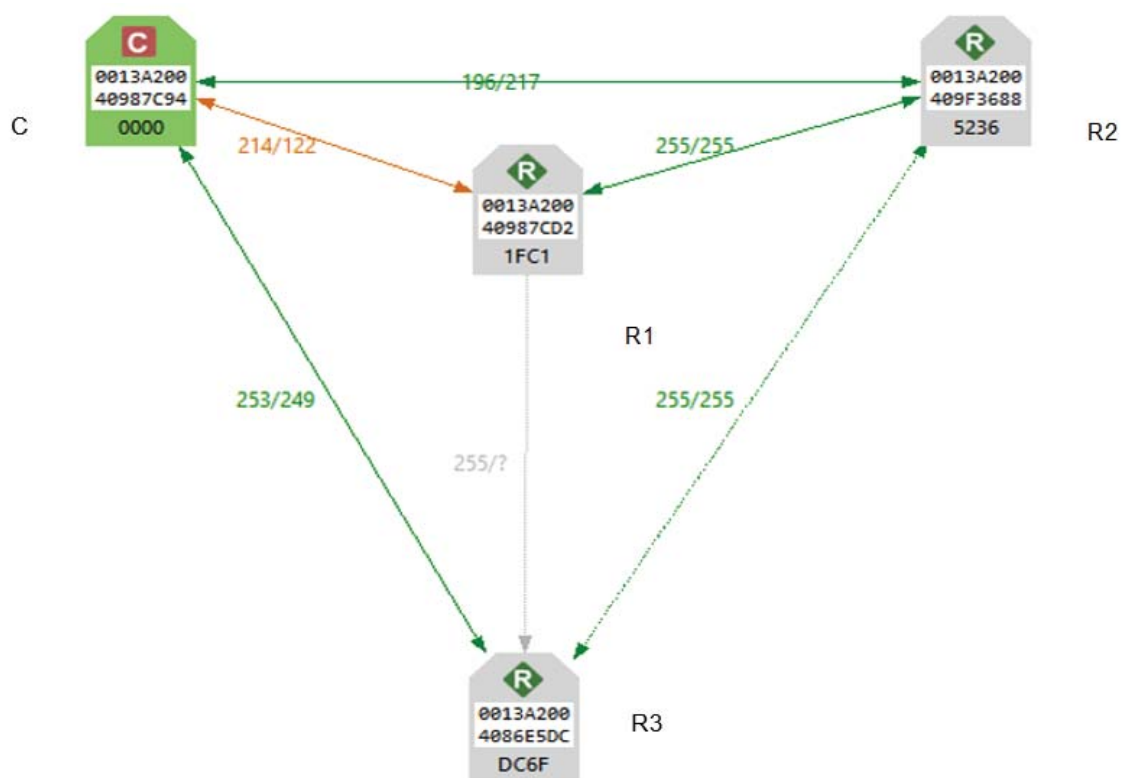


Figura 96. Prueba número dos de la Red MESH. C: coordinador, R1: nodo 1, R2: nodo 2 y R3: nodo 3.

3.3 Protocolos para las pruebas de RSSI

Para obtener el RSSI de los nodos se utilizó el siguiente método:

1. Primero se debe obtener el valor de la constante del área a medir, ese valor se lo obtiene realizando tres mediciones a diferentes distancias desde el coordinador hacia el nodo router uno. A continuación, se muestra las distancias que se va a medir. El nodo router UNO es el dispositivo que tiene que moverse en las diferentes distancias. Como se muestra en la figura 97.

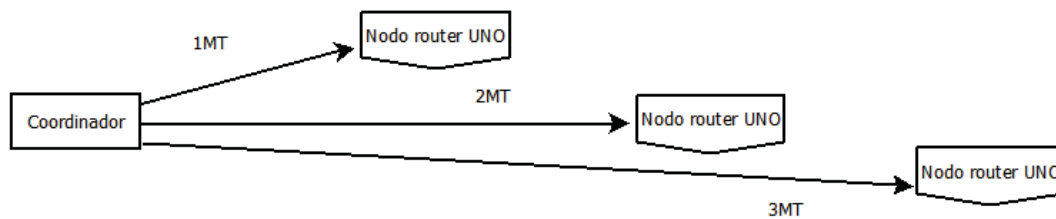


Figura 97. Valores de los parámetros LQI.

En figura 98, se muestra los valores obtenidos en dB después de la medición. Para realizar las mediciones es importante solo activar el Check list del casillero que se va medir. Ejemplo. Si se necesita el valor de la MEDIDA 1, desactivar los demás casilleros y activar solo esa casilla, luego realizar un click en LEER dB de Nodos y mostrara el valor medido. Luego para el valor de la MEDIDA 2, desactivar los demás casilleros y solo activar el que está midiendo.

La interfaz muestra los siguientes datos:

DETERMINAR CONSTANTE		DISTANCIA APROXIMADA (m)				
<input type="checkbox"/> Medida 1	-32	Distancia	1 mt	RSSI NODO 1	-35 dB	2.051 m
<input type="checkbox"/> Medida 2	-34	Distancia	2	RSSI NODO 2	-59 dB	3.457 m
<input type="checkbox"/> Medida 3	-35	Distancia	3	RSSI NODO 3	-32 dB	1.875 m

Valor Constante: 0.059 m/dB

Figura 98. Valores de RSSI de los nodos.

2. Con estos valores se calcula un valor denominado constante que permite calcular el nivel de RSSI de los demás nodos.

Para lo cual se debe presionar el icono calcular constante. En la siguiente tabla 26, se muestra las tres mediciones realizadas y el cálculo de la constante, que es automático.

Tabla 26.

Valores de RSSI tomados con respecto al Router 1 y fórmula para calcular el valor de la constante.

Mediciones	Medida 1	Medida 2	Medida 3
	Nodo 1	Nodo 1	Nodo 1
Distancia	1mt	2mt	3mt
Valor de RSSI medido en dB	-32	-34	-35
<u>Constante: 0.059mdB</u>			

El programa realiza el cálculo haciendo uso de las siguientes formulas.

Ecuación (10)

$$\text{Constante} = \left(\left(\frac{\text{Medida 1}}{-\text{dB1}} \right) + \left(\frac{\text{Medida 2}}{-\text{dB2}} \right) + \left(\frac{\text{Medida 3}}{-\text{dB3}} \right) \right) / 3$$

Ecuación (11)

$$\text{Distancia Aproximada en metros} = (\text{Constante} * -\text{dB medido de cada nodo})$$

3. Por último, se solicita el valor de RSSI de todos los nodos, los mismos que serán transformados a una distancia aproximada en metros, con respecto al Gateway.

Para lo cual se debe desactivar todas las casillas y presionar **LEER dB de Nodos** y por último presionar el icono de **Calcular distancia** y el programa calculara el valor. Los datos de la distancia no son exactos, a menores

distancias en error es de 0.50cm a 4mt. A mayor distancia el valor sube, en distancias mayores de 30mt el error es de 2 a 8 mt.

3.3.1 Pruebas de RSSI

a) Primera Prueba

La primera prueba se realizó en la UITEC de la Universidad de las Américas. En la misma se utilizó el protocolo de mediciones de RSSI para realizar estas pruebas. En la figura 99 se muestra los datos de la prueba realizada.

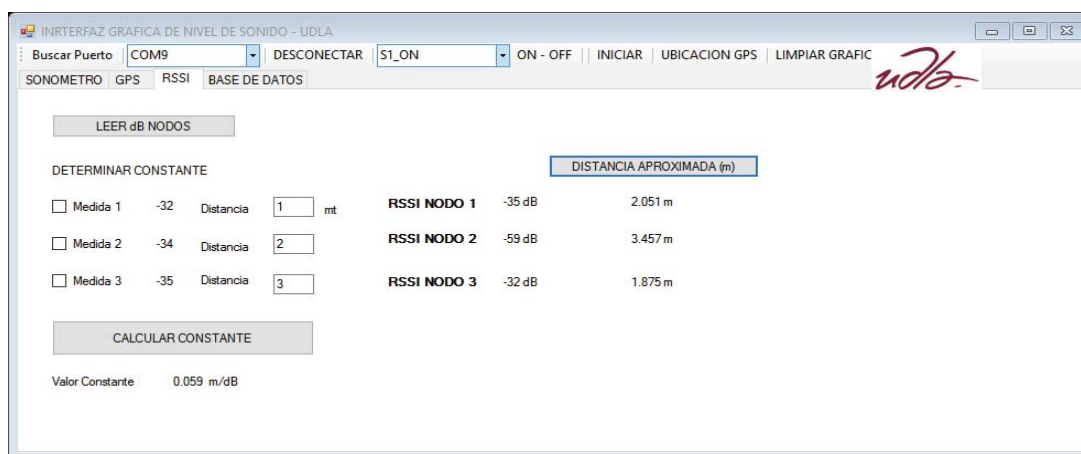


Figura 99. Prueba uno de medición de RSSI.

En la tabla 27 y figura 99, se muestra los valores de las mediciones realizadas para obtener la constante. Como se puede ver se colocó al nodo 1, con respecto al coordinador a diferentes medidas. Las mismas que son 1, 2 y 3mt, con lo cual se obtuvo las diferentes medidas de RSSI en dB y se calculó la constante.

Tabla 27.

Valores de RSSI para calcular la constante.

Mediciones	Medida 1	Medida 2	Medida 3
	Nodo 1	Nodo 1	Nodo 1
Distancia	1mt	2mt	3mt
Valor de RSSI	-32	-34	-35

medido en dB			
<u>Constante: 0.059m/dB</u>			

Una vez que se obtuvo la constante se calculó la distancia promedio, haciendo uso de la fórmula del protocolo de medición de RSSI. Dando como resultado lo siguiente, con respecto al coordinador:

- **Nodo uno:** 2.51 metros, el nodo uno se encuentra a una distancia de 2.51mt con respecto al coordinador.
- **Nodo dos:** 3.45 metros.
- **No do tres:** 1.88 metros.

b) Segunda Prueba

En la figura 100, se indica la prueba número dos del nivel de RSSI y en la tabla 28, se muestra los datos de las medidas realizadas. El nodo uno se colocó con respecto al coordinador, a diferentes distancias. Las mismas que son 2, 4 y 6mt, con lo cual se obtuvo las diferentes medidas de RSSI en dB y se calculó la constante. Luego se realiza la medición de los demás nodos y con el uso de la formula indicada en el protocolo de medición de RSSI, se calcula la distancia con respecto al coordinador.

Tabla 28.

Valores de RSSI para obtener la constante, de la prueba dos.

Mediciones	Medida 1	Medida 2	Medida 3
	Nodo 1	Nodo 1	Nodo 1
Distancia	2mt	4mt	6mt
Valor de RSSI medido en dB	-33	-35	-36
<u>Constante: 0.114m/dB</u>			

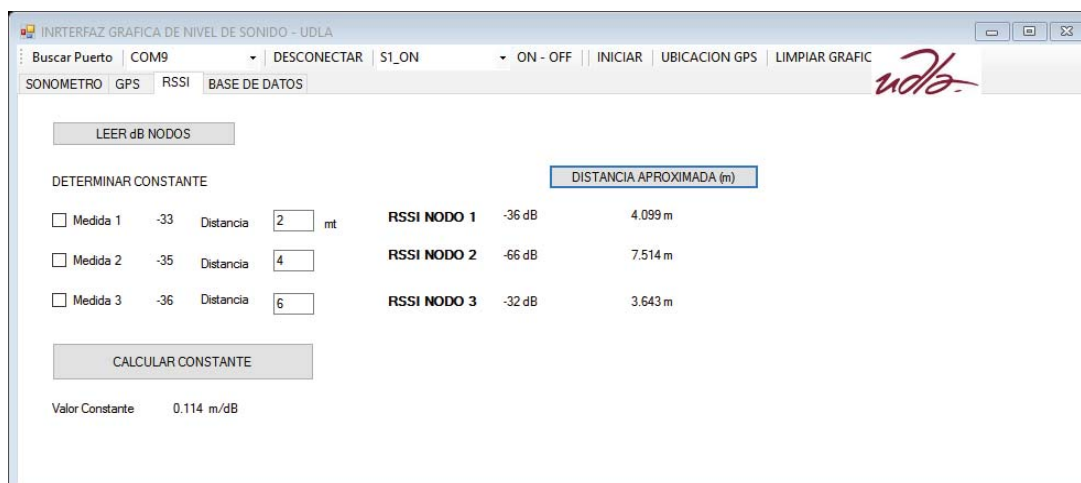


Figura 100. Prueba dos de medición de RSSI.

Los resultados en metros son los siguientes, con respecto al coordinador:

- Nodo Uno: 4 metros, el nodo uno se encuentra a una distancia de 4mt con respecto al coordinador.
- Nodo dos: 7.51 metros.
- Nodo tres: 3.64 metros.

c) Tercera prueba

En la figura 101, se muestra la prueba número tres realizada a largas distancias, aproximadamente a 40 y 50mt. En la misma se tiene un error de 3 a 5mt, en la posición real. El nodo uno se colocó con respecto al coordinador, a diferentes distancias. Las mismas que son 55, 45 y 40mt, con lo cual se obtuvo las diferentes medidas de RSSI en dB y se calculó la constante.

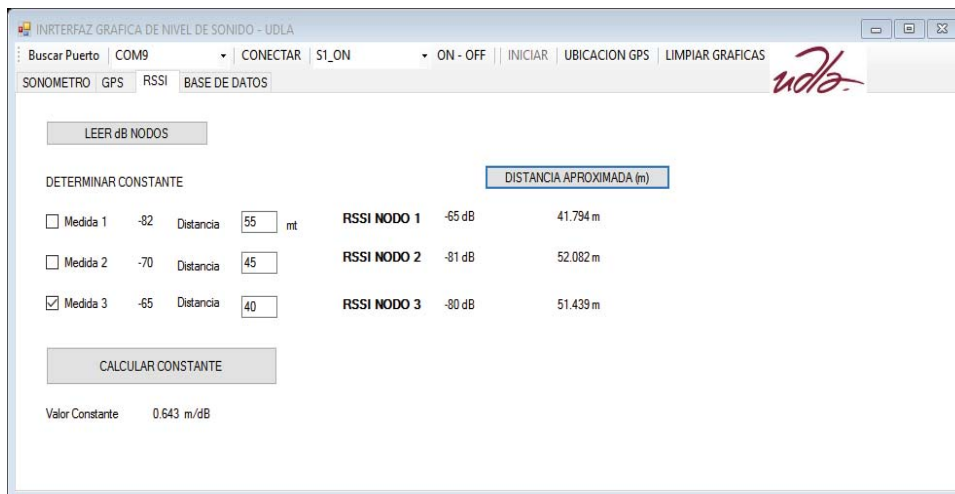


Figura 101. Prueba cuatro de medición de RSSI.

En la tabla 29, se tiene los siguientes datos para calcular la constante.

Tabla 29.

Datos para calcular la constante del nivel de RSSI, prueba tres.

Mediciones	Medida 1	Medida 2	Medida 3
	Nodo 1	Nodo 1	Nodo 1
Distancia	55mt	45mt	40mt
Valor de RSSI medido en dB	-82	-70	-65
Constante: 0.643m/dB			

Los resultados calculados en metros de los nodos son los siguientes:

- Nodo uno: 41.79 metros, el nodo uno se encuentra a una distancia de 41.79 mt con respecto al coordinador.
- Nodo dos: 52 metros.
- Nodo tres: 51.43 metros.

3.4 Pruebas del prototipo de Ruido

Para las pruebas de los niveles de ruido medidos en los sonómetros, se planteó dos escenarios. La primera prueba es haciendo uso de la PC, reproduciendo un video de varios ruidos de construcciones que se encuentra disponible en el siguiente enlace [Ruidos de Construcción](#).

El segundo escenario se escogió un lugar donde exista un alto tráfico de personas y automotores. En este caso se las realizó en la estación de buses interparroquiales a los valles cerca de la estación de la rio coca del DMQ, como se puede ver en la figura 102. Las pruebas se las realizo en la tarde en el horario de 18:00 a 18:30. Por motivos de falta de equipos no se pudo realizar las pruebas con los tres nodos ya que no se disponía de dos sonómetros.

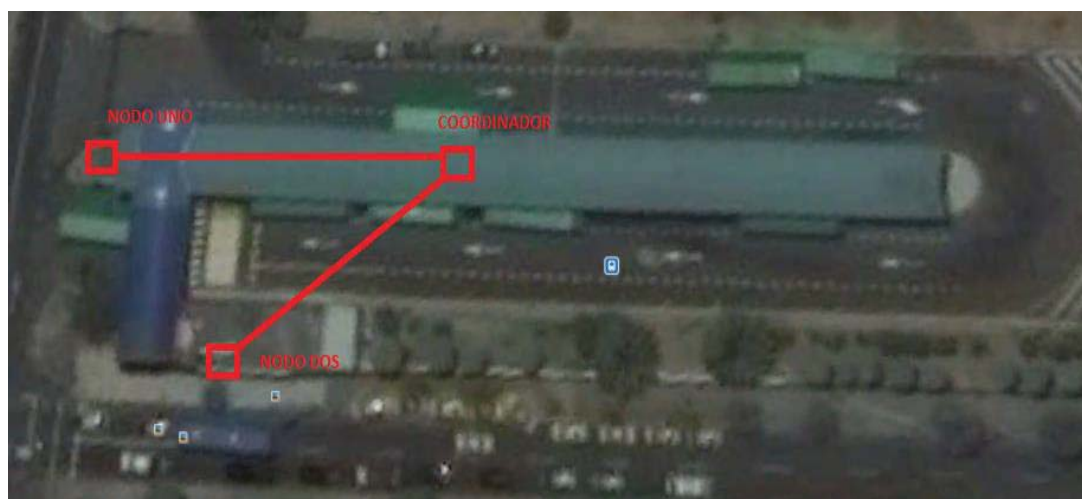


Figura 102. Mapa del sitio a medir.

Adaptado de (Google earth, s.f)

Se realizaron cinco pruebas en diferentes días. Primer día de pruebas, en el horario de 18:15, que se muestra en la figura 103, se puede ver una fluctuación entre 60 y 70 dB. Se menciona que existe varios sonidos en el área como: ruido de los automotores y personas.

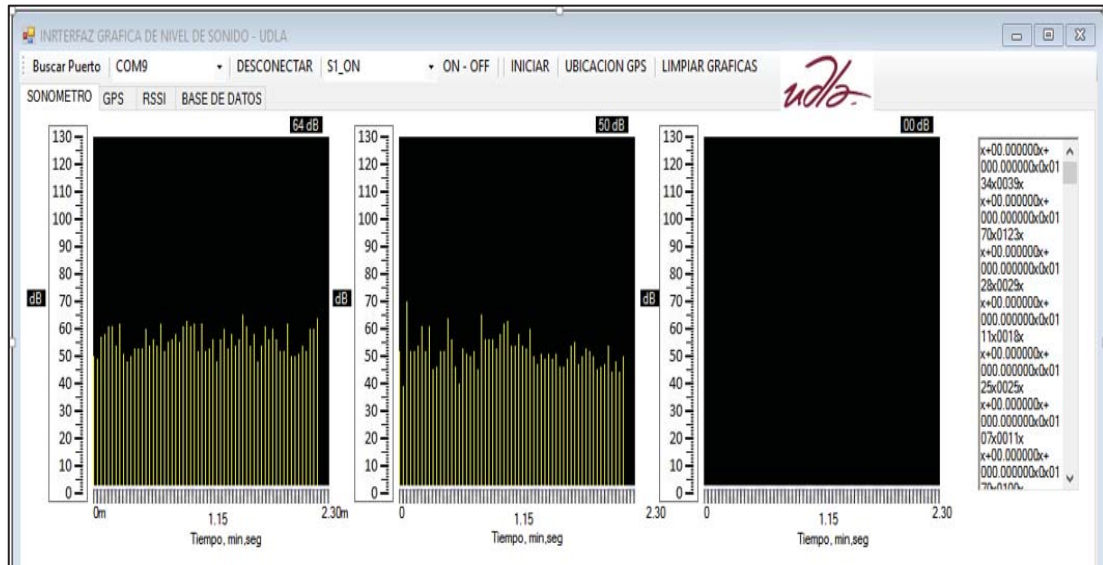


Figura 103. Pruebas de nivel de ruido, número 1.

Segundo día de pruebas, en el horario de 18:15, también se verifico que el ruido fluctúa entre 55 y 75 dB, como se ve en la figura 104.

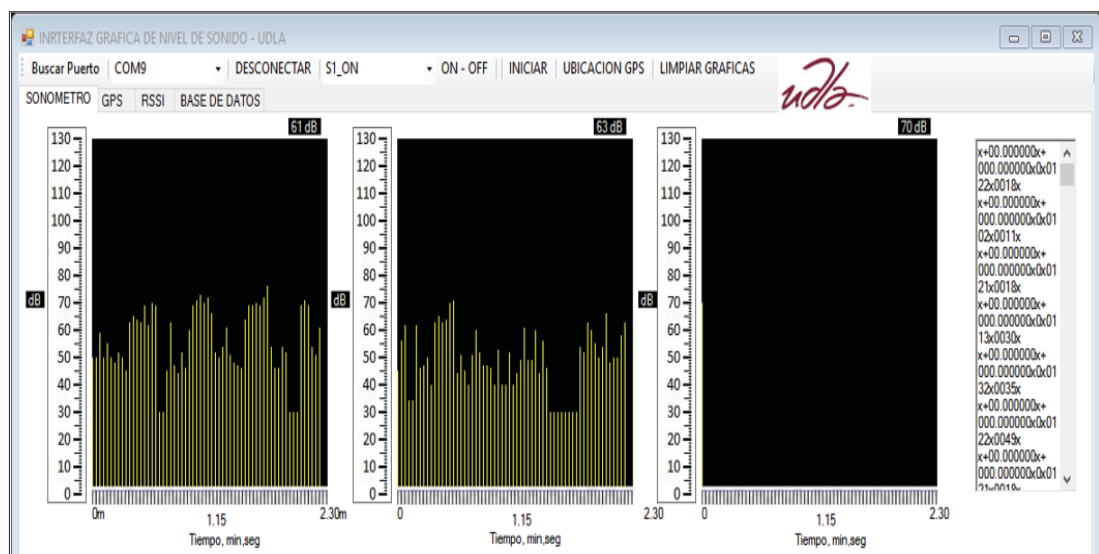


Figura 104. Pruebas de nivel de ruido, número 2.

Tercer día de pruebas, en el horario de 18:15, en la figura 105, se puede observar que el nivel de ruido varía entre 50 y 65dB.

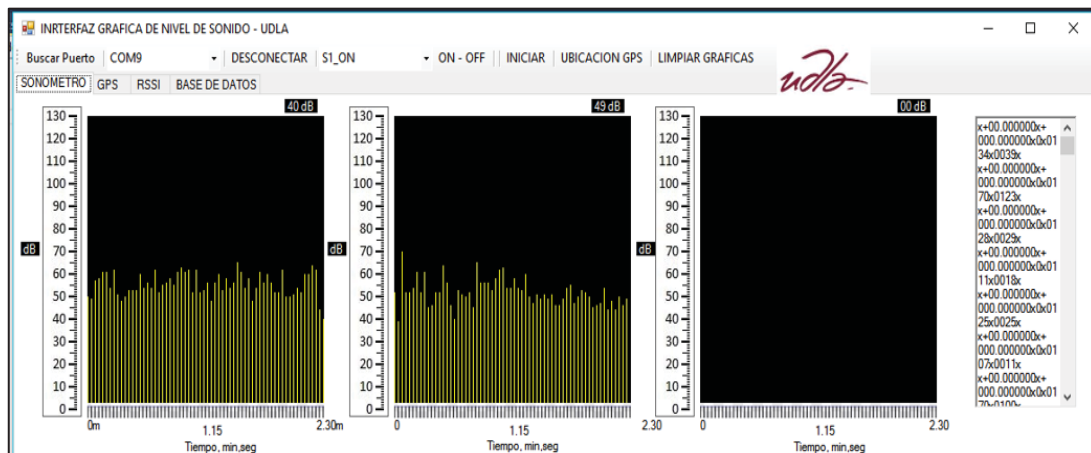


Figura 105. Pruebas de nivel de ruido, número 3.

Cuarto día de pruebas, en el horario de 18:15, se realizan las siguientes mediciones que varían entre 50 y 70dB. La misma que se puede ver en la figura 106.

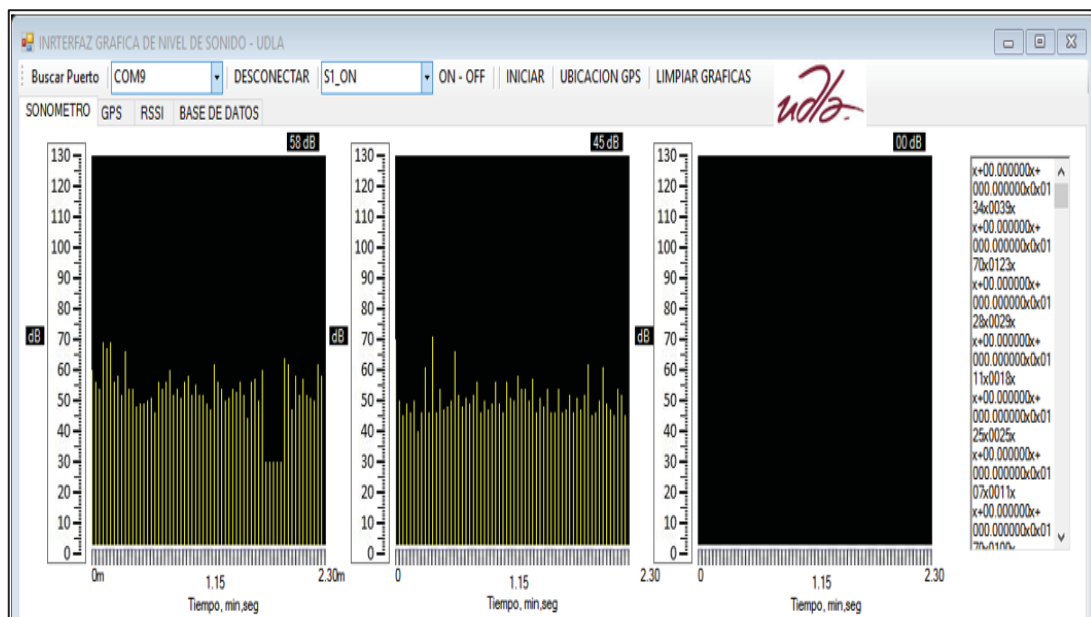


Figura 106. Pruebas de nivel de ruido, número 4.

Quinto día de pruebas, en el horario de 18:15, se observa las siguientes variaciones de nivel de ruido, que fluctúan entre 45 y 70dB, como se muestra en la figura 107.

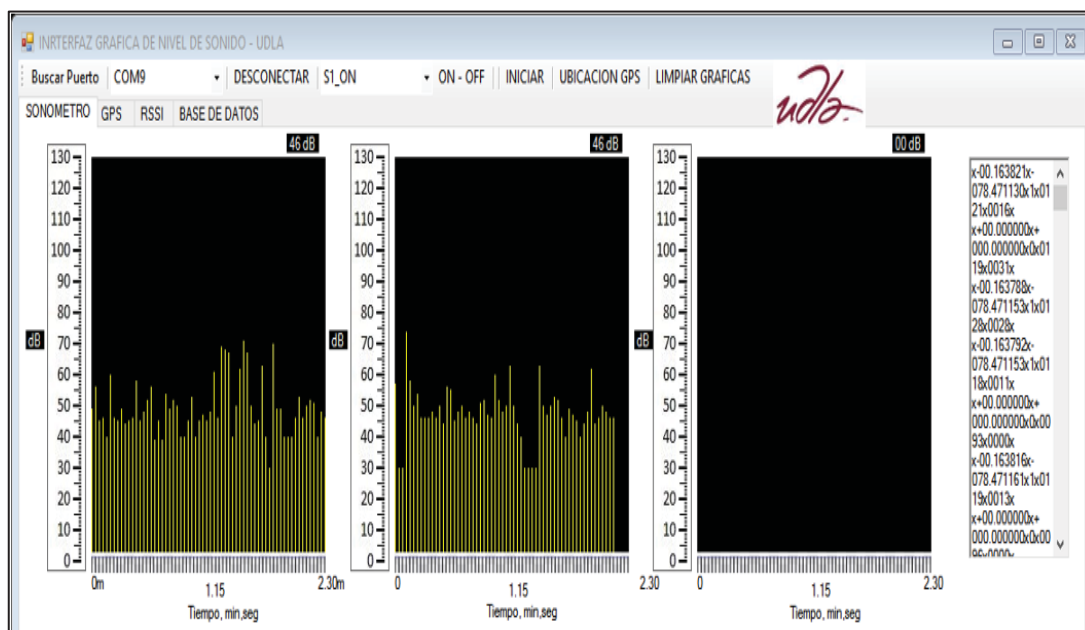


Figura 107. Pruebas de nivel de ruido, número 5.

Tabla 30.

Resultados de los valores censados

Características	Nodo UNO	Nodo DOS	Fecha
Primer día de pruebas	Fluctúa desde 55 a 65 dB	Fluctúa desde 50 a 70 dB	Miércoles 07 de Junio del 2017
Segundo día de pruebas	Fluctúa desde 50 a 73 dB	Fluctúa desde 50 a 70 dB	Jueves 08 de Junio del 2017
Tercer día de pruebas	Fluctúa desde 55 a 66 dB	Fluctúa desde 55 a 66 63	Viernes 09 de Junio del 2017
Cuarto día de pruebas	Fluctúa desde 52 a 64 dB	Fluctúa desde 48 a 58 63	Sábado 10 de Junio del 2017
Quinto día de pruebas	Fluctúa desde 45 a 65 dB	Fluctúa desde 45 a 53 dB	Domingo 11 de Junio del 2017

Después de revisar lo niveles de ruido censado por los sonómetros se determina que el intervalo se encuentra entre 48 a 76 dB. Existe minutos es los

cuales se mantiene en los 70 a 80 dB, los mismo que son; cuando los buses llegan y salen de la estación.

Por lo tanto el nivel de presión sonora se mantiene en la escala de tolerables y límites de los recomendados por la Organización Mundial de la Salud, que es 65 dB. En la tabla 30 se puede ver los resultados de las mediciones realizadas.

4. CAPÍTULO IV. ANÁLISIS DEL COSTO DE IMPLEMENTACIÓN DEL SISTEMA DE CONTAMINACIÓN AUDITIVA

En este capítulo se muestra los costos requeridos para la construcción e implementación del prototipo para la medición de contaminación auditiva. En la misma se presenta los costos de materiales, software y tiempo utilizado en desarrollar el proyecto.

4.1 Costos de materiales

En la tabla 31 se muestran los valores de los dispositivos electrónicos y materiales usados.

Tabla 31.

Valores de los costos Primarios.

Elementos	Cantidad	C. Unidad	C. Total
Arduino MiniPro	1	5	5 USD
XBee Pro S2B	3	77	231 USD
XBee Explorer	1	30	30 USD
Módulo GPS Neo	1	27	27 USD
Sonómetro	2	58	116 USD
Baterías Z ZTDM 2Pcs 18650 3.7V – 4.2V 5000MAH Li-ion Rechargeable	6	3.50	21 USD

Arduino UNO	2	10	20 USD
XBee Arduino Shield	2	15	30 USD
Materiales Electrónicos	Varios	50,00	50,00 USD
Impresión 3D	1	70,00	70,00 USD
TOTAL			600 USD

Los ítems se encuentran con valores de Estados Unidos.

4.2 Costos de Programación

A los gastos de materiales se agrega el costo de la programación de la interfaz gráfica en C#, con un total 10 días de desarrollo. En la tabla 32 se muestra el detalle de los días utilizados para desarrollar el programa y el costo requerido.

Tabla 32.

Costo de programación de la interfaz gráfica.

	Tiempo de desarrollo	Trabajo diario	Costo por hora	Costo por día	Total
Programación de la interfaz gráfica en C#	Dos semanas	8	8	8 horas x \$8= \$64	10 días * \$64= \$640

4.3 Costos de diseño

En la tabla 33, se muestra los costos requerido para el diseño del prototipo.

Tabla 33.

Costos de diseño del prototipo.

	Tiempo de desarrollo	Trabajo diario	Costo por hora	Costo por día	Total
Programación de la interfaz gráfica en C#	Un mes	8	8	8 horas x \$8= \$64	20 días * \$64= \$1280

4.4 Costo total del proyecto

En la tabla 34, se muestra los valores totales del proyecto.

Tabla 34.

Costo total del proyecto.

Costos	Valores
Costo de Materiales	600USD
Costo de Programación	640 USD
Costo de diseño	1280USD
TOTAL	2520 USD

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Una vez realizado todas las etapas descritas en el proyecto, se concluye con la construcción del nodo sensor, que permite censar el nivel de ruido y presentar la información en la interfaz gráfica. También se observa el correcto funcionamiento del GPS para la localización del nodo.

Con la implementación de la trama API, se evitó la pérdida de la información en los enlaces, que generalmente pasa en una transmisión AT.

Para diseñar la trama API, se utilizó la versión API1, ya que en el proyecto no se tuvo que utilizar caracteres especiales como: 0x7E, 0x7D, 0x11 y 0x13, para él envió de la información. En el caso de utilizar estos caracteres se debe diseñar la trama con API 2.

En la interfaz de visualización que gráfica el nivel de ruido, no muestra la información en tiempo real. El tiempo desde que capta la señal y se muestra en la gráfica tiene un intervalo de 7 segundos por cada nodo.

Con la presente investigación se determinó que, si no se toman medidas para combatir la contaminación auditiva, a futuro se tendrá muchos problemas de salud, pudiendo ocasionar la pérdida de la audición de una manera irreversible.

EL GPS Andoer NEO-6M tiene un retardo de 10 a 12 segundo para conectarse a la red satelital, y mostrar los datos de latitud y longitud. Sin embargo, el nodo remoto envía ceros en la trama, hasta que lleguen datos reales. También se debe tomar en cuenta, el lugar en la que se encuentre el nodo ya que en lugares cerrados que no tenga línea de vista al satélite, se le dificulta mucho la obtención de datos, llegando a no mostrar su posicionamiento.

Los efectos climatológicos y geográficos afectan en un decrecimiento el nivel de enlace (LQI) entre los nodos y el coordinador, en el escenario de pruebas existía ráfagas de viento que hacían variar entre 255 y 194. Afectando a una conexión en tiempo real a distancias mayores de 40mt.

La red Mesh funciono correctamente, ya que se demostró que si el nodo remoto se desconecta del coordinador, el mismo se conecta a los otros nodos, y a través de esos dispositivos llega a conectarse con el coordinador y armar nuevamente la red.

Una vez realizado el análisis de los valores de distancia calculados a partir del muestreo del parámetro RSSI, se pudo verificar que existen errores de hasta 8 metros. De las pruebas realizadas a distancias menores de 30mt el error es de 1 a 4mt, a mayores distancias aumenta el error. Para poder tener una mayor exactitud de distancia se debe realizar un mayor número de muestras del nivel de RSSI sobre el área a medir.

5.2 Recomendaciones

Después de realizar las mediciones y e informarse con la teoría, que los altos niveles de ruido provocan problemas en la salud. Se recomienda que las autoridades locales y gubernamentales, realicen campañas sobre lo perjudicial que representa para las personas, exponerse a altos niveles de ruido por tiempos prolongadas.

Si se desea implementar este proyecto en áreas industriales, se deberá tomar muy en cuenta los obstáculos y las posibles interferencias que podrían presentarse, ya que los equipos funcionan en la banda de 2.4GHz.

Con respecto a las pruebas del nivel de RSSI para calcular la distancia, se recomienda realizar un mayor estudio sobre ese tema, con varias pruebas en diferentes lugares. Ya que en el proyecto no se realizó un estudio a profundidad.

En el presente proyecto se utilizó, dispositivos de estudio como los arduinos, Xbee, sensores de ruido que no son de alta precisión. Para un proyecto que se necesite alta disponibilidad, se recomienda utilizar equipos originales.

Para la prueba de la red MESH, se recomienda utilizar más nodos routers, que permitan determinar el funcionamiento del enrutamiento de las tramas de una manera más efectiva.

Las baterías utilizadas en este proyecto no permiten tener una autonomía de energía por más de una hora. Se debería implementar unos sistemas que permita tener mayor autosuficiencia de energía, el mismo que puede ser con un panel solar o algún otro sistema.

REFERENCIAS

Escuntar, V. (2016). Diseño de un prototipo de red de sensores inalámbricos para monitorear la contaminación ambiental acústica/ sonora en la ciudad de Quito sector La Coruña y San Ignacio. Quito: Universidad Metropolitana AEDE QUITO.

Organización Mundial de la Salud (OMS). (s.f.). "Escuchar sin riesgos!". Recuperado el 28 de julio de 2015 de <http://www.who.int/topics/deafness/safe-listening/es/>

Organización mundial de la Salud. (s.f.). 1100 millones de personas corren el riesgo de sufrir pérdida de audición. Recuperado el 08 de julio de 2015 de <http://www.who.int/mediacentre/news/releases/2015/ear-care/es/>

Hisco, (s.f.). Equipos de protección auditiva en los centros de trabajo con generación de ruido. Recuperado el 11 de marzo de 2017 de <http://www.who.int/mediacentre/news/releases/2015/ear-care/es/>

World Health Organization, (1999). Guidelines for community noise. Recuperado el 11 de marzo de 2017 de <http://apps.who.int/iris/handle/10665/66217?locale=es>

Ecuador, Consejo Metropolitano de Quito. (s.f.). Código municipal para el Distrito Metropolitano de Quito. Quito, Ecuador: Ordenanza Metropolitana N° 404 del 23 de abril de 2013.

Asamblea Nacional. (s.f.). Sección segunda del medio Ambiente de la Carta Magna del Ecuador. Recuperado el 17 de mayo de 2016 de <http://pdba.georgetown.edu/Parties/Ecuador/Leyes/constitucion.pdf>

Vesilind, P., Morgan, S. y Heine, L. (2013). INTRODUCCIÓN A LA INGENIERÍA AMBIENTAL, (3.º ed.). Colombia: ISBN 13: 978-0-495-29583-9

Delgado. M. (2001). Sistemas de telecomunicación e informáticos. Electricidad-electrónica Madrid. Paraninfo, 2001. p229.

Solorzano, S. (2016). SISTEMAS DE MEDICIÓN DE CONTAMINACIÓN AUDITIVA, EMPLEANDO UNA RED DE SENSORES INALAMBRICOS Y SENSORES BAN. (Tesis de Ingeniería). Universidad de las Américas (UDLA) de Ecuador.

Leonardo, R. 2007. Riesgos físicos I: Ruido, vibraciones y presiones anormales, 2a. ed. Bogotá [Colombia]: Eco Ediciones.

Escuela de Organización industrial, 2012. Analizando el ruido ambiental en el entorno de la escuela. Recuperado el 6 de marzo de 2017 de <http://www.eoi.es/blogs/lauranavas/2012/05/10/analizando-el-ruido-ambiental-en-el-entorno-de-la-escuela/>

VAIVASUATA. (s.f.). Diferencia entre sonido y ruido. Recuperado el 04 de julio de 2015 de <http://diferenciaentre.info/diferencia-entre-sonido-y-ruido/>

American College of Occupational and Environmental Medicine. (2003). Noise-induced Hearing Loss. Journal of Occupation and Environmental Medicine. Recuperado el 01 de marzo de 2017 de <http://www.acoem.org/>

Aguirre, J. (2013). Redes inalámbricas. Recuperado el 8 de Julio de 2015 de <http://www.monografias.com/trabajos/redesinalam/redesinalam.shtml>

CCM. (2017). Redes inalámbricas. Recuperado el 05 de marzo de 2017 de <http://es.ccm.net/contents/818-redes-inalambricas>

National Instruments, (2009). Qué es una Red de Sensores Inalámbricos. Recuperado el 05 de marzo de 2017 de <http://www.ni.com/white-paper/7142/es/#toc3>

Xbee desde cero 11.0 (2015). Modo API 2. Recuperado el 05 de marzo de 2017 de <https://www.youtube.com/watch?v=yHDQI32qp5M>

Xbee desde cero 10.0 (2015). Modo API 1. Recuperado el 05 de marzo de 2017 de <https://www.youtube.com/watch?v=2XpIIaXDCpY>

Guaña, J. y Escuntar, V. (2016). Aplicación de las Redes de Sensores Inalámbricos. Quito: Universidad Metropolitana.

Digi. (s.f.). X-Ctu. Recuperado el 2 de marzo de 2017 de <http://www.digi.com/products/xbee-rf-solutions/xctu-software/xctu>

Aprendiendo Arduino, (s.f.). Recuperado el 2 de marzo de 2017 de <https://aprendiendoarduino.wordpress.com/2016/11/16/zigbeexbee/>

Maiten Chile, (2017). Equipos de comunicaciones. Recuperado el 20 de marzo de 2017 de http://www.maiten.cl/maiten_gac/inicio.html

Arduino, (2017). Arduino UNO Genuino. Recuperado el 20 de marzo de 2017 de <https://www.arduino.cc/en/main/arduinoBoardUno>

Arduino, (2017). Arduino MEGA. Recuperado el 20 de marzo de 2017 de <https://www.arduino.cc/en/Main/arduinoBoardMega>

DIGI, (2017). Received signal strength indicator (RSSI). Recuperado el 20 de mayo de 2017 de <http://docs.digi.com/pages/viewpage.action?pageId=2626044>

Andres, P. (2016). IMPLEMENTACIÓN DE UNA RED ZIGBEE MESH BASADA EN EL ESTANDAR IEEE 802.15.4 PARA LA OPERACIÓN DE UN DRON. (Tesis de Ingeniería). Universidad de la Fuerzas Armadas (ESPE).

Andres, P. (2016). IMPLEMENTACIÓN DE UNA RED ZIGBEE MESH BASADA EN EL ESTANDAR IEEE 802.15.4 PARA LA OPERACIÓN DE UN DRON. (Tesis de Ingeniería). Universidad de la Fuerzas Armadas (ESPE).

José, F. (2009). “Elaboración de un mapa de ruido ambiental para determinar la ubicación más apropiada de los puntos de monitoreo para la Red Mínima de Monitoreo del Ruido Ambiental en el DistritoMetropolitano de Quito, Zonas 2: Calderón, Carapungo, Centro, Los Chillos y Tumbaco”. (Tesis de Ingeniería). Universidad Internacional SEK (SEK).

Zigbee, (2017). XBee & XBee – Pro. Recuperado el 10 de Julio de 2017 de https://cdn.sparkfun.com/datasheets/Wireless/Zigbee/ds_xbeezbmodules.pdf

ANEXOS

ANEXO A. Datasheet del dispositivo XBee.

XBee S2 Quick Reference Guide

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

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

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

Transparent - Communication through the XBee. If data is not generated from the XBee itself then both XBees should be set to AT. **Command** - Communication to the XBee. If one XBee is sending data, that XBee should be in AT mode while the receiving one should be in API mode.

XBee Modes

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

XBee Modes	<p>Transparent - Communication through the XBee. If data is not generated from the XBee itself then both XBees should be set to AT. Command - Communication to the XBee. If one XBee is sending data, that XBee should be in AT mode while the receiving one should be in API mode.</p>																																																																											
XBee Setup	<p>Connect the XBee to a TTL Serial FTDI adapter - OR - Arduino hack: Connect RX to RX, TX to TX, RESET to ground to bypass the Arduino entirely and get serial to XBee. Use the free X-CTU software to configure the XBee. Baud: 9600 - FC: Hardware - Data Bits: 8 - Parity: None - Stop Bits: 1</p>																																																																											
Basic Settings	<p>PAN ID - The network to communicate over. If 0, the XBee will join any. DH/DL - Destination Serial number. Used to send to a specific XBee's Serial. Set to 0 to send to just the Coordinator. Set to 0x0000000000FFFF to broadcast. JV - Router/EP should be set to 1 so it rejoins the network on startup.</p>																																																																											
Pin Settings	<p>For pin settings to work, receiver XBee must be in API mode DO - Set pin 0 to start sensing IR - Collect data on sensing pins every XX millisecs</p>																																																																											
API format for Remote AT Command Request	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Byte</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>0</td><td>0x7e</td><td>Start byte - Indicates beginning of data frame</td></tr> <tr><td>1</td><td>0x00</td><td>Length - Number of bytes (ChecksumByte# - 1 - 2)</td></tr> <tr><td>2</td><td>0x10</td><td></td></tr> <tr><td>3</td><td>0x17</td><td>Frame type - 0x17 means this is a AT command Request</td></tr> <tr><td>4</td><td>0x52</td><td>Frame ID - Command sequence number</td></tr> <tr><td>5</td><td>0x00</td><td>64-bit Destination Address (Serial Number)</td></tr> <tr><td>6</td><td>0x13</td><td>MSB is byte 5, LSB is byte 12</td></tr> <tr><td>7</td><td>0xA2</td><td></td></tr> <tr><td>8</td><td>0x00</td><td>0x0000000000000000 = Coordinator</td></tr> <tr><td>9</td><td>0x40</td><td>0x0000000000000000 = Broadcast</td></tr> <tr><td>10</td><td>0x77</td><td></td></tr> <tr><td>11</td><td>0x9C</td><td></td></tr> <tr><td>12</td><td>0x49</td><td></td></tr> <tr><td>13</td><td>0xFF</td><td>Destination Network Address</td></tr> <tr><td>14</td><td>0xFE</td><td>(Set to 0xFFFF to send a broadcast)</td></tr> <tr><td>15</td><td>0x02</td><td>Remote command options (set to 0x02 to apply changes)</td></tr> <tr><td>16</td><td>0x44 (D)</td><td>AT Command Name (Two ASCII characters)</td></tr> <tr><td>17</td><td>0x02 (2)</td><td></td></tr> <tr><td>18</td><td>0x04</td><td>Command Parameter (queries if not present)</td></tr> <tr><td>19</td><td>0xF5</td><td>Checksum</td></tr> </tbody> </table>	Byte	Example	Description	0	0x7e	Start byte - Indicates beginning of data frame	1	0x00	Length - Number of bytes (ChecksumByte# - 1 - 2)	2	0x10		3	0x17	Frame type - 0x17 means this is a AT command Request	4	0x52	Frame ID - Command sequence number	5	0x00	64-bit Destination Address (Serial Number)	6	0x13	MSB is byte 5, LSB is byte 12	7	0xA2		8	0x00	0x0000000000000000 = Coordinator	9	0x40	0x0000000000000000 = Broadcast	10	0x77		11	0x9C		12	0x49		13	0xFF	Destination Network Address	14	0xFE	(Set to 0xFFFF to send a broadcast)	15	0x02	Remote command options (set to 0x02 to apply changes)	16	0x44 (D)	AT Command Name (Two ASCII characters)	17	0x02 (2)		18	0x04	Command Parameter (queries if not present)	19	0xF5	Checksum												
Byte	Example	Description																																																																										
0	0x7e	Start byte - Indicates beginning of data frame																																																																										
1	0x00	Length - Number of bytes (ChecksumByte# - 1 - 2)																																																																										
2	0x10																																																																											
3	0x17	Frame type - 0x17 means this is a AT command Request																																																																										
4	0x52	Frame ID - Command sequence number																																																																										
5	0x00	64-bit Destination Address (Serial Number)																																																																										
6	0x13	MSB is byte 5, LSB is byte 12																																																																										
7	0xA2																																																																											
8	0x00	0x0000000000000000 = Coordinator																																																																										
9	0x40	0x0000000000000000 = Broadcast																																																																										
10	0x77																																																																											
11	0x9C																																																																											
12	0x49																																																																											
13	0xFF	Destination Network Address																																																																										
14	0xFE	(Set to 0xFFFF to send a broadcast)																																																																										
15	0x02	Remote command options (set to 0x02 to apply changes)																																																																										
16	0x44 (D)	AT Command Name (Two ASCII characters)																																																																										
17	0x02 (2)																																																																											
18	0x04	Command Parameter (queries if not present)																																																																										
19	0xF5	Checksum																																																																										
API format for Remote AT Command Response	<pre> // Remote XBee: AT, Base XBee: API Serial.write(0x7E); // Sync up the start byte Serial.write(byte(0x0)); // Length MSB (always 0) Serial.write(0x10); // Length LSB Serial.write(0x17); // 0x17 is the frame ID for sending an AT command Serial.write(byte(0x0)); // Frame ID (no reply needed) Serial.write(byte(0x0)); // Send the 64 bit destination address Serial.write(byte(0x0)); // (Sending 0x0000000000000000 (broadcast)) Serial.write(byte(0x0)); Serial.write(byte(0x0)); Serial.write(byte(0x0)); Serial.write(0xFF); Serial.write(0xFF); Serial.write(0xFF); // Destination Network Serial.write(0xFE); // (Set to 0xFFFF if unknown) Serial.write(0x02); // Set to 0x02 to apply these changes Serial.write('D'); // AT Command: D1 Serial.write('1'); Serial.write(0x05); // Set D1 to be 5 (Digital Out HIGH) long checksum = 0x17 + 0xFF + 0xFF + 0xFE + 0x02 + 'D' + '1' + 0x05; Serial.write(0xFF - (checksum & 0xFF)); // Checksum </pre>																																																																											
API format for I/O Data Sample RX Indicator	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Byte</th> <th>Example</th> <th>Description</th> </tr> </thead> <tbody> <tr><td>0</td><td>0x7e</td><td>Start byte - Indicates beginning of data frame</td></tr> <tr><td>1</td><td>0x00</td><td>Length - Number of bytes (ChecksumByte# - 1 - 2)</td></tr> <tr><td>2</td><td>0x14</td><td></td></tr> <tr><td>3</td><td>0x92</td><td>Frame type - 0x92 indicates this will be a data sample</td></tr> <tr><td>4</td><td>0x00</td><td>64-bit Source Address (Serial Number)</td></tr> <tr><td>5</td><td>0x13</td><td>MSB is byte 4, LSB is byte 11</td></tr> <tr><td>6</td><td>0xA2</td><td></td></tr> <tr><td>7</td><td>0x00</td><td></td></tr> <tr><td>8</td><td>0x40</td><td></td></tr> <tr><td>9</td><td>0x77</td><td></td></tr> <tr><td>10</td><td>0x9C</td><td></td></tr> <tr><td>11</td><td>0x49</td><td></td></tr> <tr><td>12</td><td>0x36</td><td>Source Network Address - 16 Bit</td></tr> <tr><td>13</td><td>0x6A</td><td></td></tr> <tr><td>14</td><td>0x01</td><td>Receive Opts. 01=Packet Acknowledged. 02=Broadcast packet</td></tr> <tr><td>15</td><td>0x01</td><td>Number of sample sets. Always set to 1 due to XBEE limitations</td></tr> <tr><td>16</td><td>0x00</td><td>Digital Channel Mask - Indicates which pins are set to DIO</td></tr> <tr><td>17</td><td>0x20</td><td></td></tr> <tr><td>18</td><td>0x01</td><td>Analog Channel Mask - Indicates which pins are set to ADC</td></tr> <tr><td>19</td><td>0x00</td><td>Digital Sample Data (if any) - Reads the same as Digital Mask</td></tr> <tr><td>20</td><td>0x14</td><td></td></tr> <tr><td>21</td><td>0x04</td><td>Analog Sample data (if any)</td></tr> <tr><td>22</td><td>0x25</td><td>There will be two bytes here for every pin set for ADC</td></tr> <tr><td>23</td><td>0xF5</td><td>Checksum(0xFF - the 8 bit sum of the bytes from byte 3 to this byte)</td></tr> </tbody> </table>	Byte	Example	Description	0	0x7e	Start byte - Indicates beginning of data frame	1	0x00	Length - Number of bytes (ChecksumByte# - 1 - 2)	2	0x14		3	0x92	Frame type - 0x92 indicates this will be a data sample	4	0x00	64-bit Source Address (Serial Number)	5	0x13	MSB is byte 4, LSB is byte 11	6	0xA2		7	0x00		8	0x40		9	0x77		10	0x9C		11	0x49		12	0x36	Source Network Address - 16 Bit	13	0x6A		14	0x01	Receive Opts. 01=Packet Acknowledged. 02=Broadcast packet	15	0x01	Number of sample sets. Always set to 1 due to XBEE limitations	16	0x00	Digital Channel Mask - Indicates which pins are set to DIO	17	0x20		18	0x01	Analog Channel Mask - Indicates which pins are set to ADC	19	0x00	Digital Sample Data (if any) - Reads the same as Digital Mask	20	0x14		21	0x04	Analog Sample data (if any)	22	0x25	There will be two bytes here for every pin set for ADC	23	0xF5	Checksum(0xFF - the 8 bit sum of the bytes from byte 3 to this byte)
Byte	Example	Description																																																																										
0	0x7e	Start byte - Indicates beginning of data frame																																																																										
1	0x00	Length - Number of bytes (ChecksumByte# - 1 - 2)																																																																										
2	0x14																																																																											
3	0x92	Frame type - 0x92 indicates this will be a data sample																																																																										
4	0x00	64-bit Source Address (Serial Number)																																																																										
5	0x13	MSB is byte 4, LSB is byte 11																																																																										
6	0xA2																																																																											
7	0x00																																																																											
8	0x40																																																																											
9	0x77																																																																											
10	0x9C																																																																											
11	0x49																																																																											
12	0x36	Source Network Address - 16 Bit																																																																										
13	0x6A																																																																											
14	0x01	Receive Opts. 01=Packet Acknowledged. 02=Broadcast packet																																																																										
15	0x01	Number of sample sets. Always set to 1 due to XBEE limitations																																																																										
16	0x00	Digital Channel Mask - Indicates which pins are set to DIO																																																																										
17	0x20																																																																											
18	0x01	Analog Channel Mask - Indicates which pins are set to ADC																																																																										
19	0x00	Digital Sample Data (if any) - Reads the same as Digital Mask																																																																										
20	0x14																																																																											
21	0x04	Analog Sample data (if any)																																																																										
22	0x25	There will be two bytes here for every pin set for ADC																																																																										
23	0xF5	Checksum(0xFF - the 8 bit sum of the bytes from byte 3 to this byte)																																																																										

Sleep Mode	Endpoints can sleep to save power. An endpoint that only wakes up every 5 minutes to send data may only be awake for 6 seconds a day. SM - 4 = Cyclic sleep SP - Sleep time (up to 28 secs) SN - Number of sleep cycles ST - Time awake	Pin IO Options
		0 - Disabled 1 - N/A 2 - ADC 3 - Digital IN 4 - Digital OUT, LOW 5 - Digital OUT, HIGH
DIO Mask	First Byte n/a n/a n/a D12 D11 D10 n/a n/a Second Byte D7 D6 D5 D4 D3 D2 D1 D0 Example: 0x00 0x0D = 0000 0000 0000 1101 Pins D3, D2 and D0	Notes
Analog Mask	(volt) n/a n/a n/a A3 A2 A1 A0 Example: 0x05 = 0000 0101 = Pin A2 and A0	

ANEXO C. Módulo XBee.



Overview

XBee and XBee-PRO ZB embedded RF modules provide cost-effective wireless connectivity to devices in ZigBee mesh networks. Utilizing the ZigBee PRO Feature Set, these modules are interoperable with other ZigBee devices, including devices from other vendors*.

Products in the XBee family are easy to use. They require no configuration or additional development; users can have their network up and running in a matter of minutes.

Programmable versions of the XBee-PRO ZB module make customizing ZigBee applications easy. Programming directly on the module eliminates the need for a separate processor. Because the wireless software is isolated, applications can be developed with no risk to RF performance or security.

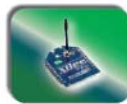
XBee modules are available in a variety of protocols and frequencies. The common hardware footprint shared by Digi's XBee modules means users can substitute one XBee for another with minimal development time and risk.

*Interoperability requires the ZigBee Feature Set or ZigBee PRO Feature Set to be deployed on all devices. Contact Digi Support for details.

Application Highlight



Related Products



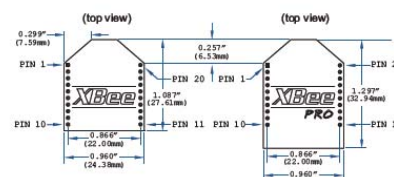
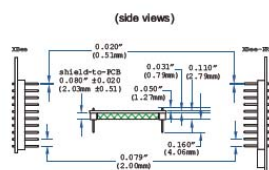
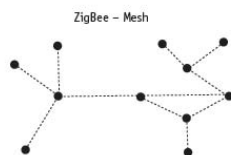
Features/Benefits

- Interoperability with ZigBee compliant devices*
- No configuration needed for out-of-the-box RF communications
- Common XBee footprint for a variety of RF modules
- ZigBee mesh networking protocol
 - Improved data traffic management
 - Remote firmware updates
 - Self-healing and discovery for network stability
- Programmable versions of the XBee-PRO ZB enable custom ZigBee application development
 - 8-bit Freescale™ S08 microprocessor brings intelligence to devices
 - CodeWarrior® development tools for easy customization



Platform	XBee® ZB	XBee-PRO® ZB	Programmable XBee-PRO® ZB
Performance			
RF Data Rate	250 Kbps		
Indoor/Urban Range	133 ft (40 m)	300 ft (90 m)	
Outdoor/RF Line-of-Sight Range	400 ft (120 m)	2 miles (3200 m) / Int'l 5000 ft (1500 m)	
Transmit Power	1.25 mW (+1 dBm) / 2 mW (+3 dBm) boost mode	63 mW (+18 dBm) / Int'l 10 mW (+10 dBm)	
Receiver Sensitivity (1% PER)	-96 dBm in boost mode	-102 dBm	
Features			
Adjustable Power	Yes		
I/O Interface	3.3V CMOS UART, ADC, DIO		3.3V CMOS UART, SPI, I2C, PWM, DIO, ADC
Configuration Method	API or AT commands, local or over-the-air		
Frequency Band	2.4 GHz		
Interference Immunity	DSSS (Direct Sequence Spread Spectrum)		
Serial Data Rate	1200 bps - 1 Mbps		
ADC Inputs	(4) 10-bit ADC inputs		
Digital I/O	10		
Antenna Options	Chip, Wire Whip, U.FL, RPSMA	PCB Embedded Antenna, Wire Whip, U.FL, RPSMA	
Operating Temperature	-40° C to +85° C, 0-95% humidity non-condensing		
Programmability			

Memory	N/A	32 KB Flash / 2 KB RAM	
CPU/Clock Speed	N/A	HCS08 / Up to 50.33 MHz	
Networking & Security			
Encryption	128-bit AES		
Reliable Packet Delivery	Retries/Acknowledgments		
IDs and Channels	PAN ID, 64-bit IEEE MAC, 16 channels	PAN ID, 64-bit IEEE MAC, 15 channels	
Power Requirements			
Supply Voltage	2.1 - 3.6VDC	2.7 - 3.6VDC	
Transmit Current	35 mA / 45 mA boost mode @ 3.3VDC	205 mA	220 mA
Receive Current	38 mA / 40 mA boost mode @ 3.3VDC	47 mA	62 mA
Power-Down Current	<1 uA @ 25° C	3.5 uA @ 25° C	4 uA @ 25° C
Regulatory Approvals			
FCC, IC (North America)	Yes		
ETSI (Europe)	Yes		
C-TICK (Australia)	Yes		
TELEC (Japan)	Yes	Yes (int'l unit only)	



Visit www.digi.com for part numbers.

DIGI SERVICE AND SUPPORT - You can purchase with confidence knowing that Digi is here to support you with expert technical support and a one-year warranty. www.digi.com/support

Digi International
877-912-3444
952-912-3444
info@digi.com

Digi International
France
+33-1-55-61-98-98
www.digi.fr

Digi International
KK
+81-3-5428-0261
www.digi-intl.co.jp

Digi International
(HK) Limited
+852-2833-1008
www.digi.cn

Digi m2m Solutions
India Pvt. Ltd
+91-80-4287-9887
info@digi.com



91001471
D3/511

BUY ONLINE • www.digi.com

© 2008-2011 Digi International Inc.
All rights reserved. Digi, Digi International, the Digi logo, the Making Wireless M2M Easy logo, ConnectPort, XBee and XBee-PRO are trademarks or registered trademarks of Digi International Inc. in the United States and other countries worldwide. All other trademarks are the property of their respective owners. All information provided is subject to change without notice.



ANEXO D. Sonómetro modelo SL1361.

Portable Display LCD Digital Portátil Medidor de Nivel de Sonido de Ruido Decibel Tester 30-130dB Meter w/Monitoreo Logger.

Especificaciones

Descripción del producto:

Este medidor se ofrece con amplio rango de medición (30-130dB). Es portátil y fácil de usar y práctico instrumento para el control de calidad de sonido en la fábrica, oficina, casa, la escuela y el sitio de construcción.

Características:

Amplia gama de medición: 30-130dB.

Respuesta rápida/Lenta y la función de retención Máxima.

Equipado con bolas de esponja.

Auto apagado después de trabajar 2 minutos (encenderá normalmente); si se presiona el botón del interruptor para 3 s, no va a poder adquisitivo si no pulsa el botón de nuevo.

AC/DC salida de señal analógica.

Fuente de alimentación: puerto usb o 4 pilas AA *.

Portátil, fácil de usar y práctico instrumento para el control de calidad de sonido en la fábrica, oficina, casa, la escuela y el sitio de construcción.

Especificaciones:

Rango de medida: 30-130dBA o 35-130dBC

precisión: \pm \$ number db

resolución: \$ number db

Rango De frecuencia: 31.5Hz a 8.5 KHz

Rango de nivel: 30 ~ 80dB, 40 ~ 90dB, 50 ~ 100dB, 60 ~ \$ number db, 70 ~ 120dB, 80 ~ 130dB, 30 ~ 130dB

Ponderación de linealidad: 50dB

Ponderación de frecuencia: A/C

Pantalla Digital: 4 Dígitos

frecuencia de Muestreo: 2 Veces/seg

MAX Hold: " MAX "

sobre la Indicación de: MÁS/MENOS

tiempo Peso: RÁPIDO/LENTO

micrófono: 1/2 pulgadas Electret Micrófono de Condensador

fuentes de Alimentación: 4*1.5 V AA Pilas (no incluidas) o USB 5 V 100mA

Entorno de funcionamiento: 0-40? 10-80% de HUMEDAD RELATIVA

Entorno de almacenamiento: -10 a 50? 10-80% de HUMEDAD RELATIVA

dimensión: 230*70*35mm/9*2.75 * 1.38in (L * W * H)

peso: 269g/9.48 oz

Tamaño del paquete: 29*14*6.5 cm/11.4*5.5 * 2.56in (L * W * H)

Peso del paquete: 459g/16.19 oz

