



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES**  
**Y REDES**

**DISEÑO E IMPLEMENTACIÓN DE UNA RED INALÁMBRICA DE**  
**SENSORES INTELIGENTES PARA EL REGISTRO DE LA**  
**CONTAMINACIÓN ACÚSTICA EN LA ESPOCH BASADO EN**  
**TARJETAS DE DESARROLLO**

Trabajo de Titulación presentado para optar al grado académico de:  
**INGENIERO EN ELECTRÓNICA TELECOMUNICACIONES Y**  
**REDES**

**AUTORES:** HÉCTOR RODRIGO ALTAMIRANO BENALCÁZAR  
AMANDA MADELAINE PUENTE MORENO  
**TUTOR:** ING. JOSÉ ENRIQUE GUERRA SALAZAR

Riobamba – Ecuador

2016

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**  
**ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y**  
**REDES**

El Tribunal del Trabajo de Titulación certifica que: la investigación: DISEÑO E IMPLEMENTACIÓN DE UNA RED INALÁMBRICA DE SENSORES INTELIGENTES PARA EL REGISTRO DE LA CONTAMINACIÓN ACÚSTICA EN LA ESPOCH BASADO EN TARJETAS DE DESARROLLO, de responsabilidad de los señores Héctor Rodrigo Altamirano Benalcázar y Amanda Madelaine Puentes Moreno, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

<b>NOMBRE</b>	<b>FIRMA</b>	<b>FECHA</b>
Dr. Miguel Tasambay PhD. <b>DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA</b>	_____	_____
Ing. Franklin Moreno <b>DIRECTOR ESCUELA DE INGENIERA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES</b>	_____	_____
Ing. José Guerra <b>DIRECTOR DE TRABAJO DE TITULACIÓN</b>	_____	_____
Ing. Wilson Zuñiga <b>MIEMBRO DEL TRIBUNAL</b>	_____	_____

Nosotros, Héctor Rodrigo Altamirano Benalcázar y Amanda Madelaine Puente Moreno, declaramos ser responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Héctor Rodrigo Altamirano Benalcázar

Amanda Madelaine Puente Moreno

## DEDICATORIA

Este trabajo se lo dedico a mi Dios y a la Virgen Santísima por darme las fuerzas necesarias cada día para lograr mis objetivos, para poder llegar a mi meta más anhelada. A mi madre Soledad Benalcázar que ha sido mi apoyo incondicional, mi pañuelo de lágrimas en momentos difíciles, la persona con la que he compartido mis triunfos. A mis hermanos Fernando y Elena que sin el aliento diario de ellos no hubiese podido conseguir nada, a los consejos de mi ñaño que es como mi segundo padre. Gracias a ellos por todo lo que me han sabido brindar y sé que en un futuro seguiré contando incondicionalmente con su apoyo.

Rodrigo

El presente trabajo de titulación se lo dedico con mucho cariño a mi madre, Luz Irene Moreno Guerra quien supo apoyarme de una u otra manera para que salga adelante, su esfuerzo por educarme y sobre todo el que continúe con mis estudios, le estoy muy agradecida por lo que obtener mi título es el mejor regalo que le puedo ofrecer en recompensa de todo lo que ha hecho por mí.

A mi padre Luis Armando Puente Carrasco, que a pesar de las dificultades, me impulsó en mis estudios, a mis hermanos Santiago, Jhoanna y Joel, a mis abuelitos Luz María Guerra y Vicente Moreno, que sin su apoyo no sería posible haber alcanzado una meta más en mi vida.

Además tuve la oportunidad de conocer a personas con las que pude contar a lo largo de mi carrera y que ahora son mis amigos y futuros colegas, ya que compartimos el mismo objetivo de ser ingenieros.

Amanda

## **AGRADECIMIENTO**

Agradecemos primeramente a Dios por habernos permitido cumplir nuestro sueño, a nuestros queridos familiares y amigos por todo su apoyo incondicional brindado a lo largo de nuestra carrera.

De la misma manera, a nuestro segundo hogar la Escuela Superior Politécnica de Chimborazo, así como también a la Escuela de Ingeniería Electrónica en Telecomunicaciones y Redes junto con sus representantes por habernos dado la oportunidad de culminar nuestros estudios profesionales.

A nuestros profesores por la paciencia y desempeño que nos han brindado ya que la mayor parte de nuestros conocimientos alcanzados se los debemos a ellos. Y sobre todo a nuestro docente tutor, el Ing. José Enrique Guerra quien supo guiarnos para cumplir una meta más en nuestras vidas, su apoyo incondicional y su ayuda han hecho posible la culminación con éxito del presente trabajo de titulación.

Rodrigo Altamirano y Amanda Puente

## TABLA DE CONTENIDO

<b>PORTADA</b> .....	<b>i</b>
<b>FIRMA DE RESPONSABILIDADES</b> .....	<b>ii</b>
<b>RESPONSABILIDAD DE AUTORES</b> .....	<b>iii</b>
<b>DEDICATORIA</b> .....	<b>iv</b>
<b>AGRADECIMIENTO</b> .....	<b>v</b>
<b>INDICE DE TABLAS</b> .....	<b>x</b>
<b>INDICE DE FIGURAS</b> .....	<b>xi</b>
<b>INDICE DE ABREVIATURAS</b> .....	<b>xiii</b>
<b>RESUMEN</b> .....	<b>xv</b>
<b>SUMMARY</b> .....	<b>xvi</b>
<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO I</b>	
<b>1. MARCO TEÓRICO</b> .....	<b>7</b>
1.1 Introducción .....	7
1.2 Red de Sensores Inalámbricos .....	7
1.2.1 Componentes de una red de sensores inalámbricos .....	8
1.2.1.1 Nodo Sensor .....	8
1.2.1.2 Gateway o Puerta de Enlace.....	9
1.2.1.3 Estación Base .....	10
1.2.2 Topologías en una red de sensores inalámbricos .....	10
1.2.2.1 Topología de red en estrella .....	10
1.2.2.2 Topología de red en malla.....	11
1.2.2.3 Topología Híbrida Estrella – Malla.....	12
1.2.3 Tecnologías para transmisión de datos .....	12
1.2.4 Parámetros a considerar en redes de sensores inalámbricos .....	14
1.3 Tarjetas de desarrollo aplicadas a redes de sensores inalámbricos .....	15

1.3.1	Arduino .....	166
1.3.2	Intel Galileo.....	17
1.3.3	Raspberry Pi.....	18
1.3.4	Libelium – Waspnote .....	18
1.4	Contaminación Acústica .....	20
1.4.1	Definición.....	200
1.4.2	Ruido Ambiental.....	221
1.4.2.1	Clasificación del ruido Ambiental.....	221
1.4.2.2	Decibelio (dB).....	22
1.4.3	Niveles y Factores sobre la Contaminación Acústica .....	22
1.4.4	Límites Permisibles de Niveles de Ruido Ambiente para Fuentes Fijas y Fuentes Móviles, y para Vibraciones .....	23
1.4.4.1	Objetivo Anexo 5 del Libro VI del TULSMA.....	24
1.4.4.2	Clasificación de los límites máximos permisibles de niveles de ruido ambiente.....	24
1.4.5	Equipos de medición de ruido.....	27
1.4.6	Trabajados realizados en Riobamba sobre supervisión de contaminación acústica....	28
 <b>CAPÍTULO II</b>		
<b>2.</b>	<b>DISEÑO DE <i>HARWARE</i> Y <i>SOFTWARE</i> DEL PROTOTIPO DEL SISTEMA ELECTRÓNICO.....</b>	<b>29</b>
2.1	Introducción .....	29
2.2	Requerimientos <i>hardware</i> del SISCA.....	29
2.2.1	Concepción de la arquitectura general del sistema .....	30
2.2.2	Diseño de la Arquitectura del SISCA .....	31
2.2.2.1	Módulo de Supervisión .....	31
2.2.2.2	Módulo de Aviso.....	32
2.2.2.3	Módulo de Almacenamiento .....	33
2.2.3	Selección de elementos que conforman el SISCA para la ESPOCH.....	34
2.2.3.1	Tarjetas de Desarrollo .....	34
2.2.3.2	Sensor de Ruido (Sound Detector).....	36

2.2.3.3	Equipos de transmisión .....	37
2.2.3.4	Reloj en Tiempo Real RTC (DS1307) .....	38
2.2.3.5	Shield micro SD .....	40
2.2.3.6	Equipos de Visualización .....	40
2.2.3.7	Módulos Relé para Arduino .....	41
2.2.4	Esquema de conexión del Sistema Inalámbrico de Sensores .....	41
2.2.4.1	Diagrama de conexión del Módulo de Supervisión .....	42
2.2.4.2	Diagrama de conexión del Módulo de Aviso .....	443
2.2.4.3	Diagrama de conexión del Módulo de Almacenamiento .....	44
2.2.5	Alimentación de la Red Inalámbrica de Sensores de contaminación acústica .....	45
2.3	Requerimientos Software de la Red Inalámbrica de Sensores de contaminación acústica .....	46
2.3.1	Software de desarrollo utilizado para la programación del sistema .....	47
2.3.1.1	Software para la programación de las tarjetas de desarrollo Arduino .....	47
2.3.1.2	Software LabVIEW .....	48
2.3.2	Programaciones realizadas en cada módulo del sistema inalámbrico .....	48
2.3.2.1	Diagrama de Flujo del Módulo de Supervisión .....	48
2.3.2.2	Bibliotecas usadas en la programación del Módulo de Supervisión .....	500
2.3.2.3	Funciones que se incluyen en el sketch del Módulo de Supervisión .....	511
2.3.2.4	Diagrama de Flujo del Módulo de Aviso .....	52
2.3.2.5	Bibliotecas usadas en la programación del Módulo de Aviso .....	54
2.3.2.6	Funciones que se incluyen en el sketch del Módulo de Aviso .....	54
2.3.2.7	Diagrama de Flujo del Módulo de Almacenamiento .....	55
2.3.2.8	Bibliotecas usadas en la programación del Módulo de Almacenamiento .....	56
2.3.2.9	Funciones que se incluyen en el sketch del Módulo de Almacenamiento .....	57
2.3.3	Software para la presentación de información en modo gráfico del SISCA .....	58
<b>CAPÍTULO III</b>		
<b>3.</b>	<b>ANÁLISIS Y RESULTADOS .....</b>	<b>62</b>

3.1	Introducción .....	62
3.2	Pruebas del Módulo de Supervisión.....	62
3.2.1	Pruebas de las mediciones realizadas por el Módulo de Supervisión .....	62
3.2.2	Repetitividad del Módulo de Supervisión.....	64
3.2.3	Caracterización de las comunicaciones entre dispositivos del Módulo de Supervisión .....	66
5		
3.2.4	Pruebas de funcionamiento del Módulo de Aviso .....	70
3.2.5	Pruebas de funcionamiento del Módulo de Almacenamiento.....	71
3.3	Funcionamiento general SISCA desarrollado para la ESPOCH. ....	72
3.4	Análisis económico de la implementación del prototipo del Sistema Inalámbrico de Sensores Inteligentes.....	73
	<b>CONCLUSIONES.....</b>	<b>74</b>
	<b>RECOMENDACIONES.....</b>	<b>76</b>
	<b>BIBLIOGRAFÍA</b>	
	<b>ANEXOS</b>	

## ÍNDICE DE TABLAS

<b>Tabla 1-1</b>	Comparativa de tarjetas de desarrollo.....	19
<b>Tabla 1-2</b>	Clasificación del ruido.....	221
<b>Tabla 1-3</b>	Ejemplos de valores de niveles acústicos por fuentes emisoras .....	22
<b>Tabla 1-4</b>	Niveles máximos de ruido permisibles según uso del suelo.....	24
<b>Tabla 1-5</b>	Niveles máximos de ruido para vehículos automotores.....	26
<b>Tabla 1-6</b>	Límite de transmisión de vibraciones .....	27
<b>Tabla 3-1</b>	Mediciones realizadas con el sensor del sistema inalámbrico y con el sonómetro digital.....	64
<b>Tabla 3-2</b>	Tabla de repetitividad de los valores del Sistema Inalámbrico.....	665
<b>Tabla 3-3</b>	Presupuesto detallando cada elemento que conforman los módulos del SISCA ....	73

## ÍNDICE DE FIGURAS

<b>Figura 1-1</b>	Elementos de la Red de Sensores Inalámbricos .....	8
<b>Figura 1-2</b>	Arquitectura del Nodo Sensor .....	9
<b>Figura 1-3</b>	Ejemplos de sensores .....	9
<b>Figura 1-4</b>	Topología de red estrella .....	11
<b>Figura 1-5</b>	Topología de red malla.....	11
<b>Figura 1-6</b>	Topología de red híbrida .....	12
<b>Figura 1-7</b>	Actuales tecnologías inalámbricas .....	13
<b>Figura 1-8</b>	Tarjetas Arduino.....	17
<b>Figura 1-9</b>	Tarjeta Intel Galileo .....	17
<b>Figura 1-10</b>	Modelos de tarjetas Raspberry Pi existentes en el mercado.....	18
<b>Figura 1-11</b>	Tarjeta Waspote .....	19
<b>Figura 1-12</b>	Equipos de medición de ruido más comunes .....	28
<b>Figura 2-1</b>	Arquitectura general del sistema inalámbrico. ....	30
<b>Figura 2-2</b>	Diagrama de bloques del Módulo de Supervisión de contaminación acústica... 332	
<b>Figura 2-3</b>	Diagrama de bloques del Módulo de Aviso del sistema sensorial. ....	33
<b>Figura 2-4</b>	Módulo de Almacenamiento del sistema inalámbrico. ....	34
<b>Figura 2-5</b>	Tarjeta de desarrollo Arduino MEGA.....	35
<b>Figura 2-6</b>	Arduino UNO.....	36
<b>Figura 2-7</b>	Sensor de sonido Sparkfun.....	37
<b>Figura 2-8</b>	Módulos de transmisión de Radio Frecuencia RF 433 MHz. ....	38
<b>Figura 2-9</b>	Módulo reloj de tiempo real RTC DS1307 I2C. ....	39
<b>Figura 2-10</b>	Shield micro SD para Arduino. ....	40
<b>Figura 2-11</b>	Pantalla de visualización .....	41
<b>Figura 2-12</b>	Módulo Relé para Arduino.....	41
<b>Figura 2-13</b>	Esquema del Módulo de supervisión.....	443
<b>Figura 2-14</b>	Esquema del Módulo de Aviso. ....	44
<b>Figura 2-15</b>	Esquema del Módulo de Almacenamiento.....	45

<b>Figura 2-16</b>	Batería recargable seca de 6 V y 2.8 A .....	45
<b>Figura 2-17</b>	Diagrama de flujo del programa para el Módulo de Supervisión. ....	50
<b>Figura 2-18</b>	Estructura de la cadena ASCII para ser enviada por RF y por puerto USB. ....	52
<b>Figura 2-19</b>	Diagrama de flujo del programa para el Módulo de Aviso. ....	554
<b>Figura 2-20</b>	Diagrama de flujo del programa para el Módulo de Almacenamiento. ....	56
<b>Figura 2-21</b>	Información almacenada en el archivo creado en la memoria micro SD. ....	58
<b>Figura 2-22</b>	Encabezado del archivo. ....	58
<b>Figura 2-23</b>	Panel frontal de la herramienta virtual desarrollada en LabVIEW. ....	59
<b>Figura 2-24</b>	Aplicación Data Dashboard al supervisando las variables globales de una herramienta virtual. ....	59
<b>Figura 2-25</b>	Diagrama de bloques de la herramienta virtual implementada. ....	61
<b>Figura 3-1</b>	Prototipo del Módulo de Supervisión, vista interna (izquierda) y vista externa (derecha) .....	63
<b>Figura 3-2</b>	Sonómetro Sound Level Meter .....	63
<b>Figura 3-3</b>	Proceso de ajuste de Fecha y Hora en pantalla Serial y en pantalla LCD. ....	66
<b>Figura 3-4</b>	Proceso de identificación del Nodo en pantalla Serial y en pantalla LCD. ....	66
<b>Figura 3-5</b>	Valores censados mostrados en la pantalla Serial (arriba) y Software LabVIEW (abajo) .....	67
<b>Figura 3-6</b>	Archivo visualizado en Microsoft Excel 2013. ....	68
<b>Figura 3-7</b>	Variables globales de la aplicación Data Dashboard .....	68
<b>Figura 3-8</b>	Cadena transmitida por el módulo supervisor, la información recibida por el módulo de almacenamiento y por el módulo de aviso .....	69
<b>Figura 3-9</b>	Diseño de antenas utilizadas en el Módulo de almacenamiento (izquierda) y en el módulo de supervisión (derecha). ....	70
<b>Figura 3-10</b>	Funcionamiento del módulo de supervisión y el vúmetro. ....	71
<b>Figura 3-11</b>	Información almacenada en el archivo creado por el módulo de almacenamiento .....	72
<b>Figura 3-12</b>	Módulo del SISCA funcionando. ....	72

## ÍNDICE DE ABREVIATURAS

<b>CPU</b>	Central Processing Unit (Unidad Central de Proceso)
<b>SD</b>	Secure Digital (Seguro Digital)
<b>USB</b>	Universal Serial Bus (Bus Universal en Serie)
<b>ESPOCH</b>	Escuela Superior Politécnica de Chimborazo
<b>PIC</b>	Peripheral Interface Controller (Control de Interfaz Periférico)
<b>WSN</b>	Wireless Sensor Network (Red Inalámbrica de Sensores)
<b>DSN</b>	Distributed Sensor Network (Red de Sensores Distribuida)
<b>DARPA</b>	Defense Advanced Research Projects Agency (Agencia de Defensa de Proyectos de Investigación Avanzada)
<b>SOSUS</b>	Sound Surveillance System (Sistema de Vigilancia de Sonido)
<b>INEC</b>	Instituto Nacional de Estadísticas y Censos
<b>OMS</b>	Organización Mundial de la Salud
<b>TCP</b>	Transmission Control Protocol (Protocolo de Control de Transmisión)
<b>IP</b>	Internet Protocol (Protocolo de Internet)
<b>TDT</b>	Televisión Digital Terrestre
<b>RF</b>	Radio Frecuencia
<b>WPAN</b>	Wireless Personal Area Network (Red Inalámbrica de Area Personal)
<b>WLAN</b>	Wireless Local Area Network (Red Inalámbrica de Area Local)
<b>IDE</b>	Integrated Development Environment (Entorno de Desarrollo Integrado)
<b>IoT</b>	Internet of Things (Internet de las cosas)
<b>GSM</b>	Global System for Mobile Communications (Sistema Global para las Comunicaciones Móviles)
<b>GPS</b>	Global Position System (Sistema de Posicionamiento Global)
<b>GPRS</b>	General Packet Radio Service (Servicio General de Paquetes Vía Radio)
<b>TULSMA</b>	Texto Unificado de Legislación Secundaria del Ministerio de Ambiente
<b>IEC</b>	International Electrotechnical Commission (Comisión Electrotécnica Internacional)

<b>SISCA</b>	Sistema Inalámbrico de Supervisión de Contaminación Acústica
<b>LCD</b>	Liquid Crystal Display (Pantalla de Cristal Líquido)
<b>PWM</b>	Pulse-Width Modulation (Modulación por Ancho de Pulsos)
<b>UART</b>	Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal)
<b>ICSP</b>	In-Circuit Serial Programming (Programación Serial en Circuito)
<b>AC</b>	Alternating Current (Corriente Alterna)
<b>DC</b>	Direct Current (Corriente Directa)
<b>EEPROM</b>	Electrically Erasable Programmable Read-Only Memory (ROM Programable y Borrable Eléctricamente)
<b>SRAM</b>	Static Random Access Memory (Memoria Estática de Acceso Aleatorio)
<b>I2C</b>	Inter-Integrated Circuit (I-Cuadrado-C)
<b>MISO</b>	Master Input Slave Output (Salida de datos del Slave y entrada al Master)
<b>MOSI</b>	Master Output Slave Input (Salida de datos del Master y entrada de datos al Slave)
<b>ASCII</b>	American Standard Code for Information Interchange (Código Estándar Estadounidense para el Intercambio de Información)
<b>SPI</b>	Serial Peripheral Interface (Interfaz Periférica Serial)
<b>RTC</b>	Real Time Clock (Reloj de Tiempo Real)
<b>VSPE</b>	Virtual Serial Port Emulator (Emulador Virtula de Puertos Seriales)

## RESUMEN

El diseño e implementación de una red inalámbrica de sensores inteligentes para el registro de la contaminación acústica en la Escuela Superior Politécnica de Chimborazo consiste en un sistema en tiempo real de sensores inalámbricos basados en tarjetas de desarrollo. Los dispositivos han sido rigurosamente estudiados según la compatibilidad con sensores, transmisores - receptores inalámbricos y señales generadas por circuitos externos; sus costos, la factibilidad y la facilidad de instalación, como resultado se eligió la tarjeta de desarrollo Arduino. El diseño e implementación se basó en una red para registrar niveles acústicos presentes en zonas con mayor actividad humana, medidos en decibelios (dB) mediante un sensor de ruido, ideal para trabajos que identifican niveles de contaminación acústica que provoca afecciones a la salud. La Facultad de Informática y Electrónica fue la zona elegida para medir las variaciones de ruido, ubicando tres módulos: de supervisión, de almacenamiento y de aviso. El tiempo medido fue continuo durante siete días calendario. La información fue almacenada en una tarjeta de memoria micro SD y por medio del puerto USB de la tarjeta Arduino, se visualizan numérica y gráficamente los valores actuales, máximos y mínimos en una herramienta virtual desarrollada en el *software* LabVIEW. Trabaja conjuntamente con una aplicación móvil Data Dashboard, que mantiene una comunicación inalámbrica a través de la red Wi-Fi. Se comprobó que el sistema electrónico de supervisión para contaminación acústica presenta un error absoluto máximo de  $\pm 1,2$  dB y una inestabilidad del 2%, demostrando que existe un rango de operación moderado y que el sistema es confiable. Los resultados representan que la red inalámbrica implementada con el sistema electrónico es funcional, con un alcance para la transmisión de 98 metros. Por las características de diseño representa una herramienta de supervisión y recopilación de información recomendable para estudios que permitan disminuir los niveles de contaminación acústica en pos de prevenir afecciones en la población. Se recomienda a las autoridades de la Escuela Superior Politécnica de Chimborazo que sea implementado el sistema diseñado para dar aviso al personal politécnico sobre los niveles de ruido al que está expuesto en sus actividades diarias.

**Palabras Clave:** <RED DE SENSORES INALÁMBRICOS>, <DISPOSITIVO SENSOR [AUDIO]>, <CONTAMINACIÓN ACÚSTICA>, <RUIDO AMBIENTAL>, <FACTORES CONTAMINANTES>, <TARJETAS DE DESARROLLO [ARDUINO]>.

## SUMMARY

The design and implementation of a wireless network of intelligent sensors for recording noise contamination at the Polytechnic School of Chimborazo is a real-time system based on wireless sensor development boards. The devices have been rigorously studied according to the compatibility with sensors, wireless transceivers and signals generated by external circuits; costs, feasibility and ease of installation, as a result the Arduino development board was elected. The design and implementation was based on a network to record noise levels shown in areas with increased human activity, measured in decibels (dB) using a noise sensor ideal for work that identified levels of noise pollution which causes health disorders. The Faculty of Computer Science and Electronics was chosen to measure changes in noise area, placing three modules: monitoring, storage and warning. The time measured was continued for seven calendar days. The information was store on a micro SD memory card and through the USB port of the Arduino board, they are displayed numerically and graphically the current, maximum and minimum values in a virtual tool developed in the LabVIEW software. It works with a mobile application Data Dashboard, which maintains a wireless communication through Wi-Fi network. It was found that the electronic monitoring system for noise contamination has a maximum absolute error of  $\pm 1,2$  dB and instability 2%, showing that there is a moderate range of operation and that the system is reliable. The results represent the wireless network implemented with the electronic system is functional with a range for the transmission of 98 meters. By design features it represents a monitoring tool and recommended gathering information for studies to reduce levels of noise contamination after foresee conditions in the population. It is recommended to the authorities of the Polytechnic School of Chimborazo which is implemented the system designed to alert the polytechnic staff on noise levels to which it is exposed in their daily activities.

**Keywords:** <WIRELESS SENSOR NETWORK>, <SENSOR DEVICE [AUDIO]>, <NOISE CONTAMINATION>, <ENVIRONMENTAL NOISE>, <CONTAMINANTS FACTS>, <DEVELOPMENT BOARDS [ARDUINO]>.

## **INTRODUCCIÓN**

### **ANTECEDENTES**

En los últimos años la contaminación acústica es un problema en la mayoría de las ciudades del mundo. Se genera por el ruido producido en el proceso de construcción de edificaciones e infraestructuras, por el incremento desmedido del parque automotriz, la utilización de bocinas, motores o el tráfico. Los ruidos son considerados como contaminación acústica por ser molestos y en algunos casos causar enfermedades al ser humano, aparecieron en las áreas urbanas y actualmente en las áreas rurales debido al desarrollo tecnológico y su uso en las labores diarias.

Un estudio desarrollado en la ciudad de Riobamba, entre octubre del 2014 y enero del 2015 por la Universidad Nacional de Chimborazo en conjunto con la Dirección de Gestión Ambiental de la municipalidad, toma mediciones de niveles acústicos los días lunes, miércoles y viernes en el horario de 12:30 a 14:00 y de 16:00 a 17:30. Se obtuvieron valores promedios de 85 dB, lo que comprueba un alto nivel de ruido en ese periodo de tiempo (Razo & Maza, 2015).

La investigación no analiza la presencia de ruido en forma continua y sus variaciones, se evidencia la necesidad de extender el área y el tiempo de supervisión para proporcionar una información pormenorizada, completa y actualizada para la toma de decisiones en pos de prevenir su aumento; se conoce que el incremento del ruido o su exposición constante provocan afecciones graves en la salud del ser humano, causando problemas en el desarrollo normal de sus actividades, estrés y la pérdida total o parcial de audición.

Considerando que en la actualidad existen muchas herramientas que permiten desarrollar sistemas electrónicos que generan y transmiten su información en forma inalámbrica, así como sensores inteligentes y tarjetas de desarrollo. Aparecieron con la llegada del Intel 4004, que fue el segundo CPU completo de un solo chip; posteriormente surgió una unidad completa de microcontrolador denominada PIC.

Finalmente se dio paso a las plataformas de electrónica abierta Arduino, Raspberry Pi, Galileo, entre otras, que en su mayoría son consideradas computadoras de una sola placa completamente funcionales. Son empleadas en diseños de prototipos electrónicos basados en Redes de Sensores Inalámbricos, es la tendencia más actual para desplegar la utilización de recursos en tamaño reducido a bajo costo.

Con la tecnología WSN (*Wireless Sensor Network por sus siglas en inglés*) se han realizado varios estudios de tipo militar, el proyecto DSN (Distributed Sensor Networks por sus siglas en inglés) implementado en 1980 en Estados Unidos. Es aplicado en DARPA (Defense Advanced Research Projects Agency por sus siglas en inglés) y en el proyecto SOSUS (Sound Surveillance System por sus siglas en inglés) que consistía en monitorear barcos enemigos con sensores de sonido ubicados en boyas en el año 1949 durante la guerra fría.

El proyecto fue considerado como la primera red de sensores en el mundo desarrollada para supervisión y control. Actualmente, el concepto de redes inalámbricas de sensores es desarrollado en múltiples aplicaciones en distintos campos como entornos industriales, domótica, entornos militares, detección ambiental, entre otros.

Con pequeños pero eficientes equipos de control, basados en protocolos inalámbricos de bajo consumo, se ha hecho viable la supervisión en tiempo real de los cambios en la mayoría de los parámetros físicos.

Entre las investigaciones realizadas en el país, se cuenta con la tesis de grado desarrollada en la Escuela Superior Politécnica del Ejército, en la ciudad de Latacunga, que implementa un sistema electrónico de medición de gases y contaminación acústica, visualizando únicamente los datos actuales sin probabilidad de almacenarlos e identificar los valores máximos y mínimos, así como emitir mensajes de alerta cuando se supera los niveles críticos (Guagchinga & Guamaní, 2012).

Por tal motivo, el presente trabajo de investigación busca estructurar un sistema electrónico a bajo costo, que basado en una red de sensores inalámbricos supervise, informe y suministre datos en tiempo real de los niveles de contaminación acústica actuales, máximos y mínimos presentes por sectores en una localidad.

En busca de proveer información relevante para la elaboración de mapas sectorizados de contaminación acústica, así como para la toma de decisiones que prevengan de futuras afecciones a los habitantes de la localidad causada por este tipo de contaminación.

## **FORMULACIÓN DEL PROBLEMA**

¿Es necesaria la implementación del sistema electrónico por medio de módulos que realicen acciones específicas que permitan registrar los niveles de contaminación de ruido como herramienta de ayuda para la toma de decisiones para evitar problemas de salud en habitantes de una localidad?

## **SISTEMATIZACIÓN DEL PROBLEMA**

- ¿Qué relación tiene el ruido con las afecciones a la salud en los ciudadanos de una localidad?
- ¿Qué consecuencias tiene el ser humano al estar expuesto a niveles altos de ruido durante sus actividades diarias?
- ¿Se puede dar información a las personas sobre los niveles de ruido que están presentes en los sectores de una localidad?
- ¿Es posible la creación de un sistema de supervisión de contaminación de ruido para la ESPOCH?

## **JUSTIFICACIÓN DEL TRABAJO DE TITULACIÓN**

### **Justificación teórica**

La necesidad de tener un sistema de supervisión y aviso sobre los niveles de contaminación acústica para los habitantes o transeúntes de una localidad es necesario, ya que en Ecuador no se le da la debida importancia, como en otros países, a los efectos nocivos provocados por los diferentes niveles de ruido.

Según los datos registrados en el Instituto Nacional de Estadísticas y Censos (INEC), en el 2014, el parque automotriz tuvo un crecimiento del 2.03% a nivel nacional, en relación a los 1.717.886

vehículos que existían en el 2013, de los cuales 43.511 unidades se encontraban en la provincia de Chimborazo.

En el mismo año se registró un total de 532 construcciones en la provincia, incrementándose a 1334 en el 2014. Estos datos revelan el creciente incremento en edificaciones y el parque automotor lo que influye en el crecimiento del índice de nivel de ruido ambiental en los sectores de la localidad.

Según la encuesta realizada por la Dirección de Gestión Ambiental en el 2015 en la ciudad de Riobamba a sus habitantes, sobre las afecciones por causa del incremento en los niveles de ruido, se obtuvo que un 35% sufren dolores de cabeza, un 27% sufre de estrés, el 21% presenta mal humor, el 5% cansancio, el 4% pérdida de audición, el 3% sufre insomnio y un 5% ansiedad, nerviosismo, vértigo y otras dolencias.

En nuestra sociedad este tipo de contaminación aún no es considerada importante y pasa desapercibida. No existen estudios pormenorizados y duraderos sobre la evolución de los niveles de ruido en la ciudad, así como campañas o acciones que indiquen los efectos en la salud que ocasiona la exposición a altos niveles de ruido en forma prolongada o periódica.

El proyecto pretende desarrollar un sistema electrónico que supervise, informe, transmita y registre en tiempo real la evolución de los niveles de ruido existentes en sectores de una localidad. La información se registra en una base de datos que permita gestionar los niveles de ruido y determinar el grado de peligro que éstos representan.

Permite dar alerta a los habitantes de un sector de la localidad por medio de un sistema electrónico de aviso. La información que se registra constituye en una herramienta de ayuda fundamental para futuros estudios que ayuden en la toma de decisiones en pos de disminuir los niveles de contaminación acústica y prevenir afecciones en la población.

El proyecto se centra en la implementación de un prototipo para la supervisión del ruido que se evaluó en los predios de la Escuela Superior Politécnica de Chimborazo y que por sus características de diseño puede ser aplicado en distintas localidades.

## **Justificación aplicativa**

En el estudio realizado por la Universidad Nacional de Chimborazo y la Dirección de Gestión Ambiental de Riobamba se obtuvo un incremento de niveles de ruido en un 0.6% anual con un valor promedio de 85 dB (Razo & Maza, 2015), lo cual es preocupante ya que según la OMS (Organización Mundial de la Salud) para que el ser humano no presente afecciones por causa del ruido, el límite máximo es 65 dB.

Si se supera este valor por más de 30 minutos se presentan complicaciones en su salud como estrés, pérdida de sueño, cansancio, hipertensión, ansiedad, dolores de cabeza, problemas digestivos, y si el ruido se encuentra en niveles extremos, puede provocar la pérdida total o parcial de la audición.

Considerando la necesidad de que existan dispositivos que permitan detectar la contaminación acústica en una localidad, se debe implementar un sistema electrónico que suministre información y genere alertas acerca del entorno por el cual se transita.

Estos sistemas pueden hacer uso de Redes de Sensores Inteligentes con comunicación inalámbrica, que reduce costos y facilita la instalación. Estas redes son escalables y se basan en nodos capaces de obtener información de su entorno, procesarla localmente y comunicarla a largas distancias.

El trabajo con Redes de Sensores Inalámbricos requiere de un control por medio de un dispositivo para determinar la forma de funcionamiento y las acciones a realizar, todo esto se puede lograr con el uso de tarjetas de desarrollo ya que la compatibilidad con dispositivos dentro de una red de sensores no es un problema.

La integración de estas tecnologías hace viable desarrollar un sistema de supervisión en tiempo real de la contaminación acústica que existe en una localidad independientemente de su área, para determinar niveles máximos, mínimos y actuales de las condiciones del ruido, además de permitir que la información sea visible y esté al alcance de sus habitantes.

## **OBJETIVOS**

### OBJETIVOS GENERALES

Diseñar e implementar una red inalámbrica de sensores inteligentes para el registro de la contaminación acústica en la ESPOCH basado en tarjetas de desarrollo.

### OBJETIVOS ESPECÍFICOS

- Analizar las características, tipos y niveles de potencia de contaminación acústica y los efectos en la salud humana.
- Investigar y seleccionar los sensores, tecnologías de comunicación inalámbrica y tarjetas de desarrollo que se ajusten a los requerimientos del sistema electrónico de supervisión acústica a implementar.
- Diseñar el sistema electrónico de supervisión acústica basado en módulos inalámbricos que ejecuten acciones específicas de recolección, aviso y registro de información.
- Crear la red de sensores inalámbrica requerida por el sistema electrónico de supervisión acústica para la ESPOCH.
- Implementar los diferentes módulos que constituirán el sistema electrónico de supervisión acústica.
- Desarrollar un sistema de alerta de los niveles de ruido existente en el sector.
- Realizar la validación del prototipo implementado comparando con un equipo comercial denominado sonómetro.

# CAPÍTULO I

## 1. MARCO TEÓRICO

### 1.1 Introducción

En este capítulo se analizan las características de las redes de sensores inalámbricos, tecnologías existentes para comunicación sin cable, que trabajan en conjunto con tarjetas de desarrollo para controlar el envío y recepción de datos. Además se investiga sobre la contaminación acústica, cómo afecta a la salud de los seres humanos y los niveles permisibles de ruido en el país.

### 1.2 Red de Sensores Inalámbricos

Las WSN (*Wireless Sensor Networks, por sus siglas en inglés*), se basan en nodos capaces de adquirir información de su entorno. Están conformados por pequeños dispositivos distribuidos físicamente para transmitir inalámbricamente, por radio, infrarrojo u otros medios.

Permite procesar y almacenar la información, manteniendo así la red en funcionamiento, cada nodo de la red contiene sensores que ayudan a recolectar la información en una estación central, miden ciertos parámetros para los que han sido creados, como por ejemplo sonido, temperatura, movimiento, contaminantes del medio, entre otros.

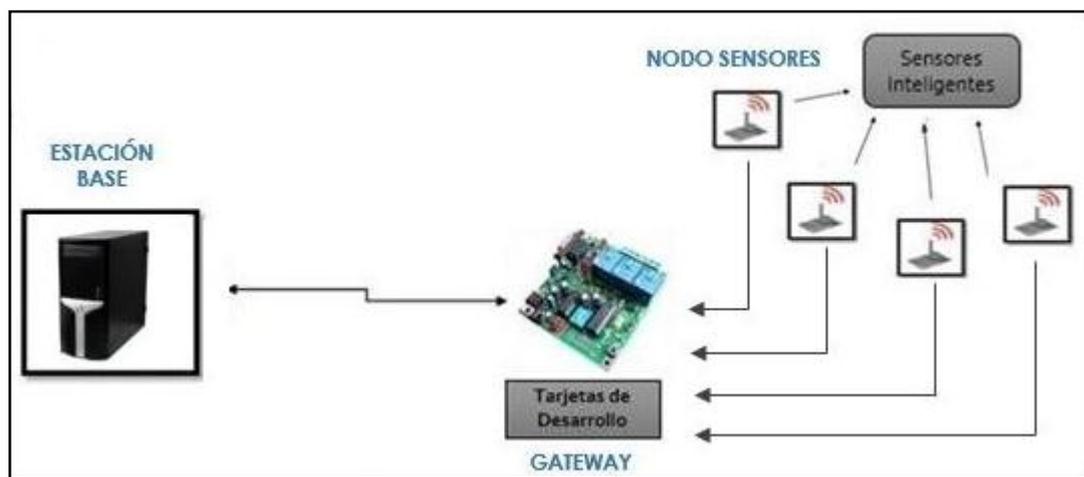
La ventaja de estas redes es que si un nodo deja de trabajar, no interrumpe su funcionamiento gracias a las topologías que presentan, ya que el objetivo se basa en que los nodos se comuniquen entre sí. Son redes inteligentes debido a que se caracterizan por la manera de resolver problemas sin la intervención de personas, ya que son desarrolladas para cumplir una aplicación en concreto (Flores, 2012).

### 1.2.1 Componentes de una red de sensores inalámbricos

Para que una red de sensores inalámbricos esté en funcionamiento es importante considerar los elementos que la conforman. Como se mencionó, una red de este tipo está compuesta por pequeños dispositivos dispersos que buscan comunicarse entre sí para encaminar los datos hacia un recolector (Fernández, et al. 2009).

Esto permite formar redes ad-hoc, ya que carecen de un nodo central, convirtiéndose en una red enteramente capaz de resolver inconvenientes por sí misma, su limitación se centra en el tiempo de operación, que depende del tiempo de vida de la fuente adjunta (Fernández, et al. 2009).

Los elementos de una red de sensores inalámbricos se pueden observar en la *Figura 1-1*, en ella se aprecia el nodo sensor, el gateway o puerta de enlace y la estación base.



**Figura 1-1** Elementos de la Red de Sensores Inalámbricos

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

#### 1.2.1.1 Nodo Sensor

Es considerado como un sistema de adquisición de datos ya que está conformado por dispositivos autónomos que permiten captar la información del medio ambiente, procesarla y transmitirla de forma inalámbrica a distintos usuarios. En su arquitectura a más de presentar elementos sensores para su correcto funcionamiento, incluye un procesador, una unidad de comunicación y una fuente de energía como se muestra en la *Figura 1-2*.



**Figura 1-2** Arquitectura del Nodo Sensor  
**Fuente:** PICO VALENCIA, Pablo, 2012

También conocido como mota, nombre asignado a este tipo de dispositivos por ser pequeños y ligeros. Los sensores que lo conforman dependen exclusivamente de la aplicación a la que se haya asignado, pueden ser fijos o móviles. Actualmente existen en el mercado placas que contienen sensores con ciertos parámetros tales como sensores de luz, humedad del suelo y aire, radiación, temperatura, sonido, velocidad, entre otros. Algunos ejemplos de sensores se observan en la *Figura 1-3* (Barcell, 2010).



**Figura 1-3** Ejemplos de sensores  
**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 1.2.1.2 Gateway o Puerta de Enlace

Dentro de una red de sensores inalámbricos, este elemento es considerado como un nodo sensor, debido a la similitud que presenta en cuanto a la arquitectura de los nodos, con la diferencia de que no contienen sensores. La característica principal que posee el *gateway* es permitir la

interconexión con los distintos tipos de dispositivos que conforman la red; no necesariamente de la misma clase, entre redes de sensores inteligentes y redes de datos TCP/IP (Flores, 2012).

### *1.2.1.3 Estación Base*

Es el elemento principal en las redes de sensores inalámbricos, ya que se encarga de recolectar toda la información que ha sido detectada gracias a las capacidades que posee, siendo superior a un nodo de sensores. La estación base puede ser un ordenador, con el fin de almacenar dicha información en una base de datos, la cual constituirá una herramienta de ayuda para futuros estudios de ser el caso.

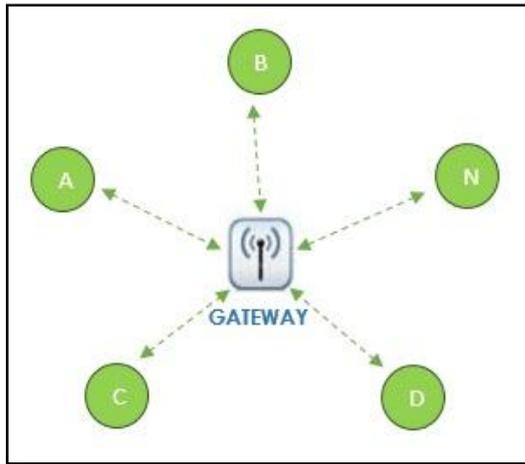
## *1.2.2 Topologías en una red de sensores inalámbricos*

Una topología de red se refiere específicamente a los dispositivos conectados físicamente dentro de la red. Existe una variedad de topologías que se utilizan en el diseño de redes de sensores inalámbricos como son topologías en estrella, malla e híbrida estrella-malla. La comunicación entre cada uno de los elementos, así como la colocación de los mismos dependerán de la aplicación requerida por la red, lo que impulsa a realizar un análisis previo para identificar que topología se ajusta a los requerimientos de la red (Burbano, 2014).

### *1.2.2.1 Topología de red en estrella*

Se caracterizan por la forma de conexión directa que tienen los nodos sensores con el *gateway*, este último recoge los datos que envían dichos nodos que no tienen la capacidad de “intercambiar información entre ellos” (Fernández, et al. 2009), la estructura de una topología de red estrella se aprecia en la *Figura 1-4*.

La desventaja que presenta es que si un nodo sensor se avería automáticamente se pierde la información detectada por la mota, ya que no tiene otro camino de comunicación con el *gateway*. Su ventaja es que poseen un menor consumo energético.

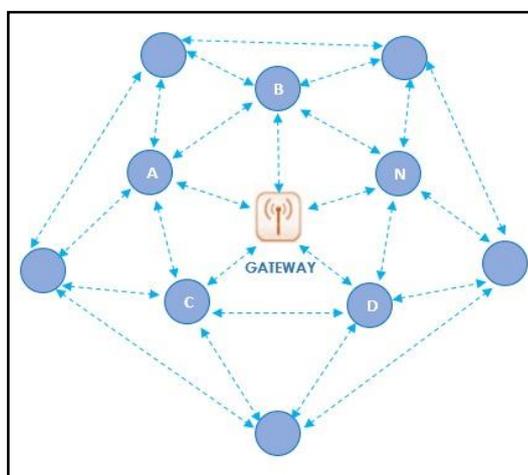


**Figura 1-4** Topología de red estrella  
**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 1.2.2.2 Topología de red en malla

Consiste en que tanto los nodos sensores como el *gateway* se pueden comunicar unos con otros, haciendo viable el intercambio de la información en toda la red. La topología en malla se caracteriza porque los nodos pueden ser tratados como *routers*, los cuáles transmiten y reciben datos de los demás nodos que conforman la red como se muestra en la *Figura 1-5*.

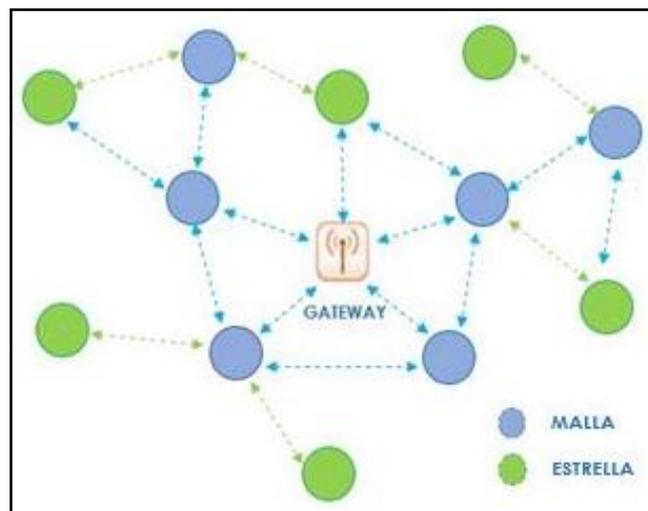
Las ventajas que presenta una red con esta topología es que la puede ser escalable, ya que son redes ilimitadas según su distribución, son consideradas sistemas multipunto porque cada nodo sensor tiene varios caminos de comunicación para llegar al *gateway* (Fernández, et al. 2009).



**Figura 1-5** Topología de red malla  
**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 1.2.2.3 Topología Híbrida Estrella – Malla

Está constituida por una red tipo estrella en torno a los nodos de una red tipo malla, unidos en una misma red que toma como nombre topología híbrida como se observa en la *Figura 1-6*. El objetivo de este tipo de redes es que la información que se transmite y reciba entre los nodos no se pierda, así como el optimizar el consumo de energía; siendo una topología más completa en todos sus aspectos (Fernández, et al. 2009).



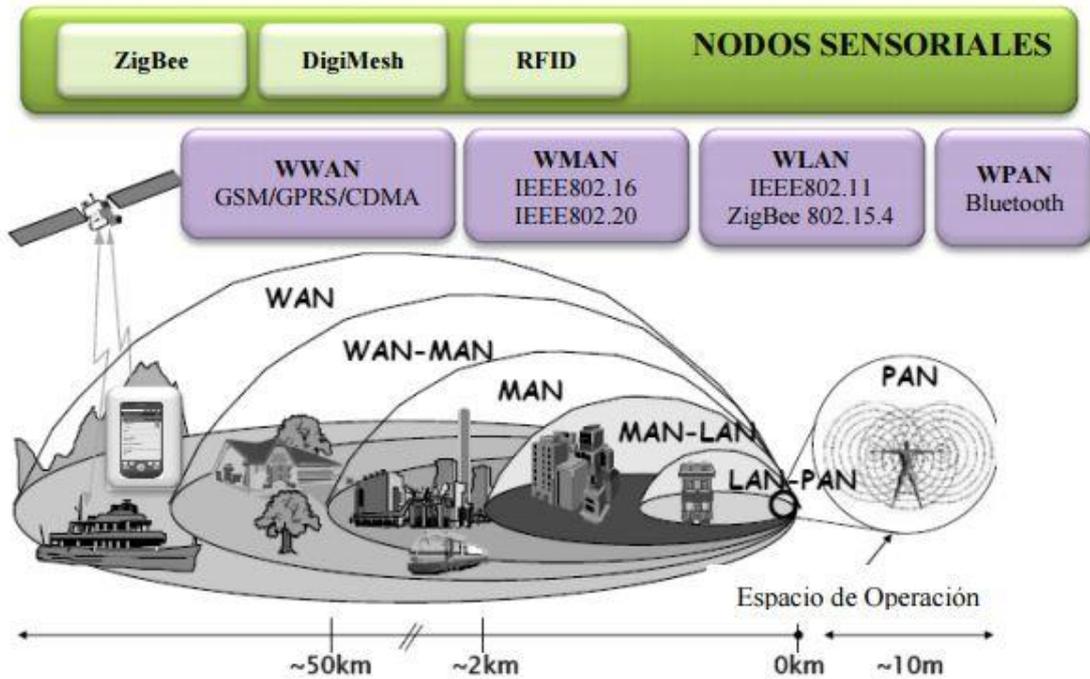
**Figura 1-6** Topología de red híbrida  
Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 1.2.3 Tecnologías para transmisión de datos

Dentro de una red de sensores inalámbricos, los nodos se comunican a través de ondas electromagnéticas, lo que permite transmitir y recibir la información entre ellos y con los demás dispositivos que forman parte de la red. Las conexiones son realizadas gracias a la aplicación que presentan algunos de los modelos para tecnologías inalámbricas, sin la necesidad de utilizar un canal físico (Pico, 2012).

Los nodos que conforman las redes sensoriales inalámbricas utilizan bandas de frecuencia específicas abiertas internacionalmente al público sin restricción, ideales para aplicaciones de este tipo de redes (Fernández, et al. 2009).

En general las tecnologías para conexión inalámbrica que existen en la actualidad se muestran en la *Figura 1-7*. Los medios más relevantes que sirven para la transmisión inalámbrica de redes de sensores son los de Radio Frecuencia (RF), Bluetooth, Wi-Fi, Zigbee.



**Figura 1-7** Actuales tecnologías inalámbricas  
Fuente: PICO VALENCIA, Pablo, 2012

- RADIO FRECUENCIA (RF)

Es el medio de comunicación inalámbrica más utilizada en redes de sensores inalámbricos para las conexiones entre los dispositivos de la red. Utilizan frecuencias que van entre 433 MHz y frecuencias de 2,4 GHz. Otras frecuencias son utilizadas como por ejemplo infrarrojos y en comunicaciones ópticas a través de láser. Estos dos últimos presentan limitaciones en la capacidad de transmisión que poseen, carecen de antenas para su uso (Fernández, et al. 2009).

Otras aplicaciones comunes que tiene la RF es en las transmisiones de radio y televisión digital terrestre (TDT), la facilidad que posee es la manera de pasar obstáculos al momento de implementar una red, ya que se caracterizan por ser omnidireccionales. Lo ideal de este tipo de tecnologías es su bajo consumo de energía lo que garantiza el funcionamiento adecuado de la red de sensores.

- BLUETOOTH

Es una tecnología inalámbrica utilizada en redes de área personal, también conocidas como WPAN (*Wireless Personal Area Network, por sus siglas en inglés*). Su aplicación lleva a ser posible la transmisión entre nodos con enlace de RF en bandas de frecuencias de los 2,4 GHz. Es de corto alcance pero suficiente para permitir el intercambio de información entre dispositivos Bluetooth, evitando así conectarse por cable a una red fija (Pico, 2012).

- Wi-Fi

Basado en el estándar IEEE 802.11, es utilizada en redes WLAN (*Wireless Local Area Network, sus siglas en inglés*). Presenta compatibilidad con dispositivos que utilizan el estándar 802.11, posee una cobertura geográfica limitada, presenta facilidad de instalación y movilidad. Dentro de la familia de especificaciones de dicho estándar se destacan: IEEE 802.11a, IEEE 802.11g, IEEE 802.11b, entre otras. La especificación final se denominó Wi-Fi que trabaja en bandas de 2,4 GHz (Pico, 2012).

- ZIGBEE

Trabaja en bandas de frecuencias de 868 MHz, 915 MHz y 2,4 GHz; está basado en el estándar 802.15.4 que son utilizadas en redes inalámbricas de área personal debido al corto alcance que generan. Su diseño exclusivo se aplica a redes de sensores inalámbricas que requieran bajas tasas de envío de datos, con bajo consumo energético. El inconveniente que presenta en las aplicaciones desarrolladas se refleja en el área de cobertura de los nodos distribuidos dentro de la red (Pico, 2012).

#### **1.2.4 Parámetros a considerar en redes de sensores inalámbricos**

Para realizar el diseño de una red sensorial inalámbrica, es importante evaluar los factores que influyen en la elección de los componentes físicos y *software*, con el objetivo de que éstos se acoplen a los requerimientos del sistema a implementar. Los parámetros más relevantes que se mencionan son los siguientes:

- TIEMPO DE VIDA: Hace referencia al consumo de energía que presenta una red de sensores inalámbricos, ya que el tiempo de vida de cada nodo dentro de la red dependen de la duración que posee la fuente de energía suministrada a los nodos, en ciertos casos utilizan baterías para cumplir con esta función. Es por tal motivo que una red de sensores inalámbricos debe ser desarrollada

con tecnologías que manejen un mínimo consumo energético, parte fundamental para que los nodos puedan operar dentro de la red, parte clave para extender la vida de la misma (Capella, 2010).

- COBERTURA DE LA RED: La cobertura en una red de sensores inalámbricos está directamente ligada al tipo de topología a utilizar. Depende de la ubicación geográfica de sus nodos sensoriales, que pueden ser distribuidos según el diseño y la ruta por la que viajan los datos a través de los nodos. Lo que le diferencia de las redes comunes, ya que la tecnología utilizada no cubre con los parámetros de diseño de las actuales redes de sensores.

- COSTO Y FACILIDAD DE INSTALACIÓN: Otro punto a considerar es el costo que tiene el desarrollar redes de sensores inalámbricos, ya que cada nodo, así como los sensores a utilizar, debe presentar un valor económico en comparación con los sistemas comunes; el fin es optimizar recursos y hacerlo a bajo costo.

Todo esto hace posible que las actuales redes no presenten inconvenientes en su instalación, ya que en el mercado existen un sin número de dispositivos que facilitan la construcción de los diferentes nodos presentes en la red.

- PRECISIÓN Y FRECUENCIA DE LAS MEDICIONES: Cada nodo sensor tiene como objetivo medir lo que ocurre a su alrededor según la aplicación de la red. Lo que destacan a estas redes es la capacidad que tienen de medir las condiciones físico-ambientales en tiempo real. Gracias a esta característica, los datos que son obtenidos por los nodos muestran cifras precisas, los resultados son evaluados en las mismas condiciones que dispositivos con similares funciones, obteniendo el valor real de la magnitud medida.

### **1.3 Tarjetas de desarrollo aplicadas a redes de sensores inalámbricos**

Las tarjetas de desarrollo son pequeñas placas electrónicas con circuitos impresos, cada una de ellas han sido fabricadas para distintas aplicaciones dependiendo de las características que poseen. Ofrecen compatibilidad con un sin número de dispositivos, entre los cuáles están los sensores, los módulos inalámbricos con tecnologías específicas, las tarjetas de memoria micro SD, entre otros; haciendo viable el uso en diseños de redes de sensores inalámbricos de forma rápida y sencilla.

Constan de E/S (entradas y salidas) análogas y digitales, puertos de conexiones, seriales, USB, plug para alimentación externa, reguladores de voltaje, indicadores de transmisión y recepción, memoria, entre otras características propias incluidas por cada diseñador. A más de que integran un *software* de programación según el sistema operativo (Windows o Linux) que esté implementado dentro de la placa.

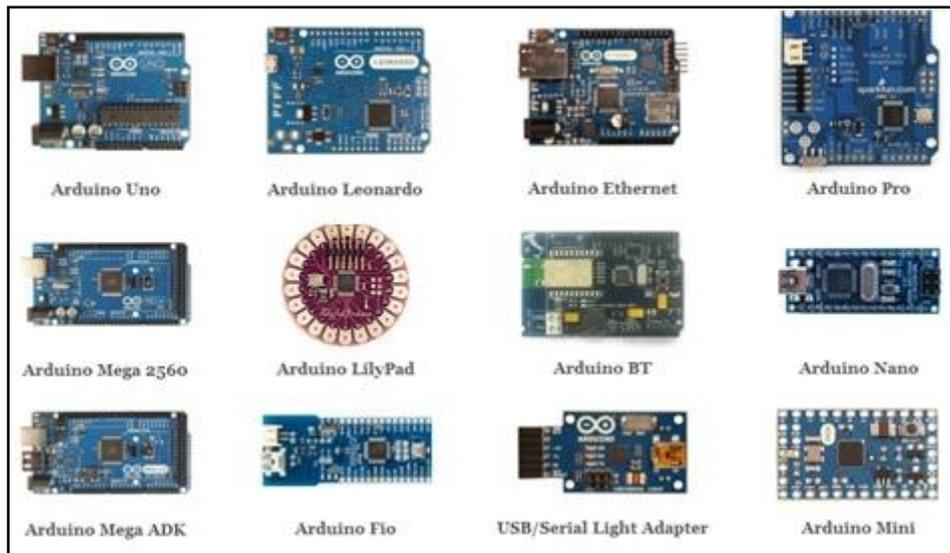
Actualmente, en el mercado, se encuentran múltiples tarjetas que han ido evolucionando y presentando mayores prestaciones y capacidad de operación que las versiones anteriores, entre las que se citan: Arduino, Intel Galileo, Raspberry Pi y Libelium Waspote.

### **1.3.1     *Arduino***

Los dispositivos Arduino se basan en los microcontroladores Atmel y tienen la particularidad de enlazar lo físico con lo virtual, así muchas aplicaciones han sido factibles de realizar gracias a las ventajas que presentan en relación con otros sistemas. Pueden recibir información del entorno mediante sus terminales de entrada, utilizando sensores compatibles con Arduino que existen en el mercado. Las características más importantes son (Tapia & Manzano, 2013):

- Factibilidad y Accesibilidad
  
- Multiplataforma
  
- Entorno de programación amigable con el usuario
  
- *Hardware* y *software* flexibles de código abierto

Existen placas Arduino con funciones específicas según el propósito a realizarse con ellas, en la *Figura 1-8* se presenta las tarjetas Arduino. Dispone de un *software* desarrollado en java que permite realizar programas con funcionalidades específicas, basta con identificar el puerto serial y el respectivo modelo de la placa con la que se vaya a trabajar, dicho entorno tiene el nombre de Arduino IDE.



**Figura 1-8** Tarjetas Arduino  
Fuente: TECNOLIFE, 2014

### 1.3.2 Intel Galileo

Debido a la acogida que tuvo las plataformas y diferentes placas de Arduino, motivó a empresas a dedicarse a crear nuevas líneas de tarjetas de desarrollo con la misma disposición de facilitar proyectos electrónicos. Es así que Intel lanzó al mercado su primer modelo, con tendencia a facilitar trabajos dirigidos a IoT (*Internet of Things, por sus siglas en inglés*), denominada Intel Galileo como se muestra en la *Figura 1-9*.



**Figura 1-9** Tarjeta Intel Galileo  
Fuente: INTEL, 2015

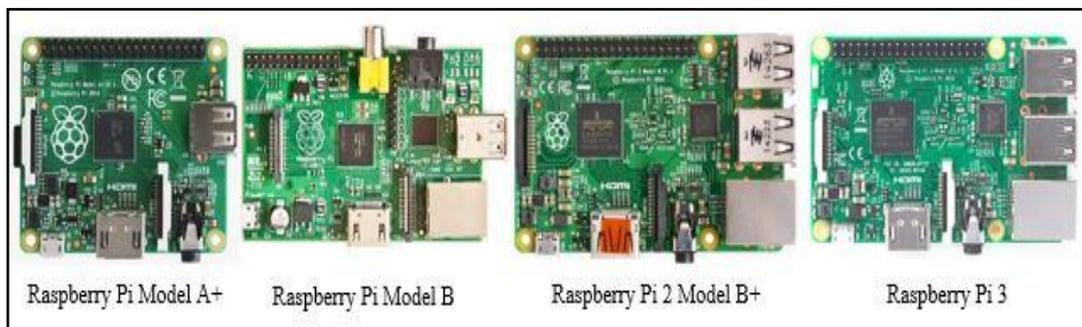
Al igual que Arduino, es una tarjeta de *hardware* y código abierto, que presenta compatibilidad con las tarjetas desarrolladas en Arduino.

La diferencia está en el hecho de combinar tanto *hardware* como *software* con Linux, surgiendo así la posibilidad de controlar dispositivos con otros lenguajes programables. Sin embargo, a pesar de ser una placa nueva, tiene ciertas desventajas al momento de realizar trabajos prácticos con IoT, una de ellas es el costo elevado en el mercado.

### 1.3.3 *Raspberry Pi*

Representa un pequeño ordenador personal de bajo coste. Esta placa fue desarrollada en Reino Unido con el objetivo de incentivar al mundo de la informática por medio de la enseñanza de la programación en numerosos trabajos como robótica, realizar servidores, entre otras cosas.

Utiliza el sistema operativo Linux, existen varios modelos que se acoplan a los requerimientos de *hardware* y *software* para diseños de sistema a implementar. Consta de puertos tanto de entradas como de salidas, facilitando aplicaciones en las que se necesiten control como por ejemplo en el campo de la domótica (Raspberrypi, 2016). La *Figura 1-10* muestra algunos ejemplos de modelos de tarjetas Raspberry Pi existentes en el mercado.



**Figura 1-10** Modelos de tarjetas Raspberry Pi existentes en el mercado.

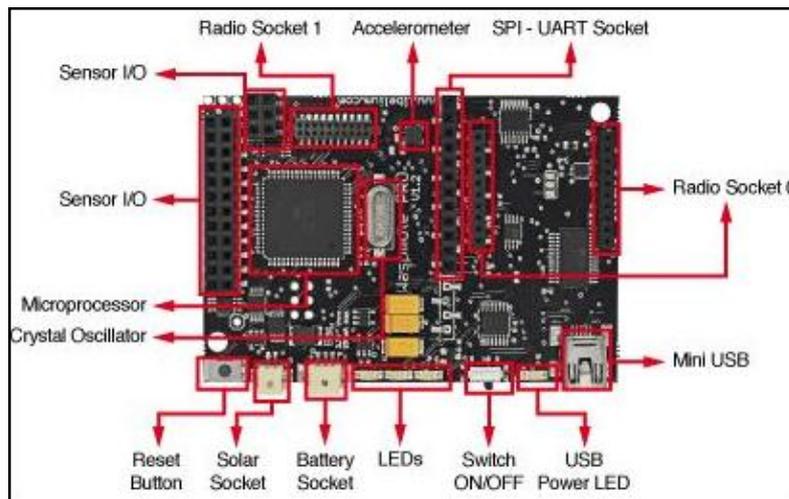
Fuente: RASPBERRYPI, 2016

### 1.3.4 *Libelium – Waspote*

Libelium representa una de las mayores compañías que presentan dispositivos con tendencia a realizar proyectos con tecnologías IoT. El producto que más se acopla para este tipo de proyectos es Waspote, que se aprecia en la *Figura 1-11*. Waspote es una tarjeta desarrollada por Libelium con características más completas que las anteriores. Basada en un diseño modular, que

tiene como funcionalidad reunir módulos inalámbricos en cada dispositivo con la capacidad de ampliarlos según los requerimientos necesarios y de ser el caso retirarlos sin problema.

Dichos módulos que lo integran son: Zigbee, GSM – 3G/GPRS, GPS, placas sensoriales, y tarjetas de memoria SD; así mismo presenta compatibilidad con dispositivos de Arduino. No obstante, a más de las ventajas que presenta dicha plataforma, tiene ciertas desventajas a la hora de implementar sistemas, como es su alto costo que provoca baja aceptación y disminución en su demanda (Fuentes, 2013).



**Figura 1-11** Tarjeta Waspote  
Fuente: LIBELIUM, 2015

- COMPARATIVA ENTRE TARJETAS DE DESARROLLO

Una vez analizada en forma individual cada una de las tarjetas de desarrollo, se procede a realizar una comparativa entre ellas de sus principales características y cuyos resultados se reflejan en la *Tabla 1-1*.

**Tabla 1-1** Comparativa de tarjetas de desarrollo

Características	Arduino Uno	Intel Galileo Gen1	Raspberry Pi (Model B)	Waspote
Procesador	ATMega328	SoC Quark X100	ARM11	ATMega1281
Voltaje de operación	5 V	3.3 V / 5 V	3.3 V / 5 V	0.5 – 3.8 V
Voltaje de entrada	7 – 12 V	5 V	5 V	5 – 7 V
RAM	2 KB	512 KB	512 MB	8 KB
Memoria Flash	32 KB	8 MB	-	128 KB
USB	1	2	2	1
Velocidad de reloj	16 MHz	400 MHz	700 MHz	16 MHz
E/S Analógicas	6	6	-	7
E/S Digitales	14	14	8	8
Ethernet	No contiene	10/100	10/100	No contiene
Precio aproximado	\$30	\$90	\$50	\$180

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda. 2016.

## **1.4 Contaminación Acústica**

La contaminación acústica se dio inicialmente en los sectores urbanos, llegando en la actualidad a las áreas rurales (Ojeda, 2014). En los últimos años ha incrementado a gran escala, provocando un daño tanto físico como psicológico. Según la Organización Mundial de la Salud (OMS) el rango máximo tolerable por el ser humano para evitar problemas audibles está entre los 55 a 65 decibelios (Organización Mundial de la Salud, 2015).

El crecimiento desmedido de las ciudades ha sido un factor para el aumento de este tipo de contaminación, la falta de gestión para evitar la existencia de altos niveles de ruido y principalmente la escasa información sobre el tema ha ocasionado problemas graves en la salud de las personas.

Entidades como el Ministerio del Ambiente, han sido creadas con el objetivo de controlar el cumplimiento de normas y actividades de calidad ambiental de agua, aire, suelo, ruido y desechos que produzcan contaminación en el medio. Según la Constitución de la República del Ecuador, los ciudadanos en general, tienen el derecho de vivir en un ambiente sano como se dicta en la segunda sección, Art. 14. (Ojeda, 2014).

Además, existen normas y leyes que exigen un control de la contaminación acústica en el Ecuador, como se presenta en el TULSMA (Texto Unificado de Legislación Secundaria del Ministerio de Ambiente). Según lo establecido dentro del texto, en el libro VI, Anexo 5; el límite máximo que pueden producir fuentes móviles como generadores de automotores es de 65 decibelios (TULSMA, 2003).

### **1.4.1 Definición**

La contaminación acústica es la presencia de ruidos de cualquier fuente y origen, que provocan una molestia a las personas, así como riesgos y daños que traen como consecuencia problemas a la salud humana, interrupción de labores diarias; y en ciertos casos, efectos en el medio ambiente (Ojeda, 2014).

Es uno de los problemas ambientales que ha crecido últimamente por causa de factores presentados en el desarrollo de distintas actividades humanas, así como de impactos naturales.

Los sonidos bruscos que se propagan en el medio afectan tanto a personas, como animales y vegetales.

Como menciona Guevara (2013) en su escrito, la contaminación acústica es producida por la cantidad de ruido, mismo que se lo puede estudiar a partir de dos puntos de vista. Desde lo físico, es un fenómeno causado por oscilaciones anormales que dependen del objeto que los ocasiona. Y, desde lo fisiológico, es un efecto producido en el oído por vibraciones formadas en el medio.

Ambos tienen en común la presencia de sonidos desagradables lo que trae como consecuencia molestias, riesgos y daños a las personas, y en ciertos casos, causar efectos específicos al medio ambiente.

#### **1.4.2 Ruido Ambiental**

Es considerado un sonido no agradable al oído, cuya intensidad presenta valores altos en comparación a un sonido normal, siendo incluso un elemento perjudicial para la salud humana.

Está formado por ondas acústicas que se pueden propagar a través del medio, sea sólido, líquido o gaseoso, dichas ondas son pequeñas oscilaciones de presión que se propagan en todas las direcciones posibles. El ruido que se emite en el medio viaja a través del aire a rápida velocidad, por lo que a ciertas distancias dejan de ser perceptibles.

##### **1.4.2.1 Clasificación del ruido ambiental**

Se clasifica por términos de origen y de fuente, que se detallan a continuación en la *Tabla 1-2*.

**Tabla 1-2** Clasificación del ruido ambiental

<b>Clasificación</b>	<b>Ejemplos</b>
Fuente	Móviles: motocicletas, generados por automotores, entre otros.
	Fijas: microempresas, talleres y sitios de esparcimiento nocturno.
Origen	Urbanas: transporte aéreo, vehículos, trenes, etc.
	Industriales: manufactura, comercio, compresoras, etc.
	Agropecuarios: animales, maquinaria agrícola, entre otras.
	Servicios: restaurantes, discotecas, tiendas, clubs, bares, etc.
	Doméstica: equipos electrodomésticos, personas, etc.

**Fuente:** ALTAMIRANO. Héctor & PUENTE. Amanda. 2016.

#### 1.4.2.2 Decibelio (dB)

Es considerada una medida del sonido en escala exponencial, que representa la relación que existe entre un valor medido con respecto a uno referencial (Martínez y Peters, 2015). El ruido es medido por su intensidad y frecuencia en decibeles, dicha unidad de medida toma como referencia el sonido más pequeño que puede distinguir el oído, siendo éste el límite perceptible por el sentido auditivo.

#### 1.4.3 Niveles y Factores sobre la Contaminación Acústica

Los niveles acústicos más característicos se presentan en la *Tabla 1-3*.

**Tabla 1-3** Ejemplos de valores de niveles acústicos por fuentes emisoras

Fuente	Nivel Acústico (dB)
Umbral auditivo	0
Respiración normal, trineo de pájaros.	10
Roce de hojas secas	20
Área rural en la noche, susurro a 5 metros	30
Suburbio	40
Área comercial	50
Conversación ordinaria	60
Calle ruidosa, ambiente de oficina.	70
Fábricas, aspiradoras	80
Tráfico rodado	85
Camión a 30 metros	90
Pito de un carro	100
Un trueno a 300 m de altura, discoteca	110
Un trueno, taladros	120
Un avión despegando a 10 m	130
Umbral del dolor	140
Sordera	150

Fuente: ALTAMIRANO. Héctor & PUENTE. Amanda. 2016.

Existen factores que pueden ser de tipo natural, pero que influyen en el aumento de la contaminación como pueden ser: el sonido de los truenos, erupciones volcánicas, entre otros componentes naturales. En general, los factores que influyen en la contaminación acústica se deben principalmente a las diferentes actividades humanas realizadas, como son:

- Tráfico de autos, aéreo y ferroviario
  
- Actividades de construcción
  
- Actividades industriales
  
- Actividades de recreación

Dichos factores ocasionan problemas a la salud humana lo que provocan efectos nocivos como son:

- EFECTOS PSICOPATOLÓGICOS: Causados por los niveles de ruido mayores a 60 dB provocando: dilatación de pupilas así como un parpadeo acelerado, agitación en la respiración y taquicardias, dolores de cabeza, aumento en la presión arterial, actividad muscular hasta llegar al punto en el que los músculos del cuerpo se vuelven tensos, lo que ocasionan dolores musculares; en cambio, niveles mayores a 85 dB provocan: gastritis, aumento del colesterol, aumento de la glucosa en la sangre (Narváez, 2014).
  
- EFECTOS PSICOLÓGICOS: Los efectos psicológicos que se presentan por causa de niveles altos de ruido son: insomnio que se produce al estar expuesto a niveles de 30 dB, cansancio, estrés, depresión, irritabilidad, ansiedad, agresividad, aislamiento social, desconcentración. Todos enfocados a la interrupción en las actividades personales y privadas de cada persona (Narváez, 2014).

#### ***1.4.4 Límites Permisibles de Niveles de Ruido Ambiente para Fuentes Fijas y Fuentes Móviles, y para Vibraciones***

TULSMA - Libro VI - Anexo 5: Recoge las distintas leyes que están relacionadas con los recursos naturales, creado con el objetivo de dar protección a los mismos. Dentro de dicho texto existen secciones, en las cuales se menciona el libro VI que trata específicamente de la calidad ambiental

del Ecuador. Dicho libro consta de anexos, cada uno regido por distintas normas principales para el debido control y prevención de la contaminación del medio ambiente, como son de agua, suelo, aire, desechos sólidos, desechos peligrosos, entre otros. (TULSMA, 2003).

La norma técnica que presenta los niveles de ruido permitidos en el país se encuentran en el Anexo 5 del Libro VI. El texto cumple con las disposiciones dictadas por la Ley de Gestión Ambiental en conjunto con el Reglamento a la Ley de Gestión Ambiental para la Prevención y Control de la Contaminación Ambiental. En el documento se establecen los máximos niveles de ruido permisibles emitidos por automóviles, además de la valoración de vibraciones realizadas en edificaciones (TULSMA, 2003).

#### 1.4.4.1 Objetivo Anexo 5 del Libro VI del TULSMA

El texto desarrollado por el Anexo 5 del Libro VI del TULSMA tiene como objetivo principal: *“Preservar la salud y bienestar de las personas, y del ambiente en general, mediante el establecimiento de niveles máximos permisibles de ruido. La norma establece además los métodos y procedimientos destinados a la determinación de los niveles de ruido en el ambiente, así como disposiciones generales en lo referente a la prevención y control de ruidos.”* (TULSMA, 2003).

#### 1.4.4.2 Clasificación de los límites máximos permisibles de niveles de ruido ambiente

##### - NIVELES MÁXIMOS PERMISIBLES DE RUIDO

Los niveles máximos según la exposición de ruido de una fuente fija tienen como límite los valores presentados en la *Tabla 1-4*.

**Tabla 1-4** Niveles máximos de ruido permisibles según uso del suelo

Tipo de zona según uso del suelo	Nivel de presión sonora equivalente NPS eq [dB]	
	De 06h00 a 20h00	De 20h00 a 06h00
Zona hospitalaria y educativa	45	35
Zona residencial	50	40

Zona residencial mixta	55	45
Zona comercial	60	50
Zona comercial mixta	65	55
Zona industrial	70	65

Fuente: TULSMA, 2003

El nivel de presión sonora (NPS) determina la intensidad del sonido que alcanza a una persona en un momento dado, se mide en decibelios (dB) y varía entre 0 dB umbral de audición y 120 dB umbral de dolor. El nivel de presión sonora equivalente está medido en decibeles, esto quiere decir que “en el mismo intervalo de tiempo, contiene la misma energía total que el ruido medido”. En áreas rurales, los valores rectificadas que han sido adquiridos en el sitio del receptor no deberán superar los 10 dB.

En casos en el que se presenten dificultades con lo estipulado en la *Tabla 1-4* sobre el uso de los suelos, según la norma dictada en el Anexo 5, la entidad encargada tiene la obligación de controlar dicho problema. Para conocer las diferentes medidas de prevención y consideraciones sobre ruido, se puede consultar en el Anexo 5 del Libro VI del TULSMA (TULSMA, 2003). Además, en el documento se añade:

*“Se prohíbe la emisión de ruidos o sonidos provenientes de equipos de amplificación u otros desde el interior de locales destinados, entre otros fines, para viviendas, comercios, servicios, discotecas y salas de baile, con niveles que sobrepasen los límites determinados para cada zona y en los horarios establecidos en la presente norma.”* (TULSMA, 2003).

#### - DE LA MEDICIÓN DE NIVELES DE RUIDO PRODUCIDOS POR UNA FUENTE FIJA

Para la realización de las mediciones de ruido producidos por fuentes fijas, en general se utilizan sonómetros calibrados que deberán cumplir con las normas establecidas por la Comisión Electrotécnica Internacional (IEC). En cuanto a la ubicación del micrófono para dicha medición, se recomienda ubicarlo de 1 m a 1,5 m de altura, y aproximadamente a 3 m de distancia de edificios para conseguir la reflexión del sonido (TULSMA, 2003).

#### - CONSIDERACIONES PARA GENERADORES ELECTRICIDAD DE EMERGENCIAS

Es importante que en infraestructuras que contengan generadores de electricidad como emergencia tomen en cuenta si estos dispositivos causan molestias e inconvenientes por los niveles de ruido que produzcan a su alrededor (TULSMA, 2003).

- RUIDOS PRODUCIDOS POR VEHÍCULOS AUTOMOTORES

En la *Tabla 1-5* se detallan los niveles máximos permisibles para vehículos automotores, en el caso de que se superen valores a 65 dB en horario diurno y 55 dB en nocturno, la entidad de control tiene la potestad de indicar que tipo de autos pueden circular o no en áreas urbanas, según el horario establecido en la norma (TULSMA, 2003).

**Tabla 1-5** Niveles máximos de ruido para vehículos automotores

<b>Categoría de Vehículo</b>	<b>Descripción</b>	<b>Máximo (dB)</b>
Motocicletas	De hasta 200 centímetros cúbicos	80
	Entre 200 y 500 c.c.	85
	Mayores a 500 c.c.	86
Vehículos	Transporte de personas, nueve asientos, incluido el conductor.	80
	Transporte de personas, nueve asientos, incluido el conductor, y peso no mayor a 3,5 toneladas.	81
	Transporte de personas, nueve asientos, incluido el conductor, y peso mayor a 3,5 toneladas.	82
	Transporte de personas, nueve asientos, incluido el conductor, peso mayor a 3,5 toneladas, y potencia de motor mayor a 200 HP.	85
Vehículos de Carga	Peso máximo hasta 3,5 toneladas.	81
	Peso máximo de 3,5 toneladas hasta 12 toneladas.	86
	Peso máximo mayor a 12 toneladas.	88

Fuente: TULSMA, 2003

- DE LAS VIBRACIONES EN EDIFICACIONES

Vibración, según la norma dictada en el Anexo 5 del Libro VI del TULSMA, la define como: *“Una oscilación en que la cantidad es un parámetro que define el movimiento de un sistema mecánico, y la cual puede ser el desplazamiento, la velocidad y la aceleración.”* (TULSMA, 2003).

Los niveles que se detallan en la *Tabla 1-6* representan los valores máximos de vibraciones que podrán enviar dispositivos o instalaciones a los componentes de la estructura receptora (TULSMA, 2003).

**Tabla 1-6** Límite de transmisión de vibraciones

Uso de edificación	Período	Curva Base
Hospitalario, Educativo y Religioso	Diurno	1
	Nocturno	1
Residencial	Diurno	2
	Nocturno	1,4
Oficinas	Diurno	4
	Nocturno	4
Comercial	Diurno	8
	Nocturno	8

Fuente: TULSMA, 2003

#### **1.4.5 Equipos de medición de ruido**

Debido a que en la actualidad el ruido se ha convertido en un problema ambiental, existen herramientas de ayuda para medir los valores de ruido presentes en el medio. Se trata de equipos de medición con una característica en común, utilizar un transductor como elemento principal para medir los niveles de ruido. Entre los equipos más relevantes del mercado, se muestra en la *Figura 1-12*, y son los siguientes (Guevara, 2013):

- **Sonómetro:** Se caracteriza por medir la presión sonora y reconocer un sonido simulando al oído humano. Este equipo es empleado para obtener valores de cualquier fuente ruidosa.
- **Dosímetro:** Considerado un sonómetro portátil debido a que presenta un menor tamaño y sirve para deducir la cantidad de ruido existente en una zona, a la cual se exponen las personas.
- **Analizador de Frecuencias:** Es un equipo de medición que sirve de ayuda para establecer “el contenido energético de cualquier sonido pero en función de la frecuencia” (Guevara, 2013).
- **Calibrador acústico:** Es un equipo diseñado para cerciorar que los sonómetros trabajen correctamente con los parámetros necesarios. El objetivo de estos instrumentos, según Guevara (2013), es: “generar un tono estable de nivel a una frecuencia predeterminada y

ajustar la lectura del sonómetro haciéndola coincidir con el nivel patrón generado por el calibrador”.



**Figura 1-12** Equipos de medición de ruido más comunes

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016

#### ***1.4.6 Trabajados realizados en Riobamba sobre supervisión de contaminación acústica***

Riobamba, a pesar de ser una ciudad pequeña, presenta problemas de contaminación acústica al igual que las grandes ciudades como Quito, Guayaquil y Cuenca. Es por lo que el desarrollo de sistemas que intervengan procesos automatizados es de suma importancia para facilitar la supervisión de zonas afectadas y proporcionar un mejor control a ciertas actividades que produzcan altos niveles de ruido en la ciudad.

Entre los trabajos que se han realizado se menciona una propuesta de la Universidad Nacional de Chimborazo sobre un modelo de gestión de ruido, en el que se realizan mediciones de ruido con un sonómetro para desarrollar un plan estratégico que permita concientizar a la ciudadanía y trabajar en conjunto con las autoridades para controlar la contaminación acústica presente en la ciudad (Razo & Maza, 2015).

Siendo así, no existen investigaciones suficientes sobre el tema, lo que impulsa a realizar el presente trabajo, que tiene como objetivo desarrollar equipos inalámbricos de supervisión con bajo costo que, al mismo tiempo, entregue un aviso al usuario sobre los actuales niveles acústicos medidos.

Además consta con un registro de datos que sirve como información para futuros estudios y posibles modelos matemáticos que ayuden a disminuir los niveles de contaminación acústica gracias a la toma de decisiones y prevenir futuras afecciones a la salud humana.

## CAPÍTULO II

### 2. DISEÑO DE *HARWARE* Y *SOFTWARE* DEL PROTOTIPO DEL SISTEMA ELECTRÓNICO

#### 2.1 Introducción

En este capítulo se presenta el diseño *hardware* y *software* del prototipo del sistema inalámbrico de supervisión de contaminación acústica (SISCA), las razones por las cuales se eligen cada uno de los componentes y los respectivos diagramas de bloques de los módulos que conforman el sistema. Las partes físicas y las que conforman el *software* son combinados para trabajar en conjunto en la supervisión y recolección de datos registrados.

#### 2.2 Requerimientos *hardware* del SISCA

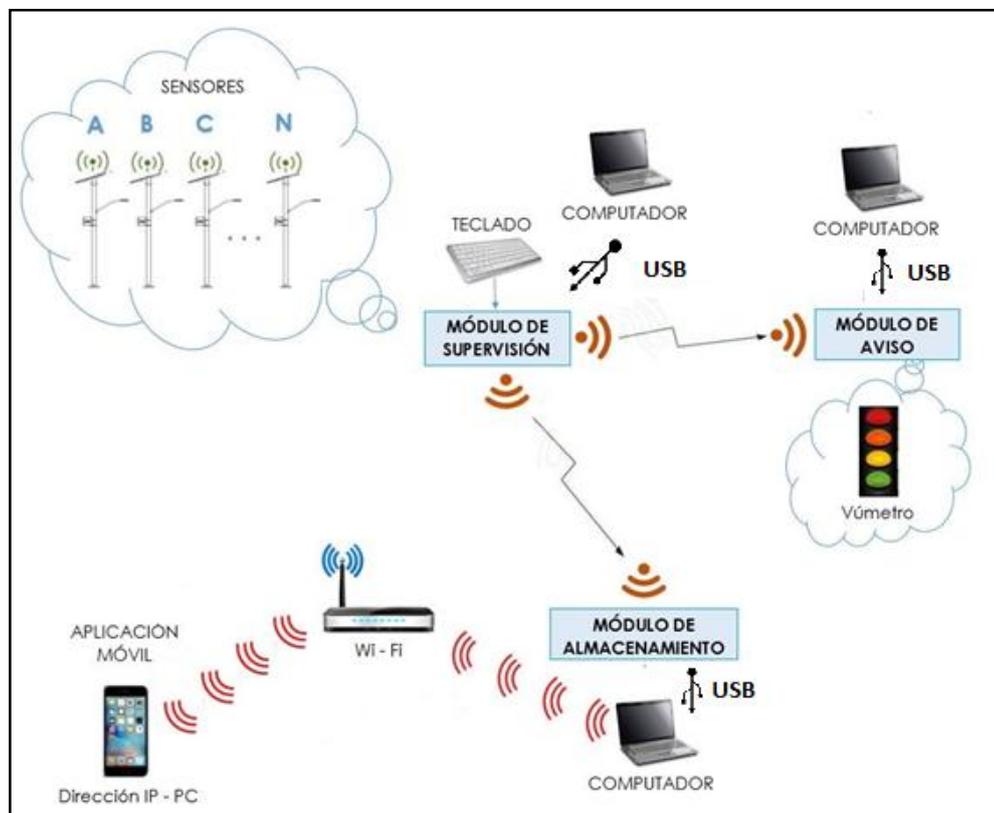
Basado en los estudios realizados en el primer capítulo se determina los requerimientos *hardware* para el sistema inalámbrico de supervisión de contaminación acústica para la ESPOCH. Dichos requerimientos son los siguientes:

- Implementación de un sistema de bajo costo con equipos de fácil instalación independientemente del área a cubrir.
- Censar los niveles de ruido encontrados en la ESPOCH determinando sus valores máximos y mínimos.
- Cubrir zonas con equipos móviles que sean capaces de captar niveles de ruido de forma inalámbrica para su fácil transportación.
- Brindar información en tiempo real de niveles de ruido para la utilización de los mismos en futuros estudios. Siendo el sistema capaz de brindar de forma visual los datos medidos en un determinado tiempo y de almacenar los mismos en una base de datos.

- Proveer de un sistema que alerte a las personas que transitan por la zona supervisada sobre niveles de ruido a los que se está exponiendo.

### 2.2.1 Concepción de la arquitectura general del sistema

La arquitectura general del sistema se encuentra en la *Figura 2-1* en donde se aprecia los tres módulos pertenecientes del sistema: módulo de supervisión, módulo de aviso y módulo de almacenamiento. Su comunicación es por medio de módulos de Radio Frecuencia (RF) que son integrados a las tarjetas de desarrollo.



**Figura 2-1** Arquitectura general del sistema inalámbrico.

**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

**Módulo de Supervisión.** Tiene el trabajo principal de la adquisición de niveles de ruido en tiempo real en una determinada zona. Cada módulo de supervisión cuenta con un bloque de alimentación, un bloque de sensores, un dispositivo reloj, un teclado matricial que facilita el ingreso de datos tales como la hora y fecha en caso de falla o inicio de funcionamiento del reloj integrado, así como la identificación del nodo.

El módulo transmite inalámbricamente la información a los módulos vecinos por medio de RF para el almacenamiento y aviso respectivo a los usuarios de la zona. El sensor utilizado es un Sparkfun Sound Detector que es el encargado de la adquisición de ondas sonoras en la zona y por medio de la tarjeta de desarrollo, los convierte en valores numéricos (decibeles).

La cantidad de módulos dependerá del área que se requiera cubrir, conectándolo a un computador se puede visualizar la información y en una pantalla LCD se muestran los valores actuales con fecha y hora.

**Módulo de Aviso.** Recibe la información transmitida por el módulo de supervisión, para convertirlo en una forma visual y que los usuarios puedan ver los niveles de ruido presentes en la zona. Este módulo consta de un receptor de RF, un vúmetro para medir los niveles de ruido, una tarjeta de desarrollo y una fuente de alimentación. Mantiene un código de colores que identifica cuando se presenta altos niveles de contaminación acústica.

**Módulo de Almacenamiento.** En este módulo se procesa la información obtenida vía RF desde el módulo de supervisión que es almacenada en una base de datos y a su vez presentada en formas gráficas los valores máximos y mínimos de los días censados. Consta de un bloque de alimentación, un bloque de almacenamiento, en este caso un *shield* SD conectada a la tarjeta de desarrollo y el respectivo bloque de recepción por RF.

## **2.2.2      *Diseño de la Arquitectura del SISCA***

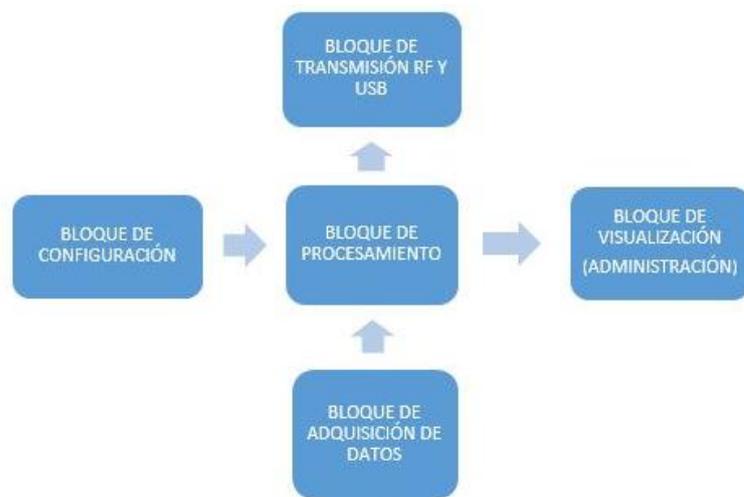
Definidos los requerimientos del sistema, se presenta el diseño de cada uno de los módulos que conforman el sistema inalámbrico para la medición de contaminación acústica.

### **2.2.2.1      *Módulo de Supervisión***

El diagrama de bloques del módulo de supervisión se los presenta en la *Figura 2-2* el cual consta de los bloques de alimentación, transmisión, reloj, visualización y control por medio de teclado de la hora, fecha y la identificación de los nodos.

Los datos obtenidos por medio del sensor de ruido son procesados por la tarjeta de desarrollo para transformarlos en valores numéricos (decibeles) para ser enviados por medio del bloque de transmisión de RF MX-FS-03V a los módulos de aviso y de almacenamiento. Consta de un reloj que proporciona la fecha y hora actual el que puede ser actualizado por medio de un teclado matricial.

Dispone de un bloque de visualización para facilitar este proceso. El bloque de procesamiento consta de un Arduino MEGA que posee terminales analógicas y digitales, el sensor de ruido utiliza una entrada analógica para la recepción de datos y se utilizan terminales digitales como entradas y salidas para el resto de bloques que conforman el módulo de supervisión.



**Figura 2-2** Diagrama de bloques del Módulo de Supervisión de contaminación acústica.

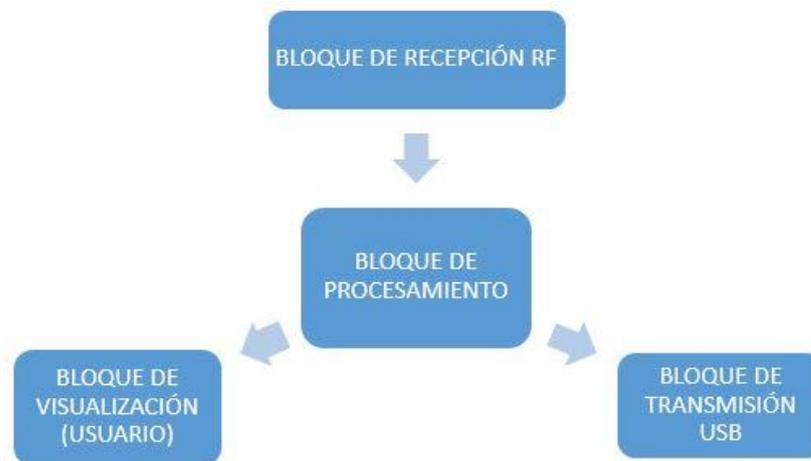
Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

#### 2.2.2.2 Módulo de Aviso

Realiza la advertencia a los usuarios de los niveles de ruido que se encuentran en la zona. Consta de cuatro bloques como se puede observar en la *Figura 2-3*, tiene el bloque de procesamiento, alimentación, bloque de visualización y el bloque de recepción. La alimentación se la realiza por medio de una fuente de alimentación de 12 V conectados a la tarjeta de desarrollo que funciona como bloque de procesamiento, en este caso un Arduino MEGA.

El bloque de visualización consta de un vúmetro que indicará los niveles del ruido que se encuentra en tiempo real en una zona. El bloque de recepción es uno de los más importantes ya

que es el encargado de captar la información enviada por el módulo de supervisión y llevarlos al procesador.



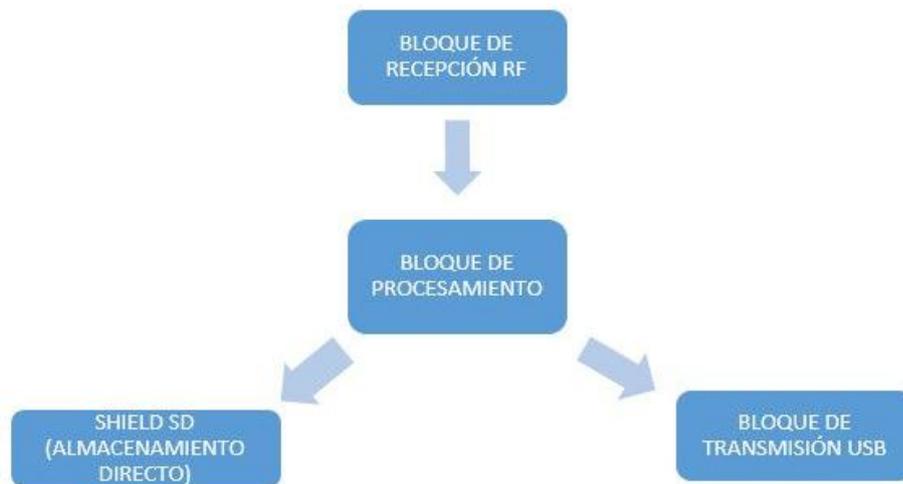
**Figura 2-3** Diagrama de bloques del Módulo de Aviso del sistema sensorial.  
Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

Este módulo está conformado por el receptor MX-JS-05V diseñado para captar señales de radio frecuencia enviados por un transmisor (Ruhman, 2014). Trabaja con el procesador Arduino MEGA 2560 y el vúmetro conformado por circuitos que permiten la activación de un sistema de bombillas de 110 V por medio de un relé alimentado con 5 V.

### 2.2.2.3 Módulo de Almacenamiento

En la *Figura 2-4* se muestra el diagrama de bloques que conforma el módulo de almacenamiento, consta de cinco bloques. Este módulo consta de un bloque de procesamiento un Arduino UNO ya que este tiene menor número de terminales que un Arduino MEGA por la razón de que no es necesario un número extenso de las mismas.

Está formado también de un bloque de recepción de RF MX-JS-05V, un bloque de alimentación y una *shield* SD conectada a la tarjeta de desarrollo para realizar el almacenamiento directo de la base de datos de los niveles de ruido registrados en un determinado periodo. El almacenamiento por medio del equipo de cómputo se lo realiza a través de una herramienta virtual desarrollada en LabVIEW que visualiza en forma numérica y grafica la información obtenida.



**Figura 2-4** Módulo de Almacenamiento del sistema inalámbrico.  
 Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 2.2.3 Selección de elementos que conforman el SISCA para la ESPOCH.

A continuación se detallan los elementos utilizados para el ensamble del prototipo del sistema inalámbrico de sensores con sus respectivas hojas técnicas detalladas en el *Anexo 1*.

#### 2.2.3.1 Tarjetas de Desarrollo

A lo largo de los años han existido dispositivos de procesamiento, empezando por el microcontrolador PIC que data de 1975 que a pesar de su pequeño tamaño ayudó a cubrir grandes necesidades de la electrónica (Quirarte, 2014). En la actualidad se dispone de otros dispositivos capaces de realizar el trabajo de control de procesos con gran cantidad de entradas y salidas dependiendo de las necesidades, denominados Arduino.

En el capítulo anterior se realizó la comparación entre tarjetas de desarrollo, como resultado las tarjetas a ser utilizadas son las Arduino MEGA y Arduino UNO. La gran ventaja de Arduino es el IDE, su *software* personalizado que ha logrado ser uno de las herramientas más fáciles de utilizar en cuanto a programación para el desarrollo de aplicaciones, con la simple configuración de sus terminales analógicas y digitales de la manera más conveniente para el usuario (Quirarte, 2014).

- Arduino MEGA 2560

Esta tarjeta de desarrollo cuenta con un microcontrolador ATmega 2560. Como se muestra en la *Figura 2-5* este dispositivo cuenta con 54 terminales digitales de entrada y salida de los cuales 15 se pueden utilizar como salidas PWM, 16 entradas analógicas, 4 puertos serie de *hardware* (UART), un velocidad de reloj de 16MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio.

Es suficiente la conexión a un computador con un cable USB o la corriente con un adaptador de AC a DC o una simple batería para ponerlo en funcionamiento. El voltaje de salida de sus terminales digitales es de 5 V, otra de 3.3 V y requiere de una alimentación de entre 5 a 12 V. Consta de tres memorias: flash, EEPROM y SRAM (Arduino, 2016).

Cabe recalcar que en este proyecto se ha utilizado el modelo genérico del Arduino MEGA, conocido como Funduino MEGA por su facilidad de adquisición que a la final son las copias certificadas por parte de Arduino con sus mismas características técnicas.



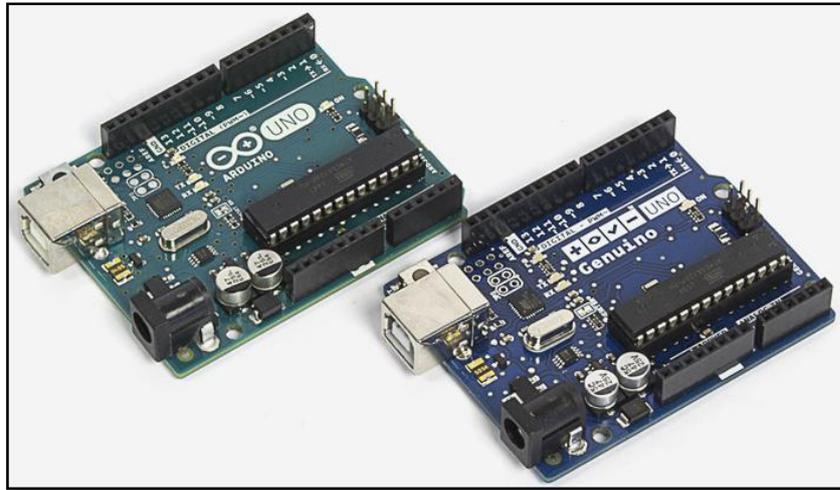
**Figura 2-5** Tarjeta de desarrollo Arduino MEGA.  
Fuente: ARDUINO, 2016

- Arduino UNO

Este dispositivo cuenta con un procesador ATmega328P, cuenta con 14 terminales digitales de entrada y de salida y 6 de ellos se los puede utilizar como PWM, tiene 6 entradas analógicas y un cristal de 16 MHz, conexión USB y un botón de reinicio. Funciona con conector para alimentación AC o DC para funcionar con una batería y es de bajo costo en el mercado por lo cual se vuelve un efectivo equipo de procesamiento.

Funciona con Arduino IDE para realizar la programación de los procesos que se requiera por medio de codificación electrónica. Tiene salidas de voltaje de 5 V para alimentar a otros

dispositivos y soporta de 6 a 20 V. Cuenta con una memoria SRAM de 2 KB y con una EEPROM de 1 KB. En la *Figura 2-6* se puede apreciar las características del Arduino UNO (Arduino, 2016).

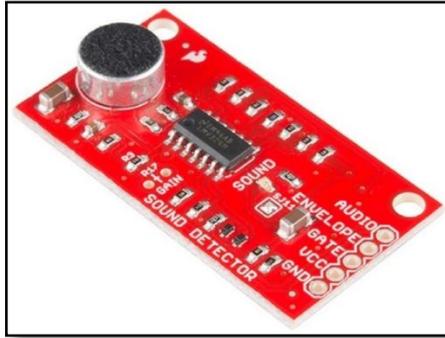


**Figura 2-6** Arduino UNO.  
Fuente: ARDUINO, 2016

### 2.2.3.2 *Sensor de Ruido (Sound Detector)*

El Sparkfun Sound Detector es un sensor utilizado para captar señales sonoras que consta de tres salidas diferentes en forma analógica como se consta en la *Figura 2-7*. La salida “*envelope*” lee la amplitud del sonido para medir la tensión analógica. El sensor combina un micrófono y algunos circuitos de procesamiento. Utiliza una tensión de alimentación VCC entre 3,5 y 5,5 voltios siendo lo ideal 5 voltios (Jfet, 2014).

Cuenta con un led que indica la entrada de ondas de sonido al sensor. La salida de audio del detector de sonido indica la tensión de audio directamente desde el micrófono. La salida de la cápsula entrega poco voltaje, por lo que la señal de la cápsula es amplificada por IC1G1, una etapa de amplificador operacional. Por defecto, el pre-amplificador tiene una ganancia aritmética de 100 (20 dB) (Jfet, 2014).



**Figura 2-7** Sensor de sonido Sparkfun.  
Fuente: JFET, 2014

### 2.2.3.3 Equipos de transmisión

Para trabajar con el SISCA es necesario utilizar equipos o dispositivos que brinden estas características buscando siempre la calidad en la transmisión y en lo posible el abaratamiento de costos. La tecnología utilizada en este sistema es la radio frecuencia.

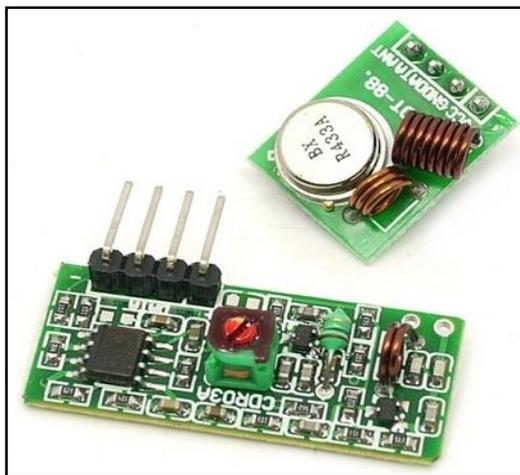
Este tipo de comunicación es la más sencilla y económica de implementar, tiene su funcionamiento en modo simplex ya que no hace un envío y recepción de información de manera simultánea. Funciona en la frecuencia de 433 MHz que permite un enlace de radio frecuencia muy funcional.

Se puede apreciar la apariencia de los módulos RF en la *Figura 2.8* a los cuales se hace necesaria la conexión de antenas externas para tener una mejor transmisión y recepción. El alcance de estos dispositivos depende de dos factores muy importantes: las antenas que se incorpore y el voltaje de funcionamiento con el que se alimente a los dispositivos (Espinosa, 2015). Las características de estos dispositivos son las siguientes:

#### **TRANSMISOR MX-FS-03V Y RECEPTOR MX-JS-05V**

- Alcance: 20 a 200 m dependiendo de las antenas y de los voltajes de alimentación
- Alimentación: 3.5-12 V
- Tasa de transmisión: 4 KB/S

- Potencia de transmisión: 10 mW
- Frecuencia de transmisión: 433 MHz
- Consumo de corriente: 4 mA
- Módulo de operación: AM
- Sensibilidad de operación: -105 dB
- Antenas recomendadas: 25 cm Tx y 32 cm Rx



**Figura 2-8** Módulos de transmisión de Radio  
Frecuencia RF 433 MHz.  
Fuente: ESPINOSA, 2015

#### 2.2.3.4 Reloj en Tiempo Real RTC (DS1307)

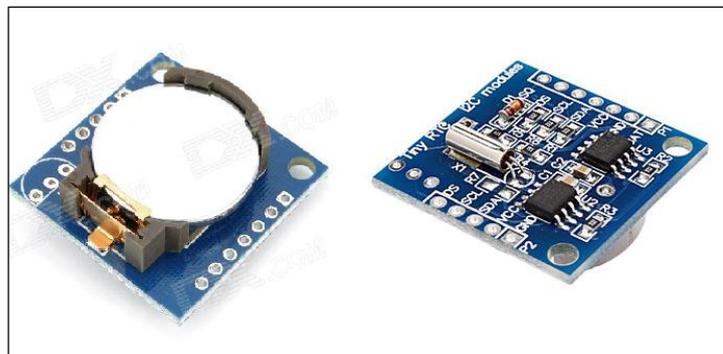
El módulo es diseñado para establecer la fecha y hora en tiempo real, trabaja con el bus I2C, es autónomo porque cuenta con una batería para memorizar los datos ingresados y una memoria EEPROM 24C32, posee muy bajo consumo de energía. El dispositivo incorpora un circuito de sensor de tensión que detecta fallas de energía (García, 2009; Narváez, 2012). Su tamaño es cómodo y sencillo de manipularlo como se ve en la *Figura 2-9*.

Las características de este dispositivo son:

- Cuenta segundo, minutos y horas y a la vez los días, meses y años.
- Interfaz Serie I2C.
- La salida entrega una onda cuadrada.
- Consume menos de 500 mA en la batería.
- Dispones de 8 terminales de conexión.
- Formato de 12 horas con indicador AM/PM o de 24 horas.
- 56 bytes de RAM no volátil, para almacenamiento de datos

La distribución de terminales es la siguiente:

- VCC y GND: Es la alimentación requerida por el RTC que es de 5 V para VCC.
- VBAT: Es entrada de alimentación para la pila de 3 V, este voltaje no puede pasar del rango de 2.5 y 3.5 para que no presente fallas en su operación.
- SCL: Entrada para la sincronización de la transferencia de datos en la interfaz serial.
- SDA: Entrada y salida de datos por medio de I2C.
- Consta de un cristal de cuarzo de 32.768 Hz.



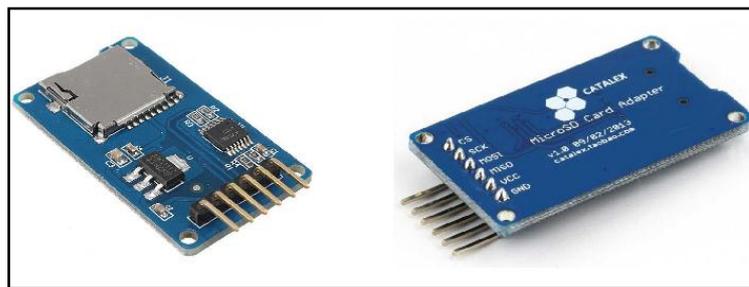
**Figura 2-9** Módulo reloj de tiempo real RTC DS1307 I2C.  
Fuente: GARCIA, 2009; NARVÁEZ, 2012

### 2.2.3.5 *Shield micro SD*

Es un módulo para la lectura, escritura y edición de archivos dentro de un dispositivo de almacenamiento SD, es compatible con la tarjeta de desarrollo Arduino en todas sus presentaciones y con tarjetas de memoria micro SD de hasta 32 GB que sean de alta velocidad, funciona con una alimentación de entre 3.3 a 5 V.

La interfaz de comunicación es SPI, que permite la comunicación entre circuitos integrados que maneje flujo de datos en bits (Owens, 2013). Las dimensiones del módulo son 4,1 x 2,4 cm como se observa en la *Figura 2-10*. Su distribución de terminales es la siguiente:

- VCC/GND (Alimentación)
- MISO (Es la salida de datos de Slave y entrada al Master)
- MOSI (Salida de datos desde el Master y entrada de datos al Slave)
- SCK (Es el pulso que marca la sincronización, se envía un bit con cada pulso de reloj)
- CS (Activación por parte del Master al Slave)



**Figura 2-10** *Shield* micro SD para Arduino.

Fuente: OWENS, 2013

### 2.2.3.6 *Equipos de Visualización*

La pantalla LCD 16x2 es un dispositivo que se emplea para la visualización de información, consta de un microcontrolador que dirige todo el funcionamiento. Dispone de 2 filas y 16

columnas, debe ser alimentado con 5 V. En la *Figura 2-11* se aprecia el aspecto de la pantalla LCD.

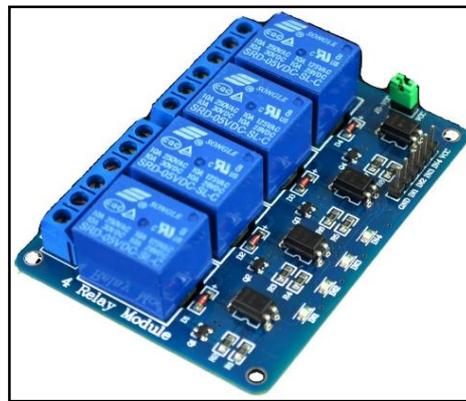


**Figura 2-11** Pantalla de visualización

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

#### 2.2.3.7 Módulos Relé para Arduino

Estos relés se activan con los 5 V de salida del terminal del Arduino para dar paso a una alimentación de 110 V. En la *Figura 2-12* se visualiza la apariencia de los módulos relé para Arduino que soportan hasta una tensión de 250 V y 10 A (Tolocka, 2015).



**Figura 2-12** Módulo Relé para Arduino.

Fuente: TOLOCKA, 2015

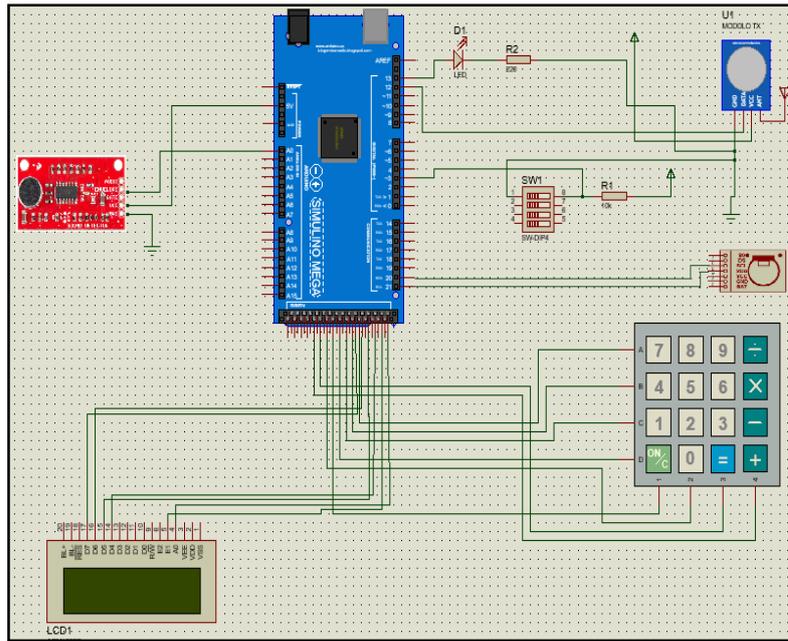
#### 2.2.4 Esquema de conexión del Sistema Inalámbrico de Sensores

Se han presentado los componentes de cada uno de los módulos que conforman el sistema inalámbrico de sensores, a continuación se detallan las conexiones realizadas.

#### 2.2.4.1 Diagrama de conexión del Módulo de Supervisión

Las conexiones realizadas en el módulo de supervisión se puede observar en la *Figura 2-13*. El equipo de procesamiento es el Arduino MEGA que tiene sus terminales conectadas a los componentes de la siguiente manera:

- El Sensor de ruido Sound Detector Sparkfun tiene conectada su salida (*envelope*) al terminal analógico A0 del Arduino con una alimentación propia de 5 V y GND.
- El dispositivo transmisor de RF MX-FS-03V es conectado a 5 V de una batería externa para lograr alcanzar una mayor distancia de transmisión y conectarlo a un solo GND en común de todo el sistema. La salida de datos desde el Arduino es por el terminal 12 para ser transmitida por RF a través de su entrada DATA.
- El reloj RTC DS1307 se lo conecta al Arduino a través de los terminales 20 (SDA) y 21 (SCL) con las respectivas alimentaciones de 5 V y GND.
- Para igualar la hora y fecha del sistema se conecta un teclado al Arduino a los terminales 31, 33, 35, 37, 39, 41, 43, 45 para el ingreso de datos, el que se realiza después de activar una interrupción a través de un *switch* que está conectado al terminal 3 del procesador.
- Para tener un control por medio de un dispositivo de visualización se conectó una pantalla LCD al Arduino por medio de los terminales 22(RS), 24(E), 26(DB4), 28(DB5), 30(DB6) y 32(DB7). La alimentación es de 5 V y con ayuda de una resistencia variable controlamos la luminosidad de la LCD.
- Al terminal 13 se conecta un *led* que visualiza la transmisión de datos y funciona conjuntamente con el dispositivo RF.



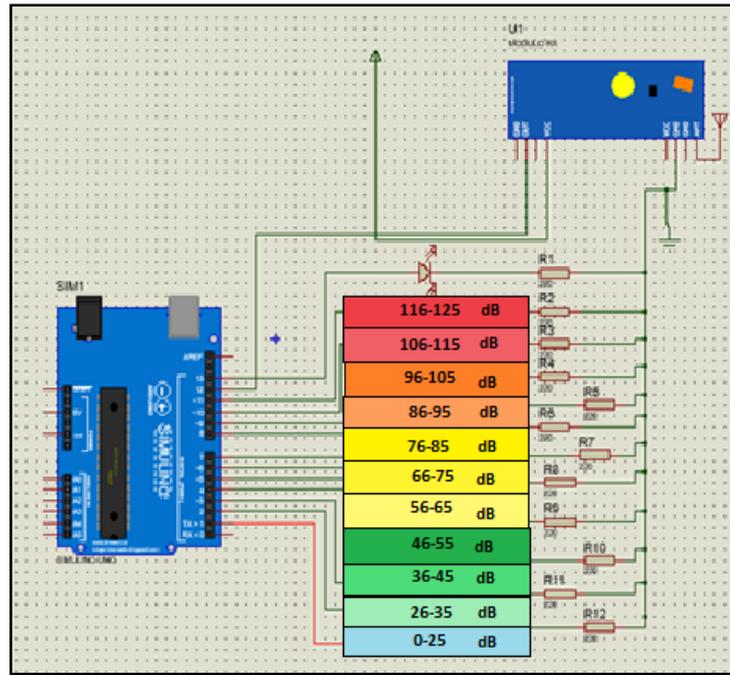
**Figura 2-13** Esquema del Módulo de supervisión.

**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

#### 2.2.4.2 Diagrama de conexión del Módulo de Aviso

Para este módulo se trabaja con un Arduino MEGA que actúa como procesador, interpreta la información recibida desde el módulo de supervisión a través de RF como se muestra en la *Figura 2-14*. Las conexiones son las siguientes:

- Terminal DATA del dispositivo receptor conectado al terminal 12 del Arduino MEGA.
- Terminal 13 del procesador conectado a un *led* que indica el funcionamiento del receptor de RF.
- La parte principal del módulo de aviso es el vúmetro que sirve para visualizar la información obtenida de las mediciones sobre los niveles de ruido, consta de 11 salidas del Arduino MEGA que representa rangos en decibeles. Están conectadas a relés que permiten la interacción de los voltajes de salida de Arduino y permitir el paso de 110 V, voltaje requerido para el funcionamiento de las lámparas del vúmetro. Los terminales digitales que se conectan son del 1 al 11.

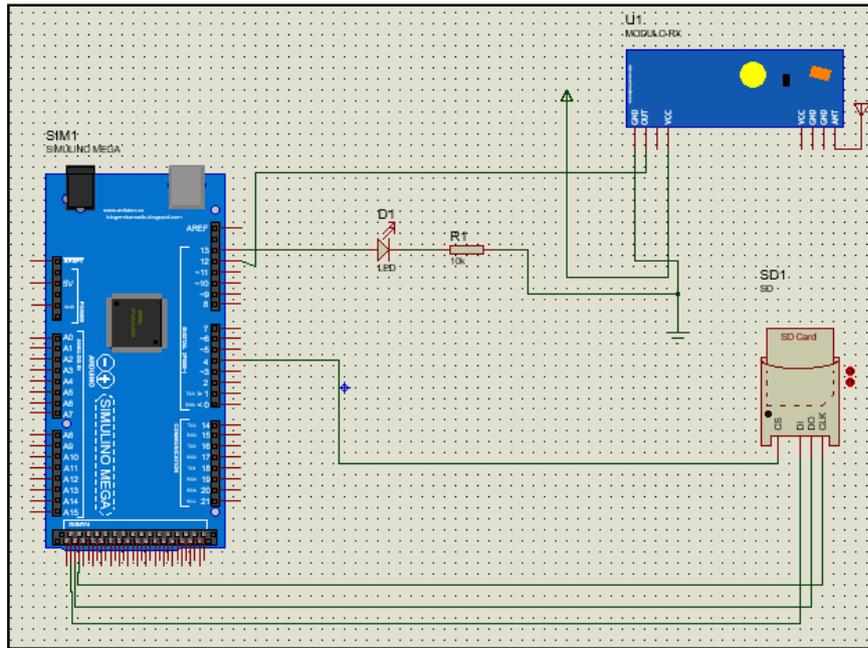


**Figura 2-14** Esquema del Módulo de Aviso.  
Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016

### 2.2.4.3 Diagrama de conexión del Módulo de Almacenamiento

En la *Figura 2-15* se observa las conexiones del módulo de almacenamiento que consta de tres dispositivos: el Arduino UNO, el dispositivo receptor de RF MX-JS-05V y de la *shield* micro SD en donde se almacenará en una base de datos los niveles de ruido obtenidos en las mediciones durante cierto periodo. Las conexiones son las siguientes:

- Al terminal 12 del Arduino UNO se encuentra conectado el terminal DATA del receptor RF con su respectiva alimentación de 5 V.
- El *shield* micro SD tiene sus terminales conectados de la siguiente manera: terminal 50 (MISO), 51 (MOSI), 52 (SCK) y el terminal 4 (CS) con una alimentación de 5 V.
- En este módulo se encuentra conectado el terminal 13 del Arduino UNO con un *led* que indica el funcionamiento del receptor de RF.

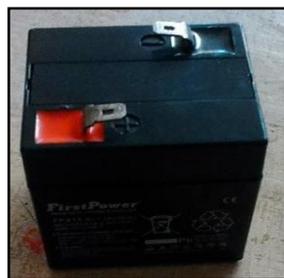


**Figura 2-15** Esquema del Módulo de Almacenamiento.  
**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 2.2.5 Alimentación de la Red Inalámbrica de Sensores de contaminación acústica.

La alimentación para los módulos del sistema se los realiza por medio de la red eléctrica disponible en la ESPOCH, lo que asegura el continuo funcionamiento del sistema. El Arduino del módulo de supervisión se conecta por medio de un adaptador AC-DC Huawei que entrega 12 V y 1 A evitando así las caídas de tensión y facilitando el funcionamiento constante.

Los Arduinos de los módulos de aviso y almacenamiento son alimentados con baterías recargables secas de 6 V y 2.8 A como se observa en la *Figura 2.16*. Los reguladores que poseen los procesadores entregan 5 V de salida necesaria para alimentar cada componente del módulo. Se utiliza una alimentación de 5 V para el transmisor RF.



**Figura 2-16** Batería recargable seca de 6 V y 2.8 A  
**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 2.3 Requerimientos Software de la Red Inalámbrica de Sensores de contaminación acústica

En esta sección se describirá los requerimientos *software* del SISCA para la Escuela Superior Politécnica de Chimborazo. Las necesidades en cuanto a *software* del sistema son los siguientes:

#### Requerimientos software para el Módulo de Supervisión

- Trabajar con la salida analógica “*envelope*” del sensor de ruido y convertirla en señales digitales que van de 0 a 1023 para luego convertirla a decibeles.
- Programar para al cambiar el estado del *switch* externo se realice una interrupción que facilite la actualización de fecha y hora del sistema por medio del teclado matricial.
- Visualizar la información de hora, fecha y valores actuales de los niveles de ruido por medio de la LCD.
- Encontrar valores máximos y mínimos de los niveles de ruido medidos por el módulo de supervisión.
- Enviar desde el módulo de supervisión la cadena de información que contiene: Nodo, fecha/hora, valor actual, valor máximo y valor mínimo por medio de la comunicación de RF a los módulos de aviso y de almacenamiento.
- Suministrar la información recibida por el puerto USB del Arduino.

#### Requerimientos software para el Módulo de Aviso

- Receptar la información desde el módulo de supervisión por medio del receptor RF conectado al procesador Arduino que se encuentra en forma de cadena de caracteres ASCII y extraer los valores actuales.
- Visualizar en el vúmetro los datos de valores actuales de ruido ambiental, como un elemento de aviso a los usuarios de los niveles de contaminación acústica.

- Suministrar la información recibida por el puerto USB del Arduino.

#### Requerimientos software para el Módulo de Almacenamiento

- De igual manera que en el módulo de aviso es necesaria la recepción de información desde los sensores por medio del dispositivo RF para administrarla y almacenarla.
- Acceder a la tarjeta de memoria micro SD para almacenar la información recibida en una base de datos desglosada en los siguientes campos: Nodo, fecha/hora, valores actuales, valores máximos y valores mínimos. todo esto se realiza en la tarjeta de memoria micro SD implementada en este módulo.
- Suministrar la información recibida por el puerto USB del Arduino.

### 2.3.1 *Software de desarrollo utilizado para la programación del sistema*

El encargado de manejar el funcionamiento de los módulos implementados es un *software* de desarrollo que contiene una programación estructurada que trabaja con funciones. Se utilizó dos herramientas de programación para lograr el funcionamiento de prototipo las cuales son:

#### 2.3.1.1 *Software para la programación de las tarjetas de desarrollo Arduino*

Con esta herramienta se realiza la programación de las tarjetas de desarrollo en este caso los Arduinos para que trabajen de acuerdo a lo requerido. Se utiliza el IDE de Arduino, es un entorno de desarrollo basado en *Processing* en un lenguaje de programación que trabaja con Wiring y con el arranque bootloader.

Este *software* simula el entorno de programación C++ para una mejor comprensión y fácil implementación. Los archivos aquí creados son llamados *sketch*, que están conformados por instrucciones que trabajan en conjunto con bibliotecas (Martinez, 2013). Este *software* libre permita el control del proceso en el momento de la ejecución de un programa cargado sobre la tarjeta de desarrollo.

Consta de dos estructuras importantes: *void setup ()* y *void loop ()*, el primero ejecuta una sola vez sus instrucciones, por lo general es para declaraciones de variables, constantes, salidas de terminales, etc. La segunda se encarga de hacer las repeticiones constantes de las instrucciones que contiene (García González, 2015).

#### 2.3.1.2 *Software LabVIEW*

Es una herramienta de programación de forma visual lo que hace más sencillo el trabajo hacia el usuario, es utilizado para la simulación, control y diseño en tiempo real, la comunicación se la realiza por medio del puerto USB del computador y con los puertos seriales.

El creador de este programa es National Instruments que utiliza el lenguaje G, permite el control externo de equipos *hardware* (National Instruments, 2016). Incluye una herramienta denominada VISA que es encargada de trabajar por medio de los puertos seriales del computador y de las tarjetas de desarrollo.

#### 2.3.2 *Programaciones realizadas en cada módulo del sistema inalámbrico*

El código realizado en el *software* de desarrollo IDE 1.6.8 para cada uno de los módulos del sistema se los presenta en los Anexos 2, 3 y 4. A continuación se presenta los diagramas de flujo, las respectivas funciones, variables y bibliotecas utilizadas en cada una de las programaciones.

##### 2.3.2.1 *Diagrama de Flujo del Módulo de Supervisión*

El diagrama de flujo del programa que se ejecuta en el módulo de almacenamiento se lo indica en la *Figura 2-17*. De acuerdo a lo programado en el *sketch* se realiza:

Declaración de variables e inicialización de procedimientos:

- Integración de bibliotecas conjuntamente con la declaración de variables necesarias para el programa.

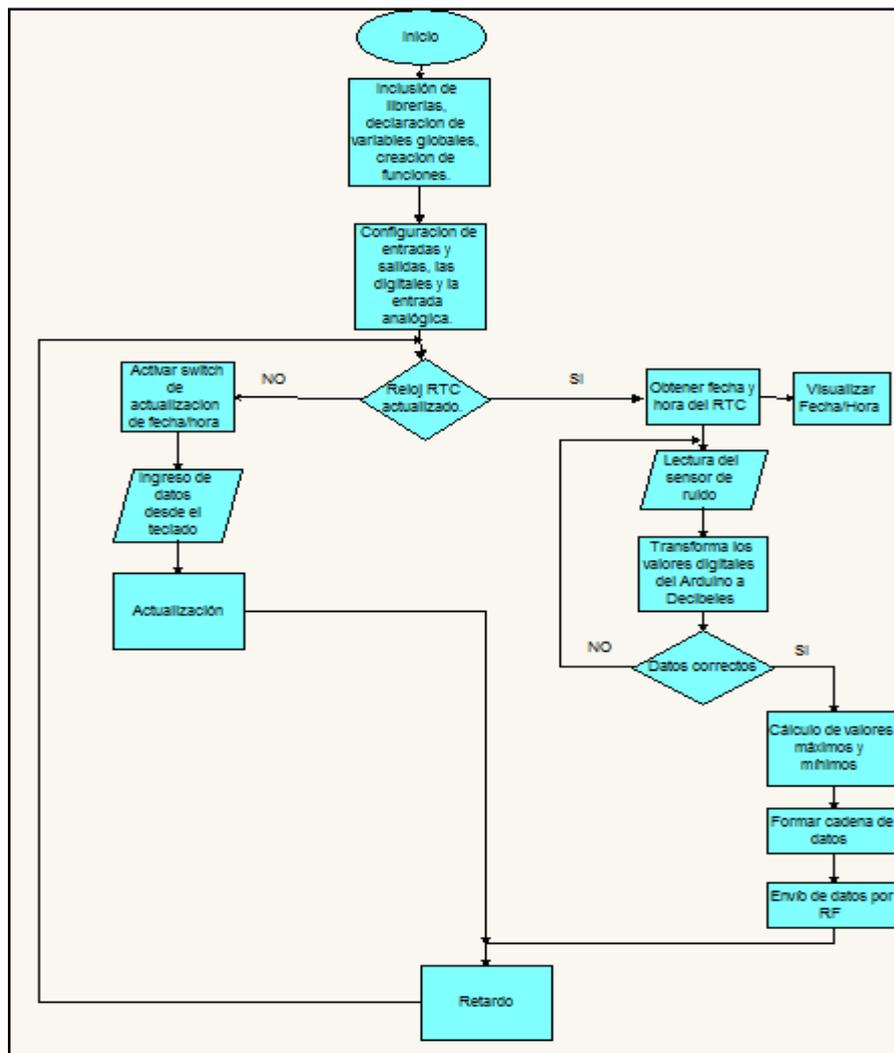
- Configuración de entradas y salidas dependiendo de las necesidades de los equipos integrados a la tarjeta de desarrollo como: teclado, LCD y la interrupción por parte de *switch*.
- Se declaran las funciones a utilizar dependiendo de los procedimientos a implementar.

Procedimientos repetitivos de manera indefinida:

- Para el procedimiento de identificación del Nodo y actualización de fecha y hora del sistema se utiliza la función **IgualarFH ()** luego de la interrupción solicitada por medio del *switch* y de la función **BotonConfiguraFecha ()**, caso contrario el RTC interviene y entrega la hora almacenada en su memoria.

Al momento de encender el modulo siempre iniciara con un número de Nodo igual a 0, para lo cual se hace la respectiva configuración para cambiar su identificación.

- La lectura del sensor se la realiza desde el terminal analógico A0, estos datos ingresan a la función **calculodecibels ()** que la convierte de valores digitales a decibelios. Aquí se identifica los valores máximos y mínimos.
- Para la transmisión de información por RF se forma una cadena de caracteres ASCII con la fecha y hora por medio de la función **Formateo ()**.
- Se crea una cadena principal que contiene información de nodo, fecha/hora, valor actual de ruido, valor máximo y valor mínimo que es transmitida por RF y al puerto USB. La función que realiza este trabajo es **CrearMensajeEnviar ()**.
- La cadena estructurada es transmitida por RF a los módulos restantes por medio de la función **EnvioRF ()**. Este proceso consta de retardos para evitar pérdida de información durante la transmisión.



**Figura 2-17** Diagrama de flujo del programa para el Módulo de Supervisión.  
Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 2.3.2.2 Bibliotecas usadas en la programación del Módulo de Supervisión

Para el *software* del módulo se requiere varias funciones que están incluidas en bibliotecas previamente establecidas, las que se declaran a continuación:

- **LiquidCrystal.h:** Es la biblioteca encargada de controlar el funcionamiento de las pantallas LCD conectadas a las placas de Arduino, se basa en el chip Hitachi HD44780 que está ubicado en cada una de las pantallas. Esta librería trabaja con 4 u 8 bits de velocidad (Arduino, 2016).

- **VirtualWire.h:** Encargado del funcionamiento de los módulos de RF por medio de funciones pre-determinadas (McCauley, 2015).
- **Wire.h:** Programado para trabajar con el módulo de reloj RTC, proporciona las funciones necesarias para acceder a fecha y hora sean estas tomadas del compilador o en este caso ajustadas desde un teclado (Arduino, 2016).
- **Keypad.h:** Con esta biblioteca se hace el control de las funciones del teclado que se utiliza para ajustar el reloj (Arduino, 2016).
- **RTCLib.h:** Por medio de funciones trabaja con el reloj RTC DS1307 (Arduino, 2016).

### 2.3.2.3 *Funciones que se incluyen en el sketch del Módulo de Supervisión*

- **BotonConfiguraFecha ():** Esta función es la encargada de realizar la interrupción para ajustar el RTC por medio del teclado. Cuando este se activa se interrumpe la transmisión de información y se procede a igualar la fecha y hora.
- **IgualarFH ():** Dentro de esta función se encuentra todo el procedimiento para recibir los datos desde el teclado para igualar el reloj RTC, cuando se pulsa el botón \* se ingresa al modo de ajuste de reloj.
- **convertir (letra):** Esta función convierte los pulsos del teclado en números enteros para ser compatibles con el ajuste del reloj, la variable **letra** recibe los pulsos del teclado.
- **calculodecibels (x, c):** Esta función convierte a decibeles un valor digital, se lo realiza por medio de la fórmula **decibelscalculados = 20 \* log10(x/dB)**, en donde la variable **c** es la letra que representa el estado del ruido sea este bajo (b), medio (m) y alto (a), y la letra **x** es la variable que representa el valor que ingresa desde el sensor.
- **FormaFeHo (FeHo []):** Es la función encargada de crear la cadena que contiene la fecha y hora del sistema. En la variable **FeHo []** es donde se almacena la cadena, los símbolos de corchetes indican la dimensión de la cadena que son 21 caracteres.

- **CrearMensajeEnviar (Nodo, FeHo [], valordB, maxdB, mindB, Cadena []):** Se encarga de crear la cadena de caracteres ASCII final que será enviada por RF, la cual requiere de los siguientes parámetros:

**Nodo:** Indica la zona donde se encuentra ubicado un módulo de sensores.

**FeHo []:** Es la cadena que almacena la fecha y hora.

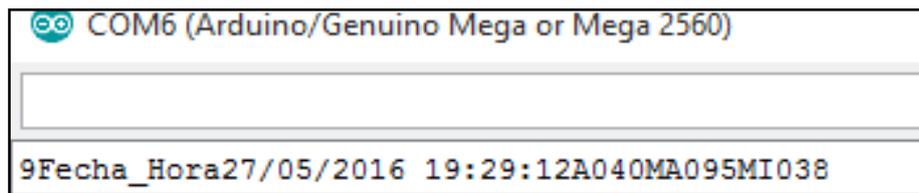
**valordB:** Es el valor actual del nivel de ruido medido.

**maxdB:** Es el valor de ruido máximo que se ha registrado en un determinado tiempo.

**mindB:** Es el valor mínimo de ruido registrado durante un periodo de tiempo.

**Cadena []:** Es la variable en donde se almacena la cadena creada con una dimensión de 46 caracteres.

En la *Figura 2-18* se presenta la estructura de la cadena ASCII para ser enviada por RF y por puerto USB.



**Figura 2-18** Estructura de la cadena ASCII para ser enviada por RF y por puerto USB.  
Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

- **EnvioporRF (Cadena []):** Esta función envía la información por medio del dispositivo transmisor RF, donde **Cadena[]** es el conjunto de caracteres a ser transmitido.
- **CadenaenSerial (Cadena []):** Esta función se encarga de escribir la cadena recibida en el puerto USB, donde **Cadena[]** es el conjunto de caracteres a ser transmitido.

#### 2.3.2.4 Diagrama de Flujo del Módulo de Aviso

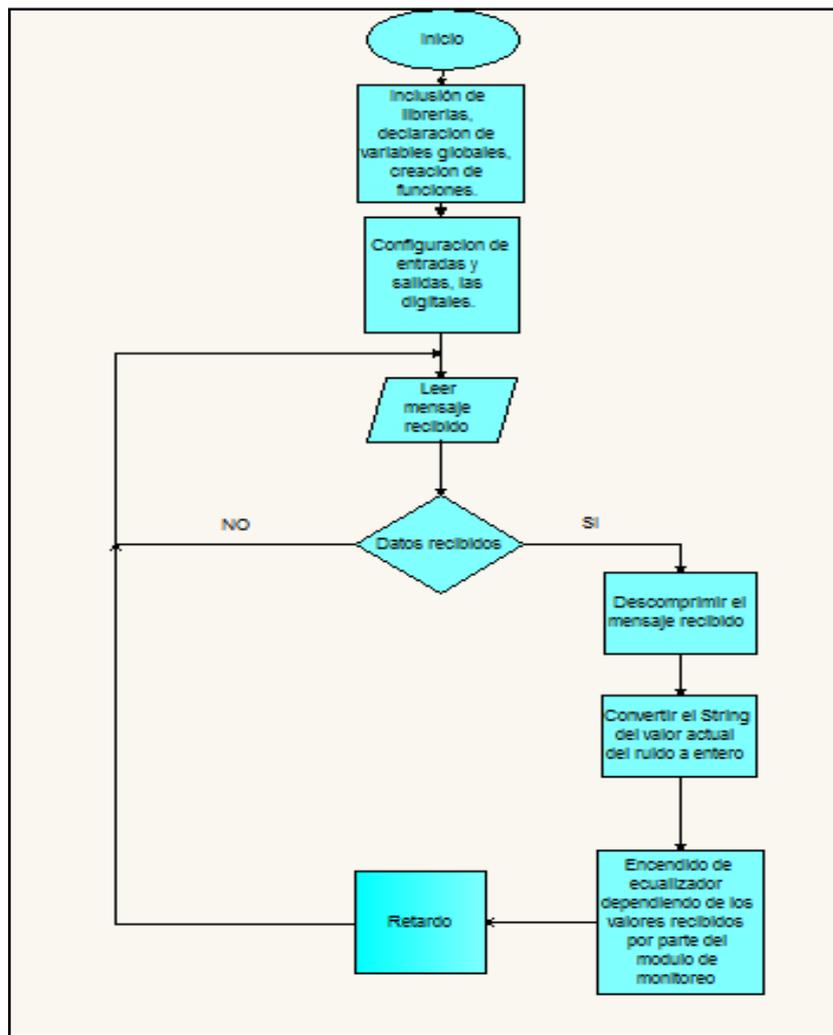
El diagrama de flujo del programa que se ejecuta en el módulo de almacenamiento se lo indica en la *Figura 2-19*. De acuerdo a lo programado en el *sketch* se realiza:

Proceso de inicialización:

- Integración de bibliotecas conjuntamente con la declaración de variables necesarias para el programa.
- Configuración de entradas y salidas dependiendo de las necesidades de los equipos integrados a la tarjeta de desarrollo.
- Se declaran las funciones a utilizar dependiendo de los procedimientos que se quieran realizar.

Proceso repetitivos de manera indefinida

- Recibir la información desde el módulo de supervisión por medio de RF con intervención del procesador Arduino.
- Leer la información recibida y por medio de las funciones **LeerMensajeRF ()** y **DescomprimirMensaje ()** extraer la información.
- Convertir la información del valor actual expresado en cadena a un valor numérico con la instrucción: **int valordb = atoi(valordB.c\_str());**
- Visualizar en el vúmetro la escala respectiva al valor numérico por medio de la función **ecualizador ()**.



**Figura 2-19** Diagrama de flujo del programa para el Módulo de Aviso.  
 Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 2.3.2.5 Bibliotecas usadas en la programación del Módulo de Aviso

Para el *software* del módulo se requiere una función que está incluida en la biblioteca **VirtualWire.h** descrita en la capítulo anterior

### 2.3.2.6 Funciones que se incluyen en el sketch del Módulo de Aviso

- **DescomprimirMensaje(Cadena, Nodo, FeHo, valordB, maxdB, mindB):** Esta función es la encargada de descomprimir la información recibida por RF, se obtiene los valores

transmitidos en forma individual y se asigna a los parámetros respectivos de la función. Los parámetros que intervienen en esta función son similares a los descritos en la función **CrerMensajeEnviar()**.

- **LeerMensajeRF (Cadena [])**: Esta función realiza la lectura de la información recibida por RF y la almacena en el parámetro **Cadena** que es de tipo *string*.
- **ecualizador (valordb)**: Esta función visualiza los niveles actuales de contaminación acústica que se ingresan con parámetro en la variable **valordb** que es de tipo entero.

### 2.3.2.7 Diagrama de Flujo del Módulo de Almacenamiento

El diagrama de flujo del programa que se ejecuta en el módulo de almacenamiento se lo indica en la *Figura 2-20*. De acuerdo a lo programado en el *sketch* se realiza:

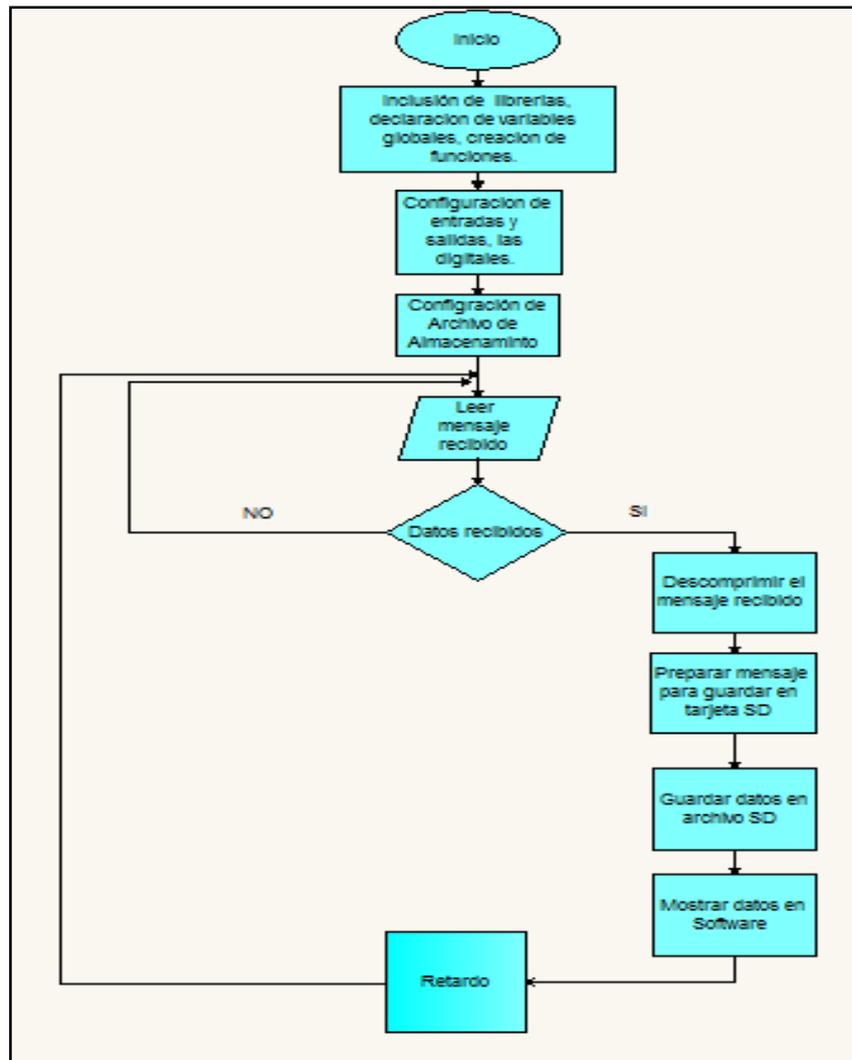
#### Proceso de inicialización

- Integración de bibliotecas conjuntamente con la declaración de variables necesarias para el programa.
- Configuración de entradas y salidas dependiendo de las necesidades de los equipos integrados a la tarjeta de desarrollo.
- Se declaran las funciones a utilizar dependiendo de los procesos a ejecutar.
- Se crea el archivo en la micro SD y su encabezado con la función **configurararchivo ()**. Cada vez que el dispositivo de almacenamiento no contenga el archivo.

#### Proceso repetitivos de manera indefinida

- Recibe la información desde el módulo de supervisión por RF.
- Lee la información recibida y separa sus valores en variables individuales por medio de las funciones **LeerMensajeRF ()** y **DescomprimirMensaje ()**.

- Forma la línea de información en el formato necesario para ser escrita en el archivo que contiene la micro SD, por medio de la función **FormarMensajeParaSD ()**.
- Escribe la línea de información en el archivo que contiene la micro SD, utilizando la función **EscribeDatosEnSD ()**.



**Figura 2-20** Diagrama de flujo del programa para el Módulo de Almacenamiento.  
**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 2.3.2.8 Bibliotecas usadas en la programación del Módulo de Almacenamiento

Para el *software* del módulo se requiere varias funciones que están incluidas en bibliotecas previamente establecidas, una de ellas es **VirtualWire.h** descrita en acápites anteriores y las bibliotecas:

- **SPI.h:** Se encarga de controlar los dispositivos SPI (Serial Peripheral Interface por sus siglas en inglés). Permite trabajar con la *shield* micro SD (Arduino, 2016).
- **SD.h:** Es la encargada de la lectura, escritura y edición de información de un archivo ubicado en la micro SD. Esta biblioteca controla el funcionamiento de la *shield* SD (Arduino, 2016).

### 2.3.2.9 Funciones que se incluyen en el sketch del Módulo de Almacenamiento

Se utilizan cinco funciones de las cuales **DescomprimirMensaje()** y **LeerMensajeRF()** han sido descritas en acápites anteriores, las restantes se las describe a continuación:

- **FormarMensajeParaSD (MSD, Nodo, FeHo, valordB, maxdB, mindB):** Esta función estructura la cadena de caracteres que contendrá el parámetro **MSD**. Que es la información recibida desde el módulo de supervisión y que es recibida como parámetros. Donde:
  - Nodo:** Indica la zona donde se encuentra ubicado un módulo de sensores.
  - FeHo:** Es la cadena que almacena la fecha y hora.
  - valordB:** Es el valor actual del nivel de ruido medido.
  - maxdB:** Es el valor de ruido máximo que se ha registrado en un determinado tiempo.
  - mindB:** Es el valor mínimo de ruido registrado durante un periodo de tiempo.
- **EscribirDatosEnSD (MSD, Datos):** Es la función encargada de realizar el proceso de escritura de la línea de información contenida en el parámetro **MSD** en el archivo cuyo nombre se encuentra en el parámetro **Datos** que se localiza previamente creada en la memoria micro SD. En la *Figura 2-21* se muestra la información almacenada en el archivo creado en la memoria micro SD.

NIVELES DE RUIDO REGISTRADOS EN LA ESPOCH					
NODO	FECHA	HORA	R-ACTUAL	R-MAXIMO	R-MINIMO
9	27/05/2016	19:25:09	042	095	038
9	27/05/2016	19:25:10	040	095	038
9	27/05/2016	19:25:11	040	095	038
9	27/05/2016	19:25:13	040	095	038
9	27/05/2016	19:25:14	038	095	038
9	27/05/2016	19:25:15	040	095	038
9	27/05/2016	19:25:16	040	095	038
9	27/05/2016	19:25:18	041	095	038
9	27/05/2016	19:25:19	041	095	038
9	27/05/2016	19:25:20	040	095	038
9	27/05/2016	19:25:21	040	095	038
9	27/05/2016	19:25:22	040	095	038
9	27/05/2016	19:25:24	041	095	038
9	27/05/2016	19:25:25	040	095	038
9	27/05/2016	19:25:26	040	095	038
9	27/05/2016	19:25:27	040	095	038
9	27/05/2016	19:25:29	040	095	038
9	27/05/2016	19:25:30	040	095	038
9	27/05/2016	19:25:31	040	095	038
9	27/05/2016	19:25:32	040	095	038
9	27/05/2016	19:25:33	041	095	038

**Figura 2-21** Información almacenada en el archivo creado en la memoria micro SD.

**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

- **Configurararchivo (Datos):** Es la encargada de crear el archivo, con el nombre almacenado en el parámetro **Datos**, que contendrá la tarjeta de memoria micro SD y escribir las dos primeras líneas de encabezado que contendrá el archivo. En la *Figura 2-22* se muestra el encabezado del archivo.

NIVELES DE RUIDO REGISTRADOS EN LA ESPOCH					
NODO	FECHA	HORA	R-ACTUAL	R-MAXIMO	R-MINIMO

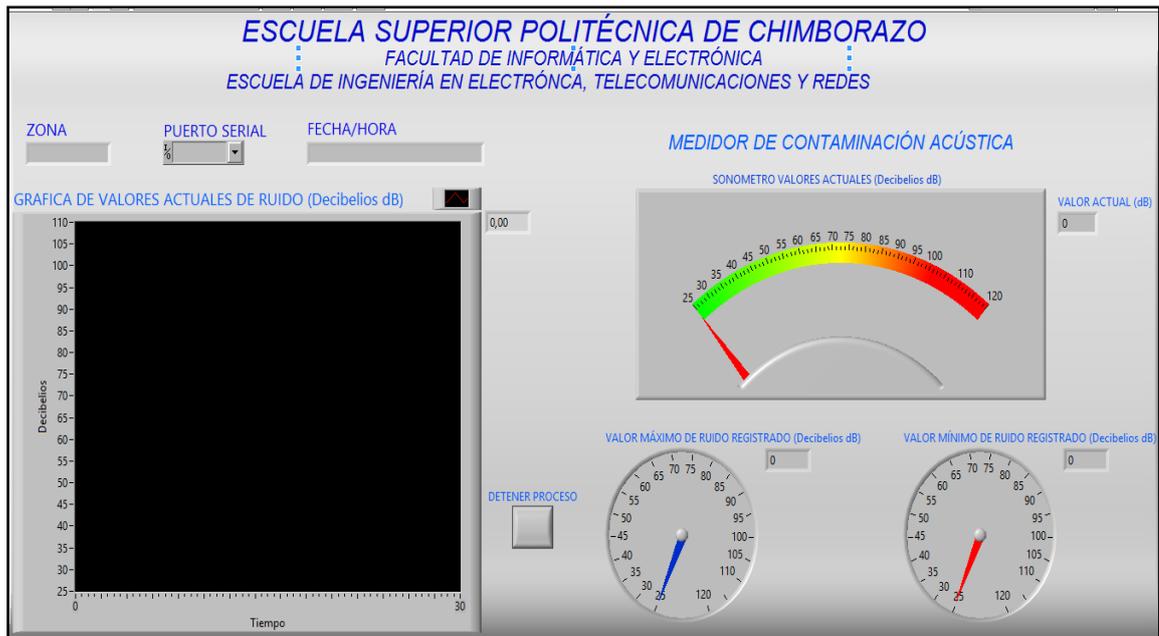
**Figura 2-22** Encabezado del archivo.

**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 2.3.3 *Software para la presentación de información en modo gráfico del SISCA.*

SISCA dispone de una herramienta virtual desarrollada en LabVIEW 2015, que permite visualizar la información censada por el módulo de supervisión y suministrada por el puerto USB de cada uno de los módulos creados.

Permite observar los niveles de ruido en forma gráfica y numérica y almacenarla en el computador como un archivo con formato similar descrito en acápite anteriores. El panel frontal de la herramienta virtual desarrollada en LabVIEW se observa en la *Figura 2-23*.

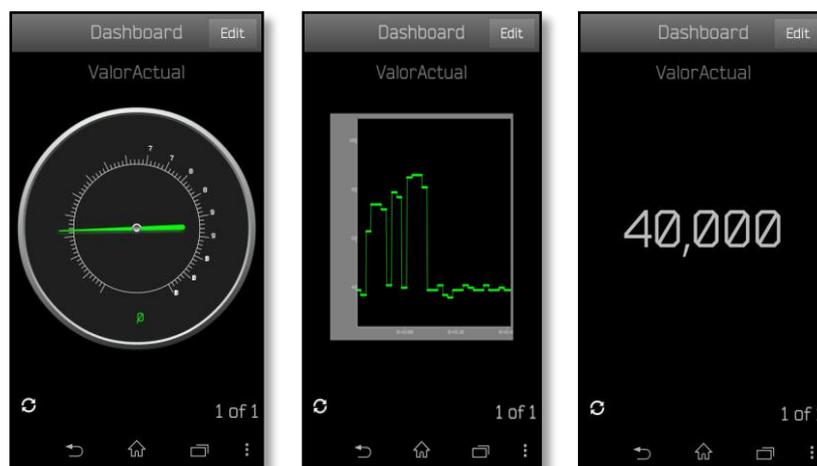


**Figura 2-23** Panel frontal de la herramienta virtual desarrollada en LabVIEW.

**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

SISCA trabaja con la aplicación móvil Data Dashboard desarrollada por *National Instruments* para dispositivos móviles con sistema operativo Android, la que permite compartir por medio de la red Wi-Fi las variables globales utilizadas por herramientas virtuales desarrolladas en LabVIEW.

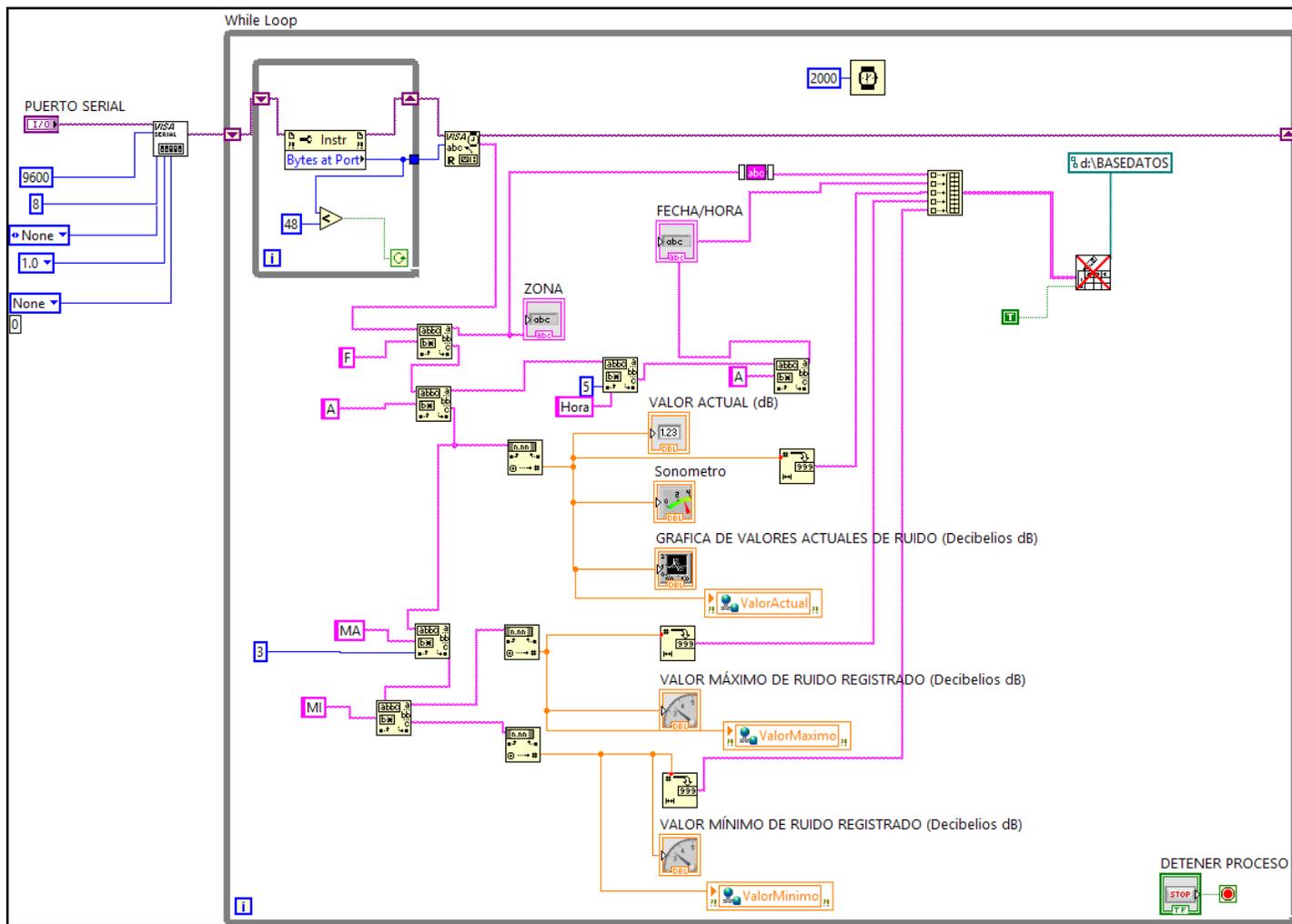
Con esta herramienta se facilita la visualización en forma gráfica y numérica de los valores censados por el módulo de supervisión en todos los dispositivos móviles que compartan la red Wi-Fi. En la *Figura 2-24* se observa la aplicación Data Dashboard supervisando las variables globales de una herramienta virtual.



**Figura 2-24** Aplicación Data Dashboard al supervisando las variables globales de una herramienta virtual.

**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

La información es compartida al conectar el puerto USB del Arduino con el del computador y visualizada por la herramienta virtual desarrollada en LabVIEW. La cadena enviada por los módulos de supervisión consta de 46 caracteres ASCII que es decodificada, presentada y almacenada mediante la programación por bloques desarrollada con el *software* LabVIEW. El diagrama de bloques de la herramienta virtual implementada se presenta en la *Figura 2-25*.



**Figura 2-25** Diagrama de bloques de la herramienta virtual implementada.

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

## CAPÍTULO III

### 3. ANÁLISIS Y RESULTADOS

#### 3.1 Introducción

En el presente capítulo se muestra los resultados obtenidos en las pruebas realizadas al prototipo del SISCA y a cada uno de los módulos al integrar el *hardware* y *software* desarrollado. Se realizaron pruebas de funcionamiento comparándoles con un equipo de similares características. Se presenta análisis de costos del equipo detallado por cada uno de los módulos.

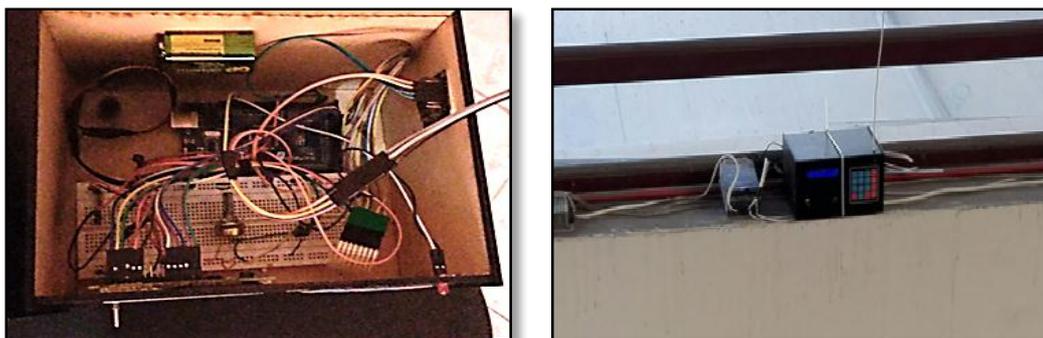
Las pruebas se realizaron con los datos de los niveles de contaminación acústica obtenidos en los predios de la Facultad de Informática y Electrónica durante un periodo de siete días.

#### 3.2 Pruebas del Módulo de Supervisión

Para corroborar al correcto funcionamiento del módulo de supervisión se realizaron pruebas de medición del sensor, la integridad de la información y el alcance de la transmisión.

##### 3.2.1 *Pruebas de las mediciones realizadas por el Módulo de Supervisión*

En la *Figura 3-1* se muestra el prototipo ensamblado del módulo de supervisión del SISCA, apreciándose sus respectivos.



**Figura 3-1** Prototipo del Módulo de Supervisión, vista interna (izquierda) y vista externa (derecha)

**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

Se realiza una comparación de los datos obtenidos por el módulo de supervisión del SISCA con un equipo de medición de emisiones acústicas denominado *Digital Sound Level Meter* el que se muestra en la *Figura 3-2*. El que tiene una precisión de  $\pm 1.5$  dB y un rango de medida de 30 dBA a 130 dBA trabajando en una frecuencia de 31.5 Hz a 8.5 KHz.



**Figura 3-2** Sonómetro Sound Level Meter

**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

Se presentan los resultados de los datos obtenidos en un periodo de tiempo de siete días calendario. De la información recabada se toman 19 datos de muestra para realizar el análisis, los cuales se detallan en la *Tabla 3-1*.

Las pruebas consisten en comparar las medidas entre los dos equipos y establecer el error lo que permitirá determinar si el equipo implementado tiene un comportamiento similar con el equipo comparado.

Del análisis se determina que el sistema implementado no incorpora un error superior a  $\pm 1.2$  dB lo que satisface los requerimientos de diseño previstos.

**Tabla 3-1** Mediciones realizadas con el sensor del sistema inalámbrico y con el sonómetro digital.

<b>Número de Muestra</b>	<b>Sonómetro Digital (dB)</b>	<b>SISCA (dB)</b>	<b>Error Absoluto (dB)</b>
<b>1</b>	71.0	72.0	1
<b>2</b>	74.6	75.0	0,4
<b>3</b>	71.2	70.0	-1,2
<b>4</b>	71.8	71.0	0,8
<b>5</b>	73.3	74.0	0,7
<b>6</b>	77.1	78.0	0,9
<b>7</b>	74.4	74.0	-0,4
<b>8</b>	71.1	72.0	0,9
<b>9</b>	62.1	61.0	-1,1
<b>10</b>	71.2	72.0	0,8
<b>11</b>	75.1	74.0	-1,1
<b>12</b>	73.2	74.0	0,8
<b>13</b>	72.7	72.0	-0,7
<b>14</b>	73.5	73.0	-0,5
<b>15</b>	61.3	62.0	0,7
<b>16</b>	60.0	61.0	1
<b>17</b>	49.5	50.0	0,5
<b>18</b>	61.5	61.0	-0,5
<b>19</b>	63.5	63.0	-0,5

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 3.2.2 *Repetitividad del Módulo de Supervisión*

Para este análisis se tomaron 15 muestras durante un periodo de un minuto que fueron registradas por la herramienta virtual desarrollada en LabVIEW y almacenadas en un archivo en el computador. Con estos valores se calculó la media, la desviación estándar y el coeficiente de variación ayudados por Excel 2013.

En la *Tabla 3-2* se muestran dichos cálculos, destacándose un coeficiente de variación del 2%, valor que determina que las mediciones realizadas por el sensor del SISCO exhibe poca variabilidad y determina la estabilidad del sistema, señalado así por (Adriana, 2009).

**Tabla 3-2** Tabla de repetitividad de los valores del Sistema Inalámbrico

<b>Número de Muestra</b>	<b>Hora</b>	<b>Valor Actual Ruido</b>
1	12:02:34	41
2	12:02:35	42
3	12:02:37	43
4	12:02:38	41
5	12:02:39	42
6	12:02:40	43
7	12:02:41	42
8	12:02:43	43
9	12:02:44	43
10	12:02:45	42
11	12:02:46	41
12	12:02:48	44
13	12:02:49	41
14	12:02:50	40
15	12:02:51	41
<b><i>Media</i></b>		<b><i>42</i></b>
<b><i>Desviación Estándar (<math>\sigma</math>)</i></b>		<b><i>1</i></b>
<b><i>Coefficiente de Variación</i></b>		<b><i>2%</i></b>

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 3.2.3 *Caracterización de las comunicaciones entre dispositivos del Módulo de Supervisión*

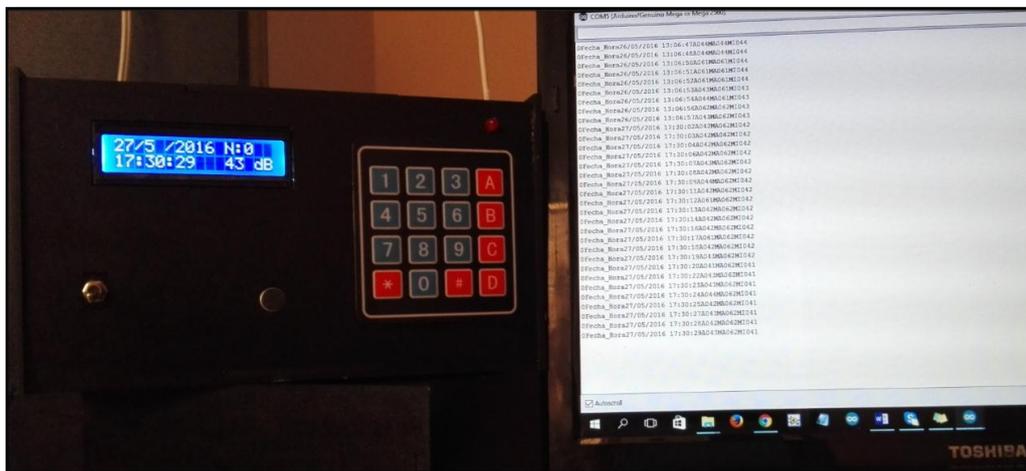
La comunicación entre los componentes que integran el módulo de supervisión es necesario evaluarla por lo que se procede a hacer las pruebas respectivas de comunicación con el RTC y el teclado.

- Pruebas de configuración del módulo

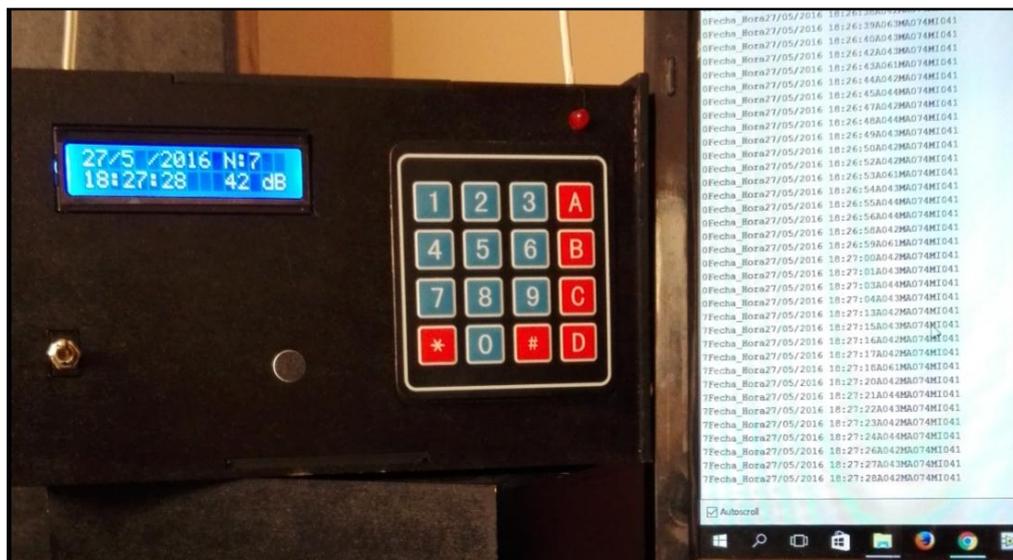
Las pruebas de configuración del módulo consisten en la identificación del nodo y el ajuste de la fecha y hora, estos procesos se describen a continuación:

- La combinación del switch y la tecla \* permite el ajuste de la fecha y hora como se observa en la *Figura 3-3*.
- La combinación del switch y la tecla # permite ingresar la identificación del nodo como se observa en la *Figura 3-4*.

Para las pruebas los mensajes de ajuste se los observa en la pantalla LCD y es supervisada por la pantalla del computador cuando se conecta el módulo por el puerto USB.



**Figura 3-3** Proceso de ajuste de Fecha y Hora en pantalla Serial y en pantalla LCD.  
Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

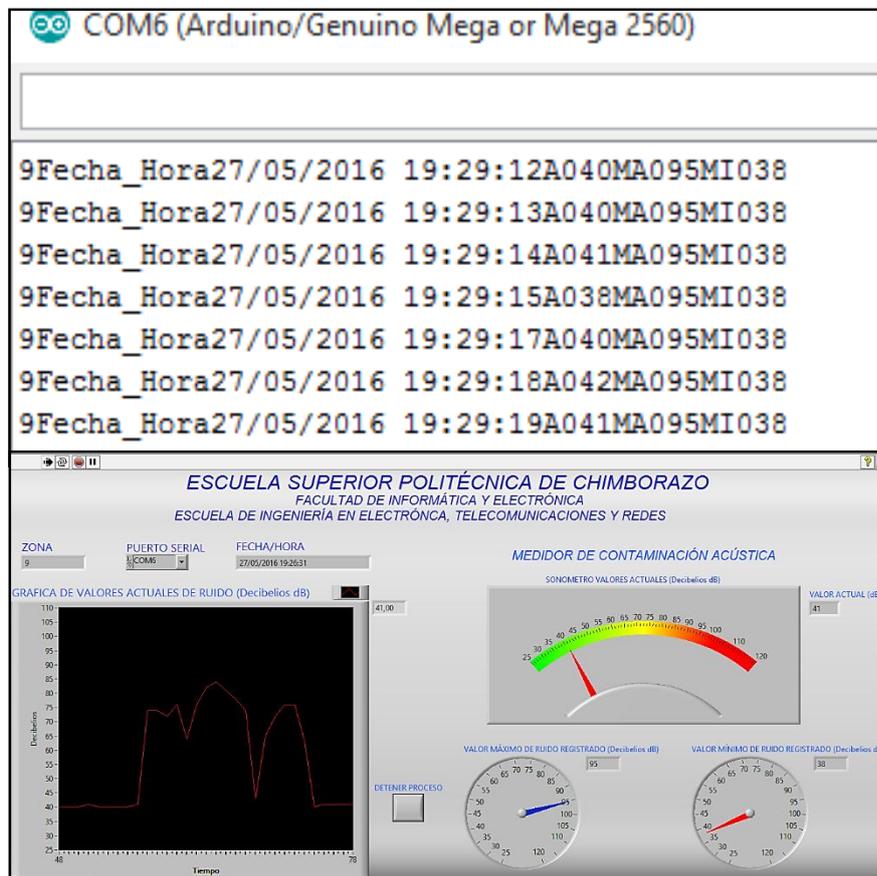


**Figura 3-4** Proceso de identificación del Nodo en pantalla Serial y en pantalla LCD.  
Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

- Evaluación de la herramienta virtual desarrollada en LabVIEW.

Las pruebas consistieron en conectar cada uno de los módulos que integran el SISCA al computador y observar la comunicación con la herramienta virtual desarrollada en LabVIEW comprobándose que no existe pérdida de datos en la comunicación por los puertos USB y el registro de la información en el archivo físico.

En la *Figura 3-5* se observa los valores censados en la pantalla serial del *software* Arduino y a su vez en la herramienta virtual desarrollada en un mismo instante de tiempo. Esto se logra con la aplicación VSPE (Virtual Serial Port Emulador por sus siglas en inglés) que permite crear puertos virtuales.



**Figura 3-5** Valores censados mostrados en la pantalla Serial (arriba) y Software LabVIEW (abajo)

**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

En la *Figura 3-6* se muestran la información almacenada en el archivo físico creado en el computador por la herramienta virtual y visualizada en la hoja de cálculo Microsoft Excel 2013.



- Pruebas de comunicación por RF entre los módulos del SISCA

La característica esencial del sistema es la comunicación inalámbrica, su correcto funcionamiento garantiza la estabilidad del sistema, evita pérdidas, distorsiones o errores de la información. Los 46 caracteres ASCII enviados como información suministrada por el módulo de supervisión hacia los demás módulos no deben presentar ningún tipo de error para asegurar el 100% de confiabilidad.

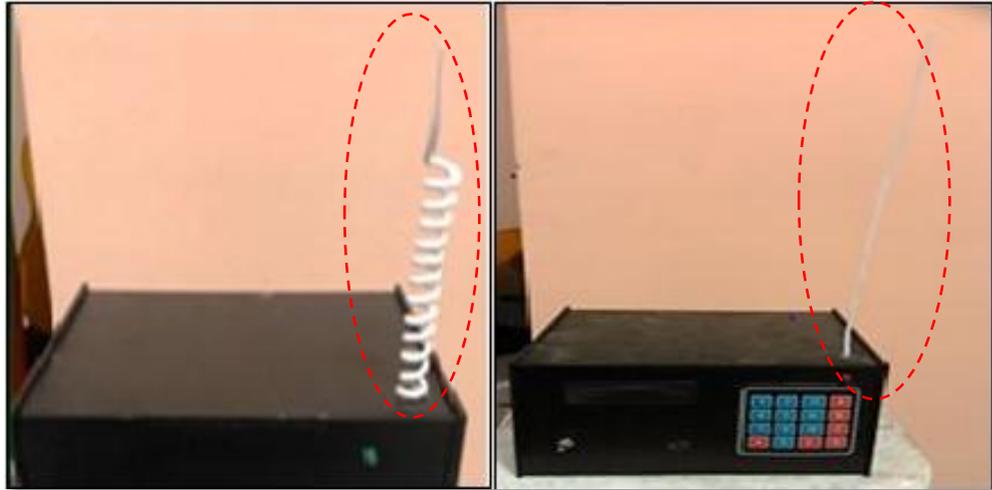
En la *Figura 3-8* se muestra la cadena transmitida por el módulo supervisor, la información recibida por el módulo de almacenamiento y por el módulo de aviso al ser monitoreados sus puertos USB por medio del computador.



**Figura 3-8** Cadena transmitida por el módulo supervisor, la información recibida por el módulo de almacenamiento y por el módulo de aviso

**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

Al evaluar la distancia máxima de comunicación RF que el SISCA mantiene entre el módulo de supervisión y los restantes se establece que al ser alimentado con 5 V el transmisor RF alcanza una distancia de 98 metros, trabajando con una antena de 30 cm en el transmisor y de 25 cm helicoidales en los receptores (Fernández, 2015). En la *Figura 3-9* se muestran el diseño de las antenas incorporadas a los módulos de almacenamiento y en el módulo de supervisión.



**Figura 3-9** Diseño de antenas utilizadas en el Módulo de almacenamiento (izquierda) y en el módulo de supervisión (derecha).

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

#### 3.2.4 Pruebas de funcionamiento del Módulo de Aviso

En este módulo la parte esencial es la visualización en el vúmetro de los niveles obtenidos por el módulo de supervisión. Su función es encender las bombillas del medidor de niveles acústicos de acuerdo a los valores censados. En la *Figura 3-10* se muestra el funcionamiento del módulo de supervisión y el vúmetro.



**Figura 3-10** Funcionamiento del módulo de supervisión y el vúmetro.  
**Fuente:** ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 3.2.5 *Pruebas de funcionamiento del Módulo de Almacenamiento*

Las pruebas consistieron en comprobar el registro de la información en la memoria micro SD. En la *Figura 3-11* se observa la información que se están almacenando en el archivo creado por el módulo de almacenamiento, el cual puede ser visualizado con diferentes aplicaciones, como el bloc de notas y aplicaciones Microsoft Office. Mediante esta evaluación se comprueba la creación, almacenamiento y acceso a la información registrada por el módulo.

DATOS: Bloc de notas						
Archivo Edición Formato Ver Ayuda						
NIVELES DE RUIDO REGISTRADOS EN LA ESPOCH						
NODO	FECHA	HORA	R-ACTUAL	R-MAXIMO	R-MINIMO	
9	27/05/2016	19:25:09	042	095	038	
9	27/05/2016	19:25:10	040	095	038	
9	27/05/2016	19:25:11	040	095	038	
9	27/05/2016	19:25:13	040	095	038	
9	27/05/2016	19:25:14	038	095	038	
9	27/05/2016	19:25:15	040	095	038	
9	27/05/2016	19:25:16	040	095	038	
9	27/05/2016	19:25:18	041	095	038	
9	27/05/2016	19:25:19	041	095	038	
9	27/05/2016	19:25:20	040	095	038	
9	27/05/2016	19:25:21	040	095	038	
9	27/05/2016	19:25:22	040	095	038	
9	27/05/2016	19:25:24	041	095	038	
9	27/05/2016	19:25:25	040	095	038	
9	27/05/2016	19:25:26	040	095	038	
9	27/05/2016	19:25:27	040	095	038	
9	27/05/2016	19:25:29	040	095	038	
9	27/05/2016	19:25:30	040	095	038	
9	27/05/2016	19:25:31	040	095	038	

**Figura 3-11** Información almacenada en el archivo creado por el módulo de almacenamiento

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 3.3 Funcionamiento general SISCA desarrollado para la ESPOCH.

Después de las pruebas realizadas en todos los módulos del sistema inalámbrico y probada en los predios de la ESPOCH se concluye que las programaciones realizadas a cada uno de los módulos según los *Anexos 2, 3, 4*; mantienen concordancia con los requerimientos establecidos en la investigación, existe una adecuada comunicación entre la parte *hardware* y las herramientas *software* desarrolladas. Se tiene un rango de actuación de 38-104 dB. En la *Figura 3-12* se muestra funcionando los módulos que integran el SISCA.



**Figura 3-12** Módulos del SISCA funcionando.

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

### 3.4 Análisis económico de la implementación del prototipo del Sistema Inalámbrico de Sensores Inteligentes.

En este apartado se presenta la parte económica del prototipo implementado, todos los gastos y costos de los dispositivos utilizados. En las *Tabla 3-3* se aprecia el presupuesto detallando cada elemento que conforman los módulos del SISCA.

**Tabla 3-3** Presupuesto detallando cada elemento que conforman los módulos del SISCA

MÓDULO	DISPOSITIVO	CANT.	COSTO UNITARIO	COSTO TOTAL
SUPERVISIÓN	Tarjeta Arduino MEGA	1	\$ 25,00	\$ 25,00
	Sensor de Ruido (Sound Detector)	1	\$ 16,00	\$ 16,00
	Pantalla LCD	1	\$ 7,00	\$ 7,00
	Teclado matricial	1	\$ 5,00	\$ 5,00
	Switch	1	\$ 0,40	\$ 0,40
	RTC (Reloj en Tiempo Real)	1	\$ 13,00	\$ 13,00
	Cables de conexión	1	\$ 5,50	\$ 5,50
	Resistencias	1	\$ 0,10	\$ 0,10
	Modulo Transmisor RF	1	\$ 4,00	\$ 4,00
	Ensamblaje	1	\$ 20,00	\$ 20,00
	<b>TOTAL MÓDULO SUPERVISIÓN</b>			
ALMACENAMIENTO	Tarjeta Arduino MEGA	1	\$ 25,00	\$ 25,00
	Módulo Receptor RF	1	\$ 4,00	\$ 4,00
	Resistencias	1	\$ 0,10	\$ 0,10
	<i>Shield</i> Micro SD	1	\$ 10,00	\$ 10,00
	Ensamblaje	1	\$ 10,00	\$ 10,00
	<b>TOTAL MÓDULO ALMACENAMIENTO</b>			
AVISO	Tarjeta Arduino UNO	1	\$ 18,00	\$ 18,00
	Módulo Receptor RF	1	\$ 4,00	\$ 4,00
	Resistencias	12	\$ 0,10	\$ 1,20
	Bombillas	11	\$ 0,40	\$ 4,40
	Boquillas	11	\$ 0,40	\$ 4,40
	Módulos relé	3	\$ 7,50	\$ 22,50
	Cables de conexión	1	\$ 10,00	\$ 10,00
	Ensamblaje	1	\$ 90,00	\$ 90,00
	<b>TOTAL MÓDULO AVISO</b>			

Fuente: ALTAMIRANO, Héctor & PUENTE, Amanda, 2016.

De los valores obtenidos en las tablas antes indicadas se puede apreciar que el módulo de almacenamiento presenta el más bajo costo, con 49,10 dólares americanos (USD). Y el que mayor costo presenta es el módulo de aviso con un total de 154,50 dólares americanos (USD), el costo total del prototipo es de 299,60 dólares americanos (USD).

## CONCLUSIONES

- Se diseñó un sistema inalámbrico de sensores de supervisión acústica cuya operatividad se basa en tres módulos: supervisión, almacenamiento y aviso que cumplen funciones específicas como nodo de la red inalámbrica implementada con comunicación por RF.
- Luego del análisis realizado para el diseño de SISCA se determinó que el sensor de audio de Sparkfun cumple con los criterios de funcionamiento requeridos en la investigación porque permite leer fácilmente la amplitud del sonido de ambientes abiertos o cerrados.
- Se recurrió a la comunicación por RF debido al bajo costo que presentan sus módulos y a la frecuencia abierta de 433 MHz con la que trabaja.
- Se seleccionó como elemento de procesamiento de los módulos las tarjetas de desarrollo Arduino UNO y MEGA por su bajo costo, facilidad de manejo y adquisición.
- Las pruebas de funcionamiento de los módulos implementados para el SISCA se realizaron supervisando los niveles de contaminación acústica en la entrada de la Facultad de Informática y Electrónica durante siete días calendario, se detectaron niveles de ruido entre 42 a 85 dB. Se comparó con las medidas obtenidas por un equipo patrón, de estas pruebas se obtuvieron que el sistema mantiene errores absolutos de  $\pm 1.2$  dB lo que se encuentra acorde con los objetivos del diseño.
- En las pruebas de comunicación por radio frecuencia se logró la transmisión sin pérdida de información entre módulos a distancias de hasta 98 metros con línea de vista, con una alimentación de 5 V. Se trabajó con antenas helicoidales de 25 centímetros para los receptores y una antena tipo látigo de 30 centímetros para el transmisor. Los módulos del SISCA pueden comunicarse en forma simultánea con el computador por medio del puerto USB sin interferir en las comunicaciones inalámbricas.
- Se diseñó un sistema de aviso con comunicación inalámbrica que se presenta en un vúmetro de 11 niveles, elemento necesario para socializar la información de los niveles de contaminación acústica, el cual puede ser replicado n veces sin alterar la estructura del sistema.

- Con la herramienta virtual desarrollada en LabVIEW, por medio de la red Wi-Fi y la aplicación Data Dashboard se puede compartir con dispositivos móviles los niveles de contaminación acústica que se registran.

## RECOMENDACIONES

- Incrementar al SISCA nuevos tipos de comunicación inalámbrica para abarcar una mayor distancia entre los nodos sensores y los de almacenamiento.
- Estudiar la posibilidad de implementar el bloque de alimentación de los módulos basado en energías renovables, de forma que su diseño sea totalmente transportable e independiente.
- Incorporar al sistema la capacidad de registrar la información en base de datos compartidas vía Web. Con la finalidad de que especialistas en el área consideren esta información como elemento fundamental en posibles investigaciones.
- Implementar sensores que permitan medir elementos contaminantes como CO<sub>2</sub>, calidad del aire, parámetros ambientales para ampliar la información suministrada por el SISCA en cuanto a niveles de contaminación ambiental.
- SISCA está diseñado para funcionar tanto en ambientes externos como internos por lo que se recomienda su uso en lugares donde los niveles de ruido son constantes y prolongados como en discotecas, bares, mercados, karaokes, coliseos, estadios como una forma de control para evitar daños en la salud de los usuarios.

## BIBLIOGRAFÍA

**ARDUINO.** *Arduino*. [En línea] 2016.

[Citado el: 27 de Abril de 2016.]

<https://www.arduino.cc/>.

**BARCELL FERNANDEZ, Manuel.** M.F.BARCELL. [pdf] 2010.

[Citado el: 23 de Abril de 2016.]

<http://www.mfbarcell.es/conferencias/wsn.pdf>.

**BURBANO GARCÍA, Jhomaira Lorena.** Red WSN para el control y monitoreo de un sistema de riego por goteo de una plantación de fresas en la granja experimental Yuyucocha (Tesis). [En línea] Universidad Técnica del Norte, Facultad de Ingeniería en Ciencias Aplicadas, Escuela de Ingeniería en Electrónica y Redes de Comunicación, Ibarra, Ecuador. 2014. p. 29.

[Citado el: 24 de Abril de 2016.]

<http://repositorio.utn.edu.ec/bitstream/123456789/3526/1/04%20RED%20031%20TESIS.pdf>.

**CAPELLA HERNÁNDEZ, Juan Vicente.** Redes inalámbricas de sensores: una nueva arquitectura eficiente y robusta basada en jerarquía dinámica de grupos (Tesis). (Doctoral). [En línea] Universidad Politécnica de Valencia, Departamento de Informática de Sistemas y Computadores, Valencia, España. 2010. p. 33.

[Citado el: 22 de Abril de 2016.]

<https://riunet.upv.es/bitstream/handle/10251/8417/tesisUPV3326.pdf>.

**COBOS HERNÁNDEZ, Javier Edgar.** Estudio de las redes sensoriales como una nueva alternativa de comunicación inalámbrica (Tesis). [En línea] Escuela Politécnica del Ejército, Departamento de Eléctrica y Electrónica, Escuela de Ingeniería Electrónica en Instrumentación, Sangolquí, Ecuador. 2007. p. 22.

[Citado el: 24 de Abril de 2016.]

<http://repositorio.espe.edu.ec/bitstream/21000/2372/1/T-ESPE-025074.pdf>.

**DIYMAKERS.** *Intel Galileo: Guía Inicial*. [blog]. 2014.

[Citado el: 27 de Abril de 2016.]

<http://diymakers.es/intel-galileo-guia-inicial/>.

**ESPINOSA, SANTIAGO.** *Saigesp*. [En línea] 2015.

[Citado el: 22 de Abril de 2016.]

<http://www.saigesp.es/rf-principios-basicos/>.

**FERNÁNDEZ, DANIEL.** *Instructables*. [En línea] 2015.

[Citado el: 27 de Abril de 2016.]

<http://www.instructables.com/id/Modulos-RF433-Mhz-con-Arduino/>.

**FERNÁNDEZ MARTÍNEZ, R., MARTÍNEZ DE PISÓN ASCACÍBAR, F. J., et. al.** *Redes Inalámbricas de Sensores: teoría y aplicación práctica*. [pdf] Universidad de La Rioja, España: Universidad de la Rioja, Servicio de Publicaciones, 2009.

[Citado el: 24 de Abril de 2016.]

<http://dialnet.unirioja.es/descarga/libro/377564.pdf>.

**FLORES CARBAJAL, Esther Elena.** *Redes de Sensores Inalámbricas Aplicado a la Medicina* (Tesis). (Maestría). [En línea] Universidad de Cantabria, Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicaciones, Santander, España. 2012. pp. 11-13.

[Citado el: 23 de Abril de 2016.]

<http://repositorio.unican.es/xmlui/bitstream/handle/10902/1288/349251.pdf?sequence=1>.

**FUENTES HABA, Enrique.** *Red de sensores inalámbricos de largo alcance para medición de la calidad del aire* (Tesis). [En línea] Universidad de Sevilla, Departamento de Ingeniería de Sistemas y Automática, Escuela Técnica Superior de Ingeniería, Sevilla, España, 2013. p. 36

[Citado el: 05 de Abril de 2016.]

<http://bibing.us.es/proyectos/abreproy/40060/fichero/SMCA+Memoria.v4.pdf>.

**GARCÍA GONZÁLEZ, ANTONY.** *Panamahitek*. [En línea] 2015.

[Citado el: 24 de Abril de 2016.]

<http://panamahitek.com/el-setup-y-el-loop-en-arduino/>.

**GARCIA, V.** *Ladelec*. [En línea] 2009.

[Citado el: 22 de Abril de 2016.]

[http://www.ladelec.com/documents/Descargas%20gratis/Hojas%20de%20datos/ds1307\\_rtc\\_esp-anol.pdf](http://www.ladelec.com/documents/Descargas%20gratis/Hojas%20de%20datos/ds1307_rtc_esp-anol.pdf).

**GUAGCHINGA MORENO, Leopoldo Vinicio, & GUAMANÍ MENA, Eduardo Marcelo.** Diseño e implementación de un sistema de medición electrónico de gases contaminantes y contaminación acústica en dos lugares de la ciudad de Latacunga utilizando un sistema de transmisión basado en tecnología Zigbee. (Tesis). [En línea] Escuela Politécnica del Ejército, Departamento de Eléctrica y Electrónica, Escuela de Ingeniería Electrónica en Instrumentación, Latacunga, Ecuador, 2012. pp. 6-10.  
[Citado el: 14 de Abril de 2016.]  
<http://repositorio.espe.edu.ec/bitstream/21000/6042/1/T-EPEL-0981.pdf>.

**GUERRA SALAZAR, José.** *Diseño e implementación de un sistema inalámbrico para el monitoreo en línea del microclima de invernaderos artesanales* (Tesis). (Maestría). Instituto Superior Politécnico "José Antonio Echeverría", Facultad de Ingeniería Eléctrica, Centro de Investigaciones en Microelectrónica, La Habana, Cuba. 2015. pp. 15-72.

**GUEVARA SÁNCHEZ, Mayra Carolina.** Eficacia de la normativa jurídica que regula la contaminación acústica generada por actividades económicas en el D.M.Q. (Tesis). [En línea] Pontificia Universidad Católica del Ecuador, Facultad de Jurisprudencia, Escuela de Ciencias Jurídicas, Quito, Ecuador. 2013. pp. 12-13.  
[Citado el: 28 de Abril de 2016.]  
<http://repositorio.puce.edu.ec/bitstream/handle/22000/5910/T-PUCE-6068.pdf;jsessionid=E10D321AB476E93D0F0AA84C21EBE136?sequence=1>.

**INTEL.** *Descargas de la placa Intel Galileo*. [En línea] 2015.  
[Citado el: 04 de Mayo de 2016.]  
<https://software.intel.com/es-es/iot/hardware/galileo/downloads>.

**JFET, BYRON.** *Sparkfun*. [En línea] 2014.  
[Citado el: 19 de Abril de 2016.]  
<https://learn.sparkfun.com/tutorials/sound-detector-hookup-guide>.

**LIBELIUM.** *Libelium*. [En línea] 2015.  
[Citado el: 04 de Mayo de 2016.]  
<http://www.libelium.com/products/waspmote/hardware/>.

**MARCONI, Liliana, & D'AMELIO, Adriana.** *Medidas de Tendencia Central y de Variabilidad.* [pdf] 2009.

[Citado el: 18 de Mayo de 2016.]

<http://www.deie.mendoza.gov.ar/aem/material/teoria/MEDIDAS%20DE%20TENDENCIA%20CENTRAL%20Y%20DE%20VARIABILIDAD.pdf>.

**MARTÍNEZ GARCÍA, Ramón.** Análisis y diseño de una red inalámbrica de sensores para un proyecto agrario (Tesis). [En línea] Universidad Abierta de Cataluña, Ingeniería Técnica de Telecomunicación, especialidad en Telemática, Cataluña, España, 2014. pp. 9-17.

[Citado el: 22 de Abril de 2016.]

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/27801/8/rmartinezgarciTFC0114memoria.pdf>.

**MARTÍNEZ LLORENTE, Jimena, & PETERS, Jens.** *Contaminación acústica y ruido.* [pdf] 3ª Edic. Madrid: Ecologistas en Acción, 2015.

[Citado el: 28 de Abril de 2016.]

[https://www.ecologistasenaccion.org/IMG/pdf/cuaderno\\_ruido\\_2013.pdf](https://www.ecologistasenaccion.org/IMG/pdf/cuaderno_ruido_2013.pdf).

**MARTINEZ, Asier.** *Ovtoaster.* [En línea] 2013.

[Citado el: 24 de Abril de 2016.]

<http://ovtoaster.com/queesarduino/>.

**McCAULEY, Mike.** *Pjrc.* [En línea] 2015.

[Citado el: 25 de Abril de 2016.]

[https://www.pjrc.com/teensy/td\\_libs\\_VirtualWire.html](https://www.pjrc.com/teensy/td_libs_VirtualWire.html).

**NARVÁEZ, Carlos.** *Microinternet.* [En línea] 2012.

[Citado el: 22 de Abril de 2016.]

<http://microinternet.com.ve/wp-content/uploads/2012/06/Reloj-de-Tiempo-Real-DS1307.pdf>.

**NARVAEZ, Vanessa.** *Problemas que genera la contaminación acústica.* [blog]. 2014.

[Citado el: 04 de Mayo de 2016.]

<http://contaminacionacustica11-4.blogspot.com/2014/09/la-es-considerada-por-la-mayoria-de-la.html>.

**NATIONAL INSTRUMENTS.** *Ni*. [En línea] 2016.

[Citado el: 24 de Abril de 2016.]

[https://www.ni.com/academic/why\\_labview/esa/](https://www.ni.com/academic/why_labview/esa/).

**OJEDA CASTILLO, Teresita del Carmen.** El derecho constitucional a disfrutar de un ambiente sano y su vulnerabilidad por la contaminación acústica (Tesis). [En línea] Universidad Nacional de Loja, Carrera de Derecho, Loja, Ecuador, 2014. pp. 11-16.

[Citado el: 27 de Abril de 2016.]

<http://dspace.unl.edu.ec/jspui/bitstream/123456789/7488/1/Teresita%20del%20Carmen%20Ojeda%20Castillo.pdf>.

**ORGANIZACIÓN MUNDIAL DE LA SALUD.** *Un nuevo informe pone de relieve cuatro formas de reducir los riesgos sanitarios debidos a los contaminantes climáticos*. [En línea] 2015.

[Citado el: 22 de Abril de 2016.]

<http://www.who.int/mediacentre/news/releases/2015/reducing-climate-pollutants/es/>

**OWENS, RYAN.** *Electronicaembajadores*. [En línea] 2013.

[Citado el: 22 de Abril de 2016.]

<http://www.electronicaembajadores.com/Productos/Detalle/19/LCA1SXX/modulo-arduino-sd-card-shield>.

**PICO VALENCIA, Pablo.** Infraestructura polivalente basada en sensores inalámbricos aplicada a la monitorización medioambiental georreferenciada (Tesis). (Maestría). [En línea] Universidad de las Palmas de Gran Canaria, Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería, Las Palmas de Gran Canaria, España. 2012. p. 33.

[Citado el: 26 de Abril de 2016.]

[http://acceda.ulpgc.es/bitstream/10553/7560/5/TFM\\_Pablo\\_Pico\\_2012.pdf](http://acceda.ulpgc.es/bitstream/10553/7560/5/TFM_Pablo_Pico_2012.pdf).

**QUIRARTE, Antonio.** *Hacedores*. [En línea] 2014.

[Citado el: 17 de Abril de 2016.]

<http://hacedores.com/que-tarjeta-de-desarrollo-elegir-parte-1/>.

**RASPBERRYPI.** *Raspberrypi*. [En línea] 2016.

[Citado el: 04 de Mayo de 2016.]

<https://www.raspberrypi.org/>.

**RAZO FIALLOS, Jaime Patricio, & MAZA APUHANGO, Mario Fernando.** *Modelo de gestión de ruido para el centro histórico de la ciudad de Riobamba* (Tesis). Universidad Nacional de Chimborazo, Facultad de Ingeniería, Carrera de Ingeniería Ambiental, Riobamba, Ecuador. 2015. pp. 2-6.

**RUHMAN, Daniel.** *Sites.google.* [En línea] 2014.

[Citado el: 17 de Abril de 2016.]

[https://sites.google.com/site/summerfuelrobots/arduino-sensor-tutorials/rf-wireless-transmitter-receiver-module-433mhz-for-arduino.](https://sites.google.com/site/summerfuelrobots/arduino-sensor-tutorials/rf-wireless-transmitter-receiver-module-433mhz-for-arduino)

**TAPIA AYALA, Carlos Hipólito, & MANZANO YUPA, Héctor Mauricio.** Evaluación de la plataforma arduino e implementación de un sistema de control de posición horizontal (Tesis). [En línea] Universidad Politécnica Salesiana, Facultad de Ingenierías, Carrera de Ingeniería Electrónica mención en Sistemas Computacionales, Guayaquil, Ecuador. 2013. p. 25.

[Citado el: 26 de Abril de 2016.]

[http://dspace.ups.edu.ec/bitstream/123456789/5522/1/UPS-GT000511.pdf.](http://dspace.ups.edu.ec/bitstream/123456789/5522/1/UPS-GT000511.pdf)

**TECNO INFE.** *Tecno Infe.* [En línea] 2014.

[Citado el: 04 de Mayo de 2016.]

[http://www.tecnoinfe.com/2014/09/arduino.html.](http://www.tecnoinfe.com/2014/09/arduino.html)

**TOLOCKA, E.** *Profetolocka.* [En línea] 2015.

[Citado el: 23 de Abril de 2016.]

[http://www.profetolocka.com.ar/2015/05/09/modulo-de-4-reles-para-arduino/.](http://www.profetolocka.com.ar/2015/05/09/modulo-de-4-reles-para-arduino/)

**TULSMA.** *Texto Unificado Legislación Secundaria, Medio Ambiente, Parte I.* [En línea] 2003.

[Citado el: 04 de Mayo de 2016.]

[http://www.agua.gob.ec/wp-](http://www.agua.gob.ec/wp-content/uploads/downloads/2012/07/TEXTO_UNIFICADO_LEGISLACION_SECUNDARIA_i.pdf)

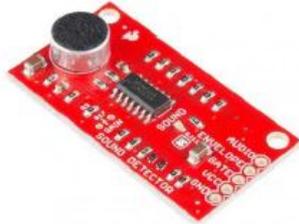
[content/uploads/downloads/2012/07/TEXTO\\_UNIFICADO\\_LEGISLACION\\_SECUNDARIA\\_i.pdf.](http://www.agua.gob.ec/wp-content/uploads/downloads/2012/07/TEXTO_UNIFICADO_LEGISLACION_SECUNDARIA_i.pdf)

# **ANEXOS**

# Anexo 1: Hojas Técnicas de los elementos que conforman el Sistema Inalámbrico de Sensores de medición de Contaminación Acústica.

## Sound Detector Hookup Guide

### Introducing The Sound Detector



#### The Sound Detector

The **Sound Detector** is a small board that combines a microphone and some processing circuitry. It provides not only an audio output, but also a binary indication of the presence of sound, and an analog representation of it's amplitude.

#### Covered in This Tutorial

This tutorial will guide you through hooking up and using the Sound Detector. It will examine how the circuit works, explain some details about getting the best performance from the Sound Detector, then present two different projects that demonstrate how to use it.

#### Suggested Reading

- [How to Use a Breadboard](#)
- Analog vs. Digital
- The Wikipedia article on [sound](#)

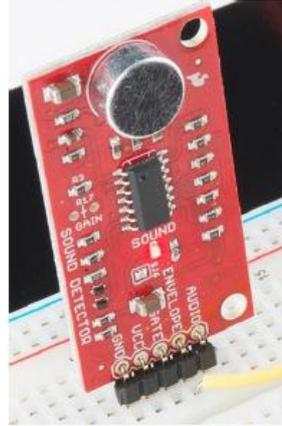
#### Quick Start

To get started with the Sound Detector, simply connect it to a power supply.

(Sound Detector --> Power Supply)

- GND --> Supply Ground.
- VCC --> Power supply voltage between 3.5 and 5.5 Volts. 5 Volts is ideal.

In a quiet room, power the board up, and then speak into the microphone. You should see the red LED on the board blink in response to your voice.

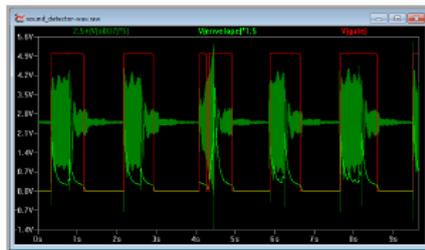


With it's 3 outputs, the board itself is a lot more flexible. To explore that flexibility, read on.

### Looking Closer

#### Three Outputs?

The Sound Detector has 3 separate outputs. It's easiest to see what each is doing with a graph. The following illustrates how the sound detector responds to a series of sound pulses.



Sample Sound Detector Output

This shows the output voltages over time.

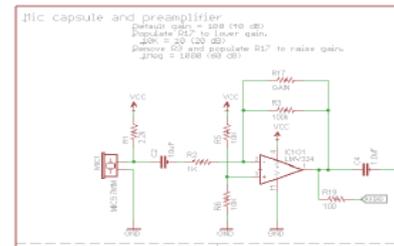
- The dark green trace is the audio output of the sound detector. The audio voltage directly from the microphone is found at this output.
- The light green trace is the envelope output. This analog voltage traces the amplitude of the sound. Of particular interest, notice that the third pulse gets noticeably louder as it goes.
- Finally, the red line is the gate output. This output is low when conditions are quiet and goes high when sound is detected.

#### How It Works

Having examined the outputs, lets also take a quick walk through the schematic, to gain an understanding of how each stage works.

##### First Stage

The first section of the circuit is an electret microphone capsule. This portion of the circuit borrows from the [Electret Microphone breakout board](#).



Microphone and Preamplifier

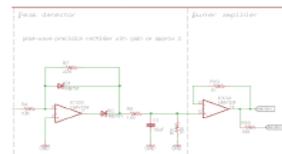
The capsule is biased by the supply voltage through R1, and it outputs an AC voltage that is riding a DC offset of approximately 1/2 the supply voltage.

The output from the capsule is an extremely small voltage, so the signal from the capsule is amplified by IC101, an operational amplifier stage. By default, the preamplifier has an arithmetic gain of 100 (20 dB), and the gain can be adjusted by populating R17 (which we'll examine in detail on the [next page](#)).

The wide output is DC coupled, riding one half the supply voltage, so it can be directly connected to the ADC of a microcontroller. In perfectly quiet conditions, it will ideally read 1/2 full scale, or 512 on a 10-bit converter.

##### Second Stage

The second stage of the circuit is an envelope follower.



### Envelope Follower

IC1G3 forms an opamp-based precision rectifier. This stage implements the equation

#### COPY CODE

```
if(Vin > 0)
  Vout = 0;
else
  Vout = Vin * -2.2
```

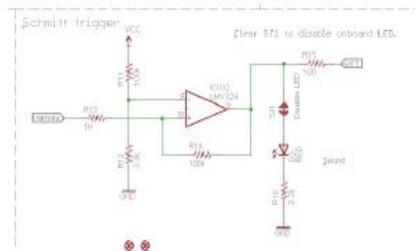
The opamp inverts and amplifies the signal. When it's output swings high, D2 turns on, and charges C1. When the opamp output is high or not swinging, D2 is turned off, and C1 discharges through R9. Thus, C1 tracks the peaks of the input signal.

IC1G4 is a buffer amplifier, so external loads on the envelope pin won't change the C1's charge/discharge behavior.

This results in a signal that tracks the peak amplitude of the input signal. A louder sound will result in a higher voltage on the Envelope pin. As with the audio pin, the envelope can be connected to the ADC of a microcontroller.

### Third Stage

The final stage implements a thresholded switch on the envelope signal.



### Schmitt Trigger

The Schmitt trigger watches the envelope signal, and toggles the output when the threshold is exceeded. A Schmitt trigger is a comparator that adjusts it's threshold voltage when the output switches, requiring a higher voltage to switch on than to switch off. This allows it to ignore some ripple in the input signal, like the ripple present in the output of the envelope follower stage.

The output of the Schmitt trigger is found on the Gate pin. You can connect it to a digital input. We'll use it to trigger interrupts in the [software example](#).

### Outputs

Each of the three output signals is present on the 1" header at the edge of the board. They are active simultaneously. If you aren't using one in your particular application, simply leave that pin disconnected.

### Configuration

#### Care and Feeding Of The Capsule

The heart of the Sound Detector is the electret microphone capsule – without it, we couldn't convert acoustic energy into electrical energy. These capsules have a couple of quirks that we need to understand in order to apply them successfully.

Inside the capsule is the diaphragm, which is actually one plate of a small capacitor. That capacitor forms a voltage divider with the external bias resistor. The diaphragm moves in response to sound, and the capacitance changes as the plates get closer together or farther apart, causing the divider to change. Since capacitors are sensitive to loading, it's internally buffered with a JFET ([junction field-effect transistor](#)).

Due to the mechanical and electronic tolerances involved, some capsules are more sensitive than others. Also, the JFET is rather sensitive to noise on the power supply. Both of these factors need to be accounted for when deploying the Sound Detector.

#### Power Supply

The Sound Detector is an analog circuit, and as such, it's more sensitive to noise on the power supply than most digital circuits. Since the capsule is effectively a voltage divider sitting across the power rails, it will transcribe any noise on the supply lines onto the capsule output. The next circuit in the chain is a high-gain amplifier, so any noise on the supply will then be amplified. Therefore, the Sound Detector may require more careful power supply configuration than many circuits.

In testing with various supplies, a significant degree of variability was discovered – some supplies are less noisy than others. One exhibited as much as 30 mV ripple on the supply output, and as a result, the the Sound Detector was rather sensitive and unstable. You can check how clean a power supply is by checking it with an oscilloscope or volt meter, set to the AC Volts (or, if provided, the AC millivolts) range. A truly clean supply will show 0.000 VAC. Based on the supplies used in testing, ripple of more than about 10 mV is problematic.

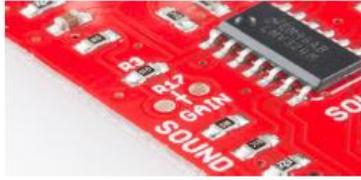
Powering my Arduino with a 9V external supply, which allows the onboard regulators to function, the Arduino's 5V output was sufficiently clean. However, powering it from the 5V available on the USB port on a PC, the regulators are bypassed, and the results were somewhat less usable, and vary greatly between different ports on different PCs. A powered USB hub will probably provide cleaner power than the ports on the PC itself.

If all else fails, three 1.5V batteries in series make a nice, clean source of 4.5V.

#### Amplitude Calibration

The Sound Detector comes set for moderate sensitivity – speaking directly into the microphone, or clapping your hands nearby should cause the gate output to fire. If you find that it doesn't work well in a specific application, you can change the circuit to be more or less sensitive.

The gain is set by changing the feedback resistors in the preamp stage. The resistors are marked in the silkscreen on the PCB.



#### Resistor Cleanup

R3 is a surface mount part, with 100K Ohm populated by default. R17 is an unpopulated position for a through hole resistor.

#### Lowering The gain

It's most likely that you'll find the detector to be too sensitive. In testing the board for this written, noisy air conditioning and music in the next office were enough to set it off. To make the board less sensitive, you can lower the preamplifier gain by populating R17 in parallel with R3.

R3 Value	R17 Value	Arithmetic Gain	Gain (dB)
100K	-	100	40
100K	100K	50	33
100K	47K	32	30
100K	22K	18	25
100K	10K	8	19
100K	4.7K	4	13
100K	2.2K	2	6

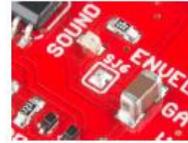
#### Raising The Gain

If you want to make the sound detector more sensitive, so that it will be activated by quieter sounds, you can remove R3, and populate R17.

R17 Value	Arithmetic Gain	Gain (dB)
100K	100	40
220K	220	46
470K	470	53
1Meg	1000	60

#### Lights Out

In some applications, the onboard LED may be distracting or undesirable. To disable it, simply use a solder sucker or wick to remove the solder blob from the jumper near the LED.



#### Physical Considerations

The electret capsule on the Sound Detector is also sensitive to mechanical vibration and wind noise.

The Sound Detector will pick up handling noise. Mounting it with a resilient material can help absorb vibration - placing it on a piece of open-cell foam helped reject vibration conducted through the testing workbench. Other resilient mounting strategies would include suspending it using rubber bands, or building shock absorbers using #4 machine screws and heat-shrink tubing.

Wind noise is caused when blasts of air enter the capsule. Examples would be a windy day, or the sharp exhalation that often accompanies syllables such "p," "v" and "b." The capsule come with a thin fabric covering, but it may not be enough to prevent stronger blasts. You can craft a more robust windscreen with open-cell foam or synthetic fur (like the covering that sneaks into the frame when a boom mic accidentally enters a film scene).

LCA1001



## ARDUINO UNO REV 3

### Características:

El Uno Arduino es una placa electronica basada en el ATmega328

Cuenta con 14 entradas / salidas digitales pines (de los cuales 6 pueden ser utilizados como salidas PWM), 6 entradas analógicas, un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera de ICSP, y un botón de reinicio.

Contiene todo lo necesario para apoyar a la micro, sólo tiene que conectarlo a un ordenador con un cable USB o el poder con un adaptador AC-DC o la batería para empezar.

El Uno difiere de todas las placas anteriores en que no utilizan el chip controlador USB FTDI a serie. Las características adicionales que vienen con la versión R3 son:

\*Lugar 8U2 como USB a serie convertidor de ATmega16U2.

\*1,0 pinout: añadido el SDA y SCL pines para la comunicación TWI coloca cerca de la pin AREF y dos pasadores de otros nuevos que se pongan cerca del pin de RESET, el IOREF que permiten a los escudos de adaptarse a la tensión proporcionada por la junta directiva y el segundo un No conecte el pin, que se reserva para usos futuros.

\*Circuito RESET.

### Especificaciones:

Microcontroller: ATmega328

Operating Voltage: 5V

Supply Voltage (recommended): 7-12V

Maximum supply voltage (not recommended): 20V

Digital I/O Pins: 14 (of which 6 provide PWM output)

Analog Input Pins: 6

DC Current per I/O Pin: 40 mA

DC Current for 3.3V Pin: 50 mA

Flash Memory: 32 KB (ATmega328) of which 0.5 KB used by bootloader

SRAM: 2 KB (ATmega328)

EEPROM: 1 KB (ATmega328)

Clock Speed: 16 MHz

LCA1010



## ARDUINO MEGA 2560

### Características:

El Arduino Mega 2560 es una placa electrónica basada en el Atmega2560

Cuenta con 54 entradas / salidas digitales pines (de los cuales 14 se pueden utilizar como salidas PWM) y 16 entradas analógicas, 4 UARTs (puertos de hardware de serie), un oscilador de cristal de 16 MHz, un puerto USB

de conexión, un conector de alimentación, una cabecera de ICSP, . y un botón de reinicio

Contiene todo lo necesario para apoyar a la micro, sólo tiene que conectarlo a un ordenador con un cable USB o el poder con un adaptador AC-DC o la batería para empezar. La Mega es compatible con la mayoría de los escudos diseñados para el Arduino Duemilanove o Diecimila

### Especificaciones:

Microcontrolador: ATmega2560

Tensión de alimentación: 5V

Tensión de entrada recomendada: 7-12V

Límite de entrada: 6-20V

Pines digitales: 54 (14 con PWM)

Entradas analógicas: 16

Corriente máxima por pin: 40 mA

Corriente máxima para el pin 3.3V: 50 mA

Memoria flash: 256 KB

SRAM: 8 KB

EEPROM: 4 KB

Velocidad de reloj: 16 MHz

**MÓDULOS DE RF PARA RADIOCONTROL**

(Versión: 10-12-11)

**Introducción**

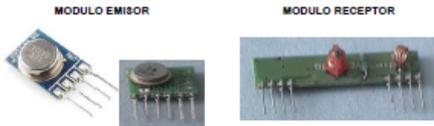
A la hora de pensar en diseñar sistemas radiocontrolados, surgen más de una opción para llevar a cabo nuestro proyecto en lo que respecta al transmisor y receptor que emplearemos para el envío y recepción de las ordenes.

Hace un tiempo atrás la única posibilidad, económica, era la construcción y calibración del circuito transmisor y receptor, lo que implicaba dedicar un tiempo importante para conseguir este objetivo. Luego podíamos dedicarnos al resto de las partes del proyecto.

Hoy en día disponemos a un costo razonable módulos de RF, transmisor y receptor diseñados para ponernos a funcionar con pocos componentes adicionales.

**NOTA:** Los módulos que se presentarán aquí son solo para transmisión de datos digitales. Es decir estos no sirven si lo que deseamos es transmitir audio o video. Y dado que la tasa de transmisión de datos es muy baja, tampoco podremos transmitir audio y video digitalizados. El diseñador que quiera transmitir audio o video deberá elegir otro tipo de componentes.

Vemos a continuación un ejemplo:



Ejemplos de módulos emisor y receptor.

Podemos encontrar en el mercado opciones interesantes en cuanto a frecuencia, potencia y alcance de estos módulos, el diseñador del sistema radiocontrolado deberá optar por alguno de acuerdo a sus necesidades y procurarse la hoja de datos correspondiente.

Un fabricante es **Wansing Electronics CO.LTD** cuyos módulos **transmisores** presentan las siguientes características.

Model	Frecuencia (MHz)		Lata (Latas)		Voltage (V)		Current (mA)		RF Power (dBm)
	Min	Max	Min	Max	Min	Max	Min	Max	
TWS-01P	315	433.92	8	8	4.5	5.5	407	32	
TWS-02	303.075	433.92	8	8	1.5	12	53	20	
TWS-03	315	433.92	8	8	2	12	6.8	8.8	
TWS-04	315	386.18	8	8	5.5	9.5	270	27	
TWS-05P	430	930	8	8	3	5.5	250	33	
TWS-02	303.075	433.92	8	8	2	12	5.4	14	

Como se puede apreciar hay diferentes opciones para la selección, tensión de trabajo, corriente que consumen, frecuencia, potencia RF emitida.

Estos módulos transmisores tienen sus correspondientes módulos receptores, especialmente preparados para su funcionamiento conjunto.

**A continuación se muestra una tabla del mismo fabricante, pero de módulos receptores:**

Model	Frecuencia (MHz)		Lata (Latas)		Voltage (V)		Current (mA)		Sensitivity (dBm)
	Min	Max	Min	Max	Min	Max	Min	Max	
RWS-04ALV	315	433.92	8	8	2.5	5.5	7	-110	
RWS-03M	315	433.92	8	8	4.0	5.5	4.9	-110	
RWS-01B	303.075	433.92	8	8	4.0	5.5	2.7	-114	
RWS-03B	315	433.92	8	8	4.0	5.5	7.5	-110	
RWS-02B	315	433.92	8	8	4.0	5.5	6.3	-112	
RWS-01A	303.075	433.92	8	8	4.0	5.5	4.8	-110	
RWS-02D	120	120	4	4	4.0	5	-	-110	
RWS-01V	303.075	433.92	8	8	4.0	5.5	4.8	-110	
RWS-08M	300	1000	5	100	2.7	5.8	12	-112	
RWS-08V	300	1000	5	100	2.7	5.8	12	-112	

Notemos que en cuanto a tensión de trabajo, este fabricante ofrece solo de 5volts.

**Concepto de dBm** (para comprender la tabla anterior)

El dBm es una unidad de medida utilizada principalmente en telecomunicación para expresar la potencia absoluta mediante una relación logarítmica.

El dBm se define como el nivel de potencia en decibelios en relación a un nivel de referencia de 1 mW.

El valor en dBm en un punto, donde tenemos una potencia P, viene dado por la fórmula siguiente:

$$dBm = 10 \log \frac{P}{1mW}$$

Al utilizarse un nivel de referencia determinado (1 mW) la potencia en dBm constituye una verdadera medición de la potencia y no una simple relación de potencias como en el caso de la medida en decibelios. Así, una lectura de 20 dBm significa que la potencia medida es 100 veces mayor que 1mW y por tanto igual a 100 mW.

Fuente que se trata de una relación logarítmica, podemos cometer error al comparar por simple inspección potencias medidas en dBm. A continuación se agrega un cuadro que por medio de ejemplos nos muestra aplicaciones reales y las potencias involucradas. \*Recopilado de internet.

Nivel dBm	Potencia	Aplicaciones
80 dBm	100 kW	Potencia típica de transmisión de una estación de radio FM con un alcance de 80 Km. aproximadamente.
60 dBm	1 mW	Radiación típica combinada de RF de un horno de microondas.
40dBm	10 W	Potencia entregada a las antenas de telefonía móvil.
30 dBm	4 W	Salida típica para banda de radio ciudadana (27MHz).
33 dBm	2 W	Máxima salida de potencia para un teléfono celular UMTS/3G (teléfono de potencia clase 1). Máxima salida de potencia para teléfono celular GSM 900/900.
30 dBm	1 w	Fuga de RF típica de un horno microondas- Máxima salida de potencia para teléfono celular GSM1800/1800.

Nivel dBm	Potencia	Aplicaciones
27 dBm	500 mw	Potencia típica de transmisión de un teléfono celular. Máxima salida de potencia para un teléfono celular UMTS/3G (teléfono de potencia clase 2)
24 dBm	250 mw	Máxima salida de potencia para un teléfono celular UMTS/3G (teléfono de potencia clase 3)
21 dBm	125 mw	Máxima salida de potencia para un teléfono celular UMTS/3G (teléfono de potencia clase 4)
20 dBm	100 mw	Bluetooth estándar Clase 1. Cobertura 100 m. Potencia típica de un router inalámbrico WiFi.
15 dBm	32 mw	Potencia típica de transmisión WiFi en computadoras portátiles.
10 dBm	10 mw	
8 dBm	4 mw	
3 dBm	2.5 mw	Bluetooth estándar Clase 2. Cobertura 10 m.
2 dBm	2 mw	
2 dBm	1.8 mw	
1 dBm	1.3 mw	
0 dBm	1mW (1000 uW)	Bluetooth estándar Clase 3. Cobertura 1 m.
-1 dBm	794 uW	
-10 dBm	100 uW	Potencia de señal típica de una red inalámbrica WiFi (-10 a -30 dBm).
-70 dBm	100 pW	Rango típico (-40 a -80 dBm) de potencia de señal inalámbrica (802.11x) recibida por una red.
-130 dBm	0.1 fw	Potencia típica recibida por un receptor GPS (en tierra)

**Conclusión:**  
 Esto nos dice que siempre desearemos que el transmisor emita con los mas dBm posibles (positivos) y que el receptor sea lo mas sensible posible, o sea mas dBm negativos.  
 Una de las especificaciones clave en los diseños de los receptores de RF de hoy en día es la sensibilidad. Esta indica la capacidad del receptor para capturar señales débiles y por tanto es una medida directa del alcance del sistema.  
**Criterios de calidad de un receptor**  
 • Sensibilidad: Nivel mínimo de señal que es capaz de detectar con la calidad deseada.  
 • Sensibilidad: Nivel mínimo de señal que es capaz de detectar con la calidad deseada.  
 • Fidelidad: Capacidad de recibir y demodular la señal sin distorsión.

**Tipo de modulación empleada por los módulos**

Algunos conceptos previos:

**Banda base**

Se denomina banda base al conjunto de señales que no sufren ningún proceso de modulación a la salida de la fuente que las origina, es decir son señales que son transmitidas en su frecuencia original. Dichas señales se pueden codificar y ello da lugar a los códigos de banda base.

Las señales empleadas en banda base se pueden clasificar de la siguiente forma:

**Unipolares**

En este caso un 1 siempre toma una polaridad, positiva o negativa, mientras que un 0 vale siempre 0.



**Polares**

En este caso la señal tomara valores positivos para un 1 lógico y negativos para un 0 lógico pero nunca toma el valor 0.



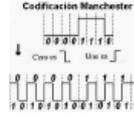
**Bipolares**

En este caso un dígito toma valor con polaridad alternada mientras que el otro permanece siempre en 0.



La transmisión en **banda base** es utilizada para cortas distancias debido a su bajo costo, por ejemplo en una red LAN cableada la información que viaja por el cable UTP lo hace en banda base.

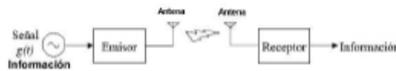
Básicamente esto quiere decir que la información (digital) viaja tal como se genera, y eventualmente lo hace codificada. Notemos que si se tiene una larga secuencia de ceros o de unos, la señal permanece constante durante un tiempo bastante largo en la línea y el receptor no puede identificar el principio y fin de cada bit. Este inconveniente se resuelve juntamente con la codificación. Por ejemplo en las redes LAN (Ethernet) cableadas se utiliza banda base pero con codificación Manchester.



Hasta ahora hemos mencionado la transmisión de señales digitales en banda base. En la práctica, ya sea por que se debe compartir el canal (por ejemplo el aire, el cable coaxial) o para poder usar antenas de dimensiones razonables (recordemos que las dimensiones de una antena dependen de la longitud de onda de la señal que desea transmitir), es necesario modular.

Modular significa modificar la amplitud, la frecuencia o la fase de una señal portadora (que puede ser una sinusoidal), en función del mensaje. Esa señal portadora siempre es una señal de frecuencia elevada, cuanto mas elevada es la frecuencia de la portadora, mas pequeña será la antena necesaria. Por supuesto que tal elección también dependerá de prestaciones y costos.

El esquema general de la radiodifusión es el que se muestra en la figura:



La **antena**, es el transductor que convierte la señal eléctrica en ondas electromagnéticas de RF. Pero la señal que llega a la antena tiene que tener la frecuencia apropiada.

La modulación consiste en trasladar el espectro (componentes de frecuencias) de la señal g(t) (que es la información que se desea transmitir) desde la componente continua, ω = 0, hasta ωC. Sucede además que el espectro se dobla alrededor de ωC, es decir, aparece una copia espejular de espectro además del inicial.

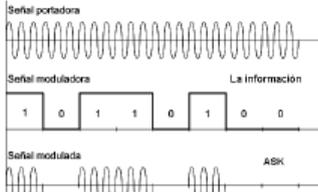


Operación de traslado de frecuencia que realiza el emisor

**Modulación empleada en los módulos que estamos estudiando**

Los módulos transmisores de RF aquí mencionados, en su mayoría, utilizan un esquema de modulación ASK (Amplitude Shift Keying -modulación por desplazamiento de amplitud). Esto les confiere ciertas particularidades propias del esquema de modulación y de su forma de detección, que deberán tener en cuenta por aquellos que diseñen su propio sistema de codificación y decodificación, o de comunicaciones.

En el esquema de modulación ASK, las señales de datos ocasionan que la amplitud de la portadora varíe entre dos estados. Generalmente, se utiliza lo que se conoce como ODSK u OOK (On-Off Shift Keying u On-Off Keying), que consiste en transmitir portadora ante un estado lógico y anulara en el otro.



Los módulos receptores, en su mayoría, suelen ser circuitos super-regenerativos, con un detector de umbral a la salida. Un receptor de este tipo presenta un ruido a la salida bastante importante, en ausencia de señal. El detector trata de ajustar el umbral automáticamente, y en ausencia de portadora se ajustará al valor medio de excursión del receptor, pudiendo responder a picos de ruido, y presentando a la salida del módulo pequeñas transiciones muy breves y esporádicas, ocasionadas por la excursión de la señal de ruido a la salida del receptor, que momentáneamente excede el umbral del detector y su histéresis. Si bien el comportamiento específico dependerá del diseño de cada módulo receptor en particular y del estado de la banda de operación en el lugar de funcionamiento, es de esperar tener pequeños y frecuentes "pulsitos" a la salida cuando no tenemos señal de entrada, es decir, cuando el módulo transmisor no está transmitiendo. En presencia de una portadora constante, la señal de salida del módulo receptor permanecerá indicando el estado correspondiente (generalmente 1 lógico) durante un tiempo, pero en muchos casos retornará nuevamente al estado inactivo, debido a que estos módulos suelen estar diseñados para trabajar con señales que varían constantemente (señales de datos) y no para transmitir estados lógicos permanentes.

En presencia de una señal de datos, el detector ajustará el umbral a la excursión del receptor y tratará de reproducir lo más fielmente la señal de datos, no obstante, debido a detalles constructivos como por ejemplo el tiempo de arranque del oscilador del transmisor, demora en la detección del receptor, y capacidades circuales que agregan tiempos de carga y descarga, es posible que aparezca una pequeña modificación del ciclo de trabajo de la señal, es decir, puede que si se transmiten unos y ceros de igual duración, resulte uno de los periodos levemente distinto del otro. Esta modificación, si existe, suele ser del orden de los microsegundos.

Ovviamente, la presencia de ruido puede llegar a ocasionar transiciones no deseadas aún en presencia de señal, que producirán la detección de bits erróneos, como en todo sistema de comunicaciones digitales.

Es por estas razones, y algunas otras más, que los sistemas de comunicaciones suelen utilizar bits "extras" para detectar errores o incluso correcciones, y los sistemas de control remoto suelen esperar a recibir varias veces seguidas una clave antes de darla por válida e incluir preámbulos al inicio de la transmisión para detectar presencia de una señal válida frente al ruido.

El receptor super-regenerativo es un circuito cuyas prestaciones distan mucho de las que ofrece el más simple de los receptores super-heterodinos (la clásica radio de AM portátil). Un receptor super-regenerativo es básicamente un receptor simple y económico, y generalmente se trata de módulos sin ningún tipo de estabilización en frecuencia (SAW o cristal), por lo que no es posible garantizar la frecuencia central de operación ni su estabilidad.

No obstante, el diseño del super-regenerativo permite ganar sensibilidad sacrificando selectividad, por lo tanto estos módulos son generalmente sensibles, pero son poco selectivos, teniendo un ancho de banda bastante amplio (1MHz es común), haciendo que la sintonía no sea demasiado crítica. Estos mismos detalles de diseño hacen que la presencia de grandes capacidades hacia el plano de tierra (cuerpos metálicos, binóculos, la mano del usuario, la antena), puedan ocasionar corrimientos de la frecuencia de sintonía. Esto puede compensarse, para situaciones particulares, operando sobre la bobina ajustable que el módulo trae. Otra característica a tener en cuenta es que los receptores super-regenerativos tienen un ancho de banda que cambia de acuerdo a la intensidad de la excitación, presentando mayor ancho de banda a señales fuertes, por lo que un transmisor de frecuencia relativamente cercana a una corta distancia puede interferir, y por supuesto ser sintonizado si no hay nada que lo enmascare.

Por último, por detalles de diseño de los receptores super-regenerativos, el "oscilador local" es forzado a funcionar de forma pulsante, y esto puede ocasionar patrones de emisión no deseados (EMI/EMC).

EMI: Interferencia electromagnética: presencia de voltajes o corrientes no deseados que pueden aparecer en un equipo o en sus circuitos, como resultado de la operación de otro aparato eléctrico, o por fenómenos naturales.

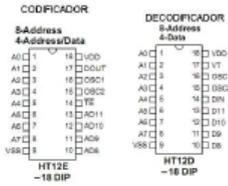
EMC: Compatibilidad Electromagnética: Es la habilidad de los equipos y sistemas para funcionar correctamente sin degradación o sin funcionamiento en el ambiente electromagnético, para el cual están diseñados. Además, el equipo o sistema no debe afectar la operación de cualquier otro equipo o sistema, ni a su vez ser negativamente afectado por este.

Terminada la breve introducción, se procederá a estudiar como utilizar estos módulos de RF.

**FUNCIONAMIENTO DE LOS MÓDULOS DE RF**

Los módulos se deben utilizar (es conveniente) con algún codificador del lado del transmisor y un decodificador del lado del receptor. De este modo se asegura que las órdenes que deseamos que se ejecuten del lado del receptor lleguen adecuadamente y sin errores, mientras estemos dentro del alcance del enlace RF creado.

Los codificadores y decodificadores recomendados son los utilizados para control remoto en sistemas de seguridad, HT12E (codificador) y HT12D (decodificador), respectivamente.



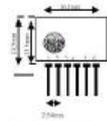
Este juego de integrados codifica y decodifica una palabra de 12 bits, compuesta por una dirección de 8 bits y una sección de datos de 4 bits. Con esta cantidad de bits se pueden comandar 256 dispositivos diferentes, enviándole hasta 16 comandos distintos a cada uno.

DIRECCIÓN	DATOS

O sea la DIRECCIÓN me identifica el dispositivo al que le vamos a enviar el DATO.

**Presentemos un módulo transmisor:** (Recuerde que hay distintos modelos)

**TWS- 434 TRANSMITTER**



- pin 1: Vcc
- pin 2: Vcc
- pin 3: Gnd
- pin 4: Gnd
- pin 5: RF Output
- pin 6: Code Input

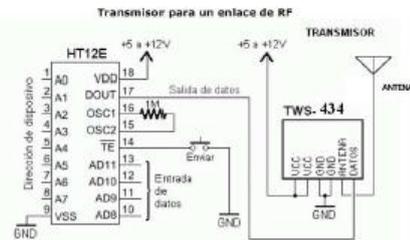
Frequency: 300 433,92 MHz Band

Supply Voltage: 1.5V - 15 Vdc  
RF Output Power: 8mW

El TWS-315 tiene una potencia de salida de hasta 8mW a 433.92MHz, alcanzando distancias de aproximadamente 140 metros en espacios abiertos y de 60 metros en espacios internos donde se tengan obstáculos.



El siguiente circuito muestra como conectar el modulo transmisor al codificador:



La figura anterior muestra un módulo transmisor usando el codificador HT-12E de Holtek. Este es un integrado que se utiliza en controles remotos con comandos de 4 bits y tiene 8 bits de direcciones. Cuando se mantiene en estado bajo el pin TE (Transmission Enable = Habilitación de la transmisión), el codificador HT-12E transmite constantemente, repitiendo el código una y otra vez. En estas condiciones, si se modifica el valor de las entradas de datos se observa que el valor va cambiando en el receptor.

El circuito transmisor permite el uso de una tensión de alimentación entre 5V y 12V. Esto habilita para la utilización de un amplio rango de baterías.

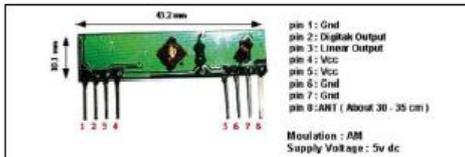
**Presentamos ahora un módulo receptor:** (Recuerde que hay distintos modelos)

**Receiver Module : RWS- 434 (433.92 MHZ)**

- \*Frequency Range: 433.92 MHz
- \*Modulate Mode: ASK
- \*Circuit Shape: LC
- \*Data Rate: 4800 bps
- \*Selectivity: -106 dB
- \*Channel Spacing: 18MHz
- \*Supply Voltage: 5V
- \* High Sensitivity Passive Design.
- \*Simple To Apply with Low External Count.

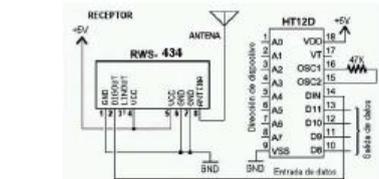


El RWS-315 es un módulo receptor que opera 433.92MHz, y tiene una sensibilidad de 3uV. Opera con una alimentación entre 4.5 y 5.5 Vols-DC y tiene tanto salida lineal como digital, además contiene un capacitor variable para el ajuste de la frecuencia de recepción utilizando un destornillador plástico.



Dado que en el módulo transmisor se ha usado un codificador HT12E, en el receptor se ha de usar el decodificador HT12D.

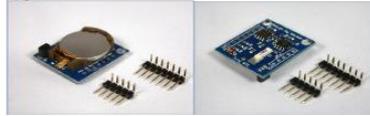
La siguiente figura muestra como se conecta el módulo receptor al decodificador mencionado.



**IMPORTANTE:** En el integrado decodificador HT12D, la señal VT significa 'Valid' (Transmisión)

## Real Time Clock Module

Image:



Description:

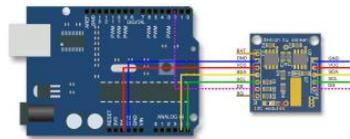
### Arduino Tiny RTC I2C Real Time Clock

This Arduino Tiny RTC I2C module incorporates the DS1307 I2C real time clock IC and the AT24C128 32K I2C EEPROM storage. What's more, it has a DS18B20 temperature sensor on board. All of this in a tiny package of 25mm x 20mm x 3.4mm. It comes with a LIR2303 rechargeable lithium battery, and a charging circuit is included in the module. When the temperature sensor is off, the RTC module can run for 1 year on a single charge.

### Specs of DS18B20 Temperature Sensor

- 3.0-5.5V input voltage
- Waterproof
- -55°C to +125°C temperature range
- ±0.5°C accuracy from -10°C to +85°C
- 1 Wire interface

### Wiring the RTC to Arduino



The most useful pins are duplicated from P1 to P2. If needed, the pin "BAT" can be fed into an ADC pin for monitoring the battery voltage. The pin-outs are explained below. Connecting "VCC" to 5 V will trickle charge the onboard battery.

### Arduino Tiny RTC I2C Real Time Clock Pinout

Pin	Description	Comment
BAT	Battery voltage	To monitor the battery voltage, or not connected
GND	Ground	Ground
VCC	5V supply	Power the module and charge the battery
SDA	I2C data	I2C data for the RTC
SCL	I2C clock	I2C clock for the RTC
DS	DS18B20 Temp. Sensor output	One wire interface
SQ	Square wave output	Normally not used

The I2C wires "SDA" and "SCL" are the data line and clock line, they should be connected to the corresponding pins depending on the Arduino board.

Board	I2C / TWI pins
Uno, Ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1

## Micro Sd De Almacenamiento Board mciro Sd Tf Tarjeta de memoria módulo del Escudo Spi Para Arduino



The module (MicroSD Card Adapter) is a Micro SD card reader module, and the SPI interface via the file system driver, microcontroller system to complete the MicroSD card read and write files. Arduino users can directly use the Arduino IDE comes with an SD card to complete the library card initialization and read-write.

Module features are as follows:

- 1 Support Micro SD Card, Micro SDHC card (high-speed card)
- 2 the level conversion circuit board that can interface level is 5V or 3.3V
- 3 power supply is 4.5V ~ 5.5V, 3.3V voltage regulator circuit board
4. Communication interface is a standard SPI interface
5. 4 M2 screw positioning holes for easy installation

**Control Interface:** A total of six pins (GND, VCC, MISO, MOSI, SCK, CS), GND to ground, VCC is the power supply, MISO, MOSI, SCK is the SPI bus, CS is the chip select signal pin;

**3.3V regulator circuit:** LDO regulator output 3.3V as level converter chip, Micro SD card supply;

**Level conversion circuit:** Micro SD card into the direction of signals into 3.3V, MicroSD card toward the direction of the control interface MISO signal is also converted to 3.3V, general AVR microcontroller system can read the signal;

**Micro SD card connector:** yes since the bomb deck for easy card insertion and removal.

## Módulo de 4 relés para Arduino



Las salidas de las placas Arduino son perfectamente útiles para controlar cargas que no consuman demasiada corriente, como un Led, pero son insuficientes para cargas mayores. ¿Cómo hacemos para controlar por ejemplo una lámpara o un motor que se alimentan de 220 voltios? Una forma es emplear un módulo de relés como el que analizo en este artículo.

Antes de comenzar con el análisis: como sucede con muchos módulos para Arduino que se consiguen en el mercado, la placa que utilicé no tiene una marca que identifique a su fabricante ni información "oficial" sobre su funcionamiento. Seguramente proviene de alguna factoría china que produce estos productos que luego son vendidos por numerosas tiendas a través de la red y del cual pueden existir distintas variedades, con diferentes características. Antes de conectarlo según las indicaciones que se dan a continuación, por favor, asegúrense de que el módulo que tienen entre sus manos sea igual al que se describe aquí.

Hecha esta salvedad, comencemos con una descripción general de la placa. Se trata de un módulo de 4 relés (o relays) que funcionan a 5 Voltios, capaces de manejar cargas de hasta 10 Amperes en 250 Voltios, convenientemente aislados mediante optoacopladores de las entradas, las que cuentan con leds individuales que sirven como indicadores de estado.

Los distintos componentes del módulo pueden verse en la siguiente imagen:

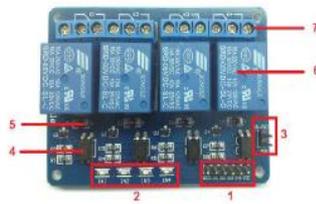


Fig. 1. Partes del módulo

el opto, Q1 se satura permitiendo el paso de la corriente a través de la bobina del relé, lo que produce que se cierren los contactos del mismo (común con normal abierto). El diodo D1 protege al transistor de la tensión que aparece en la bobina del relé cuando deja de circular corriente por la misma.

En síntesis, al ponerse la entrada a nivel BAJO se pone a la saturación el transistor Q1 a través del optoacoplador con lo que se cierra el contacto normal abierto del relé.

### Alimentación y consumo

La forma más sencilla de alimentar este módulo es desde Vcc y GND de la placa Arduino, manteniendo el Jumper en su lugar, con lo que JD-Vcc = Vcc. Esta conexión tiene dos limitaciones importantes:

- Se pierde la aislación eléctrica que brindan los optoacopladores, lo que aumenta la posibilidad de daño al Arduino si hay algún problema con las cargas de los relés.
- La corriente consumida por las bobinas de los relés debe ser provista por la placa Arduino. Cada bobina consume unos 90 mA y las cuatro juntas suman 360 mA. Si a esto le sumamos los consumos que pueden tener otras salidas, estamos muy cerca de los 500 mA que puede suministrar un puerto USB. En este caso se debería alimentar a Arduino con una fuente externa, lo que aumenta el límite de corriente a 1 A (en el caso de la Arduino UNO).

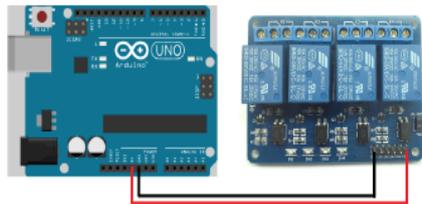


Fig. 3. Alimentación con una sola fuente

La forma más segura es remover el Jumper y alimentar la placa de relés con dos fuentes: la de la placa Arduino conectada a Vcc y una segunda fuente, con el positivo a JD-Vcc y el negativo a GND, sin estar éste unido a la placa Arduino. Esta conexión tiene como ventajas:

- Hay completa aislación entre la carga y el Arduino.
- Todo el consumo de los relés es tomado de la segunda fuente y no del Arduino o el puerto USB.

Como se puede apreciar, la placa tiene un conector de entradas (IN1 a IN4) y alimentación (GND es masa o negativo y Vcc es el positivo) [1], cuatro leds que indican el estado de las entradas [2], un Jumper selector para la alimentación de los relés [3], cuatro optoacopladores del tipo FL617C [4], cuatro diodos de protección [5], cuatro relés marca SONGLE con bobinas de 5V y contactos capaces de controlar hasta 10 Amperes en una tensión de 250V [6] y cuatro borneras, con tres contactos cada una (Común, Normal abierto y Normal cerrado), para las salidas de los relés [7].

### Esquemático

En la imagen de mas abajo se puede apreciar el circuito esquemático de un canal, el resto de los canales repite la misma configuración.

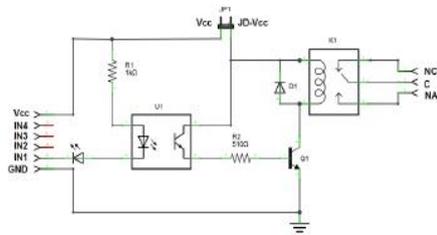


Fig. 2. Circuito esquemático

### Funcionamiento

A partir del circuito de la Fig. 2 analicemos el funcionamiento del circuito: la entrada IN1 está conectada al cátodo del diodo del optoacoplador a través del led indicador. El ánodo del diodo del optoacoplador se conecta a Vcc (positivo) por intermedio de R1, una resistencia de 1000 ohms. Estos tres componentes, el diodo indicador, el diodo del opto y la R1 forman un circuito serie por el cual circula la corriente cuando la entrada está a un nivel BAJO (conectada a GND) y no circula si la entrada está a un nivel ALTO (conectada a Vcc).

El transistor del opto tiene su colector a JD-Vcc y su emisor conectado a Q1 a través de una resistencia de 510 ohms. Este es otro circuito serie por el cual circula corriente cuando el transistor del opto conduce al ser "iluminado" por su diodo, con lo que se introduce corriente en la base de Q1 a través de R2.

Finalmente, Q1 está conectado en una típica configuración emisor común, con su emisor a masa (GND) y la bobina del relé como carga en el colector. Cuando circula corriente por la base desde

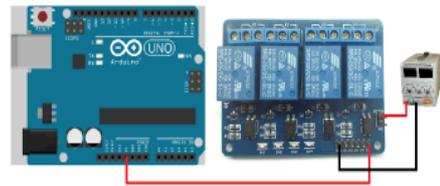


Fig. 4. Alimentación con dos fuentes

### Entradas

Las entradas a la placa puede conectarse directamente a las salidas digitales de la placa Arduino. La única precaución a tener en cuenta es que cuando Arduino arranca al ser alimentado, los pines se configuran como entradas automáticamente y puede ocurrir que, por un brevísimo lapso de tiempo entre el arranque y la correcta configuración de estos pines como salidas, las entradas de control al módulo de relé queden en un estado indeterminado. Esto se puede evitar conectando en cada entrada un pull-up con una resistencia de 10K a Vcc, lo que asegura un estado ALTO durante el arranque.

## **Anexo 2:** Software del Módulo de Supervisión

```
//MODULO DE SUPERVISION DE CONTAMINACION ACUSTICA
```

```
#include <LiquidCrystal.h>
```

```
#include <VirtualWire.h>
```

```
#include <Wire.h>
```

```
#include <Keypad.h>
```

```
#include "RTClib.h"
```

```
RTC_DS1307 RTC;
```

```
LiquidCrystal lcd(22, 24, 26, 28, 30, 32);
```

```
const byte ROWS = 4; //four rows
```

```
const byte COLS = 3; //three columns
```

```
char keys[ROWS][COLS] = {
```

```
  {'1','2','3'},
```

```
  {'4','5','6'},
```

```
  {'7','8','9'},
```

```
  {'*','0','#'}
```

```
};
```

```
byte rowPins[ROWS] = {31, 33, 35, 37}; //connect to the row pinouts of the keypad
```

```
byte colPins[COLS] = {43, 41, 39}; //connect to the column pinouts of the keypad
```

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

```
char tecla;
```

```
const double dBABajo = 10;
```

```
const double dBAMedio = 12;
```

```
const double dBAAlto = 17;
```

```
int count;
```

```
float mindB=110, maxdB=30;
```

```

int ledPin=13;

float valor=0;

int BotonFecha=0;

#define PIN_ENTRADA_ANALOGICA A0

//-----
//-----BOTON PARA CONFIGURAR FECHA-----

void BotonConfiguraFecha()
{
BotonFecha=1;
}

//-----
//-----IGUALAR FECHA Y HORA-----

void IgualarFH(int *Nodo)
{
int ano, mes, dia, hora, minuto, segundo;

unsigned long tiempo2=0;
unsigned long tiempo1=0;
unsigned long tiempo0=0;

int digito1, digito2, digito3, digito4;

int ndia, nmes, nano, nhora, nminuto;

char char1;
char char2;

tiempo2=millis();

//obtención de la fecha y hora
if (tiempo2-tiempo0>=500)

```

```
{  
    tiempo0=tiempo2;  
    DateTime now = RTC.now();  
    ano=now.year();  
    mes=now.month();  
    dia=now.day();  
    hora=now.hour();  
    minuto=now.minute();  
    segundo=now.second();  
}
```

```
tecla = keypad.waitForKey();
```

```
//entrar en ajuste
```

```
if (tecla=='*')
```

```
{
```

```
    //obtener fecha
```

```
    lcd.clear();
```

```
    lcd.print("Ajuste Fecha");
```

```
    lcd.setCursor(0,1);
```

```
    lcd.print("00/00/0000 ");
```

```
    //obtener dia
```

```
    char1=keypad.waitForKey();
```

```
    digito1=convertir(char1);
```

```
    char1=keypad.waitForKey();
```

```
    digito2=convertir(char1);
```

```
    ndia=digito1*10+digito2;
```

```
    lcd.setCursor(0,1);
```

```
    lcd.print(ndia);
```

```
    lcd.print("/00/0000 ");
```

```
    //obtener mes
```

```

char1=keypad.waitForKey();
digito1=convertir(char1);
char1=keypad.waitForKey();
digito2=convertir(char1);
nmes=digito1*10+digito2;
lcd.setCursor(0,1);
lcd.print(ndia);
lcd.print("/");
lcd.print(nmes);
lcd.print("/0000 ");

//obtener año
char1=keypad.waitForKey();
digito1=convertir(char1);
char1=keypad.waitForKey();
digito2=convertir(char1);
char1=keypad.waitForKey();
digito3=convertir(char1);
char1=keypad.waitForKey();
digito4=convertir(char1);
nano=digito1*1000+digito2*100+digito3*10+digito4;
lcd.setCursor(0,1);
lcd.print(ndia);
lcd.print("/");
lcd.print(nmes);
lcd.print("/");
lcd.print(nano);
lcd.print(" ");

//obtener hora
delay (1000);
lcd.clear();
lcd.print("Ajuste Hora");

```

```
lcd.setCursor(0,1);  
lcd.print("00:00");  
  
//obtener hora  
char1=keypad.waitForKey();  
digito1=convertir(char1);  
char1=keypad.waitForKey();  
digito2=convertir(char1);  
nhora=digito1*10+digito2;  
lcd.setCursor(0,1);  
lcd.print(nhora);  
lcd.print(":00 ");  
  
//obtener minuto  
char1=keypad.waitForKey();  
digito1=convertir(char1);  
char1=keypad.waitForKey();  
digito2=convertir(char1);  
nminuto=digito1*10+digito2;  
lcd.setCursor(0,1);  
lcd.print(nhora);  
lcd.print(":");  
lcd.print(nminuto);  
lcd.print(" ");  
  
delay(1000);  
  
//ajustar hora  
RTC.adjust(DateTime(nano,nmes,ndia,nhora,nminuto,00));  
  
lcd.clear();
```

```
lcd.print("Actualizado");  
delay (1000);
```

```
}
```

```
if (tecla=='#')
```

```
{
```

```
lcd.clear();
```

```
lcd.print("Ajuste Modulo");
```

```
lcd.setCursor(0,1);
```

```
char1=keypad.waitForKey();
```

```
digito1=convertir(char1);
```

```
*Nodo=digito1;
```

```
lcd.clear();
```

```
lcd.setCursor(0,0);
```

```
lcd.print("NODO:");
```

```
lcd.setCursor(0,1);
```

```
lcd.print(*Nodo);
```

```
delay (2000);
```

```
lcd.clear();
```

```
lcd.print("Actualizado");
```

```
delay (1000);
```

```
}
```

```
}
```

```
//-----
```

```
//-----CONVERTIR CARACTERES A ENTEROS-----
```

```
int convertir (char letra)
```

```
{
```

```
int numero;

switch (letra){

    case '1': numero=1;

    break;

    case '2': numero=2;

    break;

    case '3': numero=3;

    break;

    case '4': numero=4;

    break;

    case '5': numero=5;

    break;

    case '6': numero=6;

    break;

    case '7': numero=7;

    break;

    case '8': numero=8;

    break;

    case '9': numero=9;

    break;

    case '0': numero=0;

    break;

}

return numero;

}

//-----

//-----CAICULO DE VALORES EN DB-----

float calculodecibels(int x, char c)

{

    float decibelscalculados;
```

```

    if (c == 'b')
        decibelscalculados = 20 * log10(x/dBABajo);
    if (c == 'm')
        decibelscalculados = 20 * log10(x/dBAMedio);
    if (c == 'a')
        decibelscalculados = 20 * log10(x/dBAAlto);

    return (decibelscalculados);
}

//-----
//-----CADENA DE LA HORA Y FECHA-----

void FormaFeHo(char FeHo[])
{
    delay (500);

    DateTime FechaHora = RTC.now();//funcion que toma fecha y hora del Reloj de tiempo real
    sprintf(FeHo, "%02d/%02d/%04d %02d:%02d:%02d%c",
    FechaHora.day(),FechaHora.month(),FechaHora.year(),
    FechaHora.hour(),FechaHora.minute(), FechaHora.second(),NULL);

    lcd.setCursor(0,0);

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print(FechaHora.day());

    lcd.setCursor(2,0);

    lcd.print('/');

    lcd.setCursor(3,0);

    lcd.print(FechaHora.month());

    lcd.setCursor(5,0);

    lcd.print('/');

    lcd.setCursor(6,0);

    lcd.print(FechaHora.year());
}

```

```

    lcd.setCursor(0,1);

    lcd.print(FechaHora.hour());

    lcd.setCursor(2,1);

    lcd.print(':');

    lcd.setCursor(3,1);

    lcd.print(FechaHora.minute());

    lcd.print(':');

    lcd.print(FechaHora.second());
}

//-----
//-----MENSAJE PARA TRANSMISIÓN-----

void CrearMensajeEnviar(int Nodo,char FeHo[], int valordB, int maxdB, int mindB, char Cadena[])
{
    sprintf(Cadena,"%1dFecha_Hora%19sA%03dMA%03dMI%03d%c",Nodo,FeHo,valordB,maxdB,mindB,
    NULL);
}

//-----
//-----TRANSMISION RF-----

void EnvioporRF( char Cadena[])
{
    digitalWrite(13, true);

    char *msg = Cadena;

    // transmite mensaje RF

    vw_send((uint8_t *)msg, strlen(msg));

    vw_wait_tx(); // espera finalizar transmisión

    digitalWrite(13, false);

    delay(50);
}

```

```

//-----
//-----VISUALIZACION CADENA ENVIADA-----

void CadenaenSerial (char Cadena[])
{
Serial.print(Cadena);
Serial.print("\n");
Serial.print("\r");
}

//-----
//-----PROCESO SETUP-----

void setup() {
Serial.begin(9600);
//Serial.println("Transmisor listo");
Wire.begin();
RTC.begin();
lcd.begin(16, 2);

/*if (! RTC.isrunning()) //RELOJ
{
RTC.adjust(DateTime(__DATE__, __TIME__));
}*/

pinMode(ledPin, OUTPUT);
pinMode(A0, INPUT);

count = 0;

vw_set_tx_pin(12);

```

```

vw_set_ptt_inverted(true);

vw_setup(4000);

attachInterrupt(1,BotonConfiguraFecha,CHANGE);

}

//-----
//-----PROCESO CICLICO (LOOP)-----

void loop() {
  delay(500);
  int valor, i;
  // static int retardo=0;
  float decibelsBajo = 42;
  float decibelsMedio = 60.0;
  float decibelsAlto = 70.0;
  int valordB;
  float valorfinal;
  static int Nodo=0;
  char Cadena[50];
  char FeHo[21];

  FormaFeHo(FeHo);

  if (BotonFecha==0){

  valor = analogRead(PIN_ENTRADA_ANALOGICA); //LECTURA DE LOS DATOS ANALOGICOS

  //-----
  //-----CALCULO EN DECIBELS-----

```

```

if (valor <= 13)
{
    decibelsBajo += calculodecibels(valor, 'b');
    valordB = decibelsBajo;

    if (maxdB < valordB)
        maxdB = valordB;

    if (valordB < mindB)
        mindB = valordB;

    lcd.setCursor(11,1);
    lcd.print(valordB);
    lcd.setCursor(14,1);
    lcd.print("dB");
    lcd.setCursor(11,0);
    lcd.print("N:");
    lcd.setCursor(13,0);
    lcd.print(Nodo);

    CrearMensajeEnviar(Nodo,FeHo,valordB,maxdB,mindB,Cadena); //FORMAR EL MENSAJE PARA
    ENVIAR

    EnvioporRF(Cadena); //ENVIO DE LA CADENA

    CadenaenSerial (Cadena); //VISUALIZAR LA CADENA QUE SE ENVIA

    count = count+1;
}

else if ((valor > 13) && ( valor <=22) )
{

    decibelsMedio += calculodecibels(valor, 'm');
    valordB= decibelsMedio;

```

```
if (maxdB < valordB)
```

```
    maxdB = valordB;
```

```
if (valordB < mindB)
```

```
    mindB = valordB;
```

```
lcd.setCursor(11,1);
```

```
lcd.print(valordB);
```

```
lcd.setCursor(14,1);
```

```
lcd.print("dB");
```

```
lcd.setCursor(11,0);
```

```
lcd.print("N:");
```

```
lcd.setCursor(13,0);
```

```
lcd.print(Nodo);
```

```
CrearMensajeEnviar(Nodo,FeHo,valordB,maxdB,mindB,Cadena); //FORMAR EL MENSAJE PARA ENVIAR
```

```
EnvioporRF(Cadena); //ENVIO DE LA CADENA
```

```
CadenaenSerial (Cadena); //VISUALIZAR LA CADENA QUE SE ENVIA
```

```
    count = count+1;
```

```
}
```

```
else if(valor > 22)
```

```
{
```

```
    decibelsAlto += calculodecibels(valor, 'a');
```

```
    valordB = decibelsAlto;
```

```
if (maxdB < valordB)
```

```
    maxdB = valordB;
```

```
if (valordB < mindB)
```

```
    mindB = valordB;
```

```

    lcd.setCursor(11,1);
    lcd.print(valordB);
    lcd.setCursor(14,1);
    lcd.print("dB");
    lcd.setCursor(11,0);
    lcd.print("N:");
    lcd.setCursor(13,0);
    lcd.print(Nodo);

    CrearMensajeEnviar(Nodo,FeHo,valordB,maxdB,mindB,Cadena); //FORMAR EL MENSAJE PARA
    ENVIAR

    EnvioporRF(Cadena); //ENVIO DE LA CADENA

    CadenaenSerial(Cadena); //VISUALIZAR LA CADENA QUE SE ENVIA

    count = count+1;

}

}

else

{

/*Serial.println("USTED HA INGRESADO AL PROCESO DE CONFIGURACION DEL MODULO DE
SUPERVISION");

delay(100);

Serial.println("PRESIONES LA TECLA * PARA CONFIGURAR SU RELOJ.");

Serial.println("PRESIONES LA TECLA # PARA CONFIGURAR EL NODO DE UBICACION.");*/

IgualarFH(&Nodo);

    BotonFecha=0;

}

}

```

### **Anexo 3:** Software del módulo de Aviso

```
//MODULO DE AVISO
#include <VirtualWire.h>

int PIN_1=1;
int PIN_2=2;
int PIN_3=3;
int PIN_4=4;
int PIN_5=5;
int PIN_6=6;
int PIN_7=7;
int PIN_8=8;
int PIN_9=9;
int PIN_10=10;
int PIN_11=11;

void setup() {
  Serial.begin(9600);
  vw_set_rx_pin(12); // Pin en el que conectamos la patilla data del receptor
  vw_set_ptt_inverted(true); // Requerido por el modulo RF
  vw_setup(4000); // Velocidad de conexión bps
  vw_rx_start(); // Inicializar el receptor
  pinMode(PIN_1, OUTPUT);
  pinMode(PIN_2, OUTPUT);
  pinMode(PIN_3, OUTPUT);
  pinMode(PIN_4, OUTPUT);
  pinMode(PIN_5, OUTPUT);
  pinMode(PIN_6, OUTPUT);
  pinMode(PIN_7, OUTPUT);
  pinMode(PIN_8, OUTPUT);
  pinMode(PIN_9, OUTPUT);
  pinMode(PIN_10, OUTPUT);
  pinMode(PIN_11, OUTPUT);
  pinMode(13, OUTPUT);
}
```

```

//-----BLOQUE DE REPETICION-----
void loop() {
char Cadena [46];
LeerMensajeRF(Cadena);
}

//-----SEPARAR MENSAJE-----
void DescomprimirMensaje(String Cadena, String &Nodo,String &FeHo, String &valordB, String
&maxdB, String &mindB)
{
Nodo=Cadena.substring(0,1);
FeHo=Cadena.substring(11,30);
valordB=Cadena.substring(31,34);
maxdB=Cadena.substring(36,39);
mindB=Cadena.substring(41,45);
}

//-----LEER MENSAJE-----
void LeerMensajeRF(char Cadena[])
{
delay(100);
String Nodo=" ";
String FeHo=" ";
String valordB=" ";
String maxdB=" ";
String mindB=" ";
//delay(10);
digitalWrite(13, HIGH);
uint8_t buf[VW_MAX_MESSAGE_LEN];
uint8_t buflen = VW_MAX_MESSAGE_LEN;
if (vw_get_message(buf, &buflen) // Comprobamos si hemos
{
int i;

```

```

for (i = 0; i < buflen; i++)
{
Cadena[i]=buf[i];
}
Cadena[i]='\0';
//Serial.write("\n");
DescomprimirMensaje(Cadena,Nodo,FeHo,valor dB,max dB,min dB);
Serial.println(Cadena);
int valordb = atoi(valor dB.c_str());
//Serial.println(valordb);
ecualizador(valordb);
}
digitalWrite(13, LOW);
}

//-----
//-----ECUALIZADOR-----
void ecualizador(int valordb)
{
if ((valordb<=25))
{
digitalWrite(PIN_1, HIGH);
digitalWrite(PIN_2, LOW);
digitalWrite(PIN_3, LOW);
digitalWrite(PIN_4, LOW);
digitalWrite(PIN_5, LOW);
digitalWrite(PIN_6, LOW);
digitalWrite(PIN_7, LOW);
digitalWrite(PIN_8, LOW);
digitalWrite(PIN_9, LOW);
digitalWrite(PIN_10, LOW);
digitalWrite(PIN_11, LOW);
}
else if ((valordb >25 )&&(valordb<=35))
{
digitalWrite(PIN_1, HIGH);

```

```
digitalWrite(PIN_2, HIGH);
digitalWrite(PIN_3, LOW);
digitalWrite(PIN_4, LOW);
digitalWrite(PIN_5, LOW);
digitalWrite(PIN_6, LOW);
digitalWrite(PIN_7, LOW);
digitalWrite(PIN_8, LOW);
digitalWrite(PIN_9, LOW);
digitalWrite(PIN_10, LOW);
digitalWrite(PIN_11, LOW);
}
else if ((valordb >35 )&&(valordb<=45))
{
digitalWrite(PIN_1, HIGH);
digitalWrite(PIN_2, HIGH);
digitalWrite(PIN_3, HIGH);
digitalWrite(PIN_4, LOW);
digitalWrite(PIN_5, LOW);
digitalWrite(PIN_6, LOW);
digitalWrite(PIN_7, LOW);
digitalWrite(PIN_8, LOW);
digitalWrite(PIN_9, LOW);
digitalWrite(PIN_10, LOW);
digitalWrite(PIN_11, LOW);
}
else if ((valordb >45 )&&(valordb<=55))
{
digitalWrite(PIN_1, HIGH);
digitalWrite(PIN_2, HIGH);
digitalWrite(PIN_3, HIGH);
digitalWrite(PIN_4, HIGH);
digitalWrite(PIN_5, LOW);
digitalWrite(PIN_6, LOW);
digitalWrite(PIN_7, LOW);
digitalWrite(PIN_8, LOW);
digitalWrite(PIN_9, LOW);
```

```
digitalWrite(PIN_10, LOW);
digitalWrite(PIN_11, LOW);
}
else if ((valordb >55 )&&(valordb<=65))
{
digitalWrite(PIN_1, HIGH);
digitalWrite(PIN_2, HIGH);
digitalWrite(PIN_3, HIGH);
digitalWrite(PIN_4, HIGH);
digitalWrite(PIN_5, HIGH);
digitalWrite(PIN_6, LOW);
digitalWrite(PIN_7, LOW);
digitalWrite(PIN_8, LOW);
digitalWrite(PIN_9, LOW);
digitalWrite(PIN_10, LOW);
digitalWrite(PIN_11, LOW);
}
else if ((valordb >65 )&&(valordb<=75))
{
digitalWrite(PIN_1, HIGH);
digitalWrite(PIN_2, HIGH);
digitalWrite(PIN_3, HIGH);
digitalWrite(PIN_4, HIGH);
digitalWrite(PIN_5, HIGH);
digitalWrite(PIN_6, HIGH);
digitalWrite(PIN_7, LOW);
digitalWrite(PIN_8, LOW);
digitalWrite(PIN_9, LOW);
digitalWrite(PIN_10, LOW);
digitalWrite(PIN_11, LOW);
}
else if ((valordb >75 )&&(valordb<=85))
{
digitalWrite(PIN_1, HIGH);
digitalWrite(PIN_2, HIGH);
digitalWrite(PIN_3, HIGH);
```

```
digitalWrite(PIN_4, HIGH);
digitalWrite(PIN_5, HIGH);
digitalWrite(PIN_6, HIGH);
digitalWrite(PIN_7, HIGH);
digitalWrite(PIN_8, LOW);
digitalWrite(PIN_9, LOW);
digitalWrite(PIN_10, LOW);
digitalWrite(PIN_11, LOW);
}
else if ((valordb >85 )&&(valordb<=95))
{
digitalWrite(PIN_1, HIGH);
digitalWrite(PIN_2, HIGH);
digitalWrite(PIN_3, HIGH);
digitalWrite(PIN_4, HIGH);
digitalWrite(PIN_5, HIGH);
digitalWrite(PIN_6, HIGH);
digitalWrite(PIN_7, HIGH);
digitalWrite(PIN_8, HIGH);
digitalWrite(PIN_9, LOW);
digitalWrite(PIN_10, LOW);
digitalWrite(PIN_11, LOW);
}
else if ((valordb >95 )&&(valordb<=105))
{
digitalWrite(PIN_1, HIGH);
digitalWrite(PIN_2, HIGH);
digitalWrite(PIN_3, HIGH);
digitalWrite(PIN_4, HIGH);
digitalWrite(PIN_5, HIGH);
digitalWrite(PIN_6, HIGH);
digitalWrite(PIN_7, HIGH);
digitalWrite(PIN_8, HIGH);
digitalWrite(PIN_9, HIGH);
digitalWrite(PIN_10, LOW);
digitalWrite(PIN_11, LOW);
```

```
}  
else if ((valordb >105 )&&(valordb<=115))  
{  
    digitalWrite(PIN_1, HIGH);  
    digitalWrite(PIN_2, HIGH);  
    digitalWrite(PIN_3, HIGH);  
    digitalWrite(PIN_4, HIGH);  
    digitalWrite(PIN_5, HIGH);  
    digitalWrite(PIN_6, HIGH);  
    digitalWrite(PIN_7, HIGH);  
    digitalWrite(PIN_8, HIGH);  
    digitalWrite(PIN_9, HIGH);  
    digitalWrite(PIN_10, HIGH);  
    digitalWrite(PIN_11, LOW);  
}  
else if ((valordb>115))  
{  
    digitalWrite(PIN_1, HIGH);  
    digitalWrite(PIN_2, HIGH);  
    digitalWrite(PIN_3, HIGH);  
    digitalWrite(PIN_4, HIGH);  
    digitalWrite(PIN_5, HIGH);  
    digitalWrite(PIN_6, HIGH);  
    digitalWrite(PIN_7, HIGH);  
    digitalWrite(PIN_8, HIGH);  
    digitalWrite(PIN_9, HIGH);  
    digitalWrite(PIN_10, HIGH);  
    digitalWrite(PIN_11, HIGH);  
}  
}
```

#### **Anexo 4:** Software del módulo de Almacenamiento

```
#include <SPI.h>
#include <SD.h>
#include <VirtualWire.h>

File Archivo;

char Datos[]="Datos.txt";
//char Datos[]="Datos.xlsx";

//-----
//-----SEPARAR MENSAJE-----

void DescomprimirMensaje(String Cadena, String &Nodo,String &FeHo, String &valordB, String
&maxdB, String &mindB)
{
Nodo=Cadena.substring(0,1);
FeHo=Cadena.substring(11,30);
valordB=Cadena.substring(31,34);
maxdB=Cadena.substring(36,39);
mindB=Cadena.substring(41,45);
}

//-----
//-----FORMAR ARCHIVOS PARA LA SD-----

void FormarMensajeParaSD(String &MSD, String Nodo,String FeHo, String valordB, String maxdB,
String mindB)
{
char Tabulador= '\t';

MSD=(Nodo+ Tabulador+ FeHo.substring(0,10)+Tabulador +FeHo.substring(11,19) +Tabulador +
valordB + Tabulador + Tabulador + maxdB + Tabulador + Tabulador +mindB);

//Serial.println(MSD);
}

//-----
//-----ESCRIBE LOS ARCHIVOS EN LA SD-----

void EscribeDatosEnSD(String &MSD, char *Datos)
{
```

```
Archivo = SD.open(Datos, FILE_WRITE);//abrimos el archivo
```

```
if (Archivo) {  
    delay(100);  
    Archivo.println(MSD);  
    Archivo.flush();  
    //Serial.println(MSD);  
}  
}
```

```
//-----  
//-----LEER MENSAJE-----
```

```
void LeerMensajeRF(char Cadena[],char *Datos)
```

```
{  
    delay(500);  
    String Nodo=" ";  
    String FeHo=" ";  
    String valordB=" ";  
    String maxdB=" ";  
    String mindB=" ";  
    String MSD=" ";  
    digitalWrite(13, HIGH);  
    uint8_t buf[VW_MAX_MESSAGE_LEN];  
    uint8_t buflen = VW_MAX_MESSAGE_LEN;  
    if (vw_get_message(buf, &buflen) // Comprobamos si hemos  
    {  
        int i;  
  
        for (i = 0; i < buflen; i++)  
        {  
            //Serial.write(buf[i]); // Los datos recibidos son guardados en el buffer y enviados al el puerto serie al  
            ordenador  
            Cadena[i]=buf[i];  
        }  
    }  
}
```

```

}
Cadena[i]='\0';
//Serial.write("\n");
DescomprimirMensaje(Cadena,Nodo,FeHo,valor dB,maxdB,min dB);
Serial.println(Cadena);
FormarMensajeParaSD(MSD,Nodo,FeHo,valor dB,maxdB,min dB);
EscribeDatosEnSD(MSD,Datos);

}

digitalWrite(13, LOW); // Parpadeo para mostrar actividad

}

//-----
//-----CREAR ARCHIVO PARA ALMACENAMIENTO DE DATOS-----

void configurararchivo(char *Datos)
{
String Linea0=( "\t\tNIVELES DE RUIDO REGISTRADOS EN LA ESPOCH");
String Linea1=("NODO\tFECHA\tHORA\tR-ACTUAL\tR-MAXIMO\tR-MINIMO");

if(!SD.exists(Datos)) {
    Archivo = SD.open(Datos, FILE_WRITE);
    Archivo.println(Linea0);
    Archivo.println(Linea1);
    Archivo.flush();

}
}

void setup() {
    Serial.begin(9600);

```

```
vw_set_rx_pin(12); // Pin en el que conectamos la patilla data del receptor
vw_set_ptt_inverted(true); // Requerido por el modulo RF
vw_setup(4000); // Velocidad de conexión bps
vw_rx_start();
pinMode(13, OUTPUT);
Serial.begin(9600);
// Serial.print("Iniciando SD ...");
if (!SD.begin(4)) {
    //Serial.println("No se pudo inicializar");
    return;
}
//Serial.println("inicializacion exitosa");
configurararchivo(Datos);
}
void loop() {
char Cadena [46];
LeerMensajeRF(Cadena,Datos);
}
```