

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA MEDICIÓN DE RUIDO EN EL AMBIENTE

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y REDES DE INFORMACIÓN**

WILSON SANTIAGO GARCÍA SÁNCHEZ

DIRECTOR: ING. JORGE EDUARDO CARVAJAL RODRÍGUEZ

CO – DIRECTOR: ING. CHRISTIAN JOSÉ TIPANTUÑA TENELEMA

Quito, abril 2017

DECLARACIÓN

Yo, Wilson Santiago García Sánchez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentada para ningún grado o calificación profesional, y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa Institucional vigente.

Wilson Santiago García Sánchez

CERTIFICACIÓN

Certificamos que el trabajo fue desarrollado por Wilson Santiago García Sánchez, bajo nuestra supervisión.

Ing. Jorge Eduardo Carvajal Rodríguez, MSc

DIRECTOR DEL PROYECTO

Ing. Christian José Tipantuña Tenelema, MSc.

CO - DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

Agradezco infinitamente a Dios, mis padres por todo el apoyo brindado durante toda mi vida, a mis hermanos y a mi familia que siempre estuvieron presentes.

A Deysi por su apoyo y respaldo en todo momento para realizar el presente trabajo.

A los Ing. Jorge Eduardo Carvajal Rodríguez y Christian José Tipantuña Tenelema por todo su apoyo y paciencia para culminar con éxitos este proyecto de titulación.

A mis amig@s que formaron parte de mi carrera universitaria, gracias por todos los momentos compartidos.

DEDICATORIA

Xavier Garcia

CONTENIDO

1.1 RUIDO	1
1.1.1 NATURALEZA DEL RUIDO	1
1.1.2 CARACTERÍSTICAS PRINCIPALES DEL RUIDO	3
1.1.3 TIPOS DE RUIDO [5]	4
1.1.4 CONTAMINACIÓN SONORA	4
1.1.5 FUENTES EMISORAS DE RUIDO [8]	5
1.1.6 NIVELES PERMISIBLES DE RUIDO DE ACUERDO AL USO DE SUELO EN EL DISTRITO METROPOLITANO DE QUITO.	5
1.1.7 MANERAS DE DISMINUIR EL RUIDO [10]	6
1.2 CARACTERÍSTICAS TÉCNICAS DE ARDUINO UNO Y RASBERRY PI.	7
1.2.1 ARDUINO UNO [11]	7
1.2.2 RASBERRY PI [14]	8
1.3 SENSOR DE RUIDO [16]	10
1.4 SISTEMA DE POSICIONAMIENTO GLOBAL (GPS) [17]	10
1.4.1 PRINCIPIO DE FUNCIONAMIENTO DEL SISTEMA GPS	10
1.4.2 MÓDULO GPS [18]	11
1.5 HYPERTEXT TRANSFER PROTOCOLO [19]	11
1.5.1 FUNCIONAMIENTO DEL PROTOCOLO HTTP	12
1.5.2 SISTEMAS INTERMEDIARIOS PARA HTTP	12
1.5.3 MENSAJE HTTP	13
1.6 SOUND METER [1]	15
2.1 REQUERIMIENTOS DEL SISTEMA PROTOTIPO	17
2.2 ARQUITECTURA DEL SISTEMA PROTOTIPO	18
2.3 DISEÑO DEL SUBSISTEMA TRANSMISOR	19
2.3.1 BLOQUE DE ADQUISICIÓN DE DATOS	19
2.3.2 BLOQUE ESTRUCTURACIÓN DE DATOS	19
2.3.3 BLOQUE DE ENVIÓ DE DATOS	20

2.4 DISEÑO DEL SUBSISTEMA RECEPTOR.....	21
2.4.1 BLOQUE DE RECEPCIÓN Y ALMACENAMIENTO DE DATOS.....	22
2.4.2 BLOQUE DE PRESENTACIÓN DE RESULTADOS	22
2.5 SUBSISTEMA TRANSMISOR.....	23
2.5.1 SELECCIÓN DE LOS ELEMENTOS DE HARDWARE PARA EL BLOQUE DE ADQUISICIÓN DE DATOS	23
2.5.2 SELECCIÓN DE LOS ELEMENTOS DE HARDWARE PARA EL BLOQUE DE ESTRUCTURACIÓN DE DATOS.....	27
2.5.3 SELECCIÓN DE LOS ELEMENTOS DE HARDWARE PARA EL BLOQUE DE ENVIO DE DATOS.....	28
2.5.4 SELECCIÓN DE LOS ELEMENTOS DE SOFTWARE PARA EL SUBSISTEMA TRANSMISOR	30
2.5.5 CONFIGURACIÓN DE LA SINGLE BOARD COMPUTER.....	31
2.5.6 ACOPLAMIENTO DE LOS ELEMENTOS DE HARDWARE PARA LA CONSTRUCCIÓN DEL SUBSISTEMA TRANSMISOR.....	40
2.5.7 PROGRAMACIÓN DE LA PLATAFORMA RASPBERRY PI	44
2.6 SUBSISTEMA RECEPTOR.....	56
2.6.1 SELECCIÓN DE LOS ELEMENTOS DE HARDWARE PARA EL SUBSISTEMA RECEPTOR.	57
2.6.2 SELECCIÓN DE LOS ELEMENTOS DE SOFTWARE PARA EL BLOQUE DE RECEPCIÓN Y ALMACENAMIENTO DE DATOS	58
2.6.3 SELECCIÓN DE LOS ELEMENTOS DE SOFTWARE PARA EL BLOQUE DE PRESENTACIÓN DE RESULTADOS	60
2.6.4 PROGRAMACIÓN DEL SUBSISTEMA RECEPTOR.....	64
3.1 CALIBRACIÓN DEL SENSOR DE RUIDO	72
3.2 PRUEBAS DE FUNCIONALIDAD.....	76
3.2.1 FASE DE ADQUISICIÓN DE DATOS	77
3.2.2 FASE DE ENVIÓ DE DATOS.....	79
3.2.3 FASE DE RECEPCIÓN Y ALMACENAMIENTO DE DATOS	82
3.2.4 FASE DE PRESENTACIÓN Y VISUALIZACIÓN DE RESULTADOS.....	85

3.3 PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA PROTOTIPO	87
3.3.1 DESCRIPCIÓN DEL ESCENARIO DE PRUEBAS.....	88
3.3.2 PRUEBAS DE MEDICIÓN DE RUIDO	88
3.4 MEDICIONES DE RUIDO	90
3.4.1 MEDICIONES EN MEDIO DE LOS EDIFICIOS QUÍMICA – ELECTRÓNICA Y ELÉCTRICA	90
3.4.2 MEDICIONES EN EL ÁREA RECRATIVA DEL EDIFICIO DE INGENIERÍA EN SISTEMAS.....	92
3.4.3 MEDICIONES EN EL HALL DEL EDIFICIO DE MEDIOS EXTERNOS ...	94
3.5 ANÁLISIS DE RESULTADOS	97
3.5.1 ANÁLISIS DE RESULTADOS DE LA MEDICIÓN DE RUIDO EN MEDIO DE LOS EDIFICIOS ELÉCTRICA – QUIMICA Y ELÉCTRICA.....	97
3.5.2 ANÁLISIS DE RESULTADOS DE LA MEDICIÓN DE RUIDO EN EL ÁREA RECREATIVA DEL EDIFICIO DE INGENIERÍA EN SISTEMAS	99
3.5.3 ANÁLISIS DE RESULTADOS DE LA MEDICIÓN DE RUIDO EN EL HALL DEL EDIFICIO DE MEDIOS EXTERNOS	101
3.6 COSTO REFERENCIAL DEL SISTEMA PROTOTIPO.....	102
4.1 CONCLUSIONES.....	104
4.2 RECOMENDACIONES.....	105
REFERENCIAS BIBLIOGRÁFICAS	107

ÍNDICE DE FIGURAS

Capítulo 1

Figura 1.1 Diagrama de la placa Arduino Uno [13].....	8
Figura 1.2 Esquema de Raspberry Pi 1 Model B+ [15].....	9
Figura 1.3 Ejemplo de funcionamiento de HTTP.....	12
Figura 1.4 Sound Meter versión 3.1.4 [1]	16

Capítulo 2

Figura 2.1 Diagrama de casos de uso del sistema prototipo	18
Figura 2.2 Arquitectura del Sistema Prototipo	18
Figura 2.3 Esquema del Subsistema Transmisor	19
Figura 2.4 Estructura de los datos tomados por el Subsistema Transmisor	20
Figura 2.5 Esquema del Subsistema Receptor	21
Figura 2.6 Esquema de conexión a Internet del subsistema receptor	21
Figura 2.7 Diseño de la Base de Datos	22
Figura 2.8 Esquema del subsistema transmisor.....	23
Figura 2.9 Micrófono USB PS3 [21]	24
Figura 2.10 Módulo GPS.....	25
Figura 2.11 Esquema de la placa GY – GPSSGMV2 [18]	26
Figura 2.12 Esquema de la tarjeta Raspberry Pi 1.2 [14]	27
Figura 2.13 Distribución de pines de la placa Raspberry Pi B + [23]	28
Figura 2.14 Sistema Operativo Raspbian.....	32
Figura 2.15 Archivo Raspbian de tipo img.....	33
Figura 2.16 Escribiendo el sistema operativo Raspbian en memoria micro SD....	33
Figura 2.17 Ranura donde se inserta la memoria micro SD	34
Figura 2.18 Dirección IP que toma la placa Raspberry Pi	34
Figura 2.19 Moba Xtrem	35
Figura 2.20 Ejecución del comando sudo apt-get install wrdp.....	36
Figura 2.21 Conexión a la Raspberry Pi mediante escritorio remoto	36
Figura 2.22 Ingreso a la Raspberry Pi.....	36
Figura 2.23 Escritorio de la Raspberry Pi	37
Figura 2.24 Conexión del módulo GPS.	40
Figura 2.25 Conexión del Micrófono USB	42

Figura 2.26 Conexión del modem 3G.....	42
Figura 2.27 Diagrama de secuencias para la implementación del subsistema transmisor.....	44
Figura 2.28 Campos del mensaje NMEA	46
Figura 2.29 Diagrama de flujo del proceso de recolección de datos Geolocalización	46
Figura 2.30 Diagrama de flujo de la medición de ruido.....	48
Figura 2.31 Diagrama de flujo del túnel SSH	50
Figura 2.32 Interacción con la base de datos.....	52
Figura 2.33 Diagrama de flujo del menú principal	54
Figura 2.34 Estructura de un documento HTML [38].....	61
Figura 2.35 Diagrama de flujo del servidor web	64
Figura 2.36 Diagrama de secuencias del subsistema receptor	65

Capítulo 3

Figura 3.1 Parlantes de computadora SP-S110 [40]	72
Figura 3.2 Diagrama de bloques de la fuente generadora de ruido	73
Figura 3.3 Sistema prototipo tomando lecturas de ruido cerca de los parlantes SP-S110	74
Figura 3.4 Comparación de mediciones entre el valor de referencia y los valores obtenidos, previo a la calibración	75
Figura 3.5 Comparación de mediciones entre el valor de referencia y los valores obtenidos luego de la calibración	76
Figura 3.6 Ejecución del programa gpsRead.py.....	77
Figura 3.7 Presentación de resultados de latitud y longitud	77
Figura 3.8 Presentación de resultados en cero del módulo GPS	78
Figura 3.9 Presentación de mensaje sin resultados de latitud y longitud.....	78
Figura 3.10 Ejecución del programa audioDevices.py.....	78
Figura 3.11 Reconocimiento de los dispositivos de audio en la SBC	79
Figura 3.12 Dispositivos conectados a la SBC.....	79
Figura 3.13 Configuración de Mobile Broadband Connect	80
Figura 3.14 Configuración general de Mobile Broadband Connect.....	80
Figura 3.15 Configuración de DNS de Mobile Broadband Connect.....	81
Figura 3.16 Conexión activa de Mobile Broadband Connect	81

Figura 3.17 Ping al servidor de Google	81
Figura 3.18 Ping al dominio www.google.com.....	82
Figura 3.19 Ejecución del script mysqlHandler.py	82
Figura 3.20 Ingreso a MySql Admin	83
Figura 3.21 Almacenamiento de los datos enviados por el subsistema transmisor	84
Figura 3.22 Comparación de datos enviados y recibidos en el sistema prototipo .	84
Figura 3.23 Interfaz del servidor web	85
Figura 3.24 Selección del intervalo de tiempo de análisis	86
Figura 3.25 Desactivación de fechas anteriores a la fecha de inicio.....	86
Figura 3.26 Presentación de mediciones de acuerdo al intervalo de tiempo seleccionado	87
Figura 3.27 Presentación de resultados en el mapa mediante un mapa de calor .	87
Figura 3.28 Datos recolectados en la etapa de pruebas	88
Figura 3.29 Medición realizada con la aplicación Sound Meter [1]	89
Figura 3.30 Comparación de las lecturas tomadas por el sistema prototipo y la aplicación Sound Meter.....	90
Figura 3.31 Ruido vs Tiempo en la sesión 1, en medio de los edificios Electrónica – Química y Eléctrica	97
Figura 3.32 Ruido vs Tiempo en la sesión 1, en medio de los edificios Electrónica – Química y Eléctrica	98
Figura 3.33 Ruido vs Tiempo en la sesión 1, en el área recreativa del edificio de Ingeniería en Sistemas	99
Figura 3.34 Ruido vs Tiempo en la sesión 2, en el área recreativa del edificio de Ingeniería en Sistemas	100
Figura 3.35 Ruido vs Tiempo en la sesión 1, en el hall del edificio de Medios Externos.....	101
Figura 3.36 Ruido vs Tiempo en la sesión 2, en el hall del edificio de Medios Externos.....	102

ÍNDICE DE TABLAS

Capítulo 1

Tabla 1.1 Velocidad de sonido en el aire de acuerdo a la temperatura [4]	2
Tabla 1.2 Efectos del ruido en la salud del ser humano [7]	5
Tabla 1.3 Niveles máximos permitidos de ruido para fuentes fijas [9]	6
Tabla 1.4 Características principales de Arduino Uno [12]	7
Tabla 1.5 Características principales de Raspberry Pi 1 B+	9
Tabla 1.6 Estructura general del mensaje HTTP [19]	13

Capítulo 2

Tabla 2.1 Especificaciones Micrófono USB PS3 [21] [22]	24
Tabla 2.2 Características técnicas de la placa GPS [18]	25
Tabla 2.3 Características técnicas del chip GPS NEO-GM-0-001 [18]	26
Tabla 2.4 Características técnicas de la antena del módulo GPS [18]	27
Tabla 2.5 Características técnicas del modem Huawei E 3531 [25]	29
Tabla 2.6 Librerías importadas para Python	45
Tabla 2.7 Características técnicas del servidor	57

Capítulo 3

Tabla 3.1. Comparación de mediciones previo a la calibración	74
Tabla 3.2 Comparación de mediciones luego de la calibración	75
Tabla 3.3 Lecturas tomadas por la aplicación móvil Sound Meter	89
Tabla 3.4 Mediciones en medio de los edificios Electrónica – Química y Eléctrica	91
Tabla 3.5 Mediciones en medio de los edificios Electrónica – Química y Eléctrica	92
Tabla 3.6 Mediciones en el área recreativa del edificio de Ingeniería en Sistemas	93
Tabla 3.7 Mediciones en el área recreativa del edificio de Ingeniería en Sistemas	94
Tabla 3.8 Mediciones en el hall del edificio de Medios Externos	95
Tabla 3.9 Mediciones en el hall del edificio de Medios Externos	96
Tabla 3.10 Costo referencial del sistema prototipo	103

ÍNDICE DE CÓDIGOS

Capítulo 2

Código 2.1 Instalando software necesario en la Raspberry Pi.....	37
Código 2.2 Paquetes de instalación para GUI.....	38
Código 2.3 Instalación de VNC	38
Código 2.4 Instalación de software para la comunicación	39
Código 2.5 Creación de la base de datos.....	39
Código 2.6 Liberación del puerto serial en la Raspberry Pi	41
Código 2.7 Instalación de paquetes GPS	41
Código 2.8 Creación y arranque de demonio gpsd	41
Código 2.9 Instalación del Modem 3G	43
Código 2.10 Inicialización del módulo GPS.....	46
Código 2.11 Proceso de escucha para el modulo GPS.....	47
Código 2.12 Escribir los datos de geolocalización en un archivo de texto	47
Código 2.13 Finalización del proceso de escucha del módulo GPS.....	47
Código 2.14 Creación de la clase Microphone	49
Código 2.15 Creación de stream de lecturas de ruido.....	49
Código 2.16 Proceso de llenado del stream.....	49
Código 2.17 Ingreso de parámetros para el túnel SSH	51
Código 2.18 Proceso de autenticación SSH.....	51
Código 2.19 Apertura y cierre de la conexión SSH	51
Código 2.20 Parámetros para la conexión con la base de datos	52
Código 2.21 Autenticación con la base de datos.....	52
Código 2.22 Ingreso de información a la base de datos	53
Código 2.23 Consultas a la base de datos	53
Código 2.24 Cierre de la conexión con la base de datos.....	53
Código 2.25 Inicialización y creación de objetos	55
Código 2.26 Establecimiento de la conexión y cargar del valor rms del ruido	55
Código 2.27 Impresión en pantalla de datos tomados	55
Código 2.28 Decisión si el modulo GPS no ha logrado conectarse	56
Código 2.29 Se guarda información en la base de datos	56
Código 2.30 Manejo de excepciones para finalizar el proceso	56

Código 2.31 Llamado de scripts en el documento HTML	66
Código 2.32 Botones Consultar y Capturar del documento HTML	66
Código 2.33 Script que define el estilo de la página web	67
Código 2.34 Declaración de variables	67
Código 2.35 Coordenadas de inicio para el mapa	68
Código 2.36 Se cargan las nuevas coordenadas	68
Código 2.37 Función de zoom del mapa	68
Código 2.38 Ingreso de la variable date1	69
Código 2.39 Ingreso de la variable date2	69
Código 2.40 Solicitud de datos al servidor	69
Código 2.41 Selección y autenticación con la base de datos	70
Código 2.42 Consulta a la base de datos	70
Código 2.43 Cargar los puntos solicitados en un array	70
Código 2.44 Selección y autenticación con la base de datos	71
Código 2.45 Consulta de actualización a la base de datos	71

ÍNDICE DE ECUACIONES

Capítulo 3

Ecuación 3.1. Cálculo de potencia P_{in}	73
Ecuación 3.2. Cálculo de la intensidad de ruido teórico	73

RESUMEN

El presente proyecto muestra el diseño e implementación de un sistema prototipo para la medición de ruido en el ambiente, haciendo uso de plataformas de hardware y software libre.

Inicialmente se realiza una descripción general del fenómeno del ruido, cuáles son sus características principales, su naturaleza, las fuentes que lo provocan y maneras de disminuirlo. Además se indica los niveles permisibles de ruido de acuerdo al uso del suelo según la Ordenanza Municipal N 146.

Se describen las características técnicas de las plataformas Arduino Uno y Raspberry Pi. Esto se realiza como fundamento para su posterior selección.

Se presenta el principio de funcionamiento del sensor de sonido, en el cual se indica el proceso de captación, conversión y cuantificación de la señal de ruido.

Se describe el protocolo *Hypertext Transfer Protocol* (HTTP), en el cual está basado el funcionamiento del servidor web que trabaja en el subsistema receptor.

Se realiza el diseño y la implementación del sistema prototipo, para lo cual se realiza un análisis de requerimientos, se define la arquitectura que tendrá, para lo cual se divide en un subsistema transmisor encargado de recolectar los datos y enviar a un subsistema receptor, el cual se encarga de recibir y presentar los datos recolectados.

Se realiza la calibración de los micrófonos que se utiliza para la medición de ruido. Además de las pruebas de funcionamiento general del sistema prototipo. Se realizan las mediciones de ruido en el ambiente de acuerdo al cronograma establecido en la metodología del plan de trabajo. Se presenta el análisis de resultados de las mediciones de ruido efectuadas.

Finalmente se indica el costo referencial del proyecto, en el cual no se incluyen costos de implementación y mantenimiento. Al final del proyecto se presentan las conclusiones obtenidas durante el desarrollo del presente proyecto y recomendaciones que podrían aportar en la ejecución de este y otros proyectos relacionados.

PRESENTACIÓN

El presente proyecto muestra el diseño y la implementación de un sistema prototipo inalámbrico para la medición del ruido en el ambiente y presentación de resultados de forma gráfica, desarrollado bajo las plataformas de hardware y software libre. Para cumplir con los objetivos planteados, se ha dividido al proyecto en cuatro capítulos.

En el capítulo uno se establecerá el fundamento teórico del presente proyecto, partiendo por la descripción de: el fenómeno del ruido, características técnicas de las plataformas Arduino Uno y Raspberry Pi, principio de funcionamiento del sensor de sonido, protocolo HTTP y la aplicación para medir el ruido Sound Meter que funciona en dispositivos móviles.

En el capítulo dos se realiza el diseño e implementación del sistema prototipo. Para realizar el diseño se hace un análisis de requerimientos y se define la arquitectura del sistema prototipo, la cual consta de un subsistema transmisor y un subsistema receptor. Se realiza la implementación del sistema prototipo mediante la configuración, instalación y programación de los diferentes elementos de hardware y software utilizados para este propósito.

En el capítulo tres se realizan las pruebas de funcionamiento del sistema prototipo, las mediciones de ruido en el ambiente de acuerdo al cronograma planteado en el plan de trabajo. Además se presenta un análisis de resultados de las mediciones realizadas.

En el capítulo cuatro se presentan las conclusiones obtenidas a lo largo del desarrollo del presente proyecto, además se incluyen las recomendaciones que podrían aportar en la ejecución de este y otros proyectos similares en un futuro.

CAPÍTULO I: FUNDAMENTO TEÓRICO

En este capítulo se describe el fenómeno del ruido, su naturaleza, características y tipos de ruido. Se describe también la contaminación sonora, fuentes y maneras de disminuir el ruido. Además se presentan los niveles permisibles del ruido de acuerdo al uso del suelo en el Distrito Metropolitano de la ciudad de Quito.

Se presentan características técnicas de dos SBC como son: Arduino Uno y Raspberry Pi. Además, se describe el principio de funcionamiento de los sensores de ruido y módulo GPS.

Finalmente se describen, el protocolo HTTP en el cual está basado el servidor web que se utiliza en el presente trabajo, y la aplicación móvil SOUND METER [1] que se utiliza en dispositivos móviles para medir el ruido.

1.1 RUIDO

“Proviene del latín “rugitus” que significa rugido. Es el sonido no deseado que provoca una sensación auditiva desagradable” [2], de acuerdo a su grado de intensidad y duración puede producir contaminación sonora y afectaciones en la salud.

Técnicamente se define como ruido a las sucesivas perturbaciones que no se producen a intervalos regulares y no mantienen su forma de onda constante generando ondas aperiódicas [3].

En el área de comunicaciones se considera ruido a toda señal no deseada que se mezcla con una señal deseada, provocando perturbaciones que guarda la información cuando esta se presenta dentro del ancho de banda¹.

1.1.1 NATURALEZA DEL RUIDO

El ruido tiene la misma naturaleza que el sonido, consecuentemente el ruido es una perturbación que se propaga a través de un medio elástico². La perturbación que

¹ Ancho de Banda: Intervalo de frecuencias donde la señal tiene la mayor potencia, se mide en Hertzios.

² Medio Elástico: Espacio donde se permite la circulación de cuerpos o el paso de energía.

viaja a través del medio se denomina onda³, para el sonido, si el medio es gaseoso o líquido se llama onda acústica y si es audible se llama onda sonora.

La velocidad del sonido varía de acuerdo al medio y la temperatura, por ejemplo en el aire a una temperatura ambiente puede alcanzar los 345 m/s y se estima que en el agua el sonido cuadruplica este valor [4].

En la Tabla 1.1 se presenta la velocidad del sonido en el aire de acuerdo a la temperatura.

Temperatura (C ⁰)	Velocidad del Sonido (m/s)
0	332
10	338
20	344
30	349
40	355

Tabla 1.1 Velocidad de sonido en el aire de acuerdo a la temperatura [4]

Las ondas sonoras se encuentran entre las frecuencias de los 20 Hz y 20000 Hz. Una frecuencia menor de 20 Hz se denomina infrasonidos y superior a 20000 Hz se denominan ultrasonidos. Los ultrasonidos y los infrasonidos son inaudibles para el ser humano [3].

En la naturaleza existen animales capaces de escuchar los infrasonidos y los ultrasonidos, por ejemplo el perro es capaz de escuchar los infrasonidos que provoca un terremoto y por eso pueden dar una señal de alerta previo a que se sienta el fenómeno, en tanto que, el murciélago es capaz de escuchar ultrasonidos los cuales son utilizados para su orientación y desplazamiento [3].

1.1.1.1 Espectro⁴ de sonido [3]

De acuerdo al teorema de Fourier, toda forma de onda periódica puede descomponerse en serie de formas de onda senoidales, cada una de estas ondas

³ Onda: Es una perturbación de alguna propiedad del espacio que tiene la capacidad de propagarse.

⁴ Espectro: Es la distribución de amplitudes para cada frecuencia en un fenómeno ondulatorio.

tiene una frecuencia múltiplo de la frecuencia fundamental⁵ de la onda original, a estas frecuencias se las denomina armónicos del sonido original.

Cuando la onda original ya tiene una forma de onda senoidal y a está se aplica la transformada de Fourier, resulta un solo armónico de la misma frecuencia de la onda original, en este caso la onda original se conoce como tono puro.

Cuando la onda original es aperiódica, y se aplica la transformada en tiempo discreto de Fourier se tiene espectro continuo y cuando se aplica la transformada rápida Fourier se tiene un espectro discreto.

En un espectro continuo se tienen todas las frecuencias y en un espectro discreto se tienen distintos valores de frecuencia pero ya no son múltiplos de la frecuencia fundamental.

1.1.1.2 Intensidad sonora [3]

Es el máximo pico de presión en cada período de onda. También es conocido como amplitud⁶ y se mantiene constante. Para una onda aperiódica (ruido) la intensidad puede variar y en este caso sería el promedio de la amplitud de cada periodo de onda.

1.1.2 CARACTERÍSTICAS PRINCIPALES DEL RUIDO

A continuación se presentan las características principales que posee el ruido:

- Necesita una cantidad mínima de energía para ser emitido.
- Necesita de un medio para su transmisión.
- No genera residuos en el ambiente.
- No tiene un efecto acumulativo en el medio que lo transmite.
- El efecto que produce es localizado.
- La velocidad de propagación varía de acuerdo a la temperatura y medio que lo transmite.
- Generalmente se usan los Decibelios⁷ para su medición.

⁵ Frecuencia Fundamental: Es la frecuencia más baja del espectro, de la cual se expresan como múltiplo el resto de frecuencias dominantes.

⁶ Amplitud de onda: Es la distancia que alcanza desde su línea de equilibrio hasta su pico.

⁷ Decibelio: Unidad utilizada para medir la ganancia o atenuación de un cuadripolo.

1.1.3 TIPOS DE RUIDO [5]

De acuerdo al tiempo de presentación.

- Continuo: El nivel de ruido es prácticamente constante en el tiempo. Por ejemplo el ruido que genera un ventilador.
- Variable: El nivel de ruido varía irregularmente en el tiempo. Por ejemplo el ruido en una fábrica.
- Intermitente: El nivel de ruido varía de forma escalonada y siguiendo un patrón definido. Por ejemplo el ruido que genera la alarma de un reloj de mano.
- De impacto: El nivel de ruido presenta un pico de alta intensidad y de corta duración. Por ejemplo el ruido de un golpe del martillo.

De acuerdo a la frecuencia.

- Blanco: Presenta una distribución uniforme en el espectro audible (20 Hz a 20 KHz), es decir presenta la misma intensidad para todas sus frecuencias.
- Rosa o Rosado: Presenta un nivel constante en todas las bandas de octava⁸. Generalmente se utiliza para realizar mediciones acústicas.
- Marrón: Está compuesto principalmente por frecuencias graves y medias.

1.1.4 CONTAMINACIÓN SONORA

Según la OMS⁹ (Organización Mundial de la Salud) es perjudicial para el ser humano el ruido que supere los 85 dB y con una exposición por más de 8 horas. Continuando con esta relación, el tiempo de exposición a 100 dB no debe sobrepasar los 15 minutos [6].

En la Tabla 1.2 se indican los efectos sobre la salud del ser humano según el nivel de exposición al ruido en los diferentes entornos.

⁸ Bandas de Octava: El conjunto de frecuencias que forma una banda queda definido por dos frecuencias donde la frecuencia superior es el doble de la frecuencia inferior.

⁹ OMS: Es la autoridad directiva y coordinadora de la acción sanitaria en el sistema de las Naciones Unidas.

Entorno	Nivel de Sonido (dB)	Tiempo (h)	Efectos sobre la salud
Exterior de viviendas	50-55	16	Molestia
Interior de viviendas	35	16	Problemas de comunicación
Dormitorios	30	8	Interrupción del sueño
Aulas escolares	35	No Aplica	Problemas de comunicación
Áreas industriales,	70	24	Dolor auditivo
Música en auriculares	85	1	Dolor auditivo
Actividades de ocio	100	4	Dolor auditivo

Tabla 1.2 Efectos del ruido en la salud del ser humano [7]

1.1.5 FUENTES EMISORAS DE RUIDO [8]

Una fuente emisora de ruido es aquella capaz de generar un nivel de ruido que contamina el ambiente.

Las fuentes generadoras de ruido, se pueden clasificar por: fijas, móviles y de vibración.

- Fuente Fija: Todo tipo de establecimiento fijo como: parque industrial, terminales aéreos, terrestres y marítimos, centros comerciales y de servicios, lugares de actividades lúdicas (bares, discotecas, conciertos) y otras similares.
- Fuente Móvil: Todo tipo de equipo móvil como: automotores, motociclistas, maquinaria pesada, aviones, ferrocarriles, helicópteros y semejantes.
- Fuente Vibradora: Todo tipo de equipo capaz de vibrar como: motores, turbinas y semejantes.

1.1.6 NIVELES PERMISIBLES DE RUIDO DE ACUERDO AL USO DE SUELO EN EL DISTRITO METROPOLITANO DE QUITO.

La Municipalidad del Distrito Metropolitano de Quito a través de la Dirección Metropolitana del Ambiente expide la Ordenanza 146 para la prevención y control del medio ambiente, donde se incluye el control de la contaminación sonora, en

este documento se presenta el uso de suelo con sus correspondientes horarios y niveles de ruido [9].

En la Tabla 1.3 se indican los niveles de ruido permisibles de acuerdo al uso de suelo. En la tabla se presentan los horarios para cada una de las zonas.

Tipo de Zona Según el Uso de Suelo	Nivel de Presión Sonora Equivalente: NPS eq [dB(A)]	
	De 06H00 A 20H00	De 20H00 A 06H00
Zona Equipamientos y Protección 1	45	35
Zona Residencial	50	40
Zona Residencial Múltiple (2)	55	45
Zona Industrial 1	60	50
Zona Industrial 2 (3)	65	55
Zona Industrial 3 y 4 (4)	70	60

Tabla1.3 Niveles máximos permitidos de ruido para fuentes fijas [9]

Donde:

- 1 Equipamientos de Servicios Sociales.
- 2 Incluye uso comercial y de servicios, uso agrícola residencial, y equipamiento de servicios públicos.
- 3 Incluye uso de aprovechamiento de recursos renovables.
- 4 Incluye uso de aprovechamiento de recursos no renovables.

1.1.7 MANERAS DE DISMINUIR EL RUIDO [10]

Existen principalmente 3 maneras de controlar el ruido, este control debe basarse en la normativa vigente propuesta por el organismo de control correspondiente, de acuerdo al uso de suelo. Estos mecanismos son:

- En la fuente: se puede controlar el nivel de ruido generado aislando la fuente que lo genera.
- En la propagación: se puede controlar el nivel de ruido generado instalando barreras acústicas entre la fuente y el receptor.

- En el receptor: se puede controlar el nivel de ruido generado utilizando protección para los oídos como tapones, audífonos y cabinas.

1.2 CARACTERÍSTICAS TÉCNICAS DE ARDUINO UNO Y RASBERRY PI.

1.2.1 ARDUINO UNO [11]

Es una plataforma que basa su funcionamiento en el microcontrolador¹⁰ ATmega328p, tiene 14 pines de entradas/salidas digitales y 6 entradas analógicas, las cuales permiten trabajar con señales emitidas por sensores.

En la Tabla 1.4 se detallan las características técnicas de Arduino Uno.

Microcontrolador	ATmega328p
Voltaje de Operación	5 V
Voltaje de Entrada (Recomendado)	(7 - 12) V
Voltaje de Entrada (Critico)	(6 - 20) V
Pines digitales I/O	14 (6 Pines proporcionan salidas PWM)
Pines de entrada analógica	6
Corriente DC para pines de I/O	20 mA
Corriente DC por 3,3 V por pin	50 mA
Memoria Flash	32 KB (ATmega328p) de los cuales 0,5 KB son usados para el gestor de arranque
SRAM	2 KB (ATmega328p)
EEPROM	1 KB (ATmega328p)
Frecuencia del reloj	16 MHz
Largo	68,6 mm
Ancho	53,4 mm
Peso	25 g

Tabla 1.4 Características principales de Arduino Uno [12]

¹⁰ Microcontrolador: Es un circuito programable capaz de ejecutar las ordenes guardadas en él.

Arduino es una plataforma de hardware¹¹ y software libre¹² donde la programación y configuración de esta placa se realiza bajo el entorno de desarrollo IDE Arduino.

La plataforma Arduino Uno fue desarrollada con el propósito de que principiantes y expertos pudieran realizar proyectos de electrónica de bajo costo, brindando resultados interactivos a través del uso de sensores.

En la Figura 1.1 se presenta un diagrama donde se muestran los elementos con los cuales está compuesta la placa Arduino Uno.

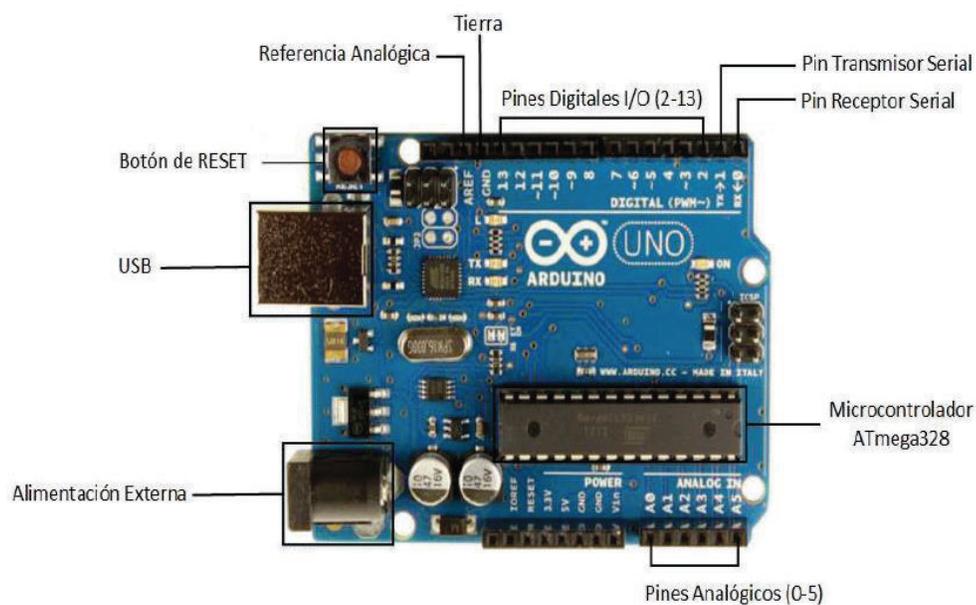


Figura 1.1 Diagrama de la placa Arduino Uno [13]

1.2.2 RASPBERRY PI [14]

Es una plataforma SBC (*Single Board Computer*), es decir es un computador de bajo costo. Tiene como objetivo estimular a los estudiantes incursionar en las ciencias de la computación, actualmente posee varios modelos, el primero en ser lanzado fue el modelo Raspberry Pi 1 modelo A, llegando actualmente al modelo Raspberry Pi 3 modelo B.

¹¹ Hardware Libre: Dispositivos que tienen sus especificaciones y diagramas de libre acceso ya sea gratuito o con algún tipo de pago.

¹² Software Libre: Software que puede ser copiado, modificado y utilizado libremente previo autorización de su autor.

El modelo Raspberry Pi 1 modelo B+ posee varios pines de entrada y salida que permiten el desarrollo de diferentes tipos de proyectos de electrónica.

En la Tabla 1.5 se presentan las características técnicas del modelo mencionado en el párrafo anterior.

Procesador	Broadcom BCM2835
RAM	512 MB LPDDR SDRAM 400 MHz
Puertos USB	4 Puertos USB 2.0
Energía	600mA hasta 1.8A a 5V
Salida de Video	HDMI 1.4 @ 1920 x 1200 pixeles
Ethernet	10/100 Mbps
Almacenamiento	MicroSD
Pines GPIO	40 I/O
Largo	8.56 cm
Ancho	5.39 cm
Sistema Operativo	GNU/Linux: Debian (Rasbian), Fedora
CPU	700 MHz
GPU	Video Core IV 250 MHz

Tabla 1.5 Características principales de Raspberry Pi 1 B+

En la Figura 1.2 se presenta un diagrama en el cual se indican los elementos de una placa Raspberry Pi 1 modelo B+.

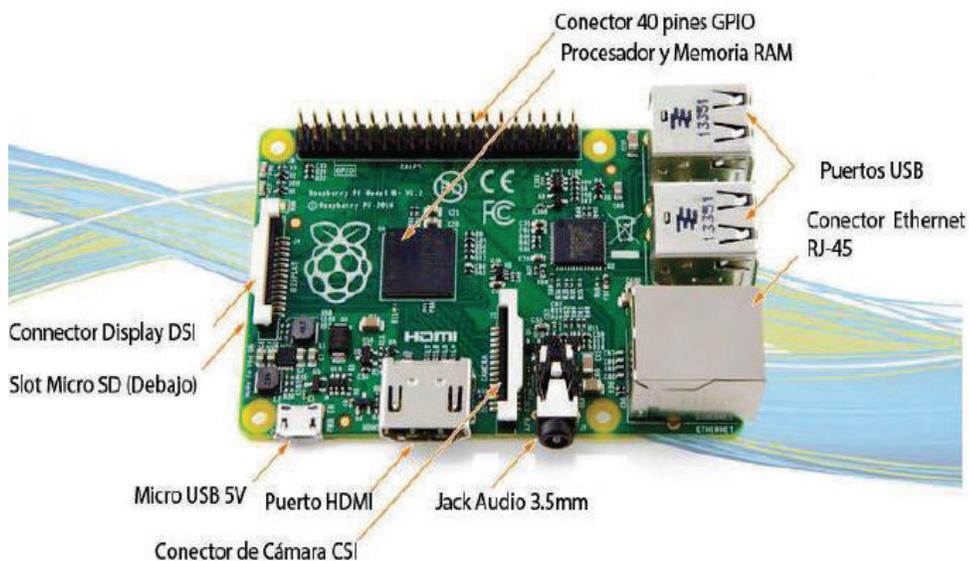


Figura 1.2 Esquema de Raspberry Pi 1 Model B+ [15]

1.3 SENSOR DE RUIDO [16]

El sensor de ruido tiene el mismo principio de funcionamiento que un micrófono, capta las ondas sonoras que se encuentran en el ambiente y las transforma en señales eléctricas, las señales son digitalizadas de acuerdo al umbral de voltaje calibrado previamente, la unidad que se utiliza para medir el ruido es el decibelio (dB).

Los elementos que constituyen un sensor de ruido son:

- Un amplificador de potencia que genere voltaje senoidal.
- Una resistencia ajustable (potenciómetro) que determina la sensibilidad del sensor.

1.4 SISTEMA DE POSICIONAMIENTO GLOBAL (GPS) [17]

Es un receptor que utiliza la tecnología GPS, permite la ubicación de cualquier objeto sobre la tierra hasta con pocos metros de margen de error y en cualquier condición atmosférica. El sistema de posicionamiento global funciona las 24 horas al día, los 7 días a la semana y los 365 días del año.

1.4.1 PRINCIPIO DE FUNCIONAMIENTO DEL SISTEMA GPS

El sistema de posicionamiento global está compuesto de 24 satélites¹³ que se encuentran rodeando la tierra a una altura de 20200 Km de altitud. Las trayectorias de los satélites están sincronizadas de tal manera que cubran toda la superficie de la tierra.

Cuando un receptor requiere conocer la ubicación, este al menos hace contacto con tres satélites de la red, recibiendo la hora del reloj y la identificación de cada uno de ellos, en base a estas señales sincroniza su reloj interno y se calcula el tiempo en que tardan en llegar las señales al equipo. De esta forma se mide la distancia al satélite mediante el método de trilateración inversa¹⁴, el cual se basa

¹³ Satélite: Pueden ser naturales o artificiales, son cuerpos que orbitan alrededor d otro cuerpo.

¹⁴ Trilateración Inversa: Es un método matemático para determinar posiciones relativas de objetos usando la geometría de triángulos.

en conocer la distancia de cada satélite con el receptor. Una vez conocidas las distancias se determina la posición relativa del receptor respecto a los satélites, Conociendo las coordenadas de los satélites mediante las señales que emiten se obtiene la posición absoluta con las coordenadas reales del punto de medición del receptor. Un receptor GPS para determinar la ubicación exacta de un objeto en el espacio necesita conocer: altitud, longitud, latitud y sincronismo.

1.4.2 MÓDULO GPS [18]

En la actualidad existe en el mercado una gran variedad de módulos GPS¹⁵, la selección del mismo dependerá del propósito para el cual se vaya a utilizar. Existen módulos para rastreo vehicular, para navegación, módulos que vienen incorporados en teléfonos celulares y tablets, etc.

A continuación se presentan las principales características y parámetros del módulo GPS que se deben considerar al momento de hacer una elección.

- Sensibilidad de seguimiento.
- Sensibilidad de rastreo.
- Número de canales.
- Numero de satélites.
- Consumo de energía.
- Exactitud en la posición.
- Precisión de la velocidad.
- Velocidad máxima.
- Protocolos.
- Tipo de antena.
- Costo.

1.5 HYPERTEXT TRANSFER PROTOCOLO [19]

“Es un protocolo que permite transmitir cualquier tipo de información con la eficiencia necesaria para ser enviada a través de Internet” [19]. HTTP esta estandarizado en su última versión HTTP 1.1 RFC 2616 [20].

¹⁵ GPS: Sistema de Posicionamiento Global

1.5.1 FUNCIONAMIENTO DEL PROTOCOLO HTTP

El usuario hace uso de un navegador web (por ejemplo Internet Explorer) y se conecta con un servidor web ubicado en cualquier parte, esta conexión es de tipo cliente – servidor. El protocolo HTTP hace uso del protocolo *Transmission Control Protocol* (TCP) para brindar fiabilidad a la conexión, por consiguiente, para cada transacción entre el navegador web y el servidor web se crea una nueva conexión y una vez terminada la transacción se cierra la conexión, esto permite que cada transacción sea tratada independientemente.

A continuación se detalla un ejemplo del proceso de conexión entre el cliente y el servidor.

El servidor web posee una página de inicio de interés, para lo cual, el cliente inicia una conexión extremo a extremo con el servidor web, el cliente envía una solicitud de tipo HTTP la cual tiene una orden concreta, una *Uniform Resource Locator* (URL) y un mensaje de tipo *Multipurpose Internet Mail Extensions* (MIME) que contiene parámetros de la solicitud e información del cliente. El servidor web recibe la solicitud, intenta llevar a cabo la acción requerida y envía una respuesta al cliente de tipo HTTP. La respuesta consta de información de estado, un código de éxito o error y un mensaje de tipo MIME que contiene información acerca del servidor web, información acerca de la respuesta requerida. Finalmente se cierra la conexión TCP.

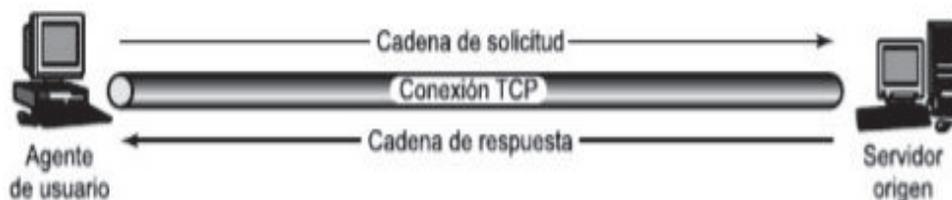


Figura 1.3 Ejemplo de funcionamiento de HTTP

1.5.2 SISTEMAS INTERMEDIARIOS PARA HTTP

El cliente y el servidor web pueden estar separados por sistemas intermediarios, cada sistema intermediario funciona como un retransmisor y cumple determinada

función dentro de la conexión. Existen tres tipos de sistemas intermediarios para HTTP.

- *Proxy*: actúa en nombre de otros clientes y presenta solicitudes de estos a un servidor, puede actuar como cliente y como servidor.
- *Gateway*: actúa en nombre de otros servidores, generalmente se da cuando el cliente no se puede conectar directamente con un servidor de origen.
- *Túnel*: No realiza operaciones sobre la solicitud HTTP, simplemente actúa como punto de retransmisión.

1.5.3 MENSAJE HTTP

HTTP tiene dos tipos de mensajes, las solicitudes enviadas por los clientes a los servidores y las respuestas enviadas a los clientes por los servidores.

En la Tabla 1.6 se presenta la estructura del mensaje HTTP tanto para las solicitudes como para las respuestas.

Línea de Solicitud
Cabecera General
Cabecera de Solicitud o Cabecera de Respuesta
Cabecera de Entidad
Cuerpo de Entidad

Tabla 1.6 Estructura general del mensaje HTTP [19]

1.5.3.1 Mensaje de solicitud HTTP

Este mensaje contiene: una línea de solicitud *request*, un método que contiene la orden de la solicitud, *request* – URL y la versión HTTP que utiliza el cliente.

Para la versión HTTP 1.1 se definen los siguientes métodos: OPTIONS, GET, HEAD, POST, PUT, PATCH, COPY, MOVE, DELETE, LINK, UNLINK, TRACE, WRAPPED, Extension Method.

1.5.3.2 Campos de la cabecera general

Los campos de la cabecera general son utilizados tanto para las solicitudes y las respuestas. Contienen información que no se aplica directamente a la entidad receptora. Este campo puede ser: Cache – Control, Connections, Date, Forwarded, Keep – Alive, MIME version, Pragma, Upgrade.

1.5.3.3 Campos de la cabecera solicitud

Estos campos funcionan como modificadores de la solicitud y proporcionan información adicional relacionada con la petición. Para HTTP 1.1 se tienen los siguientes tipos de campos: Accept, Accept - Charset, Accept – Encoding, Accept – Language, Authorization, From, Host, if – Modified – Since, Proxy – Authorization, Range, Referrer, Unless, User – Agent.

1.5.3.4 Mensaje de respuesta HTTP

Un mensaje de respuesta HTTP tiene una línea de estado seguida de una o más cabeceras generales, de respuesta y de entidad, y finalmente por un cuerpo opcional de entidad.

1.5.3.5 Códigos de estado

En el campo línea de estado para una respuesta tiene el siguiente formato: número de versión HTTP que utiliza el emisor, código de estado que indica la respuesta a la solicitud y consta de 3 dígitos y una *Reason – Phrase* que contiene una explicación del código de estado.

1.5.3.6 Campos de la cabecera respuesta

Proporciona información adicional que no ha sido especificada en el campo status – line. Para la versión HTTP 1.1 se tiene sol siguientes campos: Location, Proxy – Authenticate, Public, Retry – After, Server, WWW – Authenticate.

1.5.3.7 Campos de la cabecera de entidad

Proporcionan información adicional acerca del cuerpo de la entidad, si no está presente el cuerpo de entidad presenta sobre el recurso identificado por la solicitud.

1.5.3.8 Cuerpo de entidad

Contiene una secuencia arbitraria de bytes. Debido a que HTTP puede transferir cualquier tipo de información (voz, datos, video, etc); este campo puede contener información o estar vacío. En caso de poseer información los bytes son interpretados de acuerdo a los campos de la cabecera entidad, Content – Encoding, Content – Type y Transfer – Encoding. Estos campos definen un modelo de codificación en tres capas.

Se debe indicar que los campos de la cabecera de entidad más el campo del cuerpo de entidad se conoce como entidades. Las entidades pueden estar presentes en los mensajes de solicitud o de respuesta. Una entidad puede contener un recurso de datos o información proporcionada con otra solicitud o respuesta.

1.6 SOUND METER [1]

Es una aplicación para dispositivos móviles que usan el sistema Android¹⁶ y sirve para medir la intensidad de ruido en el ambiente. Está aplicación usa el micrófono del dispositivo móvil para captar las ondas sonoras y presentar el nivel de ruido en dBs. Se debe indicar que el micrófono del dispositivo móvil esta calibrado para la voz humana, que está dentro del rango de (300 - 3400) Hz, por lo tanto la aplicación podrá captar los niveles de ruido que se encuentren dentro de este rango, también se debe mencionar que de acuerdo al fabricante del dispositivo móvil el rango de medición puede variar. Por lo general en la mayoría de dispositivos móviles están limitados para captar un máximo de 90 dB; sin embargo dependiendo de las características de los dispositivos celulares, estos pueden llegar medir 110 dB.

A continuación se presentan las características que tiene esta aplicación:

- Muestra los decibelios por el manómetro¹⁷.
- Muestra una referencia del ruido al instante.
- Presenta una gráfica indicando el comportamiento del ruido en el tiempo.
- Se puede calibrar para cada dispositivo móvil.
- Pantalla MIN / AVG / MAX en decibelios.

¹⁶ Sistema Operativo Android: Sistema operativo diseñado para teléfonos inteligentes y tablets, está basado en el núcleo del sistema operativo Linux.

¹⁷ Manómetro: Instrumento para medir la presión de gases o fluidos en recipientes cerrados.

En la Figura 1.4 se presenta la interfaz gráfica que posee la aplicación Sound Meter.



Figura 1.4 Sound Meter versión 3.1.4 [1]

CAPÍTULO 2: DESARROLLO DEL PROTOTIPO

En este capítulo se describe el análisis de requerimientos para el desarrollo del prototipo, se presentan los elementos de hardware y software que se emplean en la implementación del prototipo. Adicionalmente se describe el diseño y la implementación del sistema prototipo.

En el diseño se indican los elementos y el esquema que tendrá el sistema prototipo, para cumplir con los requerimientos planteados se divide en el subsistema transmisor y el subsistema receptor.

En la implementación del sistema prototipo se describe el proceso de configuración de la SBC, el envío de datos al servidor web y el desarrollo de la aplicación web, la cual que funcionará en el servidor para la recepción de datos y presentación de resultados.

2.1 REQUERIMIENTOS DEL SISTEMA PROTOTIPO

Los requerimientos que debe cumplir el sistema prototipo se obtienen a partir de los objetivos planteados en el plan de trabajo y a las necesidades de realizar un control de la contaminación sonora de acuerdo a lo planteado en la Ordenanza Municipal N 146.

A continuación se detallan las tareas que debe cumplir el sistema prototipo.

- a) Tomar niveles de ruido presente en el ambiente a través del sensor de sonido.
- b) Tomar la ubicación del lugar donde se está realizando la medición del ruido a través del módulo GPS.
- c) Organizar y estructurar los datos en la SBC para su posterior transmisión al servidor.
- d) Establecer la comunicación entre la SBC y el servidor web.
- e) Enviar datos desde la SBC hacia el servidor web.
- f) Recibir los datos en el servidor.
- g) Almacenar los datos en una base de datos relacional.
- h) Presentar los datos almacenados, a través de la aplicación Google Maps de forma gráfica.

En la Figura 2.1 se presenta un diagrama donde se resume las tareas anteriormente mencionadas.

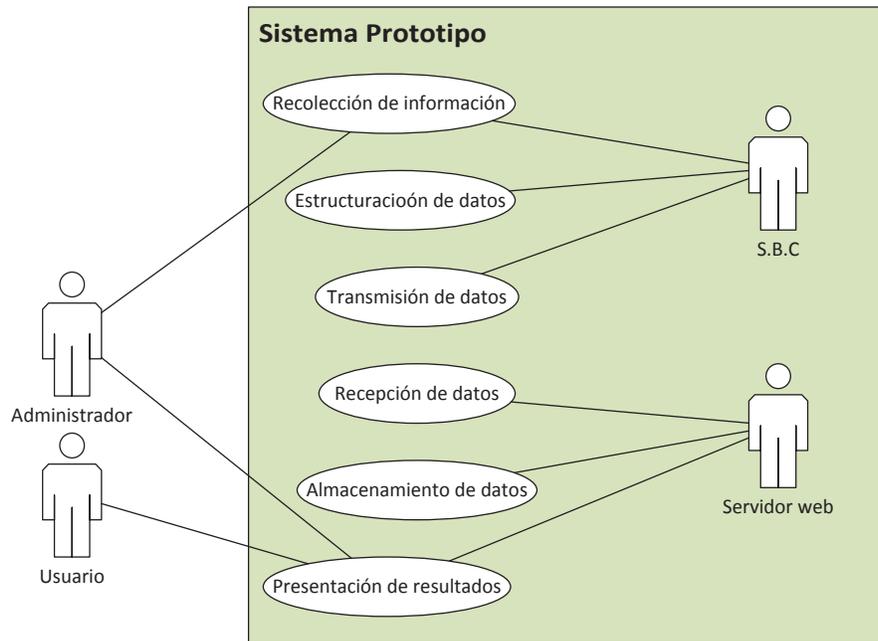


Figura 2.1 Diagrama de casos de uso del sistema prototipo

2.2 ARQUITECTURA DEL SISTEMA PROTOTIPO

El funcionamiento del sistema prototipo está dividido en dos partes principales: subsistema transmisor y subsistema receptor.

En la Figura 2.2 se presenta el esquema de los diferentes elementos que deben intervenir para cumplir con los requerimientos planteados en el apartado 2.1.

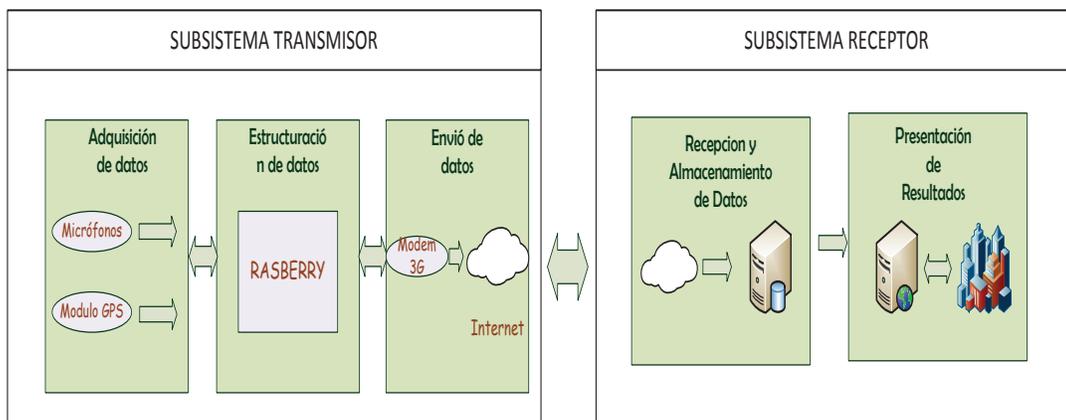


Figura 2.2 Arquitectura del Sistema Prototipo

2.3 DISEÑO DEL SUBSISTEMA TRANSMISOR

El subsistema transmisor debe cumplir con las tareas a, b, c, d, e del subcapítulo 2.1, por lo que se ha dividido en tres bloques.

- Bloque de adquisición de datos.
- Bloque de estructuración de datos.
- Envío de datos.

En la Figura 2.3 se presenta un esquema donde se muestra como están relacionados los diferentes elementos que intervienen en el subsistema transmisor.

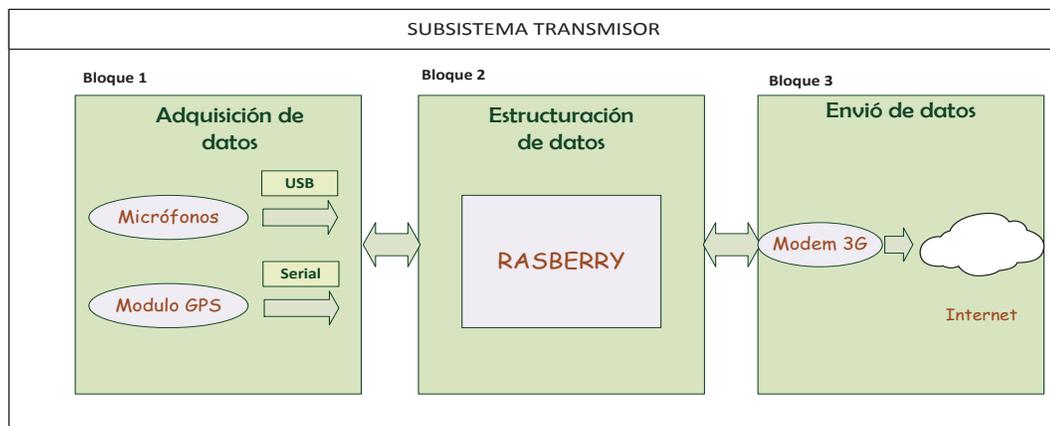


Figura 2.3 Esquema del Subsistema Transmisor

2.3.1 BLOQUE DE ADQUISICIÓN DE DATOS

Este bloque será el encargado de cumplir con las tareas a y b del subcapítulo 2.1, para ello se tomarán los valores de ruido presente en el ambiente a través de los micrófonos que estarán conectados directamente a los puertos USB de la Raspberry. También, este bloque se encargará de tomar las lecturas que realice el modulo GPS, donde se indique la longitud y latitud del lugar en donde se están realizando las mediciones de ruido.

2.3.2 BLOQUE ESTRUCTURACIÓN DE DATOS

Este bloque se encargará de cumplir con la tarea c del subcapítulo 2.1, estará compuesto por la SBC Raspberry PI cuyas características técnicas serán detalladas en las siguientes secciones.

La función del bloque de adquisición será tomar las lecturas realizadas por los micrófonos y el modulo GPS, realizar la estructuración de datos para enviarlos al subsistema receptor a través del modem 3G.

En la Figura 2.4 se presenta la estructura que tendrán los datos enviados al subsistema receptor.

ID	Date	Latitud	Longitud	RMS
-----------	-------------	----------------	-----------------	------------

Figura 2.4 Estructura de los datos tomados por el Subsistema Transmisor

- **ID:** Es un identificador único para cada lectura de posición y ruido que tome el modulo GPS y los micrófonos. Este identificador tiene el propósito de permitir el almacenamiento de información en la base de datos en el subsistema receptor
- **Date:** El subsistema transmisor toma la fecha y hora que tiene la SBC, por lo cual cada vez que el subsistema transmisor vaya a realizar mediciones estas deberán estar actualizadas. La fecha y hora serán enviadas con cada lectura de posición y ruido, esto se hace con la finalidad de realizar un posterior análisis de los resultados obtenidos.
- **Latitud:** Es la lectura que toma el modulo GPS para determinar la ubicación geográfica del subsistema transmisor en donde se están realizando las mediciones de ruido, se expresa en formato decimal.
- **Longitud:** Es la lectura que toma el modulo GPS para determinar la ubicación geográfica del subsistema transmisor en donde se están realizando las mediciones de ruido, se expresa en formato decimal.
- **RMS:** Es el valor promedio de las lecturas tomadas por los micrófonos, se expresa en dBs.

2.3.3 BLOQUE DE ENVIÓ DE DATOS

Este bloque se encargará de cumplir con las tareas d y f del subcapítulo 2.1. Estará compuesto por el modem 3G conectado a la SBC, el cual tiene un plan de Internet que permitirá el envío de los datos hacia el subsistema receptor. El modem debe pertenecer a una operadora de telefonía celular que ofrezca el servicio de conexión a Internet, tenga los mejores precios y cobertura.

2.4 DISEÑO DEL SUBSISTEMA RECEPTOR

El subsistema receptor deberá cumplir con las tareas f, g y h del subcapítulo 2.1, por lo cual se ha dividido en dos bloques.

- Bloque de recepción y almacenamiento de datos.
- Bloque de presentación de resultados.

En la Figura 2.5 se presenta un esquema donde se muestra como están relacionados los diferentes elementos que intervienen en el subsistema receptor.

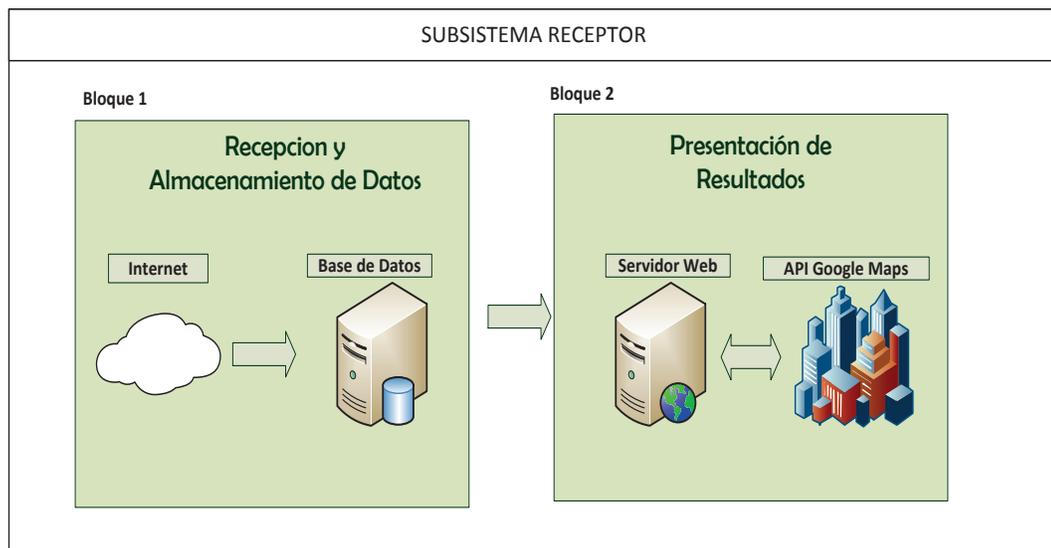


Figura 2.5 Esquema del Subsistema Receptor

Los dos bloques que se describen en la figura 2.5 funcionan en un computador con conexión a Internet.

En la Figura 2.6 se presenta un esquema que indica el modo de conexión a Internet.



Figura 2.6 Esquema de conexión a Internet del subsistema receptor

2.4.1 BLOQUE DE RECEPCIÓN Y ALMACENAMIENTO DE DATOS

Este bloque se encargará de cumplir con las tareas f y g del subcapítulo 2.1, el subsistema receptor deberá recibir los datos enviados por el bloque envió de datos del subsistema transmisor a través de Internet, para lo cual el subsistema receptor deberá tener conexión a Internet.

La información será almacenada en una base de datos que garantice la integridad y disponibilidad de los mismos. Para el almacenamiento de datos se utilizará el motor de base de datos MySQL.

En la Figura 2.7 se presenta la propuesta que tendrá la base de datos y los campos mínimos con los que deberán contar.

Sonometro
Id Usuario
Fecha y Hora
Latitud
Longitud
RMS

Figura 2.7 Diseño de la Base de Datos

2.4.2 BLOQUE DE PRESENTACIÓN DE RESULTADOS

Este bloque se encargará de cumplir con la tarea f señalada en el apartado 2.1, consiste en presentar los resultados de las mediciones realizadas por el bloque de adquisición de datos.

Para presentar los resultados de forma gráfica se hace uso del API de Google Maps, donde un explorador web se descarga el mapa en el cual se marcan las coordenadas tomadas por el modulo GPS. El lugar donde se realicen las mediciones será marcado en el mapa mediante un mapa de calor, los colores que se utilicen indicarán el grado de intensidad del ruido.

Para presentar los resultados de forma textual se usará una tabla con los siguientes campos: Identificador, Fecha y Hora, Latitud, Longitud, RMS. La tabla se mostrará junto con el mapa de Google Maps y el mapa de calor.

Para facilitar la visualización de los resultados al usuario se presentarán tanto de forma gráfica como textual y será mostrado al mismo tiempo en la página del explorador web.

2.5 SUBSISTEMA TRANSMISOR

El subsistema transmisor está compuesto por los elementos de hardware y software que permitan cumplir con lo descrito en el subcapítulo 2.2 y subcapítulo 2.3.

En la Figura 2.8 se presenta un diagrama donde se indica los elementos de hardware necesarios para cumplir con lo descrito en el subcapítulo 2.3.



Figura 2.8 Esquema del subsistema transmisor

A continuación se detallan los elementos de hardware y software que se utilizarán en la implementación del subsistema transmisor.

2.5.1 SELECCIÓN DE LOS ELEMENTOS DE HARDWARE PARA EL BLOQUE DE ADQUISICIÓN DE DATOS

Para el cumplir con las tareas: a, b, c, d de la lista de requerimientos que debe realizar el subsistema transmisor son necesarios los siguientes dispositivos.

La selección del hardware se realizó bajo los siguientes criterios:

- Plataforma de libre distribución.
- Sistema operativo de tipo *Open Source*.
- Debe soportar GNU/Linux.
- Bajo Costo.

2.5.1.1 Sensor de ruido

Para la obtención de datos de ruido en el ambiente se utiliza el micrófono USB PS3. Este micrófono cuenta con una entrada USB que permite la conexión directa con el la placa SBC y utiliza como fuente de energía la proporcionada por el puerto USB de la placa, además posee un bajo costo en el mercado, estas son las razones por las cuales fue seleccionado este modelo.

Para tener una lectura más acertada del ruido presente en el ambiente se utiliza un arreglo de dos micrófonos, estos están direccionados de tal manera que tomen las señales presentes en el ambiente provenientes de diferentes direcciones. Los dos micrófonos serán de las mismas características [21].

En la Figura 2.9 se muestra el micrófono que se utilizará para medir el sonido.



Figura 2.9 Micrófono USB PS3 [21]

En la Tabla 2.1 se indican las características técnicas del micrófono.

Micrófono USB PS3	Especificaciones
Interfaz	USB 2.0
Sensibilidad	-67 dBV/pBar, -47dBV/Pascal, +/- 4dB
Respuesta de frecuencia	(100 – 16) KHz
SNR	67 dB

Tabla 2.1 Especificaciones Micrófono USB PS3 [21] [22]

2.5.1.2 Módulo GPS

Para determinar la ubicación del sitio donde se realiza la medición del ruido se utiliza el módulo GPS que consta de una placa GY – GPSGMV2 y con un chip NEO – GM -0-001.

Se decidió utilizar este módulo GPS debido a que presenta las siguientes ventajas: fácil acoplamiento con la placa SBC, su tamaño permite portabilidad, tiene un bajo costo y fácil adquisición en el mercado, finalmente existe abundante información acerca de este dispositivo electrónico [18].

En la Figura 2.10 se muestra el dispositivo electrónico módulo GPS.



Figura 2.10 Módulo GPS

En la Tabla 2.2 se presentan las características técnicas de la placa GY – GPSGMV2 que se emplea en el presente trabajo.

Placa GY – GPSGMV2	Especificaciones
Energía	(3 - 5) V
Velocidad de transmisión	9600 bps
Tamaño	25 mm * 35 mm
Antena Cerámica	25 mm * 25 mm
Memoria	EEPROM ¹⁸
Protocolo de Comunicación	UART ¹⁹

Tabla 2.2 Características técnicas de la placa GPS [18]

¹⁸ EEPROM: Electrically Erasable Programmable Read-Only Memory.

¹⁹ UART: Transmisor-Receptor Asíncrono Universal.

En la Figura 2.11 se muestra el diagrama del circuito electrónico de la placa GY – GPSGMV2.

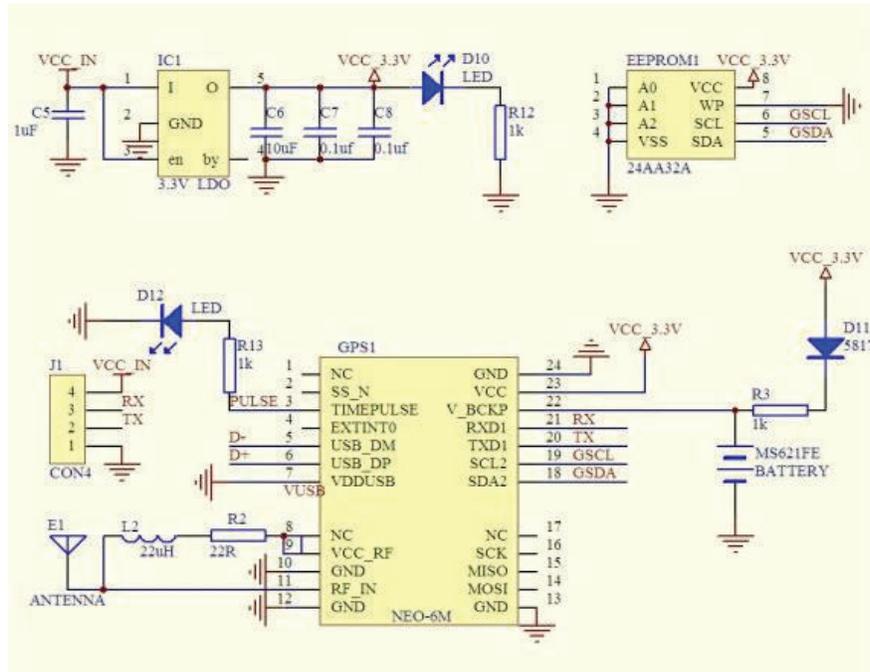


Figura 2.11 Esquema de la placa GY – GPSGMV2 [18]

En la Tabla 2.3 se presentan las principales características técnicas del chip GPS NEO – GM -0-001 que se emplea en el presente trabajo.

Chip NEO – GM -0-001	Especificaciones
Tipo de receptor	50 Canales GPS L1 Frecuencia, C/A Code SBAS ²⁰ : WAAS ²¹ , EGNOS ²² , MSAS ²³ .
Tiempo de arranque	Cold Start: 27 s. Warm Start: 27 s. Hot Start: 1 s.
Sensibilidad	Rastreo y Navegación: -161 dBm. Readquisición: -160 dBm. Cold start: -147 dBm Hot start: -156 dBm.
Maxima velocidad de actualización	1 Hz.
Rango de frecuencia configurable	0.25 Hz – 10 MHz.

Tabla 2.3 Características técnicas del chip GPS NEO-GM-0-001 [18]

²⁰ SBAS: Satellite Based Augmentation System.

²¹ WAAS: Wide Area Augmentation System.

²² EGNOS: European Geostationary Navigation Overlay Service.

²³ MSAS: Multi-funtional Satellite Augmentation System.

En la Tabla 2.4 se muestran las características que tiene la antena del módulo GPS.

Antena	Especificaciones
Tipo	Antena pasiva y activa
Ganancia mínima	15 dB
Ganancia máxima	50 dB
Ruido máximo	1.5 dB

Tabla 2.4 Características técnicas de la antena del módulo GPS [18]

2.5.2 SELECCIÓN DE LOS ELEMENTOS DE HARDWARE PARA EL BLOQUE DE ESTRUCTURACIÓN DE DATOS

La selección del hardware se realizó bajo los siguientes criterios:

- Plataforma de libre distribución.
- Sistema operativo de tipo *Open Source*.
- Bajo Costo.

2.5.2.1 Single Board Computer [14]

La estructuración de datos que debe realizar el subsistema transmisor utiliza la *Single Board Computer* Raspberry Pi modelo B + versión 1.2. La placa permite trabajar con hardware y software libre, también presenta suficientes puertos USB que facilitan la conexión de los micrófonos y modem que se utilizan en el presente trabajo, tiene en la distribución de pines la posibilidad de conexión con el protocolo UART que usa el modulo GPS para la comunicación, además la placa tiene gran difusión en el desarrollo de proyectos de electrónica y es de bajo costo.

En la Figura 2.12 se presenta la *Single Board Computer* Raspberry Pi B + 1.2.

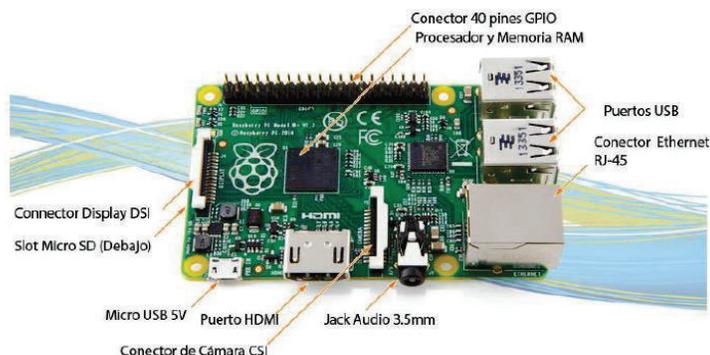


Figura 2.12 Esquema de la tarjeta Raspberry Pi 1.2 [14]

La placa para interactuar con el medio exterior tiene la distribución de 40 pines de entrada y salida.

En la Figura 2.13 se presentan el nombre y función de cada uno de ellos.

Raspberry Pi 2 Model B (J8 Header)						
GPIO#	NAME			NAME	GPIO#	
	3.3 VDC Power	1			2	5.0 VDC Power
8	GPIO 8 SDA1 (I2C)	3			4	5.0 VDC Power
9	GPIO 9 SCL1 (I2C)	5			6	Ground
7	GPIO 7 GPCLK0	7			8	GPIO 15 TxD (UART) 15
	Ground	9			10	GPIO 16 RxD (UART) 16
0	GPIO 0	11			12	GPIO 1 PCM_CLK/PWM0 1
2	GPIO 2	13			14	Ground
3	GPIO 3	15			16	GPIO 4 4
	3.3 VDC Power	17			18	GPIO 5 5
12	GPIO 12 MOSI (SPI)	19			20	Ground
13	GPIO 13 MISO (SPI)	21			22	GPIO 6 6
14	GPIO 14 SCLK (SPI)	23			24	GPIO 10 CE0 (SPI) 10
	Ground	25			26	GPIO 11 CE1 (SPI) 11
30	SDA0 (I2C ID EEPROM)	27			28	SCL0 (I2C ID EEPROM) 31
21	GPIO 21 GPCLK1	29			30	Ground
22	GPIO 22 GPCLK2	31			32	GPIO 26 PWM0 26
23	GPIO 23 PWM1	33			34	Ground
24	GPIO 24 PCM_FS/PWM1	35			36	GPIO 27 27
25	GPIO 25	37			38	GPIO 28 PCM_DIN 28
	Ground	39			40	GPIO 29 PCM_DOUT 29

Figura 2.13 Distribución de pines de la placa Raspberry Pi B + [23]

2.5.3 SELECCIÓN DE LOS ELEMENTOS DE HARDWARE PARA EL BLOQUE DE ENVIO DE DATOS

La selección del hardware se realizó bajo los siguientes criterios:

- Plataforma de libre distribución.
- Sistema operativo de tipo *Open Source*.
- Bajo Costo.

2.5.3.1 Modem 3G [24]

El modo de envío de datos entre el subsistema transmisor y el subsistema receptor debe proveer una conexión a Internet en cualquier lugar y en todo momento, puesto que el prototipo tiene como propósito desplazarse a lo largo de la ciudad de Quito para realizar las mediciones de ruido en el ambiente.

Para cumplir con este requerimiento, el “bloque envío de datos” del subsistema transmisor utiliza el modem HUAWEI 3G modelo E 3531.

Este dispositivo tiene la capacidad de proveer el servicio de internet al “bloque de envío de datos”, de esta manera la placa SBC envía la trama con los datos recolectados a través de internet al bloque de recepción y almacenamiento de datos del subsistema receptor.

El modem 3G que se utiliza en el presente trabajo pertenece a la operadora de telefonía celular CONECEL²⁴ comercialmente conocida como Claro.

Se decidió por esta operadora principalmente por el área de cobertura que maneja, ya que de esta manera se garantiza que el modem 3G podrá ser usado en gran parte del país de ser necesario.

En la Tabla 2.5 se presentan las características técnicas que tiene el modem 3G Huawei E3531.

Huawei E 3531	Especificaciones
Interfaz	USB 2.0.
Sistema de comunicación	WCDMA ²⁵ /HSPA ²⁶⁺ 850/900/1900/2100 MHz. GSM ²⁷ /GPRS ²⁸ /EDGE ²⁹ 850/900/1800/1900 MHz.
Energía	5 V, 500 mA.
LED	Indica el estado del modem.
Dimensiones	84mm*27mm*10.5mm

Tabla 2.5 Características técnicas del modem Huawei E 3531 [25]

²⁴ CONECEL: Consorcio Ecuatoriano de Telecomunicaciones.

²⁵ WCDMA: Wideband Code Division Multiple Access.

²⁶ HSPA: High Speed Packet Access.

²⁷ GSM: Global System for Mobile.

²⁸ GPRS: General Packet Radio Service.

²⁹ EDGE: Enhanced Data Rates for GSM Evolution.

2.5.4 SELECCIÓN DE LOS ELEMENTOS DE SOFTWARE PARA EL SUBSISTEMA TRANSMISOR

Para cumplir con lo descrito en los subcapítulos 2.1 y 2.4 es necesario indicar los criterios de selección para las herramientas de software.

- 1.- *Open Source* y licenciamiento gratuito.
- 2.- Soportar el paradigma orientado a objetos.
- 3.- Poseer abundante información de consulta.

Para el subsistema transmisor el software se compone de: el sistema operativo³⁰ que funciona en los Raspberry Pi B+ versión 1.2, el lenguaje de programación compatible con la SBC y controladores del modem 3G para funcionar en Linux.

Debido a la versatilidad que posee la placa Raspberry Pi actualmente existen varios sistemas operativos que pueden funcionar en la placa, los más importantes son: Raspbian, Kano, Pipa, PiBang, Risc. De los sistemas operativos mencionados el más utilizado y de mejor desempeño en la placa Raspberry Pi es Raspbian.

2.5.4.1 Raspbian [26]

Es un sistema operativo de libre distribución de GNU/Linux basado en el sistema operativo Debían Wheezy creado para la placa Raspberry Pi. Al ser una distribución de GNU/Linux soporta todo tipo de software de código abierto.

El sistema operativo Raspbian viene con un conjunto de herramientas orientadas al desarrollo de programas que permitan interactuar a la placa con el medio, una de estas herramientas es Python.

Raspbian es la versión optimizada de Debían Wheezy, esta funciona en la placa Raspberry Pi, actualmente viene con 35000 paquetes recompilados para facilitar la instalación del paquete se llegue a necesitar en la Raspberry Pi.

Se debe indicar que Raspbian no pertenece a la Fundación Raspberry Pi; Raspbian fue creado por un grupo de desarrolladores aficionados de a la placa Raspberry Pi

³⁰ Sistema Operativo: Conjunto de órdenes, comandos y programas que controlan los procesos de una computadora.

y del proyecto Debían. A continuación se presentan las características principales de Raspbian, sistema operativo usado por excelencia para la placa Raspberry Pi [27].

- Entorno de escritorio LXDE y MIDORI como navegador web por defecto.
- Herramientas de desarrollo como: IDE Python y Scratch.
- Programas instalados: Libre Office, Calculadora, Visor de imágenes, PDF, editor de texto, etc.

2.5.4.2 Python [28]

Es un lenguaje de programación interpretado³¹, tiene una sintaxis favorable al programador que hace que sea fácilmente legible. Es un lenguaje multiparadigma, soporta programación orientada a objetos, programación imperativa y programación funcional.

Esta administrado por la Fundación Software Python y tiene una licencia de código abierto y es compatible con la licencia publica general GNU, es decir compatible con las distribuciones de Linux.

Fue creado a finales de los años ochenta y toma el nombre de Python por la afición de su creador a Monty Python [29].

Se utiliza este lenguaje principalmente porque soporta programación orientada a objetos. Además porque Python viene por defecto en el paquete de herramientas de desarrollo del sistema operativo Raspbian.

Python es uno de los lenguajes más utilizados para el desarrollo de proyectos de electrónica con la placa Raspberry Pi por lo cual hay abundante material de consulta que puede ser útil para el desarrollo del presente trabajo [30].

2.5.5 CONFIGURACIÓN DE LA SINGLE BOARD COMPUTER

Para cumplir con las tareas que debe realizar el subsistema transmisor se debe configurar la Raspberry Pi B+ versión dos, a continuación se indican los pasos que se realizaron para la consecución de dicho proposito:

³¹ Lenguaje de programación interpretado: Son lenguajes que no requieren de un código a ser compilado, ya que los programas consisten en *scripts* que son interpretados directamente.

1. Instalación del sistema operativo Raspbian en la SBC.
2. Instalación de paquetes, librerías y herramientas necesarias para el correcto funcionamiento de la SBC en el presente trabajo.
3. Instalación de paquetes y herramientas para el canal de comunicaciones entre el subsistema transmisor y el subsistema receptor.
4. Instalación de herramientas para la interacción con la base de datos.

2.5.5.1 Instalación del sistema operativo Raspbian

Para el proceso de instalación del sistema operativo Raspbian es necesario indicar, que existen varias maneras en la que puede funcionar el sistema operativo en la placa Raspberry Pi, las más comunes son: en una tarjeta micro SD y en un dispositivo de almacenamiento externo que se puede conectar a través de los puertos USB de la placa.

El sistema operativo se instala mediante una tarjeta micro SD que posteriormente se coloca en el puerto correspondiente de la placa. Se utiliza esta forma de instalación principalmente porque los puertos USB que tiene la placa se utilizan para conectar los micrófonos y el modem, además la tarjeta micro SB ocupa un reducido espacio y esto ayuda a una mayor portabilidad del prototipo. A continuación se indica el proceso de instalación del sistema operativo Raspbian.

1. Se descarga el sistema operativo Raspbian desde el sitio oficial: <https://www.raspberrypi.org/downloads/>. Se elige Raspbian y dar click en descargar el archivo tipo zip. La Figura 2.14 muestra la página web de la cual se descarga Raspbian.

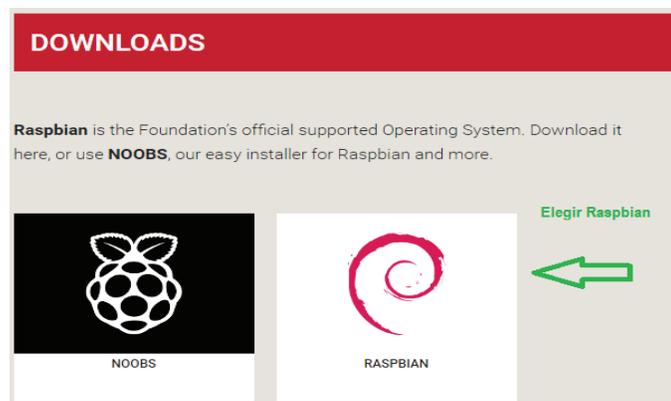


Figura 2.14 Sistema Operativo Raspbian

2. Se descomprime el archivo descargado y se obtiene un archivo de tipo .img.

En la Figura 2.15 se muestra el tipo de archivo que debe tener la descarga del sistema operativo.

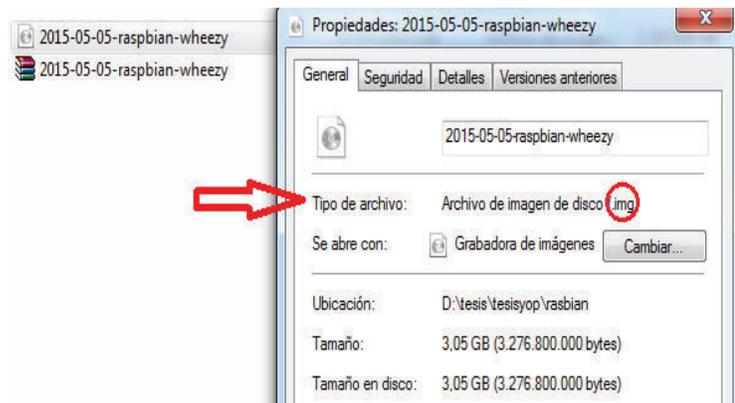


Figura 2.15 Archivo Raspbian de tipo img

3. Se guarda el archivo de tipo .img en la tarjeta micro SD, para lo cual hay que considerar que el archivo tiene un tamaño de 3.05 GB por lo tanto la tarjeta debe tener una capacidad de almacenamiento de al menos 4 GB.

Se recomienda que tenga una capacidad de almacenamiento superior para que el sistema operativo pueda funcionar ampliamente.

Se guarda el archivo, se verifica que la tarjeta micro SD sea boteable y luego se utiliza la herramienta Win32DiskImager para escribir el archivo en la tarjeta micro SD.

En la Figura 2.16 se muestra como se carga el sistema Raspbian en la tarjeta micro SD.

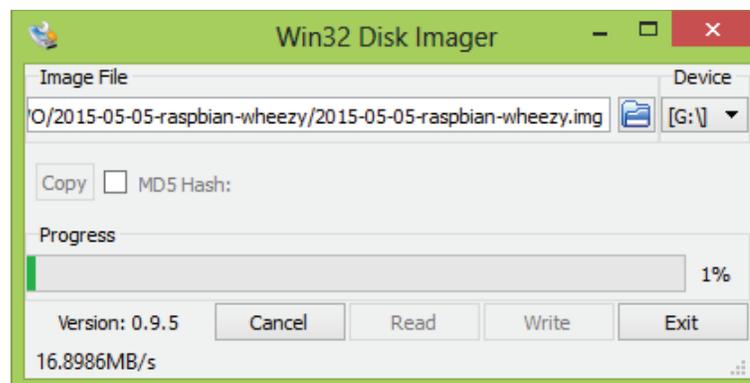


Figura 2.16 Escribiendo el sistema operativo Raspbian en memoria micro SD

4. La Figura 2.17 muestra cómo se inserta la memoria micro SD en la ranura que se encuentra en la parte posterior de la placa Raspberry Pi.

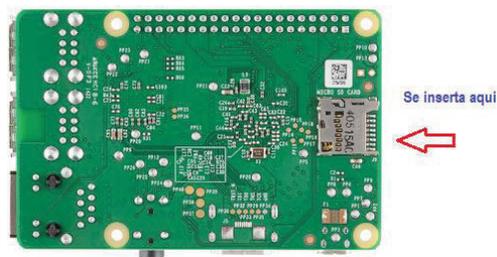


Figura 2.17 Ranura donde se inserta la memoria micro SD

5. En este punto es necesario ingresar a la placa Raspberry Pi, para lo cual existen varias maneras.

Para ingresar a la placa Raspberry Pi se conecta un cable de red desde la placa al modem Huawei E 3531 que sea parte de la red, el cual se encuentra también conectado al computador, esto se realiza para tener en la misma red. Previamente han sido instaladas en el computador las herramientas Advanced IP³² Scanner y Moba Xterm³³, Advanced IP Scanner muestra todas las direcciones IP que se encuentran en la red interna al cual está conectado el computador. En la Figura 2.18 se muestran los equipos que se encuentran en la red.

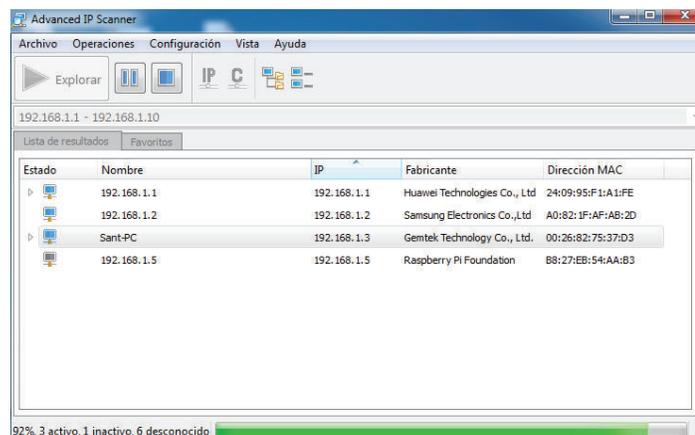


Figura 2.18 Dirección IP que toma la placa Raspberry Pi

³² Advanced Ip Scanner: Muestra las direcciones IP de los dispositivos que se encuentran conectados en la red interna.

³³ Moba Xterm: Permite la conexión remota entre un terminal y otro mediante varias formas: SSH, RDP, VNC, etc.

Con esto se consigue saber que la dirección IP que fue asignada para la placa SBC y su la dirección MAC que tiene.

A continuación se ejecuta la herramienta Moba Xterm que permite la conexión entre el computador y la placa Raspberry Pi de varias formas. Como aún no está configurado el sistema operativo en la Raspberry Pi para que se muestre la interfaz gráfica del escritorio remoto la conexión se realiza mediante SSH.

En la Figura 2.19 se muestra como se establece la sesión con la placa Raspberry Pi.

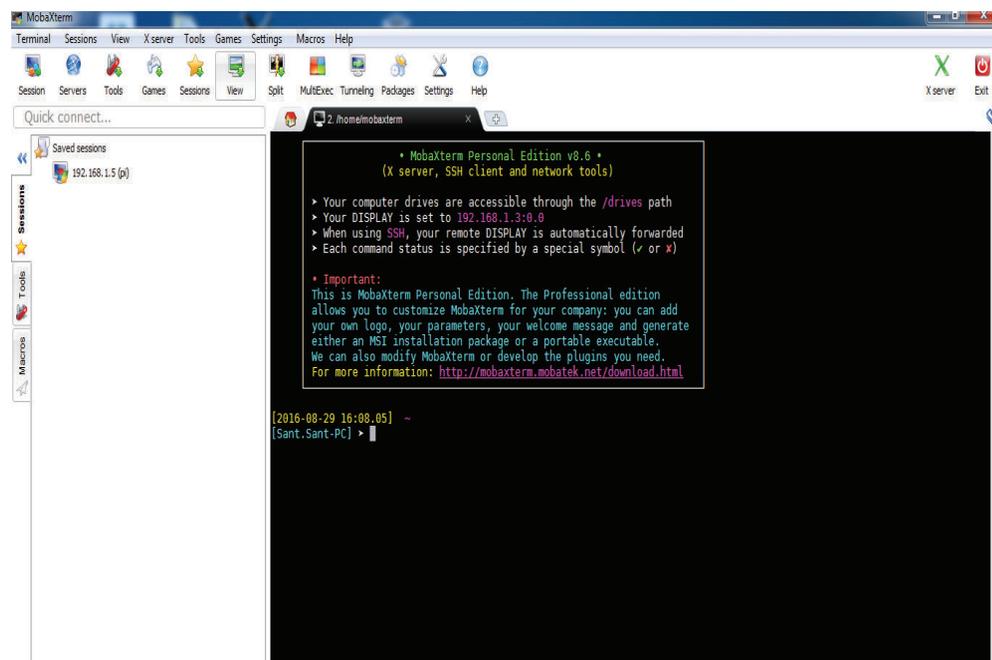
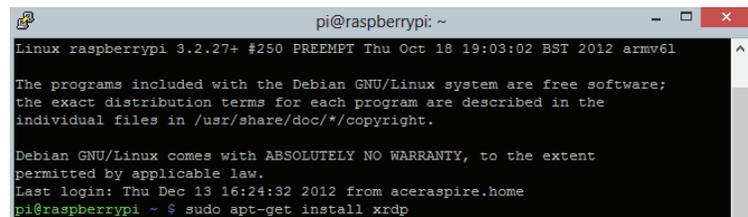


Figura 2.19 Moba Xtrem

Se ingresa la dirección IP de la Raspberry Pi, a continuación se ingresa el nombre del usuario y la contraseña de la placa que por defecto es "pi" y "raspberrypi" respectivamente.

6. La sesión SSH que se ha creado permite ejecutar comandos en el terminal de la Raspberry Pi que permiten la configuración de la misma.
Sudo apt-get install xrdp: Sirve para instalar el servidor de escritorio remoto en la Raspberry Pi, se presenta una pregunta de confirmación y se tiene la posibilidad de conexión.

En la Figura 2.20 se muestra el comando que se ejecuta en el terminal de la Raspberry Pi para la instalación de la interfaz del escritorio.



```

pi@raspberrypi: ~
Linux raspberrypi 3.2.27+ #250 PREEMPT Thu Oct 18 19:03:02 BST 2012 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec 13 16:24:32 2012 from aceraspire.home
pi@raspberrypi ~ $ sudo apt-get install xrdp

```

Figura 2.20 Ejecución del comando `sudo apt-get install wrdp`

Una vez instalado este comando se procede a la conexión remota con la placa, ejecutando escritorio remoto de Windows, donde se ingresa la dirección IP de la placa Raspberry Pi. En la Figura 2.21 se muestra la ventana de escritorio remoto de Windows que permite la conexión con la SBC.



Figura 2.21 Conexión a la Raspberry Pi mediante escritorio remoto

Finalmente se ingresa el usuario "pi" y la contraseña "raspberrypi", la placa Raspberry Pi está lista para su uso. En la Figura 2.22 se muestra la ventana de autenticación que tiene la Raspberry Pi para iniciar sesión.

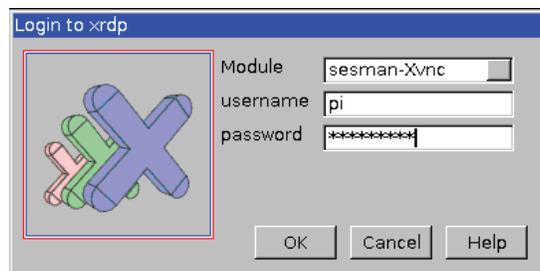


Figura 2.22 Ingreso a la Raspberry Pi

En la Figura 2.23 se muestra la interfaz que tiene el escritorio de la Raspberry Pi.

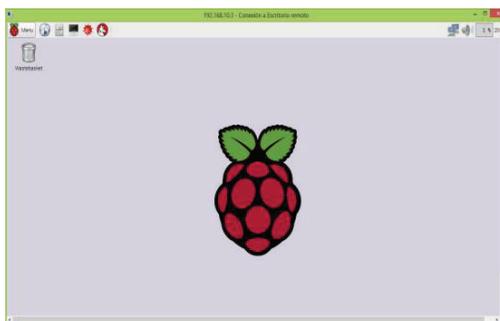


Figura 2.23 Escritorio de la Raspberry Pi

2.5.5.2 Instalación de Paquetes y Herramientas en la Raspberry PI

En el proceso de configuración de la Raspberry Pi es necesario instalar herramientas de software que permitan el adecuado funcionamiento de la placa respecto al hardware con el cual va a trabajar. Para lo cual se ha creado un *script*³⁴ que se ejecutará una vez que la placa arranque.

Se debe mencionar que Raspbian al ser una distribución de GNU/Linux tiene los mismos comandos que las demás distribuciones basadas en Debian.

En el Código 2.1 se muestra parte del *script*, donde se instalan varias herramientas de software y posteriormente se indica la funcionalidad que tienen cada una de ellas.

```
## INSTALANDO SOFTWARE NECESARIO
sudo apt-get install -y minicom mlocate nmap telnet ntp
sudo apt-get install -y python-pip python-matplotlib python-matplotlib-data python-pyaudio python-scipy
```

Código 2.1 Instalando software necesario en la Raspberry Pi

- Minicom: Se utiliza para comunicación serial de la Raspberry PI, permite una conexión remota mediante consola.
- Mlocate: Se utiliza para buscar archivos en el sistema.
- Nmap: *Network Mapper*, sirve para explorar y administrar la seguridad de redes de los computadores.

³⁴ Script: Es un archivo el cual contiene un conjunto de comandos que se ejecutan sin necesidad de abrir el archivo.

- Telnet: *Telecommunication Network*, protocolo que permite la conexión remota a una máquina y manejarla directamente.
- Ntp: *Network Time Protocol*, permite sincronizar los relojes de los sistemas informáticos a través del enrutamiento de paquetes para redes con latencia variable.

Se instalan las siguientes librerías Python.

- Pip: Sirve para instalar librerías Python.
- Matplotlib: Sirve para hacer gráficos en Python.
- Pyaudio: Sirve para manejar dispositivos de audio.
- Scipy: Para hacer cálculos avanzados en Python.

Para realizar la instalación de la Interfaz Gráfica de Usuario y tener una interfaz agradable para manejar el sistema se procede a instalar los siguientes paquetes, como se muestra en el Código 2.2.

```
## INSTALANDO GUI
sudo apt-get install -y --no-install-recommends xserver-xorg
sudo apt-get install -y --no-install-recommends xinit
sudo apt-get install -y lxde-core lxappearance
sudo apt-get install -y lightdm
```

Código 2.2 Paquetes de instalación para GUI

El Código 2.3 muestra la instalación de VNC³⁵ que permitir la conexión entre el cliente y el servidor, donde el cliente puede conectarse de forma remota al servidor y tomar el control del mismo.

```
## INSTALANDO VNC
sudo apt-get install tightvncserver -y
sudo cp -v ~/archivos/tightvncd /etc/init.d/
sudo chmod +x /etc/init.d/tightvncd
sudo update-rc.d tightvncd defaults
```

Código 2.3 Instalación de VNC

³⁵ VNC: Virtual Network Computing.

2.5.5.3 Instalación de paquetes y herramientas para la comunicación

Para establecer la comunicación entre el bloque de envío de datos del subsistema transmisor y el bloque de recepción y almacenamiento de datos del subsistema receptor se utiliza el modem 3G, para lo cual es necesario instalar algunas librerías que permitan la interacción con la base de datos que se encuentra en el bloque de recepción y almacenamiento de datos, el establecimiento del túnel SSH de modo cliente – servidor, tal como muestra el Código 2.4.

```
## MANEJO DE LA BD Y TUNEL SSH PARA LA COMUNICACION
sudo apt-get install mysql-client python-mysqldb libffi5-dev libffi6 libffi6-dbg
sudo apt-get install libssl-dev python-paramiko
sudo pip install cryptography sshtunnel
```

Código 2.4 Instalación de software para la comunicación

2.5.5.4 Instalación y creación de la base de datos

Este *script* se ejecuta una vez que arranque la Raspberry Pi, por lo tanto aquí se encuentran los comandos para empezar con la creación de la base de datos y las consultas que se deben realizar a la misma. En la creación de la base de datos se indica: el nombre de la tabla y los campos que tendrán la misma, el tipo de datos que serán insertados en los campos.

El Código 2.5 muestra la creación de la base de datos junto con las tablas y campos que tendrá.

```
CREATE DATABASE sonometroDB;
USE sonometroDB;
CREATE TABLE IF NOT EXISTS `data` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `fecha` datetime NOT NULL,
  `latitud` double NOT NULL,
  `longitud` double NOT NULL,
  `medicion` double NOT NULL,
  PRIMARY KEY (`id`)
) DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1;

CREATE TABLE IF NOT EXISTS `estado` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `tiempo` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci AUTO_INCREMENT=1 ;

INSERT INTO `sonometroDB`.`estado` (`id`, `tiempo`) VALUES (NULL, '0');
```

Código 2.5 Creación de la base de datos

2.5.6 ACOPLAMIENTO DE LOS ELEMENTOS DE HARDWARE PARA LA CONSTRUCCIÓN DEL SUBSISTEMA TRANSMISOR.

Una vez instalado el sistema operativo Raspbian en la placa SBC, se empieza con el acoplamiento de los elementos de hardware seleccionados en los subcapítulos 2.5.1, 2.5.2 y 2.5.3.

2.5.6.1 Acoplamiento del Módulo GPS con la SBC [31].

Para realizar la correcta conexión del módulo GPS NEO – GM -0-001 y la placa GY – GPSPGMV2 con la SBC, se procede a consultar la hoja de datos del módulo GPS.

La conexión del módulo GPS con la Raspberry Pi se realiza como muestra en la Figura 2.24. Hay que tener en cuenta que el pin de transmisión del módulo GPS se conecta en el pin de recepción en la placa y viceversa. Los pines VCC y GND del módulo GPS se conectan directamente a los pines VCC y GND de la Raspberry Pi, como se muestra en la Figura 2.24.

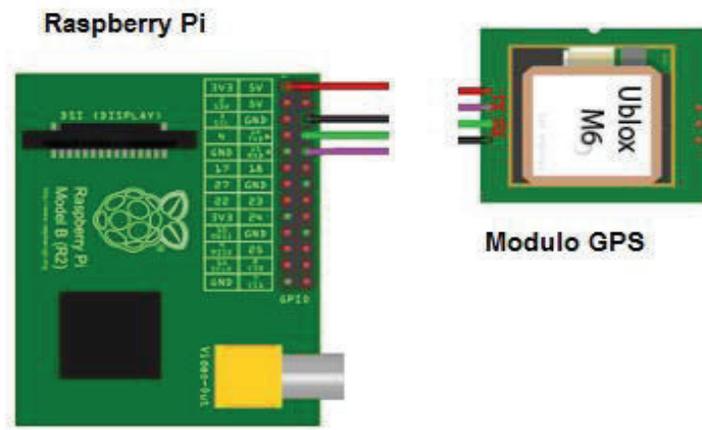


Figura 2.24 Conexión del módulo GPS.

2.5.6.1.1 Instalación y Configuración del Módulo GPS

Se procede a liberar el puerto serial para la conexión del módulo GPS, por defecto la Raspberry Pi para la comunicación serial utiliza los pines de tipo UART que en la

placa toman el nombre del puerto serial `dev/ttyAMA0`. Esto se realiza en el *script* general que tiene la instalación y configuración de la placa. Como muestra el código 2.6.

```
## LIBERANDO EL PUERTO SERIAL
#sudo nano /boot/cmdline.txt
sudo sed -i 's/dwc_otg/#dwc_otg/g' /boot/cmdline.txt
sudo bash -c "echo 'dwc_otg.lpm_enable=0 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait' >> /boot/cmdline.txt"
sudo sed -i 's#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100#T0:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100#g' /etc/inittab
```

Código 2.6 Liberación del puerto serial en la Raspberry Pi

El Código 2.7 muestra la línea de comandos que permite que los cambios realizados tengan efecto cuando se reinicia la placa y se instalan los paquetes que permitan conectar, manejar e interpretar los datos que se entregue el modulo GPS.

```
## INSTALANDO SOFTWARE DE MANEJO DE PUERTO SERIAL Y GPS
sudo apt-get install gpsd gpsd-clients python-gps
```

Código 2.7 Instalación de paquetes GPS

En el Código 2.8 se muestra la creación del demonio `gpsd` para escuchar el puerto `dev/ttyAMA0` cada vez que la Raspberry Pi arranque y se reciba los datos que toma el modulo GPS.

```
## CONFIGURO GPSD COMO DEMONIO Y PONGO A ESCUCHAR EL PUERTO /dev/ttyAMA0
sudo dpkg-reconfigure gpsd

## ARRANCO DEMONIO DE GPS
#sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock
```

Código 2.8 Creación y arranque de demonio `gpsd`

2.5.6.2 Acoplamiento de los micrófonos USB con la SBC

Los micrófonos que se utilizan en el presente trabajo tienen salidas USB 2.0, por lo tanto la conexión con la SBC es directa y no necesitan ser instalados, ya que la Raspberry Pi realiza el reconocimiento automático de los dispositivos USB. Además los paquetes y herramientas necesarias para el funcionamiento de estos dispositivos de audio fueron descargados e instalados en la sección de herramientas y paquetes en el subcapítulo 2.5.5.2.

En la Figura 2.25 se muestra la conexión del micrófono con la SBC.



Figura 2.25 Conexión del Micrófono USB

2.5.6.3 Acoplamiento del modem 3G Huawei E 3531 con la SBC

Para realizar la conexión del modem 3G con la Raspberry Pi, simplemente se inserta el dispositivo en cualquiera de los puertos USB disponibles de la SBC. La Raspberry Pi realiza el reconocimiento automático de los dispositivos que se conectan a los puertos USB.

En la Figura 2.26 se muestra la conexión del modem 3G Huawei E 3531 con la SBC.



Figura 2.26 Conexión del modem 3G

Se debe indicar que el modem 3G está conectado correctamente a la red celular cuando el Led que tiene incorporado se encuentra encendido de forma permanente y no parpadea constantemente.

2.5.6.3.1 Instalación y Configuración del Modem 3g Huawei E-3531.

Para realizar la instalación del modem en una distribución de Linux se procede a copiar la carpeta de archivos que viene en el dispositivo en el escritorio de la SBC, esto se realiza como medida de prevención para que los archivos se encuentren en lugar seguro al momento de la instalación. Entre los archivos copiados se encuentra la carpeta Linux en la cual están los archivos que se deben instalar para funcionamiento del modem.

En el *script* general se incluye la instalación de modeswitch que permite el cambio de servicio del modem, de ser un dispositivo de almacenamiento a ser un modem proveedor de Internet.

Modem manager sirve para gestionar y administrar los módems que se conecten en la placa raspberry.

Se incluye el paquete que permite la administración de conexiones de red del modem.

En el Código 2.9 se muestra la instalación del modem 3G en la Raspberry Pi.

```
## INSTALANDO SOFTWARE NECESARIO PARA EL MODEM USB
sudo apt-get install usb-modeswitch* modemmanager network-manager*

## CONFIGURANDO AUTOARRANQUE DEL MODEM
sudo cp -v ~/archivos/mobile-broadband-connect /etc/init.d/
sudo chmod +x /etc/init.d/mobile-broadband-connect
sudo update-rc.d mobile-broadband-connect defaults

## CREANDO REGLAS PARA QUE EL MODEM CAMBIE DE MODO ALMACENAMIENTO A MODEM GSM AUTOMATICAMENTE AL ARRANQUE
sudo bash -c "echo \$'ATTRS{idVendor}==\"12d1\", ATTRS{idProduct}==\"15ca\", RUN+=\"usb_modeswitch \\\b/%k\"' >
/etc/udev/rules.d/41-usb_modeswitch.rules"
```

Código 2.9 Instalación del Modem 3G

2.5.7 PROGRAMACIÓN DE LA PLATAFORMA RASPBERRY PI

Las actividades que debe realizar la placa Raspberry Pi en interacción con los demás elementos de hardware que hacen parte del subsistema transmisor se presentan en la Figura 2.27.

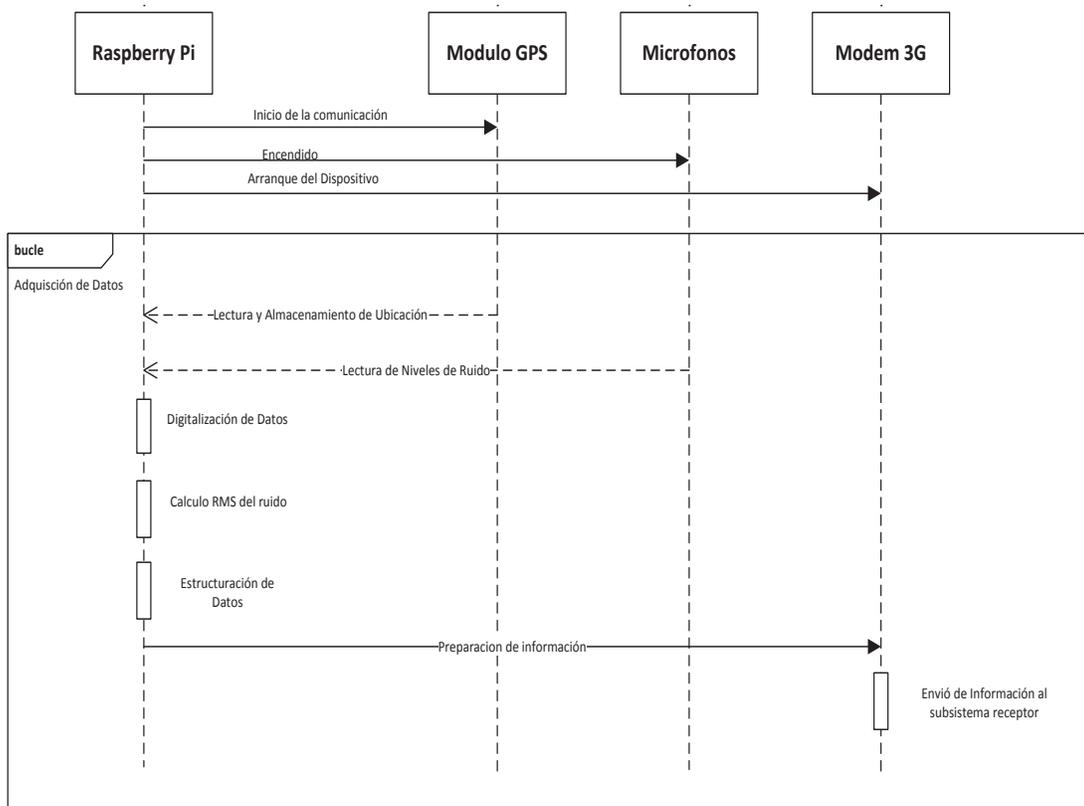


Figura 2.27 Diagrama de secuencias para la implementación del subsistema transmisor

Para cumplir con las actividades mencionadas en el diagrama de secuencias de la Figura 2.27 se realiza un *script* en Python en la placa Raspberry Pi.

Se debe indicar que el detalle de la programación que hace referencia a las actividades mencionadas en el diagrama de secuencias que debe realizar la SBC Raspberry Pi se encuentra en el Anexo 1. A continuación se presentan las partes destacadas del *script*.

2.5.7.1 Declaración de librerías

Como ya se determinó en el subcapítulo 2.5.4 para realizar la programación de la SBC Raspberry Pi se utiliza Python.

Las librerías son bloques de código que permiten usar funciones previamente establecidas en el lenguaje de programación para facilitar determinadas tareas, algunas librerías vienen por defecto y otras hay que importar para su uso.

En la Tabla 2.6 se presentan las librerías que fueron importadas para realizar la programación.

Librería	Función
<code>datetime</code>	Manipulación de la fecha y hora
<code>math</code>	Proporciona acceso a las funciones matemáticas
<code>MySQLdb</code>	Permite la conexión, lectura y escritura en la base de datos
<code>os</code>	Permite acceder a funcionalidades del sistema operativo
<code>pyaudio</code>	Proporciona enlaces para PortAudio v19
<code>struct</code>	Interpreta cadenas como datos binarios empaquetados
<code>threading</code>	Permite utilizar hilos
<code>Time</code>	Ofrece funciones relacionadas con el tiempo

Tabla 2.6 Librerías importadas para Python

2.5.7.2 Lectura y almacenamiento de los datos de geolocalización

Para tomar las lecturas del módulo GPS y almacenarlas en un archivo en la Raspberry Pi se crea un proceso donde: el modulo GPS arranque, se conecte con los satélites, tome las lecturas de latitud y longitud y las almacene en un archivo de texto para su posterior envío.

La velocidad de comunicación entre la Raspberry Pi y el modulo GPS está dado por defecto 9600 baudios que es la velocidad a la cual trabaja el puerto serial UART, en la Raspberry Pi este puerto toma el nombre de ttyAMA0.

El modulo GPS trabaja con varios tipos de mensajes para presentar la información recolectada para lo cual utiliza el protocolo NMEA³⁶.

En la Figura 2.28 se presentan los tipos de mensajes y características que maneja el modulo GPS.

³⁶ NMEA: National Marine Electronic Association

CFG_COM1	CFG_COM0	Protocol	Messages	UARTBaud rate	USB power
1	1	NMEA	GSV, RMC, GSA, GGA, GLL, VTG, TXT	9600	BUS Powered

Figura 2.28 Campos del mensaje NMEA

En la Figura 2.29 se presenta el diagrama de flujo donde se indica el proceso de arranque y lectura de datos de geolocalización del módulo GPS.

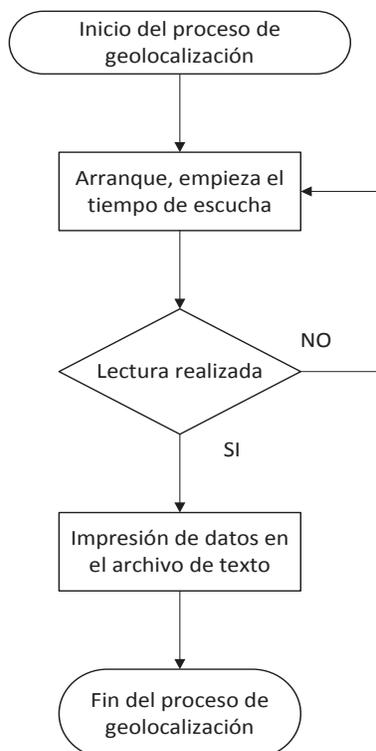


Figura 2.29 Diagrama de flujo del proceso de recolección de datos Geolocalización

En el Código 2.10 se crea la clase `GpsPoller`, donde se crea el hilo para que el modulo empiece la escucha y el modulo pueda engancharse, asi los satélites provean información de geolocalización. Se activa el *streaming* de datos para el GPS.

```

class GpsPoller(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
        global gpsd #Declaramos GPSD como una variable global
        gpsd = gps(mode=WATCH_ENABLE) #Iniciamos el streaming de datos GPS
        self.current_value = None
        self.running = True
  
```

Código 2.10 Inicialización del módulo GPS

En el Código 2.11 se inicia el proceso de escucha y en caso de conectarse el modulo GPS se imprimen las lecturas, caso contrario se continua en el proceso de escucha.

```

gpsp.start() # Y lo arrancamos
while True: #Iniciamos un bucle

    os.system('clear')#Limpiamos la terminal
#Este paso es opcional ya que no hace falta que muestre ningun mensaje en la terminal
#Si ponemos el script como un 'daemon'

    print 'Latitud: ' , gpsd.fix.latitude #Sacamos por pantalla la latitud
    print 'Longitud: ' , gpsd.fix.longitude #Sacamos por pantalla la longitud

    #Si la latitud y la longitud son igual a 0.0
    if gpsd.fix.latitude == 0.0 and gpsd.fix.longitude == 0.0:
#Sacamos por pantalla este mensaje
        print "Esperando GPS..."

```

Código 2.11 Proceso de escucha para el modulo GPS

En el Código 2.12 se escribe en un archivo de texto las lecturas de geolocalización que tome el módulo GPS.

```

print "GPS OK"
#Guardamos los datos en un archivo
#(uso 'a' para agregar el texto. si usase 'w' solo se guardarían las últimas coordenadas)
data = open("locations.txt", "a")
data.write("%s,%s\n" % (gpsd.fix.latitude, gpsd.fix.longitude))
data.close()

```

Código 2.12 Escribir los datos de geolocalización en un archivo de texto

En el Código 2.13 se maneja la finalización del hilo y del proceso de escucha del módulo GPS.

```

except (KeyboardInterrupt, SystemExit): #Al pulsar ctrl+c
    print "\nDesconectando GPS..."
    gpsp.running = False
    gpsp.join() # Espera a que el thread finalice
print "Ok.\nSaliendo..."

```

Código 2.13 Finalización del proceso de escucha del módulo GPS

2.5.7.3 Lectura y almacenamiento de los niveles de ruido

Para la medición de ruido se realiza un proceso matemático que permite determinar un valor promedio de todas las lecturas de ruido tomadas por los micrófonos para posterior envío.

Se realiza un valor promedio (RMS) de las lecturas de ruido con un fin práctico, para cumplir con el propósito del presente trabajo que es determinar el nivel de ruido en el ambiente.

A continuación se explica el fin práctico: el prototipo tendrá la capacidad de movilidad por diferentes lugares, si las lecturas que se presentan no son promediadas se puede dar lugar a lecturas de ruido aisladas o particulares que no pertenezcan al comportamiento habitual del sitio.

En la Figura 2.30 se presenta en un diagrama de flujo con el proceso de medición y cálculo del valor RMS del ruido.

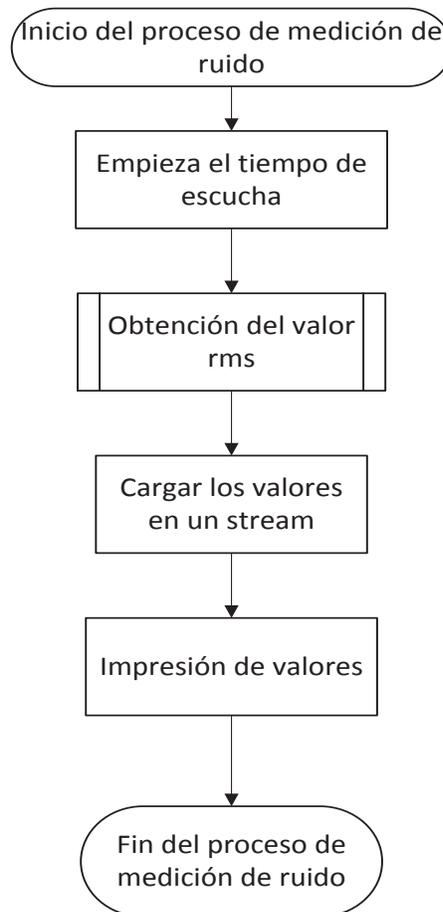


Figura 2.30 Diagrama de flujo de la medición de ruido

En el Código 2.14 se crea la clase Microphone, se establece el tiempo de escucha de los micrófonos, se realiza la definición de la función para establecer el valor RMS de las lecturas tomadas.

```

class Microphone:

    LISTEN_TIME = 5

    def rms(self, frame):
        count = len(frame)/2
        format = "%dh"% (count)
        shorts = struct.unpack( format, frame )
        sum_squares = 0.0
        for sample in shorts:
            n = sample * (1.0/32768.0)
            sum_squares += n*n
        rms = math.pow(sum_squares/count,0.5);
        return rms * 1000

```

Código 2.14 Creación de la clase Microphone

En el Código 2.15 se carga en un *stream* los valores RMS de las lecturas de ruido realizadas.

```

CHUNK = 8192; RATE = 44100; LISTEN_TIME = self.LISTEN_TIME
print "Escuchando..."

p = pyaudio.PyAudio()
stream = p.open(format=pyaudio.paInt16,
                channels=1,
                rate=RATE,
                input_device_index=2,
                input=True,
                frames_per_buffer=CHUNK)

```

Código 2.15 Creación de stream de lecturas de ruido

En el Código 2.16 se llena el *stream*, se imprime los resultados de las lecturas realizadas y cierra el *stream*.

```

for i in range(0, RATE / CHUNK * LISTEN_TIME):
    input = stream.read(CHUNK)
    all.append(input)
stream.stop_stream()
stream.close()
data = ''.join(all)
rms_value = self.rms(data)
print rms_value
return rms_value

```

Código 2.16 Proceso de llenado del stream

2.5.7.4 Establecimiento del túnel SSH

Para establecer la conexión entre el subsistema transmisor y el subsistema receptor se establece un túnel de comunicaciones SSH.

En la Figura 2.31 se presenta el diagrama de flujo que se utiliza para el proceso de establecimiento del túnel SSH.

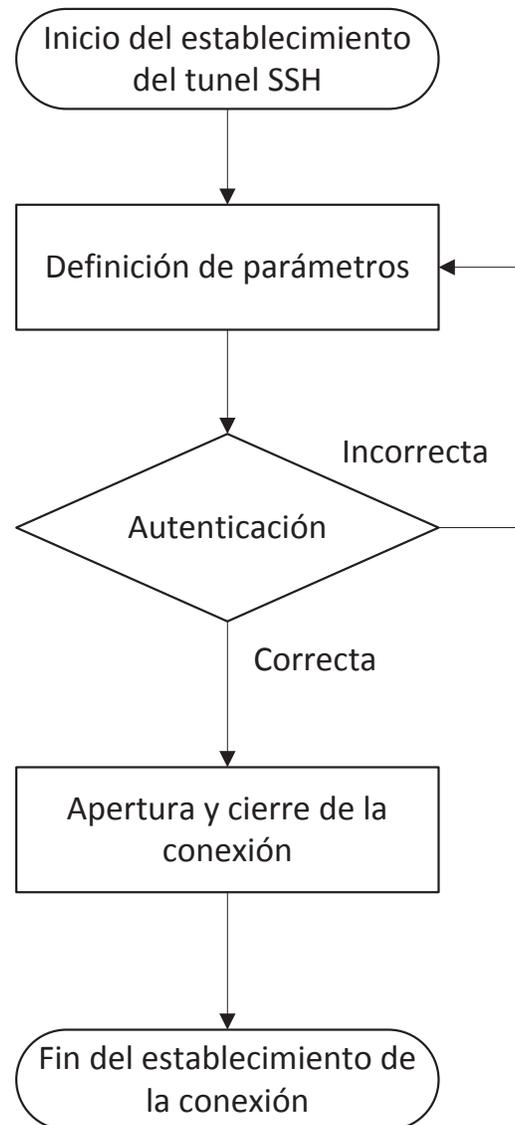


Figura 2.31 Diagrama de flujo del túnel SSH

En el Código 2.17 se muestra como se deben ingresar los parámetros que definen: el dominio, el puerto, usuario y contraseña tanto para el host local como para el servidor remoto.

```

def ssh_open(self):
    SSH_HOST = '192.168.2.1'
    SSH_PORT = 22
    SSH_USER = 'casa'
    SSH_PASS = 'casa'
    SSH_LHOST = '127.0.0.1'
    SSH_LPORT = 7900
    SSH_BHOST = '127.0.0.1'
    SSH_BPORT = 3306

```

Código 2.17 Ingreso de parámetros para el túnel SSH

En el Código 2.18 se realiza el proceso de autenticación para el establecimiento de la conexión mediante el túnel SSH.

```

self.ssh_server = SSHTunnelForwarder(
    (SSH_HOST, SSH_PORT),
    ssh_username=SSH_USER,
    ssh_password=SSH_PASS,
    local_bind_address=(SSH_LHOST, SSH_LPORT),
    remote_bind_address=(SSH_BHOST, SSH_BPORT)
)

```

Código 2.18 Proceso de autenticación SSH

En el Código 2.19 se establece la conexión y se presenta un mensaje indicando lo ocurrido, se cierra la conexión y se presenta un mensaje indicando lo ocurrido.

```

self.ssh_server.start()
print "Conexion SSH abierta"

def ssh_close(self):
    self.ssh_server.stop()
    print "Conexion SSH cerrada"

```

Código 2.19 Apertura y cierre de la conexión SSH

2.5.7.5 Establecimiento de la conexión e interacción con la base de datos

En la Figura 2.32 se presenta el diagrama de flujo donde se indica el proceso de interacción entre el subsistema transmisor con la base de datos que está en el subsistema receptor.

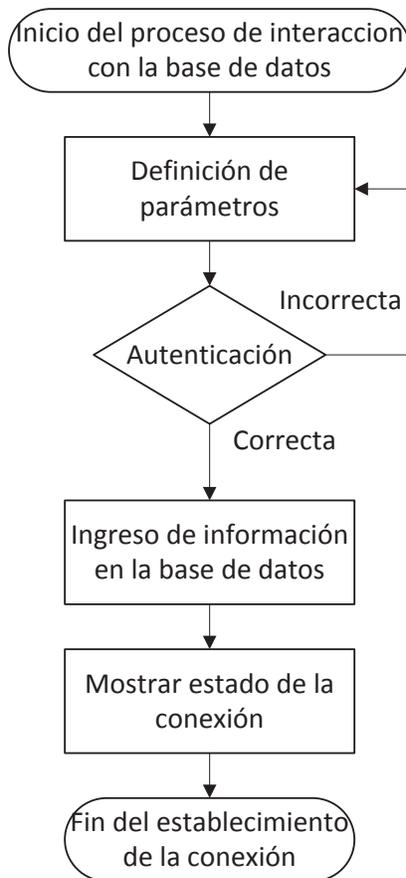


Figura 2.32 Interacción con la base de datos

En el Código 2.20 se definen los parámetros para el establecimiento de la conexión con la base de datos.

```

DB_HOST = '127.0.0.1'
DB_PORT = 7900
DB_USER = 'sonometro'
DB_PASS = 'tesispass'
DB_NAME = 'sonometroDB'
  
```

Código 2.20 Parámetros para la conexión con la base de datos

En el Código 2.21 se realiza la autenticación para la conexión con la base de datos, se ingresa el usuario y contraseña. Si la autenticación ha sido correcta se presenta un mensaje indicando que se ha establecido la conexión con la base de datos.

```

self.bd_con = mdb.connect(host=DB_HOST,port=DB_PORT,user=DB_USER,passwd=DB_PASS,db=DB_NAME)
print "Conexion BD abierta"
  
```

Código 2.21 Autenticación con la base de datos

En el Código 2.22 se realiza el ingreso de información a la base de datos, el llenado de la base de datos está dado por la siguiente información: fecha, latitud, longitud y medición.

```
def db_set_data(self, fecha, lat, lng, rms):
    with self.bd_con:
        cur = self.bd_con.cursor()
        sql = "INSERT INTO data(fecha,latitud,longitud,medicion) VALUES('"+fecha+"', '"+str(lat)+"', '"+str(lng)+"', '"+str(rms)+"'"
        print "sql: ", sql
        cur.execute(sql)
        self.bd_con.commit()
```

Código 2.22 Ingreso de información a la base de datos

En el Código 2.23 se realizan consultas a la base de datos para determinar el estado de la base de datos, también se realiza la actualización de la base de datos cada cierto tiempo.

```
def db_get_estado(self):
    with self.bd_con:
        cur = self.bd_con.cursor()
        cur.execute("SELECT * FROM estado")
        rows = cur.fetchall()
        for row in rows:
            segundos = row[1]
        return segundos

def db_set_estado(self, tiempo):
    with self.bd_con:
        cur = self.bd_con.cursor()
        cur.execute("UPDATE estado SET tiempo = %s WHERE Id = 1", (str(tiempo)))
        self.bd_con.commit()
```

Código 2.23 Consultas a la base de datos

En el Código 2.24 se cierra la conexión con la base de datos.

```
def db_close(self):
    self.bd_con.close()
    print "Conexion BD cerrada"
```

Código 2.24 Cierre de la conexión con la base de datos

2.5.7.6 Funcionamiento del menú principal

Una vez creadas las funciones que debe realizar el programa, se procede a entrelazar entre ellas, para crear el funcionamiento general del programa que cumple con el propósito de la Raspberry Pi en el subsistema transmisor.

En la Figura 2.33 se presenta el diagrama de flujo que cumple el programa.

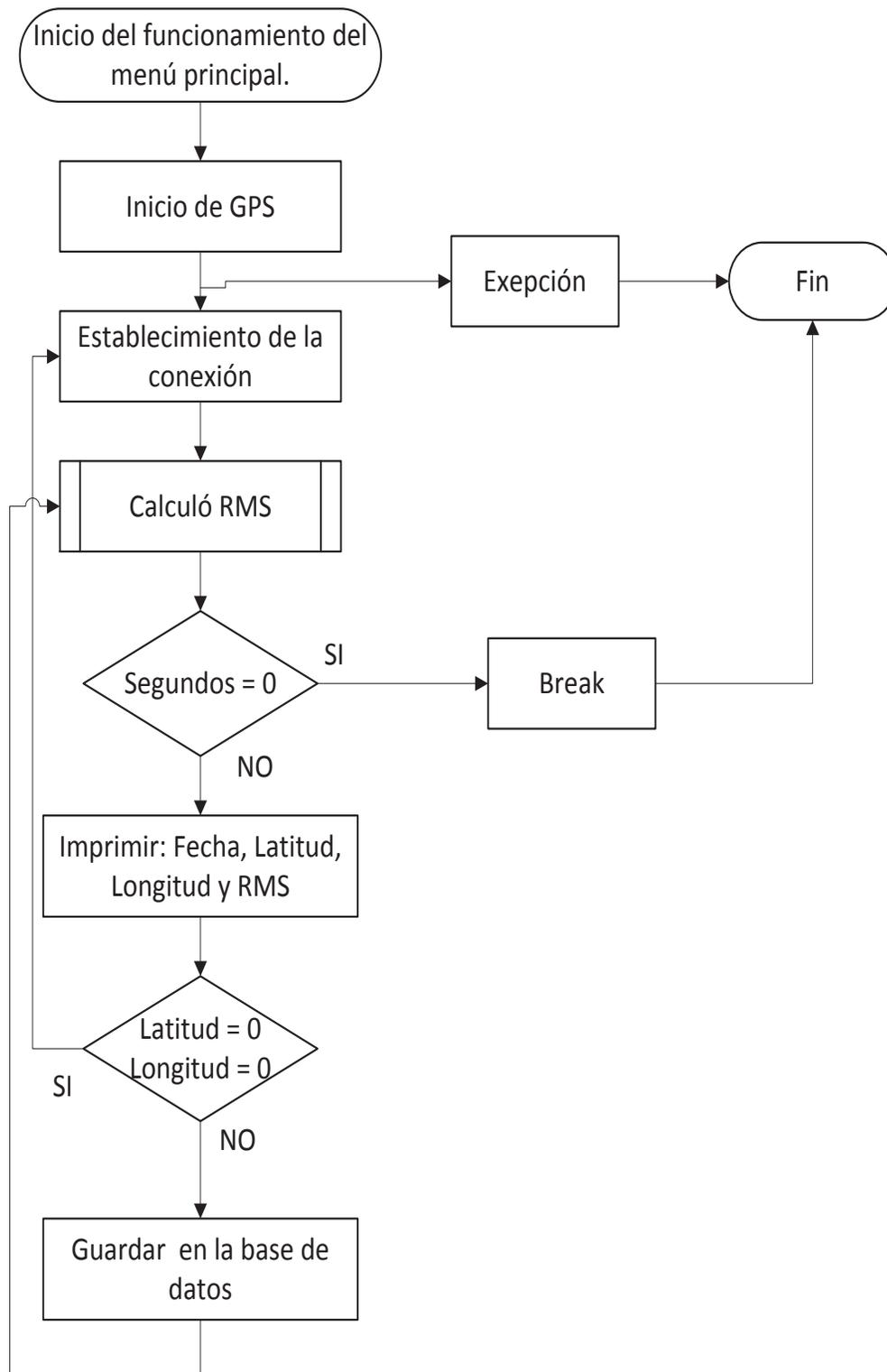


Figura 2.33 Diagrama de flujo del menú principal

En el Código 2.25 se inicia el objeto `gpsp` y se crean los objetos de las clases `Microphone` y `MySQLoverSSHTunnel`.

```
gpsp.start()

mic = Microphone()
con = MySQLoverSSHTunnel()
```

Código 2.25 Inicialización y creación de objetos

En el Código 2.26 se establece la conexión a través del túnel SSH con la base de datos y carga del valor RMS del ruido.

```
con.ssh_open()
con.db_connection()
sleep(1)
segundos = con.db_get_estado()
con.db_close()
con.ssh_close()

print "Estado: segundos = " + str(segundos)
for i in range(0, segundos // mic.LISTEN_TIME):
    rms = mic.passiveListen()
    os.system('clear')
```

Código 2.26 Establecimiento de la conexión y cargar del valor RMS del ruido

En el Código 2.27 se imprime la siguiente información: Fecha, latitud, longitud y RMS.

```
print 'Hora UTC: ', gpsd.utc, ' + ', gpsd.fix.time
print 'Latitud: ', gpsd.fix.latitude
print 'Longitud: ', gpsd.fix.longitude
print 'RMS: ', rms
```

Código 2.27 Impresión en pantalla de datos tomados

Si el modulo GPS no ha podido conectarse y realizar alguna lectura de geolocalización y los campos son cero, en este caso el Modulo estará esperando conectarse a alguno de los satélites y en este caso se muestra un mensaje indicando lo ocurrido. Si el modulo GPS ha logrado conectarse y tomar datos de geolocalización se muestra un mensaje de lo ocurrido y se toman las lecturas normalmente. Como se muestra en el código 2.28.

```

if gpsd.fix.latitude == 0.0 and gpsd.fix.longitude == 0.
    print "Esperando GPS..."
else:
    print "GPS OK"
    con.ssh_open()
    con.db_connection()

```

Código 2.28 Decisión si el modulo GPS no ha logrado conectarse

El Código 2.29 se utiliza para mostrar si el modulo GPS se ha conectado y ha tomado datos de geolocalización, en ese caso se envía la siguiente información a la base de datos: Fecha, Latitud, Longitud y RMS.

```

#Guardamos los datos en una BD
ahora = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
con.db_set_data(ahora,gpsd.fix.latitude,gpsd.fix.longitude,rms)
segundos = con.db_get_estado()

```

Código 2.29 Se guarda información en la base de datos

En el Código 2.30 muestra que en caso de interrumpirse la conexión, se presente alguna excepción o un bucle infinito, se manejan las excepciones para finalizar con el proceso.

```

except (KeyboardInterrupt, SystemExit):
    print "\nDesconectando GPS..."
    gpsp.running = False
    gpsp.join()
except:
    pass

print "\nSaliendo..."
if gpsp.running:
    gpsp.running = False
    gpsp.join()

```

Código 2.30 Manejo de excepciones para finalizar el proceso

2.6 SUBSISTEMA RECEPTOR

El subsistema receptor está compuesto por los elementos de hardware y software que permitan cumplir con lo descrito en el subcapítulo 2.2 y el subcapítulo 2.4.

A continuación se detalla los elementos de hardware y software que se utilizarán en la implementación del subsistema receptor.

2.6.1 SELECCIÓN DE LOS ELEMENTOS DE HARDWARE PARA EL SUBSISTEMA RECEPTOR.

Como se muestra en el subcapítulo 2.4, el subsistema receptor está compuesto por los bloques: Recepción y almacenamiento de datos y presentación de resultados. Para lo cual se necesitará únicamente de un computador con conexión a Internet.

La selección del hardware se realizó bajo los siguientes criterios:

- Plataforma de libre distribución.
- Sistema operativo de tipo *Open Source*.
- Debe soportar GNU/Linux.
- Bajo Costo.

2.6.1.1 Computador [32]

En la Tabla 2.7 se presentan las características del equipo.

Laptop	Especificaciones
Procesador	Intel core i5 CPU 2,27 MHz.
RAM	4 GB
Marca	HP Pavilion dm4
Tamaño	14.5"
Puertos	2.0 USB
Puerto Ethernet	10/100 MB

Tabla 2.7 Características técnicas del servidor

El computador con conexión a Internet soporta las aplicaciones: Virtual Box³⁷, Mysql y Google Maps. En este caso la conexión a Internet puede ser mediante cable Ethernet o a través de la tecnología Wi-Fi³⁸, la forma de conexión a Internet que el computador tenga es transparente para el propósito del presente trabajo.

³⁷ Virtual Box: Software que permite la instalación y uso de máquinas virtuales en un computador.

³⁸ Wi-Fi: Conexión inalámbrica de dispositivos electrónicos a un dispositivo proveedor de internet.

El servicio de Internet que usa el computador puede variar de acuerdo al sitio donde esté se encuentre, tendrá preferencia la conexión inalámbrica debido a la portabilidad que ofrece.

Por la cantidad de datos que el subsistema transmisor envía, el servicio de Internet que use el subsistema receptor puede ser de bajo ancho de banda

2.6.2 SELECCIÓN DE LOS ELEMENTOS DE SOFTWARE PARA EL BLOQUE DE RECEPCIÓN Y ALMACENAMIENTO DE DATOS

Para cumplir con lo descrito en el subcapítulo 2.4 es necesario indicar los criterios de selección para las herramientas de software.

- 1.- *Open Source* y licenciamiento gratuito.
- 2.- Soportar el paradigma orientado a objetos.
- 3.- Poseer abundante información de consulta.

2.6.2.1 VirtualBOx [33]

Es un software de código abierto bajo los términos de licencia pública general GNU versión 2.

Es un software que permite la instalación de máquinas virtuales en máquinas físicas. La instalación de este software puede ser en los sistemas operativos Windows 7, Windows 8, Solaris, OpenBSD, etc. VirtualBox tiene la capacidad de ser instalado en sistemas operativos licenciados sin problemas.

En las máquinas virtuales se pueden instalar varios tipos de sistemas operativos como son: Windows, Linux, etc. En el proceso de instalación de las máquinas virtuales se asignan capacidades físicas como son: capacidad de almacenamiento, memoria, etc. Además, se configuran las características de red y dispositivos externos que tendrán cada una de estas máquinas.

2.6.2.2 Sistema Operativo Debían [34]

Es una distribución de GNU/Linux de código abierto y licencia pública. Cuenta con más de 43000 paquetes precompilados que permiten una fácil instalación. También cuenta con un gestor de paquetes APT que hace posible gestionar miles de paquetes.

2.6.2.3 MySQL [35]

Es un sistema que permite almacenar, administrar y gestionar información. Desarrollado bajo licencia dual por ORACLE *Corporation* [36] y es la base de datos *open source*³⁹ más usada.

En sus inicios MySQL fue desarrollado por MySQL AB, en 2008 fue adquirida por Sun Microsystem y finalmente es comprada por ORACLE en 2010. Al ser una empresa privada que posee el copyright de la mayor parte del código causa que el licenciamiento sea dual, es decir puede ser pagado o gratuito, dependerá de cada versión del producto.

Por ejemplo, las versiones Enterprise tienen un costo de licenciamiento ya que tienen servicios de monitoreo de la base de datos y soporte oficial. Para las versiones *Community* está distribuida bajo licencia pública general de GNU.

Los datos en MySQL son almacenados en tablas, una tabla es una colección de datos que están relacionados entre sí mediante filas y columnas. Para interactuar con la base de datos MySQL se dispone una serie de sentencias que permiten realizar: comandos, consultas, crear campos, crear tablas, eliminar campos, eliminar tablas, etc.

A continuación se presentan las principales características de MySQL, las cuales.

- Orientado para entornos de desarrollo web.
- Permite la conexión con lenguajes de programación: C, C++, Pascal, Java, PHP, Python, etc.
- Disponibilidad en múltiples plataformas (Windows, Linux, etc.).
- Es un amplio subconjunto del lenguaje SQL.
- Generalmente se ejecuta en un servidor.
- Tiene la capacidad de manejar pequeños y grandes volúmenes de datos.
- Tiene gran rendimiento gracias a su implementación multihilo.
- Alto nivel de seguridad en los datos.
- Gran cantidad de librerías y herramientas que permiten su uso.

³⁹ Open Source: Software usado y distribuido libremente.

2.6.3 SELECCIÓN DE LOS ELEMENTOS DE SOFTWARE PARA EL BLOQUE DE PRESENTACIÓN DE RESULTADOS

Para cumplir con lo descrito en el subcapítulo 2.4 es necesario indicar los criterios de selección para las herramientas de software.

- 1.- *Open Source* y licenciamiento gratuito.
- 2.- Soportar el paradigma orientado a objetos.
- 3.- Poseer abundante información de consulta.

2.6.3.1 Lenguaje de programación PHP [37]

Pre *Hypertext Processor*, es un lenguaje de programación de uso general, fue creado para el desarrollo web y orientado a funcionar en el lado del servidor. Puede ser incorporado como parte del documento HTML⁴⁰ en vez de que se llame a un archivo externo para que procese los datos. Hasta el momento PHP se considera como uno de los lenguajes más flexibles y de gran rendimiento para ser usado en los servidores web.

A continuación se presentan las características principales que tiene este lenguaje de programación y por la cuales fue seleccionado para el desarrollo del presente trabajo.

- Orientado al desarrollo de aplicaciones web con conexión a una base de datos.
- Soporta el paradigma de programación orientada a objetos.
- El código escrito en PHP es invisible al navegador web y al cliente.
- Puede emplearse en los principales sistemas operativos como son: Linux, Microsoft Windows y Mac. Además admite la mayoría de servidores web como son: Apache, IIS, etc.
- Tiene la capacidad de comunicarse con otros servicios haciendo uso de los protocolos: LDAP, IMAP, SNMP, NNTP, etc. Además para la comunicación se pueden crear sockets de red
- Posee gran cantidad de información almacenada en su página oficial.

⁴⁰ HTML: HyperText Markup Lenguaje, sirve para el marcado de páginas Web.

2.6.3.2 HTML [38]

Es un lenguaje de programación que se utiliza para elaborar el marcado en las páginas web. Las principales características de HTML son:

- El programa desarrollado es un conjunto de etiquetas de marcado.
- Los documentos escritos en HTML están descritos por las etiquetas HTML.
- Cada etiqueta HTML describe diferente contenido del documento.
- Los documentos HTML son interpretados por un navegador web.

Las etiquetas HTML son palabras clave que están rodeadas por corchetes angulares. Generalmente las etiquetas vienen en pares, la primera es la etiqueta de inicio y la segunda es la etiqueta de cierre, la etiqueta de cierre viene con (/) antes del nombre.

En la Figura 2.34 se detalla la estructura que maneja HTML para la creación de páginas web.

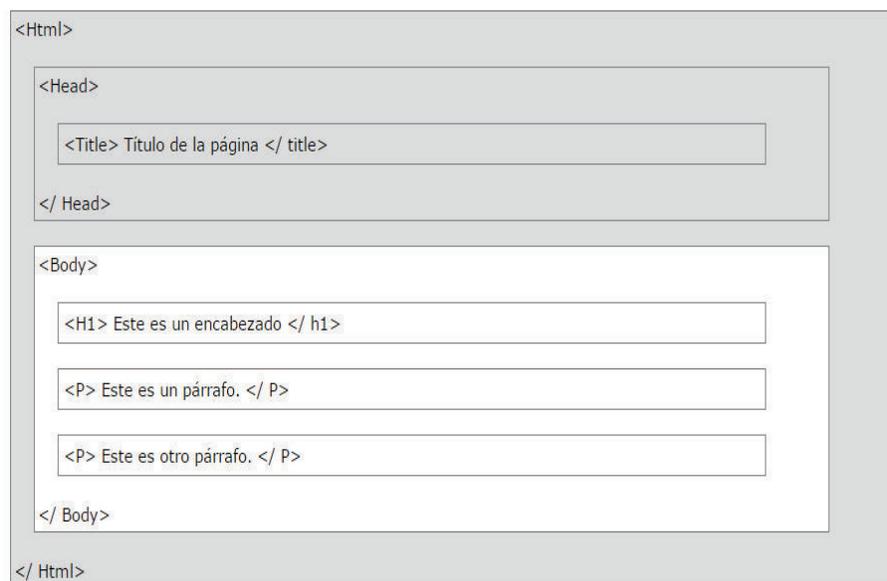


Figura 2.34 Estructura de un documento HTML [38]

A continuación se describe las funciones de cada etiqueta que se presenta en la estructura de un documento HTML.

<! DOCTYPE> Se encuentra al inicio de todo documento HTML y define el tipo de documento, ayuda al navegador web a presentar la página correctamente

<Html> </Html> Describe el inicio y fin de un documento HTML.

<Head> </Head> Provee información del documento HTML pero no se muestra en la página web.

<Title> </Title> Indica el título del documento.

<Body> </Body> Muestra el contenido visible en la página web.

<H1> </h1> Indica el encabezado del contenido visible.

<P> </P> Indica un párrafo del contenido visible.

2.6.3.3 Java Script

Es un lenguaje de programación interpretado, soporta el paradigma de programación orientado a objetos.

Este lenguaje es muy utilizado en el lado del cliente para ser parte de un navegador web logrando mejorar la interfaz de usuario y permitiendo tener páginas web dinámicas. Hoy en día todos los navegadores web pueden interpretar el código Java Script integrado en las páginas web.

En un principio Java Script se utilizaba únicamente para realizar operaciones en páginas web HTML al lado del cliente sin tener acceso a funciones del servidor, ahora con Java Sript se puede enviar y recibir información al servidor haciendo uso de la tecnología AJAX⁴¹.

A continuación se presentan las características principales que tiene este lenguaje de programación y por la cuales fue seleccionado para el desarrollo del presente trabajo.

- Es un lenguaje de programación que trabaja con HTML y se utiliza en la web.
- Generalmente funciona en el lado del cliente.
- Define el comportamiento de las páginas web.
- Puede cambiar los estilos HTML a través de CSS⁴².
- El código de Java Script puede ser colocado en la cabecera y cuerpo de un documento HTML.

⁴¹ AJAX: Asynchronous JavaScript And XML, técnica de desarrollo web para crear aplicaciones interactivas.

⁴² CCS: Cascading Style Sheets, es un lenguaje usado para definir y crear la presentación de un documento estructurado en HTML y XML.

- El código de Java Script debe ser colocado dentro de las etiquetas <script> </script>.
- Compatible con otros lenguajes de programación.

2.6.3.4 API de Google Maps [39]

Google Maps es un servidor que funciona en la web y ofrece aplicaciones de mapas y fotografías de todo el mundo tomadas por satélites, pertenece a Alphabet Inc y fue anunciado por primera vez el 8 de Febrero de 2005. Google API permite la comunicación que con los servicios que Google ofrece y se define como un conjunto de interfaces de programación de aplicaciones.

Las API de Google Maps son gratuitas y están limitadas en su uso. Estas herramientas están disponibles para usarse en Android, iOS, navegadores web y a través de servicios web HTTP.

2.6.3.4.1 Maps Java Script API

Para presentar en forma gráfica los datos tomados por el subsistema transmisor el subsistema receptor utiliza Maps Java Script API en donde el usuario puede realizar cualquier cambio, modificación o implementación sobre el API.

Maps Java Script API ofrece varias maneras de visualización de datos en los mapas, estas son: capa de datos, tablas de fusión, mapas de calor, etc.

2.6.3.4.2 Capa de Mapas de Calor [24]

Los mapas de calor tienen la capacidad de representar la intensidad de los datos en los puntos geográficos. Una desventaja que tiene esta manera de presentar los resultados es: cuando existe un gran número de puntos a representar el rendimiento disminuye. Entre las ventajas que presenta esta manera es: el juego de colores a usar para la representación de puntos es totalmente personalizable, además el radio de los puntos representados también puede ser modificado.

Los puntos representados al momento de realizar zoom sobre el mapa se ajustan al tamaño y de esta manera no sufren distorsión alguna. Los datos a ser representados tienen la facilidad que pueden ser almacenados en el documento HTML o en un servidor y pueden usarse en cualquier momento.

2.6.4 PROGRAMACIÓN DEL SUBSISTEMA RECEPTOR

El subsistema receptor está compuesto principalmente del servidor web para presentar los resultados y la base de datos en donde se almacena la información enviada por el subsistema transmisor. La programación del subsistema receptor empieza por el documento HTML del servidor web, aquí se encuentran las conexiones con todas las funciones del subsistema receptor.

EL documento HTML tiene la programación lógica de las funciones del servidor a través del lenguaje PHP y la programación que permite el diseño y presentación de la página web a través de HTML, CSS y Java Script en el cliente.

En la Figura 2.36 se presenta el diagrama de flujo que define el funcionamiento de la aplicación web en el servidor del subsistema receptor.

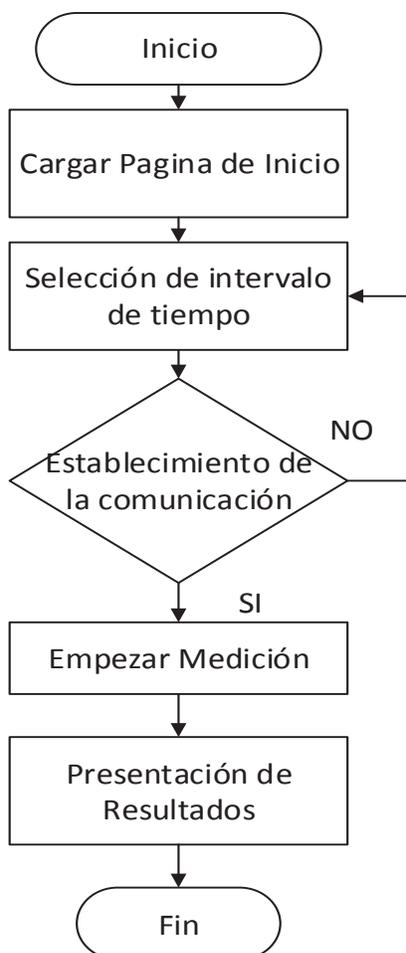


Figura 2.35 Diagrama de flujo del servidor web

En la Figura 2.36 se presentan las actividades que debe realizar el subsistema receptor compuesto por los bloques de recepción y almacenamiento de datos y el bloque de presentación de resultados.

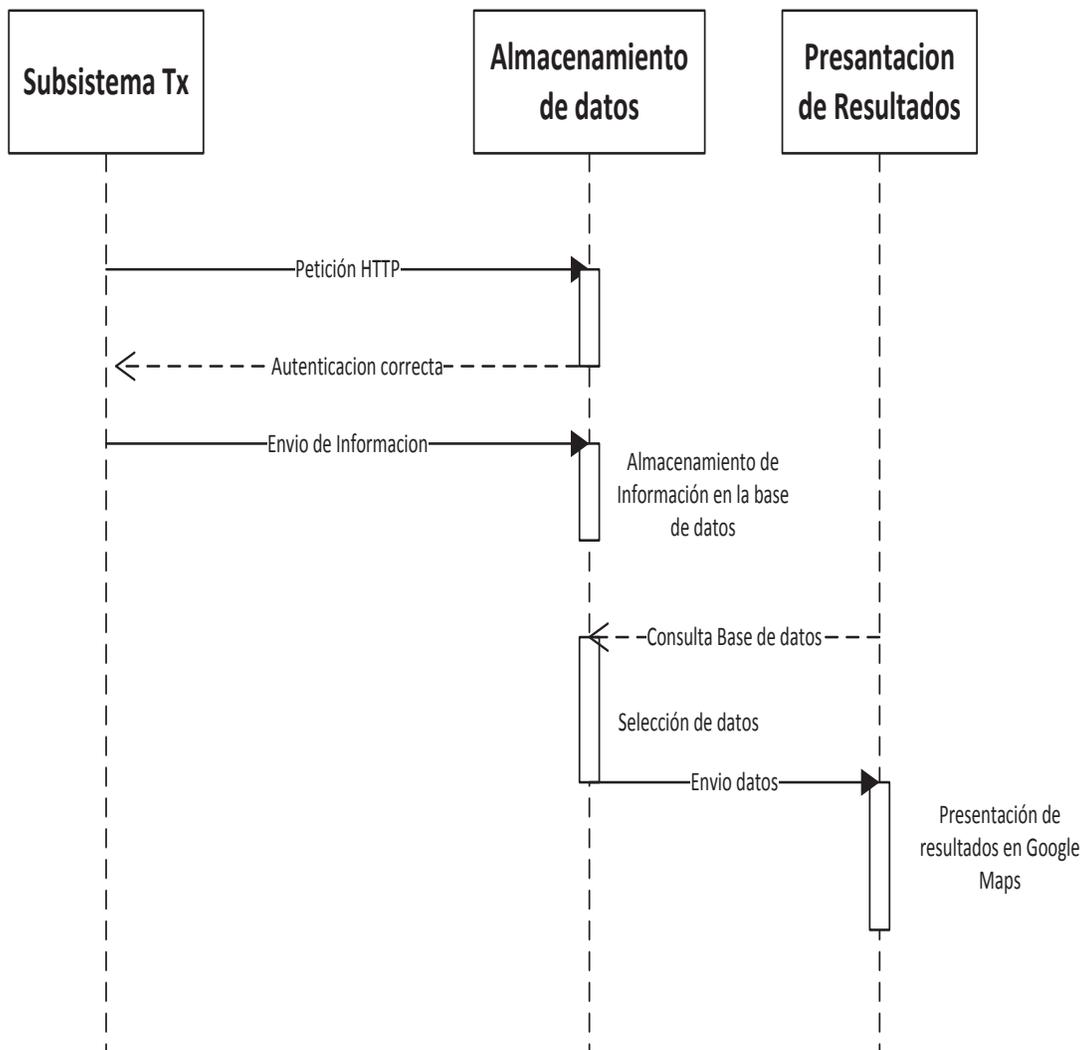


Figura 2.36 Diagrama de secuencias del subsistema receptor

2.6.4.1 Programación del documento HTML

El documento HTML principalmente da el formato a la página web, por esta razón mediante el uso de etiquetas en la cabecera se hace el llamado de los diferentes *scripts* que intervienen en la programación.

En el Código 2.32 se muestra el llamado de *scripts* mediante etiquetas en el lenguaje HTML.

```
<head>
  <meta name='viewport' content='initial-scale=1.0, user-scalable=no'>
  <meta charset='utf-8'>
  <title>Sonómetro</title>
  <link rel='stylesheet' type='text/css' href='./css/jquery.datetimepicker.css' />
  <link rel='stylesheet' type='text/css' href='./css/mystyle.css' />
  <link href='tablecloth/tablecloth.css' rel='stylesheet' type='text/css' media='screen' />

  <script type='text/javascript' src='tablecloth/tablecloth.js'></script>
  <script src='./js/jquery.js'></script>
  <script src='./js/jquery.datetimepicker.full.js'></script>
  <script src='./js/moment.min.js'></script>
```

Llamado de Scripts

Código 2.31 Llamado de scripts en el documento HTML

En este documento se descarga el mapa para representar los datos enviados por el subsistema receptor. La dirección es la siguiente:

https://maps.googleapis.com/maps/api/js?v=3.exp&libraries=visualization,geometry&key=AlzaSyC6w3XN7_5Lz8xoBHsXTNfMv5VYI2GSEMc'.

En este documento HTML también constan los botones consultar y capturar. El botón Consultar se utiliza para realizar una consulta a la base de datos sobre los datos tomados por el subsistema transmisor en determinado intervalo de tiempo.

El botón capturar se utiliza para enviar una solicitud de lectura al subsistema transmisor y poder representar los resultados en el mapa y en la tabla.

El Código 2.32 muestra la creación del botones en la página web.

```
<button id='queryBtn'>Consultar</button>
<button id='captureBtn'>Capturar</button>
```

Código 2.32 Botones Consultar y Capturar del documento HTML

2.6.4.2 Programación del script mystyle

Java Script utiliza CSS para cambiar los estilos de los documentos HTML, CSS define los márgenes, los colores, la alineación del texto, etc. En este caso se definen los márgenes que tendrán el mapa y los campos de la tabla que se presentan en la página web.

En el Código 2.33 se presenta el *script* que define el estilo del documento HTML.

```

html, body, #map {
    height: 90%;
    width: 99%;
    margin: auto;
    padding: 0px;
}

#floating-panel {
    position: absolute;
    top: 10px;
    left: 30%;
    z-index: 5;
    background-color: #fff;
    padding: 5px;
    border: 1px solid #999;
    text-align: center;
    font-family: 'Roboto', 'sans-serif';
    line-height: 30px;
    padding-left: 10px;
}

a {
    font-size: 10px;
    text-indent: 100px;
}

```

Código 2.33 Script que define el estilo de la página web

Se debe indicar que este tipo de scripts vienen por defecto para todo documento HTML de una página web. Cada página web cuenta con un script CSS que define su estilo.

2.6.4.3 Programación del script myscript

Se definen las variables con las cuales se va a trabajar. Las variables `date1` y `date2` se utilizan para definir un intervalo de tiempo para seleccionar los puntos a ser representados. El array `puntos` se utiliza para cargar los datos a ser representados. La variable `timeMedia` se utiliza para promediar los puntos a ser representados.

El Código 2.34 muestra la creación de las variables: `date 1`, `date 2`, `puntos` y `timeMedia`.

```

var date1 = moment().format('YYYY/MM/DD');
var date2 = moment().format('YYYY/MM/DD');
var puntos = [];
var timeMedia = 3600;

```

Código 2.34 Declaración de variables

2.6.4.4 Programación del script mygoogle

En este *script* se realizan las funciones que tendrá la página web en donde se muestren los resultados tanto de forma gráfica como en la tabla.

Se definen las coordenadas de inicio en el mapa. Se ha seleccionado las coordenadas de la universidad Escuela Politécnica Nacional únicamente porque es el sitio de análisis en el próximo capítulo. Si el prototipo se encuentra en un sitio diferente a la universidad se cargara el mapa con este nuevo sitio.

En el Código 2.35 se crean las variables: map, mapCenter, desiredRadiusPerPointInMeters, TILE-SIZE.

```
var map, heatmap;
var mapCenter = {lat: -0.210023333, lng: -78.4900295};
var desiredRadiusPerPointInMeters = 5;
var TILE_SIZE = 256;
```

Código 2.35 Coordenadas de inicio para el mapa

En el Código 2.36 se declaran variables y se cargan las nuevas coordenadas y se retorna el nuevo mapa.

```
var origin = me.pixelOrigin_;
var lng = (point.x - origin.x) / me.pixelsPerLonDegree_;
var latRadians = (point.y - origin.y) / -me.pixelsPerLonRadian_;
var lat = radiansToDegrees(2 * Math.atan(Math.exp(latRadians)) - Math.PI / 2);

return new google.maps.LatLng(lat, lng);
```

Código 2.36 Se cargan las nuevas coordenadas

En el Código 2.37 se define la función del zoom de tal manera que no se distorsionen los puntos representados la vista del mapa se acerque o se aleje.

```
google.maps.event.addListener(map, 'zoom_changed',
function () {
heatmap.setOptions({radius:getNewRadius()});
```

Código 2.37 Función de zoom del mapa

2.6.4.5 Programación del script myquery

En este *script* se da la funcionalidad para seleccionar las fechas que definen un intervalo de tiempo en el cual se seleccionan los puntos a ser representados.

En el Código 2.38 se determina el ingreso de la fecha de inicio del intervalo, se define el formato a utilizar (año, mes, día) y se carga en la variable date 1.

```
$('#range_date_start').datetimepicker({
  dayOfWeekStart : 1,
  timepicker:false,
  format:'Y/m/d',
  startDate: date1,
  maxDate: 0,
  onChangeDateTime:function(dp,input){
    date1 = $('#range date start').val();
  }
});
```

Código 2.38 Ingreso de la variable date1

En el Código 2.39 se determina el ingreso de la fecha fin del intervalo, se define el formato a utilizar (año, mes, día) y se carga en la variable date 2.

```
$('#range_date_end').datetimepicker({
  dayOfWeekStart : 1,
  timepicker:false,
  format:'Y/m/d',
  startDate: date2,
  maxDate: 0,
  onChangeDateTime:function(dp,input){
    date2 = $('#range_date_end').val();
  }
});
```

Código 2.39 Ingreso de la variable date2

En el Código 2.40 se solicita el servidor web de acuerdo a intervalo de tiempo definido por las variables date1 y date2 los datos de: id, fecha, latitud, longitud y RMS. Para presentar en la tabla y ser ubicados representados en el mapa.

```
if (response.success) {
  var output = "<h1> Se han encontrado " + response.data.message + " puntos</h1>";

  output += '<table cellspacing="0" cellpadding="0">';
  output += '<tr>';
  output += '<th>Id</th>';
  output += '<th>Fecha</th>';
  output += '<th>Latitud</th>';
  output += '<th>Longitud</th>';
  output += '<th>RMS</th>';
  output += '</tr>';
}
```

Código 2.40 Solicitud de datos al servidor

2.6.4.6 Programación del script para obtener los puntos de la base de datos

Este *script* está en el lado del servidor, utiliza el lenguaje de programación PHP y tiene la finalidad de permitir la interacción con la base de datos y la página web.

En el Código 2.41 se selecciona la base de datos a la cual se pretende obtener la información mediante consultas. Se realiza la autenticación ingresando el usuario y contraseña.

```
$dbserver = '127.0.0.1';
$dbuser = 'sonometro';
$password = 'tesispass';
$dbname = 'sonometroDB';
```

Código 2.41 Selección y autenticación con la base de datos

En el Código 2.42 se solicita información de la base de datos mediante una consulta, se debe indicar que la consulta se usa la sintaxis de MySQL que es el motor de la base de datos que se está utilizando. La información que se solicita es: longitud, latitud y de acuerdo a esto se presentan los puntos, el intervalo de tiempo para el cual se solicita esta información está definido por las variables date1 y date2.

```
$sql="SELECT * FROM data
WHERE `fecha` >= '". $date1.'" 00:00:00'
AND `fecha` <= '". $date2.'" 23:59:59'
AND `latitud` <> 0
AND `longitud` <> 0";
```

Código 2.42 Consulta a la base de datos

En el Código 2.43 se muestra que si los datos solicitados son encontrados son cargados en un array y se muestra un mensaje de lo ocurrido caso contrario se presenta un mensaje indicando que no se ha encontrado ningún resultado.

```
$jsondata["success"] = true;

$jsondata["data"]["message"] = $result->num_rows;
$jsondata["data"]["puntos"] = array();
while( $row = $result->fetch_object() ) {
    $jsondata["data"]["puntos"][] = $row;
}
else {
    $jsondata["success"] = false;
    $jsondata["data"] = array(
        'message' => 'No se encontró ningún resultado.'
```

Código 2.43 Cargar los puntos solicitados en un array

2.6.4.7 Programación del script para definir el estado de la base de datos.

Este *script* tiene como finalidad obtener el estado de la base de datos y así realizar acciones sobre la misma.

En el Código 2.44 se selecciona la base de datos a la cual se pretende obtener la información mediante consultas. Se realiza la autenticación ingresando el usuario y contraseña.

```
$dbserver = '127.0.0.1';
$dbuser = 'sonometro';
$password = 'tesispass';
$dbname = 'sonometroDB';
```

Código 2.44 Selección y autenticación con la base de datos

En el Código 2.45 se realiza la consulta a la base de datos que permite la actualización de la misma y se define el tiempo de escucha de acuerdo al intervalo de tiempo seleccionado.

```
$sql = "UPDATE `estado` SET `tiempo`=".$tiempo." WHERE `id` = 1";

if ( $result = $database->query( $sql) ) {
    $jsondata["success"] = true;
    $jsondata["data"] = array(
        'message' => 'Se establecio el tiempo de escucha.'
    );
    //$result->close();
} else {
    $jsondata["success"] = false;
    $jsondata["data"] = array(
        'message' => $database->error
    );
}
```

Código 2.45 Consulta de actualización a la base de datos

CAPÍTULO 3: PRUEBAS DE FUNCIONAMIENTO Y MEDICIONES

En este capítulo se presenta: el proceso de calibración de los sensores de ruido, las pruebas de funcionalidad del sistema prototipo, las mediciones de acuerdo al cronograma planteado en la metodología, el análisis de resultados y finalmente un costo referencial del sistema prototipo.

3.1 CALIBRACIÓN DEL SENSOR DE RUIDO

Para realizar la calibración de los micrófonos se realiza el siguiente procedimiento:

1.- Se utiliza como fuente generadora de ruido a los parlantes PC SP-S110 [40], los cuales pueden emitir sonidos a una potencia de 1W. En la figura 3.1 se muestran los parlantes SP-S110.



Figura 3.1 Parlantes de computadora SP-S110 [40]

2.- Los parlantes se conectan a la computadora y se reproduce un audio a máxima potencia.

3.- El control de volumen de los parlantes es colocado a máxima potencia, en este caso sería 1 W.

4.- Se determina el valor teórico del nivel de ruido, el cual se esperaba obtener en las mediciones, para ello se utilizan los valores característicos que tienen los dispositivos electrónicos involucrados en la calibración.

En la figura 3.2 se muestra el diagrama de bloques de los elementos que intervienen en la calibración.

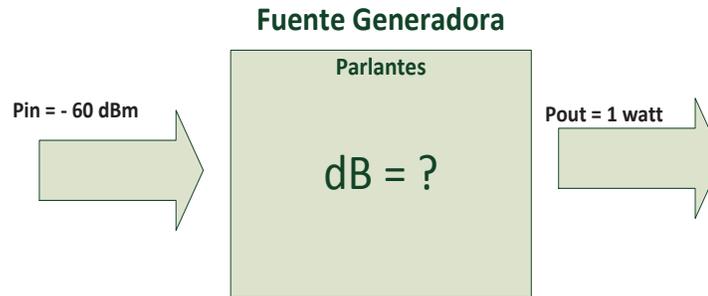


Figura 3.2 Diagrama de bloques de la fuente generadora de ruido

La potencia de entrada $P_{in} = - 60 \text{ dBm}$, es la potencia que ofrece la tarjeta de sonido 92HD81 [41].

La potencia de salida $P_{out} = 1 \text{ watt}$, es la potencia máxima que ofrecen los parlantes SP – S-110 [40].

A continuación se calcula el nivel de ruido teórico que ofrecen los parlantes.

En la ecuación 3.1 se calcula la potencia de entrada en watt.

$$dBm = 10 \log \frac{P_{in}}{1mw}$$

Ecuación 3.1. Cálculo de potencia P_{in}

$$-60 = 10 \log \frac{P_{in}}{1mw}$$

$$P_{in} = 1 \times 10^{-9} \text{ w}$$

En la ecuación 3.2 se calcula la intensidad de sonido en dB.

$$dB = 10 \log \frac{P_{out}}{P_{in}}$$

Ecuación 3.2. Cálculo de la intensidad de ruido teórico

$$dB = 10 \log \frac{1 \text{ w}}{1 \times 10^{-9} \text{ w}}$$

$$dB = 90$$

El nivel de intensidad del ruido calculado teóricamente es de 90 dB, este valor es ideal puesto que los valores utilizados en el cálculo son valores pico.

El valor de 90 dB se toma como referencia para realizar la calibración. El valor que resulte de la medición deberá ser cercano a 90 dB pero no lo podrá superar, puesto que este valor es ideal.

5.- Se coloca el sistema prototipo cerca de los parlantes que están transmitiendo a máxima potencia para realizar las mediciones con el sistema prototipo y comparar los valores obtenidos con el valor teórico.

En la figura 3.3 se muestra el sistema prototipo junto a los parlantes SP-S110.

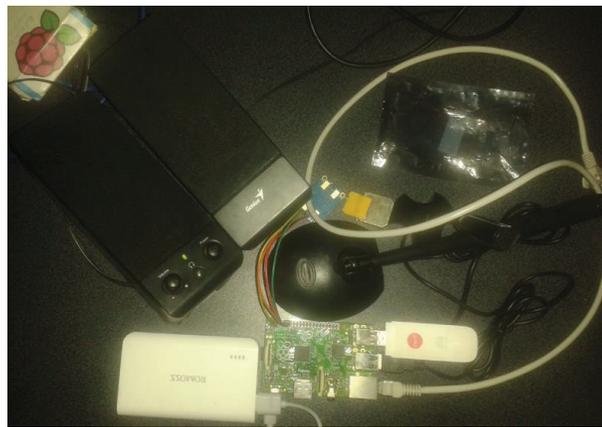


Figura 3.3 Sistema prototipo tomando lecturas de ruido cerca de los parlantes SP-S110

En la tabla 3.1 se presentan las mediciones de ruido del sistema prototipo y el valor teórico de referencia.

Valor Referencia (dB)	Valor Obtenido (dB)
90	19
90	21
90	22
90	20
90	25
90	17
90	21

Tabla 3.1. Comparación de mediciones previo a la calibración

En la figura 3.4 se muestra la comparación de mediciones entre el valor de referencia y los valores obtenidos.

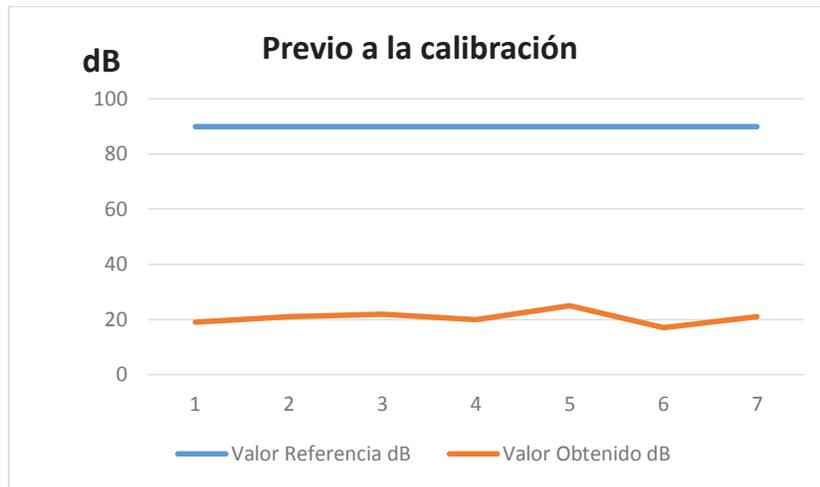


Figura 3.4 Comparación de mediciones entre el valor de referencia y los valores obtenidos, previo a la calibración

Como se puede observar en la figura 3.4, los valores obtenidos por el sistema prototipo están por debajo del valor esperado. Lo cual indica que es necesario realizar la calibración que permita obtener valores cercanos al esperado.

6.- Se realiza el ajuste en el sistema prototipo y se comprueba que las lecturas estén en el rango esperado.

En la tabla 3.2 se muestran las mediciones de ruido del sistema prototipo y el valor teórico de referencia una vez realizada la calibración.

Valor Referencia (dB)	Valor Obtenido (dB)
90	78
90	80
90	81
90	76
90	82
90	81
90	74

Tabla 3.2 Comparación de mediciones luego de la calibración

En la figura 3.5 se muestra la comparación de mediciones entre el valor de referencia y los valores obtenidos una vez realizada la calibración.

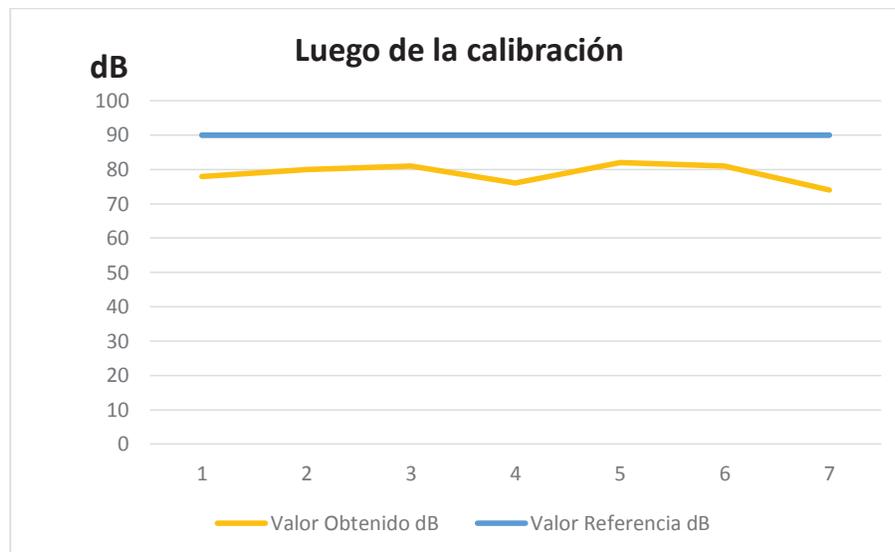


Figura 3.5 Comparación de mediciones entre el valor de referencia y los valores obtenidos luego de la calibración

Como se observa en la figura 3.5 los valores de ruido obtenidos por el sistema prototipo están próximos al valor esperado. Al ser 90 dB un valor ideal sucede que el valor obtenido solo podrá aproximarse a dicho valor, no puede ser igual o superior, porque el valor ideal se obtuvo con valores pico.

La variación en los valores obtenidos por el sistema prototipo se debe a que el audio en reproducción no tiene un ritmo constante. Además, se debe indicar que los micrófonos utilizados presentan una sensibilidad de 4 dB lo que incide al momento de realizar las mediciones.

3.2 PRUEBAS DE FUNCIONALIDAD

Las pruebas de funcionalidad del sistema prototipo se dividen en las siguientes fases:

- Fase de adquisición de datos.
- Fase de envío de datos.
- Fase de recepción y almacenamiento de datos.
- Fase de visualización y presentación de resultados.

3.2.1 FASE DE ADQUISICIÓN DE DATOS

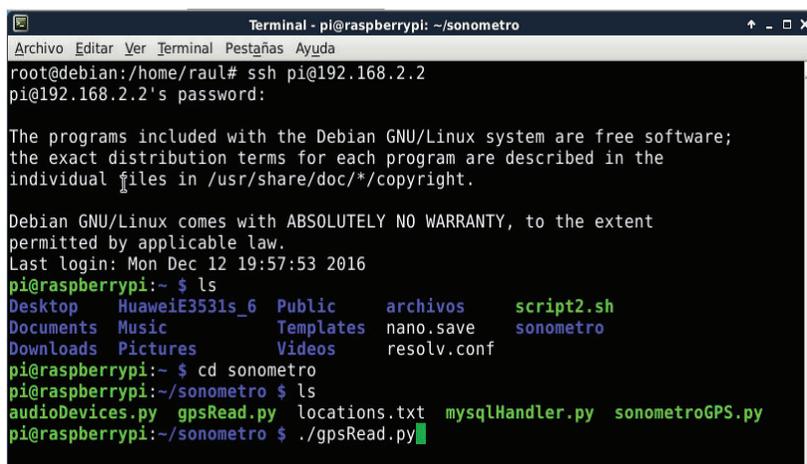
La fase de adquisición de datos comprueba el proceso de recolección de información acerca de: geolocalización mediante el modulo GPS y niveles de ruido en el ambiente a través de los micrófonos USB.

3.2.1.1 Fase de adquisición de datos de geolocalización

Para comprobar que el proceso de adquisición de datos de geolocalización sea correcto se utiliza el programa `gpsRead.py`. El código del programa `gpsRead.py` se encuentra disponible en la sección ANEXOS.

Se ingresa al terminal de la Raspberry Pi, se ingresa al directorio sonómetro y finalmente se ejecuta el programa `gpsRead.py`.

En la figura 3.6 se presenta la ejecución del programa `gpsRead.py` en la placa Raspberry Pi.



```

Terminal - pi@raspberrypi: ~/sonometro
Archivo Editar Ver Terminal Pestañas Ayuda
root@debian:/home/raul# ssh pi@192.168.2.2
pi@192.168.2.2's password:

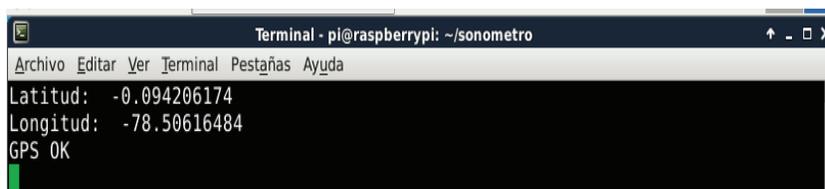
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 12 19:57:53 2016
pi@raspberrypi:~$ ls
Desktop  HuaweiE3531s_6  Public      archivos      script2.sh
Documents Music            Templates  nano.save     sonometro
Downloads Pictures         Videos     resolv.conf
pi@raspberrypi:~$ cd sonometro
pi@raspberrypi:~/sonometro$ ls
audioDevices.py  gpsRead.py  locations.txt  mysqlHandler.py  sonometroGPS.py
pi@raspberrypi:~/sonometro$ ./gpsRead.py

```

Figura 3.6 Ejecución del programa `gpsRead.py`

En la figura 3.7 se presentan los resultados de la ejecución del programa `gpsRead.py`.



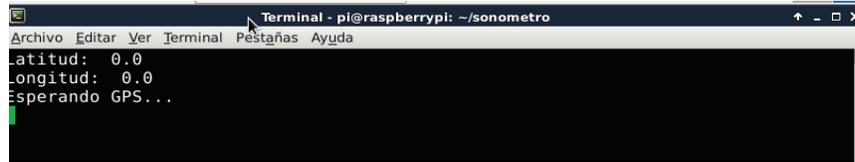
```

Terminal - pi@raspberrypi: ~/sonometro
Archivo Editar Ver Terminal Pestañas Ayuda
Latitud: -0.094206174
Longitud: -78.50616484
GPS OK

```

Figura 3.7 Presentación de resultados de latitud y longitud

En la figura 3.8 se presentan los resultados cuando el modulo GPS ha logrado engancharse al sistema GPS pero no tiene ningún resultado para mostrar, el módulo GPS sigue esperando en la conexión hasta tener resultados que mostrar.



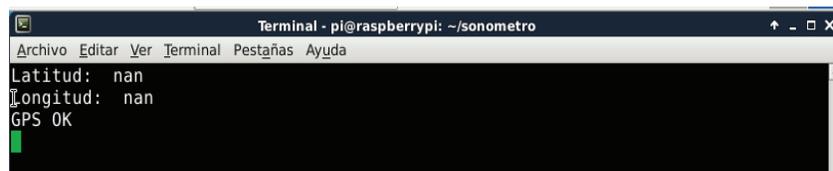
```

Terminal - pi@raspberrypi: ~/sonometro
Archivo Editar Ver Terminal Pestañas Ayuda
Latitud: 0.0
Longitud: 0.0
Esperando GPS...

```

Figura 3.8 Presentación de resultados en cero del módulo GPS

En la figura 3.9 se muestra cuando el modulo GPS no ha logrado engancharse al sistema GPS.



```

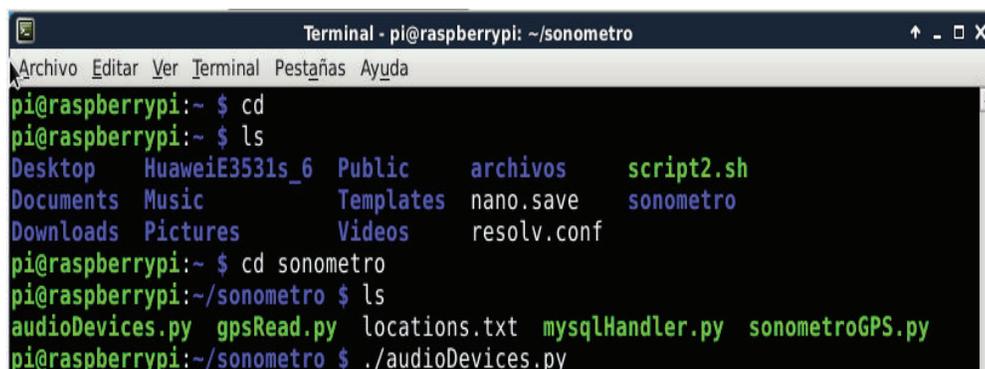
Terminal - pi@raspberrypi: ~/sonometro
Archivo Editar Ver Terminal Pestañas Ayuda
Latitud: nan
Longitud: nan
GPS OK

```

Figura 3.9 Presentación de mensaje sin resultados de latitud y longitud

Para comprobar el correcto funcionamiento de la fase de adquisición de los niveles de ruido presente en el ambiente se utiliza el programa `audioDevices.py`, desarrollado con la finalidad del reconocimiento de los dispositivos de audio conectados a la SBC. El código del programa se encuentra disponible en la sección ANEXOS. Para ejecutar `audioDevices.py` se ingresa en el terminal de la SBC y se dirige al directorio `sonómetro` y se ejecuta el programa.

En la figura 3.10 se presenta la ejecución del programa `audioDevices.py`.



```

Terminal - pi@raspberrypi: ~/sonometro
Archivo Editar Ver Terminal Pestañas Ayuda
pi@raspberrypi:~ $ cd
pi@raspberrypi:~ $ ls
Desktop  HuaweiE3531s_6  Public      archivos      script2.sh
Documents Music            Templates  nano.save     sonometro
Downloads Pictures         Videos     resolv.conf
pi@raspberrypi:~ $ cd sonometro
pi@raspberrypi:~/sonometro $ ls
audioDevices.py  gpsRead.py  locations.txt  mysqlHandler.py  sonometroGPS.py
pi@raspberrypi:~/sonometro $ ./audioDevices.py

```

Figura 3.10 Ejecución del programa `audioDevices.py`

En la figura 3.11 se presenta el reconocimiento de los dispositivos de audio conectados a la placa SBC.

```
(2, u'USB PnP Sound Device: Audio (hw:1,0)', 1L)
(3, u'USB PnP Sound Device: Audio (hw:2,0)', 1L)
```

Figura 3.11 Reconocimiento de los dispositivos de audio en la SBC

El listado de los dispositivos USB conectados a la placa Raspberry Pi se puede realizar a través del comando `lsusb`, de esta manera también se puede distinguir a los micrófonos conectados en la SBC.

En la figura 3.12 se presenta la lista de dispositivos USB conectados a la SBC.

```
pi@raspberrypi:~/sonometro $ lsusb
Bus 001 Device 005: ID 0d8c:013c C-Media Electronics, Inc. CM108 Audio Controller
Bus 001 Device 007: ID 12d1:1506 Huawei Technologies Co., Ltd. E398 LTE/UMTS/GSM Modem/Networkcard
Bus 001 Device 004: ID 0d8c:013c C-Media Electronics, Inc. CM108 Audio Controller
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Figura 3.12 Dispositivos conectados a la SBC

3.2.2 FASE DE ENVIÓ DE DATOS

El envío de datos desde el subsistema transmisor hacia el subsistema receptor se realiza a través del modem 3G Huawei E3531 [25]. El modem envía los datos al servidor a través del internet, por tal razón la comprobación del envío de datos empieza por la verificación de la conexión del modem a internet.

El modem 3G Huawei E3531 [25] posee un led de color azul, el cual indica el estado del dispositivo. Si el encendido del led es intermitente indica que el modem no se encuentra conectado a la red celular, si el encendido del led es constante indica que el modem está conectado a la red celular.

El control de la conexión del modem a la red celular se realiza a través del *script* Mobile Broadband Connect, este *script* permite realizar una conexión automática del modem a la red celular. Si por alguna razón la conexión del modem se interrumpe el *script* realiza la reconexión con la red celular.

Mobile Broadband Connect se ejecuta como demonio para que arranque cada vez que se encienda la SBC y finalmente en el editor de conexiones se selecciona y configura la conexión Mobile Broadband Connect.

En la figura 3.13 se muestra la configuración del *Acces Point Name*⁴³ (APN) de la conexión Mobile Broadband Connect.

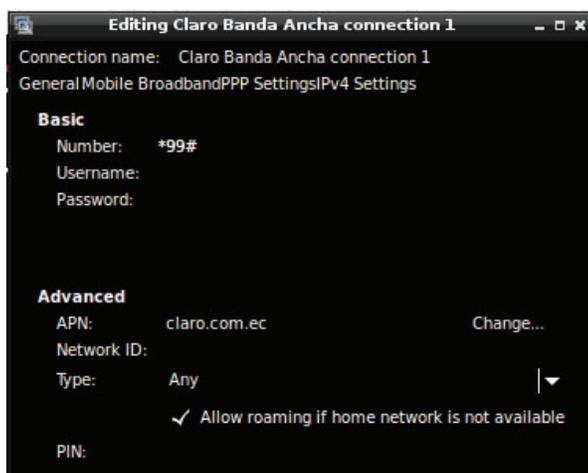


Figura 3.13 Configuración de Mobile Broadband Connect

En la figura 3.14 se muestra la configuración para que el modem se conecte automáticamente cuando la red este activa y todos los usuarios puedan conectarse sin restricción.

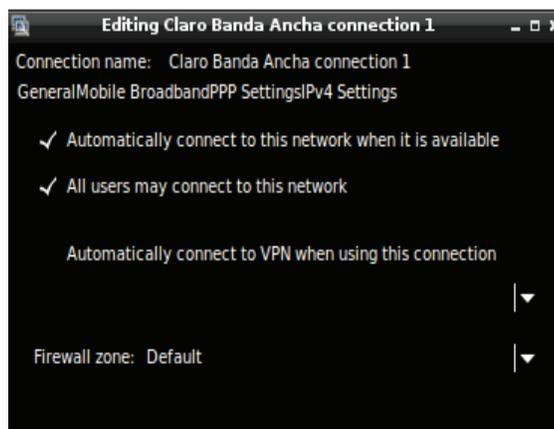


Figura 3.14 Configuración general de Mobile Broadband Connect

⁴³ Acces Point Name: Es la configuración que permite al dispositivo móvil navegar en internet a través de una red celular.

En la figura 3.15 se muestra la configuración del *Domain Name System*⁴⁴ (DNS) para que el modem pueda salir a internet. Se ingresa la dirección IP 8.8.8.8 en DNS server.

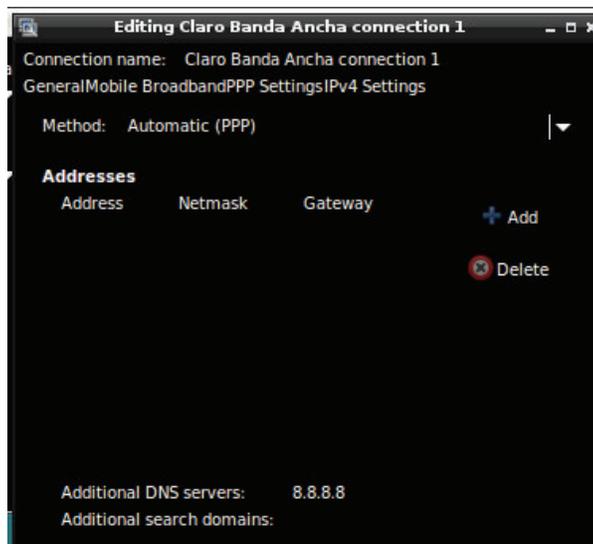


Figura 3.15 Configuración de DNS de Mobile Broadband Connect

Una vez realizado la configuración, en la parte inferior derecha de la pantalla del escritorio de la Raspberry Pi se podrá observar el icono correspondiente indicando que la conexión del modem 3G está activa, como se indica en la figura 3.16.

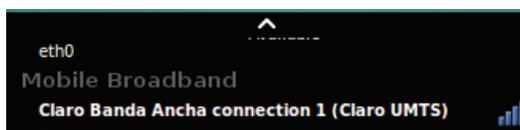


Figura 3.16 Conexión activa de Mobile Broadband Connect

En la figura 3.17 se muestra la realización de un ping desde la Raspberry Pi al servidor de Google para comprobar la conexión a internet a través del modem 3G.

```
pi@raspberrypi:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=779 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=499 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=429 ms
```

Figura 3.17 Ping al servidor de Google

⁴⁴ DNS: Es un sistema de nomenclatura jerárquico descentralizado para dispositivos conectados a redes IP.

También se verifica la conexión a internet haciendo ping al dominio de Google. En la figura 3.18 se muestra el ping desde la SBC al dominio `www.google.com`.

```

pi@raspberrypi:~$ ping www.google.com
PING www.google.com (74.125.141.106) 56(84) bytes of data:
64 bytes from vl-in-f106.1e100.net (74.125.141.106): icmp_seq=1 ttl=42 time=606
ms
64 bytes from vl-in-f106.1e100.net (74.125.141.106): icmp_seq=2 ttl=42 time=438
ms
64 bytes from vl-in-f106.1e100.net (74.125.141.106): icmp_seq=3 ttl=42 time=489
ms
64 bytes from vl-in-f106.1e100.net (74.125.141.106): icmp_seq=4 ttl=42 time=438
ms

```

Figura 3.18 Ping al dominio `www.google.com`

3.2.3 FASE DE RECEPCIÓN Y ALMACENAMIENTO DE DATOS

Para comprobar que la recepción y almacenamiento de datos en el subsistema receptor es correcta, se realizan dos procedimientos, uno en el subsistema transmisor y el otro en el subsistema receptor.

3.2.3.1 Procedimiento en el subsistema transmisor

Para el almacenamiento de información es necesario instalar un motor de base de datos en el subsistema receptor y en el subsistema transmisor la instalación del cliente de la base de datos.

Para comprobar que el envío de información es correcta, se ingresa al terminal de la SBC y se ejecuta el *script* `mysqlHandler.py`. Este *script* tiene la finalidad de establecer la conexión con el servidor, enviar los datos tomados por el subsistema transmisor y cerrar la conexión. El código de este *script* está disponible en la sección de ANEXOS.

En la figura 3.19 se muestra la ejecución del *script* `mysqlHandler.py`, el establecimiento de la conexión, el envío de información y el cierre de la conexión con el servidor.

```

pi@raspberrypi:~/sonometro $ ./mysqlHandler.py
Conexion SSH abierta
(1L, datetime.datetime(2016, 8, 4, 7, 27, 10), -0.119083333, -78.4815495, 5.9184
065174)
(3L, datetime.datetime(2016, 8, 4, 17, 0, 29), -0.119168333, -78.4816275, 19.025
163898)
(4L, datetime.datetime(2016, 8, 14, 18, 53, 52), -0.118961167, -78.481763167, 20
.100266023)
Conexion SSH cerrada
FINISH!

```

Figura 3.19 Ejecución del *script* `mysqlHandler.py`

3.2.3.2 Procedimiento en el subsistema receptor.

Para verificar que la información ha llegado y almacenado correctamente en el servidor, se ingresa al motor de base de datos MySQL.

El ingreso al motor de la base de datos se realiza de la siguiente manera:

- Abrir un navegador web.
- Ingresar la dirección Ip y el puerto establecidos previamente.

Se presenta una pantalla solicitando el usuario y contraseña, se ingresan los datos solicitados y se accede a MySQL.

En la figura 3.20 se muestra el ingreso al motor de la base de datos MySQL.

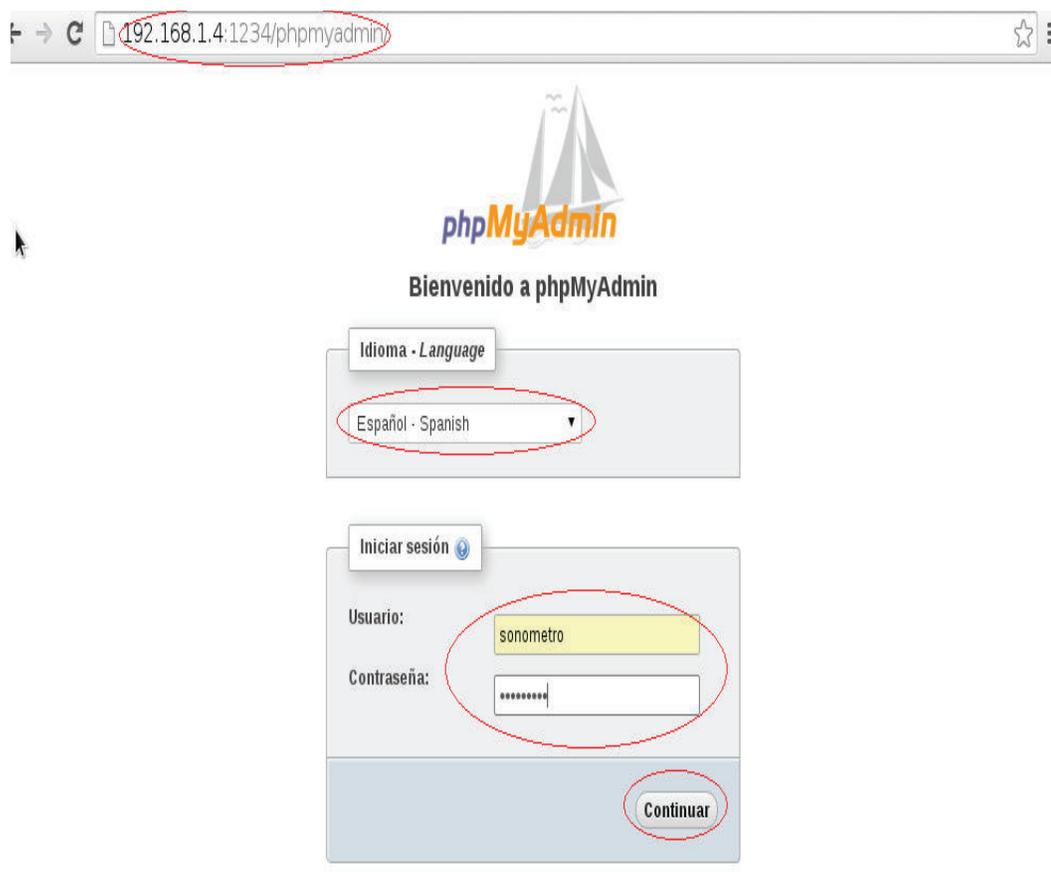


Figura 3.20 Ingreso a MySQL Admin

Una vez ingresado a la base de datos sonómetro, se dirige a la tabla data y se comprueba que los valores almacenados sean los mismos que envió el subsistema transmisor.

En la figura 3.21 se muestra la tabla que almacena los datos enviados por el subsistema transmisor.

Mostrando filas 0 - 24 (total de 100, La consulta tardó 0.0004 segundos)

SELECT * FROM `data`

Mostrar todo >>> Número de filas: 25 Filtrar filas: Buscar en esta tabla

Ordenar según la clave: Ninguna

	id	fecha	latitud	longitud	medicion
<input type="checkbox"/>	1	2016-08-04 07:27:10	-0.119083333	-78.4815495	5.9184065174
<input type="checkbox"/>	3	2016-08-04 17:00:29	-0.119168333	-78.4816275	19.025163898
<input type="checkbox"/>	4	2016-08-14 18:53:52	-0.118961167	-78.481763167	20.100266023
<input type="checkbox"/>	5	2016-08-14 18:54:04	-0.119004833	-78.481784167	16.455777802
<input type="checkbox"/>	6	2016-08-14 18:54:17	-0.119082333	-78.481814667	17.1291143952
<input type="checkbox"/>	7	2016-08-14 18:54:27	-0.119060833	-78.481768667	19.2586085022
<input type="checkbox"/>	8	2016-08-14 18:54:37	-0.119025667	-78.481757333	15.3324932084

Figura 3.21 Almacenamiento de los datos enviados por el subsistema transmisor

En la figura 3.22 se muestra la comparación de los datos enviados por el subsistema transmisor con los datos almacenados en el subsistema receptor.

```

pi@raspberrypi:~/sonometro $ ./mysqlHandler.py
Conexion SSH abierta
(1L, datetime.datetime(2016, 8, 4, 7, 27, 10), -0.119083333, -78.4815495, 5.9184065174)
(3L, datetime.datetime(2016, 8, 4, 17, 0, 29), -0.119168333, -78.4816275, 19.025163898)
(4L, datetime.datetime(2016, 8, 14, 18, 53, 52), -0.118961167, -78.481763167, 20.100266023)
(5L, datetime.datetime(2016, 8, 14, 18, 54, 4), -0.119004833, -78.481784167, 16.455777802)

```

	id	fecha	latitud	longitud	medicion
<input type="checkbox"/>	1	2016-08-04 07:27:10	-0.119083333	-78.4815495	5.9184065174
<input type="checkbox"/>	3	2016-08-04 17:00:29	-0.119168333	-78.4816275	19.025163898
<input type="checkbox"/>	4	2016-08-14 18:53:52	-0.118961167	-78.481763167	20.100266023
<input type="checkbox"/>	5	2016-08-14 18:54:04	-0.119004833	-78.481784167	16.455777802

Figura 3.22 Comparación de datos enviados y recibidos en el sistema prototipo

En la figura 3.22 se puede verificar que los datos enviados por el subsistema transmisor son los mismos que se encuentran almacenados en el subsistema receptor.

3.2.4 FASE DE PRESENTACIÓN Y VISUALIZACIÓN DE RESULTADOS

La visualización de resultados se realiza a través de una tabla con los datos recolectados por el subsistema transmisor, también se presenta un mapa donde se indica el lugar donde se realizó la medición y mediante un mapa de calor se indica la intensidad de ruido presente en el ambiente. La tabla de resultados y el mapa de calor se presentan en una misma página web para facilidad de visualización del usuario.

Para comprobar que la visualización de resultados es correcta, se ingresa a un navegador web y se ingresa la dirección del servidor web.

En la figura 3.23 se muestra la interfaz del servidor web para la presentación de resultados.

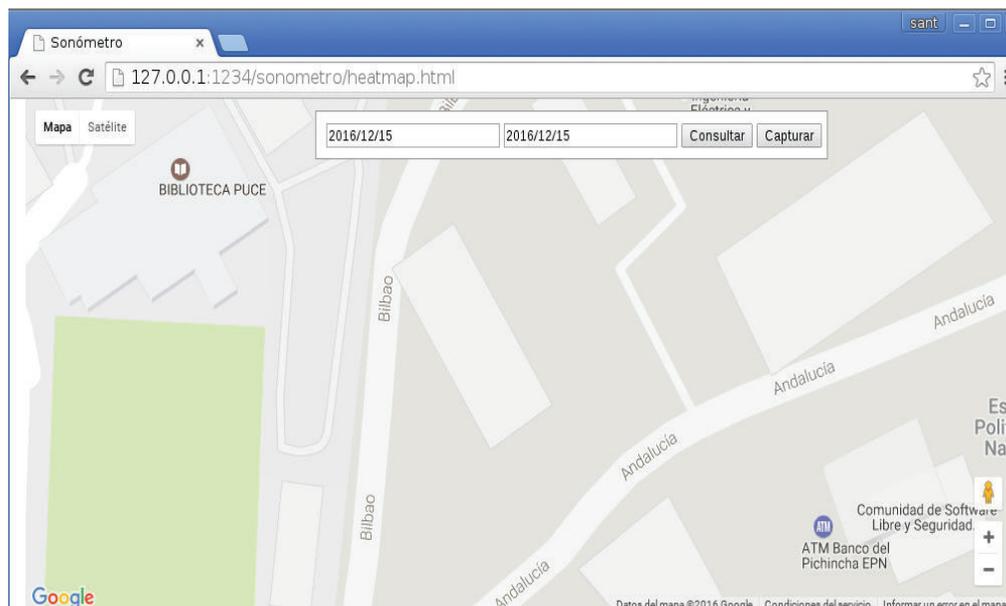
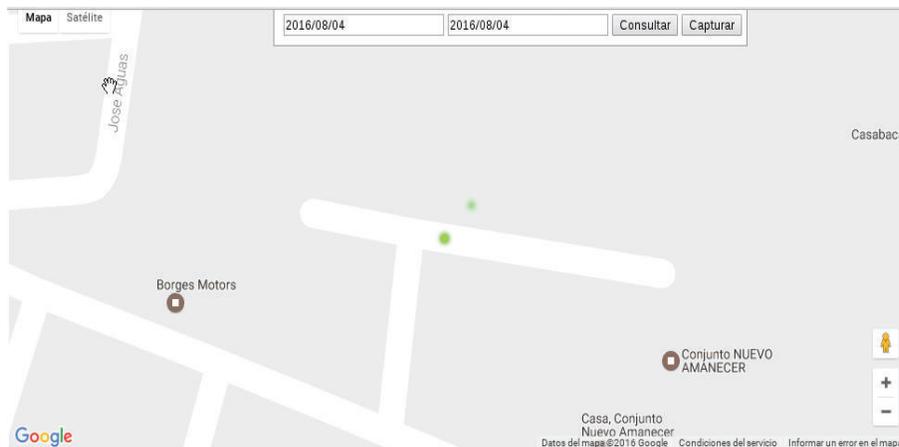


Figura 3.23 Interfaz del servidor web

En la figura 3.23 muestra la interfaz de inicio de la página web, el mapa de inicio pertenece a las coordenadas de la Escuela Politécnica Nacional.

La interfaz muestra dos calendarios que permiten seleccionar la fecha de inicio y la fecha final de análisis. La fecha final puede ser igual a la fecha de inicio mas no puede ser anterior a dicha fecha, esto se controla desactivando la selección de fechas anteriores a la fecha de inicio.

Para visualizar los resultados almacenados se selecciona el intervalo de tiempo requerido. En la figura 3.24 se muestran los resultados correspondientes al día 4 de Agosto del 2016.



Se han encontrado 2 puntos

Id	Fecha	Latitud	Longitud	RMS
1	2016-08-04 07:27:10	-0.119083333	-78.4815495	5.9184065174
3	2016-08-04 17:00:29	-0.119168333	-78.4816275	19.025163898

Figura 3.24 Selección del intervalo de tiempo de análisis

En la figura 3.25 se muestra la desactivación de fechas anteriores a la fecha de inicio.

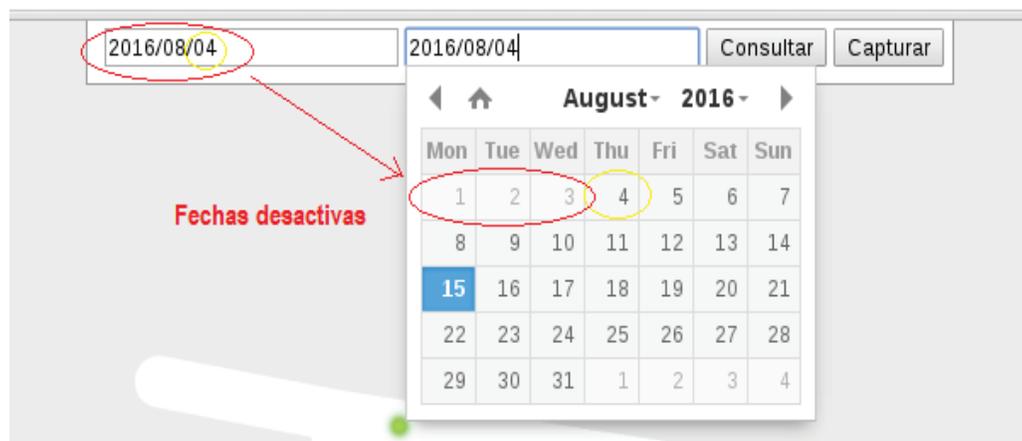


Figura 3.25 Desactivación de fechas anteriores a la fecha de inicio

Una vez seleccionado el intervalo de tiempo de análisis, se presentan las mediciones realizadas, correspondientes a dicho intervalo.

En la figura 3.26 se muestran las mediciones realizadas en una tabla donde se indican los valores de: fecha, latitud, longitud y ruido, correspondientes al intervalo de tiempo seleccionado.

Se han encontrado 2 puntos

Id	Fecha	Latitud	Longitud	RMS
1	2016-08-04 07:27:10	-0.119083333	-78.4815495	5.9184065174
3	2016-08-04 17:00:29	-0.119168333	-78.4816275	19.025163898

Figura 3.26 Presentación de mediciones de acuerdo al intervalo de tiempo seleccionado

En la figura 3.27 se muestra el mapa donde se indica el lugar donde se realizaron las mediciones y mediante un mapa de calor se indica la intensidad del ruido.



Se han encontrado 2 puntos

Id	Fecha	Latitud	Longitud	RMS
1	2016-08-04 07:27:10	-0.119083333	-78.4815495	5.9184065174
3	2016-08-04 17:00:29	-0.119168333	-78.4816275	19.025163898

Figura 3.27 Presentación de resultados en el mapa mediante un mapa de calor

3.3 PRUEBAS DE FUNCIONAMIENTO DEL SISTEMA PROTOTIPO

Para realizar las pruebas de funcionamiento del sistema prototipo es necesario elegir un escenario que presente características similares en las cuales va trabajar normalmente el sistema prototipo.

3.3.1 DESCRIPCIÓN DEL ESCENARIO DE PRUEBAS

El escenario de pruebas es una habitación que tiene las siguientes características:

- Área de 48 metros cuadrados.
- Tres paredes de concreto.
- Una pared de ventanales.
- Techo de loza.

3.3.2 PRUEBAS DE MEDICIÓN DE RUIDO

Para comprobar que las mediciones de ruido realizadas con el sistema prototipo son acertadas se utiliza la aplicación Sound Meter [1], esta aplicación permite medir los niveles de ruido presente en el ambiente a través de dispositivos móviles.

A continuación se presenta la comparación de las medidas tomadas por el sistema prototipo y la aplicación Sound Meter.

En la figura 3.28 se presentan las mediciones tomadas por el sistema prototipo.

id	fecha	latitud	longitud	medicion
102	2016-12-12 08:14:35	-0.094143653	-78.5061982	42.987987657
103	2016-12-12 08:36:46	-0.094143634	-78.5061986	36.349348628
104	2016-12-12 08:36:58	-0.094143655	-78.5061989	39.829283849
105	2016-12-12 08:37:13	-0.094143633	-78.5061982	44.349348628
106	2016-12-12 08:37:58	-0.094143633	-78.5061982	41.349348628
107	2016-12-12 08:42:23	-0.094143631	-78.5061933	45.988345327
108	2016-12-12 08:42:23	-0.094143664	-78.5061931	49.721049877
109	2016-12-12 08:45:28	-0.094143639	-78.5061981	43.989876564
110	2016-12-12 08:47:31	-0.094143641	-78.5061988	46.938495994

Figura 3.28 Datos recolectados en la etapa de pruebas

En la figura 3.29 se presentan las mediciones tomadas por la aplicación Sound Meter.



Figura 3.29 Medición realizada con la aplicación Sound Meter [1]

Como se puede observar en la figura 3.24 y 3.23, la aplicación del dispositivo móvil realiza la lectura con 1 dB menor que lectura tomada por sistema prototipo. En la tabla 3.3 se presentan las mediciones tomadas por el dispositivo móvil.

Fecha y Hora	Medición (dB)
2016-12-12 8:14:11	41
2016-12-12 8:36:05	34
2016-12-12 8:36:21	41
2016-12-12 8:37:34	44
2016-12-12 8:37:48	39
2016-12-12 8:42:55	44
2016-12-12 8:42:12	48
2016-12-12 8:45:23	44
2016-12-12 8:47:17	47

Tabla 3.3 Lecturas tomadas por la aplicación móvil Sound Meter

En la figura 3.20 se realiza la comparación de las lecturas de ruido tomadas por el sistema prototipo y la aplicación del dispositivo móvil.

SOUND METER		SISTEMA PROTOTIPO				
Fecha y Hora	Medición	id	fecha	latitud	longitud	medicion
2016-12-12 8:14:11	41	102	2016-12-12 08:14:35	-0.094143653	-78.5061982	42.987987657
2016-12-12 8:36:05	34	103	2016-12-12 08:36:46	-0.094143634	-78.5061986	36.349348628
2016-12-12 8:36:21	41	104	2016-12-12 08:36:58	-0.094143655	-78.5061989	39.829283849
2016-12-12 8:37:34	44	105	2016-12-12 08:37:13	-0.094143633	-78.5061982	44.349348628
2016-12-12 8:37:48	39	106	2016-12-12 08:37:58	-0.094143633	-78.5061982	41.349348628
2016-12-12 8:42:55	44	107	2016-12-12 08:42:23	-0.094143631	-78.5061933	45.988345327
2016-12-12 8:42:12	48	108	2016-12-12 08:42:23	-0.094143664	-78.5061931	49.721049877
2016-12-12 8:45:23	44	109	2016-12-12 08:45:28	-0.094143639	-78.5061981	43.989876564
2016-12-12 8:47:17	47	110	2016-12-12 08:47:31	-0.094143641	-78.5061988	46.938495994

Figura 3.30 Comparación de las lecturas tomadas por el sistema prototipo y la aplicación Sound Meter

En la figura 3.30 se puede observar que las lecturas tomadas por la aplicación Sound Meter son similares a las tomadas por el sistema prototipo, la mayoría de lecturas difieren en 1 dB, llegando a ser 2 dB la diferencia máxima. La diferencia presentada entre las lecturas está dentro de la sensibilidad que presentan los micrófonos que usa el sistema prototipo. Por tal razón se puede concluir que las lecturas tomadas por el sistema prototipo son acertadas.

3.4 MEDICIONES DE RUIDO

Las mediciones de ruido con el sistema prototipo se realizaron de acuerdo al cronograma planteado en la metodología en el plan de trabajo.

3.4.1 MEDICIONES EN MEDIO DE LOS EDIFICIOS QUÍMICA – ELECTRÓNICA Y ELÉCTRICA

A continuación se presentan las mediciones de ruido realizadas en medio de los edificios Química – Electrónica y Eléctrica.

3.4.1.1 Sesión 1

La primera sesión para las mediciones de ruido en este sitio fue, el día miércoles 14 de diciembre del 2016. La sesión tiene una duración de 2 horas.

En la tabla 3.4 se muestran los datos obtenidos en la sesión 1.

Fecha y Hora	Latitud (S^0)	Longitud (W^0)	Ruido (dB)
2016-12-14 13:02:13	-0.209356753	-78.4895501	56
2016-12-14 13:07:11	-0.209507534	-78.4895371	49
2016-12-14 13:12:03	-0.209267347	-78.4895635	57
2016-12-14 13:17:05	-0.209341676	-78.4895135	48
2016-12-14 13:22:07	-0.209342123	-78.4895835	55
2016-12-14 13:27:02	-0.209341087	-78.4895501	49
2016-12-14 13:32:06	-0.209341876	-78.4894663	50
2016-12-14 13:37:10	-0.209341236	-78.4896105	50
2016-12-14 13:42:08	-0.209341745	-78.4896233	53
2016-12-14 13:47:10	-0.209330724	-78.4896432	44
2016-12-14 13:52:09	-0.209330123	-78.4896654	53
2016-12-14 13:57:11	-0.209330989	-78.4896432	47
2016-12-14 14:02:03	-0.209330213	-78.4896878	51
2016-12-14 14:07:07	-0.209345903	-78.4896541	53
2016-12-14 14:12:04	-0.209346237	-78.4896589	54
2016-12-14 14:17:06	-0.209341246	-78.4895959	54
2016-12-14 14:22:11	-0.209361234	-78.4895932	52
2016-12-14 14:27:09	-0.209361352	-78.4895945	44
2016-12-14 14:32:02	-0.209361269	-78.4895912	49
2016-12-14 14:37:03	-0.209361764	-78.4895976	54
2016-12-14 14:42:08	-0.209385213	-78.4895943	42
2016-12-14 14:47:06	-0.209385243	-78.4895425	45
2016-12-14 14:52:05	-0.209385342	-78.4895422	44
2016-12-14 14:57:01	-0.209385648	-78.4895433	51

Tabla 3.4 Mediciones en medio de los edificios Electrónica – Química y Eléctrica

3.4.1.2 Sesión 2

La segunda sesión para las mediciones de ruido en este sitio fue, el día jueves 5 de enero del 2017. La sesión tiene una duración de 2 horas.

En la tabla 3.5 se muestran los datos obtenidos en la sesión 2.

Fecha y Hora	Latitud (S^0)	Longitud (W^0)	Ruido (dB)
2017-01-05 11:00:02	-0.209323349	-78.4894853	53
2017-01-05 11:05:04	-0.209323495	-78.4894855	54
2017-01-05 11:10:05	-0.209323876	-78.4894855	52
2017-01-05 11:15:03	-0.209323823	-78.4894857	50
2017-01-05 11:20:12	-0.209323125	-78.4894858	55
2017-01-05 11:25:08	-0.209309364	-78.4894933	56
2017-01-05 11:30:05	-0.209309127	-78.4894930	49
2017-01-05 11:35:11	-0.209309098	-78.4894937	47
2017-01-05 11:40:03	-0.209309456	-78.4894935	56
2017-01-05 11:45:01	-0.209309234	-78.4894934	53
2017-01-05 11:50:07	-0.209309456	-78.4894934	55
2017-01-05 11:55:09	-0.209284345	-78.4894412	54
2017-01-05 12:00:10	-0.209284590	-78.4894411	52
2017-01-05 12:05:11	-0.209284098	-78.4894412	55
2017-01-05 12:10:12	-0.209284233	-78.4894418	57
2017-01-05 12:15:03	-0.209284674	-78.4894419	51
2017-01-05 12:20:08	-0.209387234	-78.4895934	56
2017-01-05 12:25:09	-0.209387454	-78.4895935	52
2017-01-05 12:30:10	-0.209387098	-78.4895934	53
2017-01-05 12:35:11	-0.209387456	-78.4895933	54
2017-01-05 12:40:12	-0.209387349	-78.4895935	55
2017-01-05 12:45:06	-0.209264234	-78.4893962	52
2017-01-05 12:50:13	-0.209264472	-78.4893965	54
2017-01-05 12:55:36	-0.209264238	-78.4893969	55

Tabla 3.5 Mediciones en medio de los edificios Electrónica – Química y Eléctrica

3.4.2 MEDICIONES EN EL ÁREA RECRATIVA DEL EDIFICIO DE INGENIERÍA EN SISTEMAS

A continuación se presentan las mediciones de ruido realizadas en el área recreativa del edificio de Ingeniería en Sistemas.

3.4.2.1 Sesión 1

La primera sesión para realizar las mediciones de ruido en este sitio fue, el día jueves 15 de diciembre del 2016. La sesión tiene una duración de 2 horas.

En la tabla 3.6 se muestran los datos obtenidos en la sesión 1.

Fecha y Hora	Latitud (S^0)	Longitud (W^0)	Ruido (dB)
2016-12-15 13:01:01	-0.210245234	-78.4891932	57
2016-12-15 13:06:10	-0.210245343	-78.4891931	55
2016-12-15 13:11:15	-0.210245456	-78.4891934	55
2016-12-15 13:16:18	-0.210245595	-78.4891933	58
2016-12-15 13:21:19	-0.210245569	-78.4891933	54
2016-12-15 13:26:04	-0.210234876	-78.4891545	50
2016-12-15 13:31:05	-0.210234785	-78.4891540	52
2016-12-15 13:36:12	-0.210234345	-78.4891547	51
2016-12-15 13:41:14	-0.210234435	-78.4891546	53
2016-12-15 13:46:11	-0.210234094	-78.4891547	54
2016-12-15 13:51:01	-0.210190124	-78.4891215	55
2016-12-15 13:56:13	-0.210190547	-78.4891214	50
2016-12-15 14:01:14	-0.210190832	-78.4891214	50
2016-12-15 14:06:12	-0.210190120	-78.4891217	53
2016-12-15 14:11:10	-0.210190123	-78.4891218	55
2016-12-15 14:16:16	-0.210176632	-78.4890851	51
2016-12-15 14:21:07	-0.210176347	-78.4890859	56
2016-12-15 14:26:08	-0.210176986	-78.4890853	53
2016-12-15 14:31:09	-0.210176498	-78.4890856	55
2016-12-15 14:36:05	-0.210176094	-78.4890856	57
2016-12-15 14:41:02	-0.210168985	-78.4890432	55
2016-12-15 14:46:04	-0.210168982	-78.4890439	54
2016-12-15 14:51:03	-0.210168091	-78.4890432	53
2016-12-15 14:56:07	-0.210168732	-78.4890432	50

Tabla 3.6 Mediciones en el área recreativa del edificio de Ingeniería en Sistemas

3.4.2.2 Sesión 2

La segunda sesión para realizar las mediciones de ruido en este sitio fue, el día miércoles 04 de enero del 2017. La sesión tiene una duración de 2 horas.

En la tabla 3.7 se muestran los datos obtenidos en la sesión 2.

Fecha y Hora	Latitud (S^0)	Longitud (W^0)	Ruido (dB)
2017-01-04 15:03:08	-0.210237121	-78.4892491	50
2017-01-04 15:08:12	-0.210237132	-78.4892492	51
2017-01-04 15:13:10	-0.210237232	-78.4892491	55
2017-01-04 15:18:19	-0.210237475	-78.4892492	48
2017-01-04 15:23:07	-0.210237347	-78.4892492	46
2017-01-04 15:28:02	-0.210231298	-78.4892064	49
2017-01-04 15:33:00	-0.210231236	-78.4892063	50
2017-01-04 15:38:03	-0.210231381	-78.4892069	52
2017-01-04 15:43:05	-0.210231109	-78.4892067	53
2017-01-04 15:48:00	-0.210231879	-78.4892066	51
2017-01-04 15:53:17	-0.210211123	-78.4891567	49
2017-01-04 15:58:11	-0.210211092	-78.4891560	53
2017-01-04 16:03:15	-0.210211029	-78.4891561	57
2017-01-04 16:08:00	-0.210211873	-78.4891562	69
2017-01-04 16:13:08	-0.210211673	-78.4891563	72
2017-01-04 16:18:01	-0.210203872	-78.4891202	54
2017-01-04 16:23:23	-0.210203293	-78.489120	50
2017-01-04 16:28:29	-0.210203309	-78.489120	48
2017-01-04 16:33:11	-0.210203987	-78.489120	49
2017-01-04 16:38:22	-0.210203091	-78.489120	51
2017-01-04 16:43:07	-0.210158042	-78.489036	50
2017-01-04 16:48:34	-0.210158035	-78.489036	50
2017-01-04 16:53:09	-0.210158045	-78.489036	53
2017-01-04 16:58:02	-0.210158012	-78.489036	52

Tabla 3.7 Mediciones en el área recreativa del edificio de Ingeniería en Sistemas

3.4.3 MEDICIONES EN EL HALL DEL EDIFICIO DE MEDIOS EXTERNOS

A continuación se presentan las mediciones de ruido realizadas hall del edificio de medios externos.

3.4.3.1 Sesión 1

La primera sesión para realizar las mediciones de ruido en este sitio fue, el día viernes 16 de diciembre del 2016. La sesión tiene una duración de 2 horas. En la tabla 3.8 se muestran los datos obtenidos en la sesión 1.

Fecha y Hora	Latitud (S^0)	Longitud (W^0)	Ruido (dB)
2016-12-16 13:04:26	-0.209103231	-78.4870273	50
2016-12-16 13:09:06	-0.209103124	-78.4870272	52
2016-12-16 13:14:13	-0.209103235	-78.4870274	53
2016-12-16 13:19:12	-0.209103156	-78.4870273	52
2016-12-16 13:24:09	-0.209103189	-78.4870276	50
2016-12-16 13:29:03	-0.209140221	-78.4870547	56
2016-12-16 13:34:02	-0.209140233	-78.4870546	53
2016-12-16 13:39:00	-0.209140211	-78.4870548	55
2016-12-16 13:44:07	-0.209140256	-78.4870546	54
2016-12-16 13:49:09	-0.209140312	-78.4870548	56
2016-12-16 13:54:05	-0.209112390	-78.4870219	54
2016-12-16 13:59:03	-0.209112210	-78.4870218	57
2016-12-16 14:04:13	-0.209112932	-78.4870219	53
2016-12-16 14:09:12	-0.209112785	-78.4870217	52
2016-12-16 14:14:05	-0.209112978	-78.4870219	57
2016-12-16 14:19:04	-0.209156231	-78.4870355	56
2016-12-16 14:24:13	-0.209156257	-78.4870354	58
2016-12-16 14:29:19	-0.209156122	-78.4870355	54
2016-12-16 14:34:20	-0.209156109	-78.4870357	55
2016-12-16 14:39:02	-0.209156286	-78.4870354	55
2016-12-16 14:44:07	-0.209184957	-78.4870772	53
2016-12-16 14:49:04	-0.209184901	-78.4870771	51
2016-12-16 14:54:19	-0.209184934	-78.4870775	50
2016-12-16 14:59:10	-0.209184899	-78.4870774	56

Tabla 3.8 Mediciones en el hall del edificio de Medios Externos

3.4.3.2 Sesión 2

La segunda sesión para realizar las mediciones de ruido en este sitio fue, el día viernes 06 de enero del 2017. La sesión tiene una duración de 2 horas. En la tabla 3.9 se muestran los datos obtenidos en la sesión 2.

Fecha y Hora	Latitud (S^0)	Longitud (W^0)	Ruido (dB)
2017-01-06 17:02:28	-0.209195234	-78.4870463	50
2017-01-06 17:07:00	-0.209195278	-78.4870464	53
2017-01-06 17:12:09	-0.209195345	-78.4870466	52
2017-01-06 17:17:11	-0.209195645	-78.4870464	52
2017-01-06 17:22:17	-0.209195794	-78.4870466	57
2017-01-06 17:27:25	-0.209145478	-78.4870803	55
2017-01-06 17:32:22	-0.209145235	-78.4870802	54
2017-01-06 17:37:09	-0.209145457	-78.4870807	56
2017-01-06 17:42:02	-0.209145123	-78.4870805	46
2017-01-06 17:47:13	-0.209145345	-78.4870803	44
2017-01-06 17:52:17	-0.209131389	-78.4870293	47
2017-01-06 17:57:06	-0.209131023	-78.4870291	45
2017-01-06 18:02:09	-0.209131128	-78.4870292	45
2017-01-06 18:07:08	-0.209131178	-78.4870290	40
2017-01-06 18:12:12	-0.209131124	-78.4870299	43
2017-01-06 18:17:24	-0.209092239	-78.4869828	44
2017-01-06 18:22:27	-0.209092209	-78.4869829	49
2017-01-06 18:27:38	-0.209092187	-78.4869827	51
2017-01-06 18:32:44	-0.209092167	-78.4869826	55
2017-01-06 18:37:56	-0.209092155	-78.4869825	52
2017-01-06 18:42:23	-0.209087453	-78.4870525	50
2017-01-06 18:47:02	-0.209087420	-78.4870527	46
2017-01-06 18:52:08	-0.209087418	-78.4870528	48
2017-01-06 18:57:15	-0.209087411	-78.4870529	55

Tabla 3.9 Mediciones en el hall del edificio de Medios Externos

3.5 ANÁLISIS DE RESULTADOS

El análisis de resultados presenta los niveles de ruido obtenidos en cada sesión de medición. El análisis está basado de acuerdo a la Ordenanza Municipal N 146, cada sesión para la medición de ruido tiene un análisis independiente debido a las características presentadas.

3.5.1 ANÁLISIS DE RESULTADOS DE LA MEDICIÓN DE RUIDO EN MEDIO DE LOS EDIFICIOS ELÉCTRICA – QUÍMICA Y ELÉCTRICA

3.5.1.1 Análisis de la sesión 1

A continuación se presentan las características del sitio al momento de realizar las mediciones.

- Cielo parcialmente nublado.
- No hay presencia de vehículos.
- Presencia constante de personas.

En la figura 3.31 se muestra un comportamiento del ruido a medida que transcurre el tiempo.

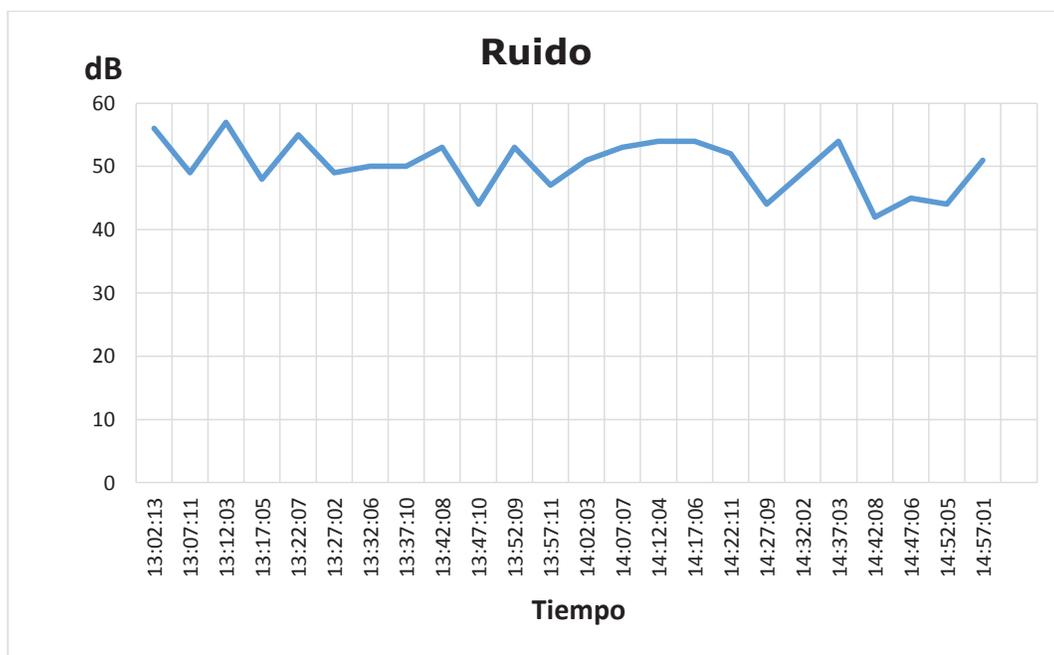


Figura 3.31 Ruido vs Tiempo en la sesión 1, en medio de los edificios Electrónica – Química y Eléctrica

Se observa en la figura 3.28 que el ruido en este sitio oscila entre 40 dB y 60 dB. Lo cual indica que en este sitio durante el tiempo de medición presenta un promedio de ruido elevado, sin llegar a ser crítico, siendo las personas la principal fuente generadora de ruido.

3.5.1.2 Análisis de la sesión 2

A continuación se presentan las características del sitio al momento de realizar las mediciones.

- Cielo nublado.
- No hay presencia de vehículos.
- Presencia mínima de personas.
- Actividades de limpieza.

En la figura 3.32 se muestra un comportamiento del ruido a medida que transcurre el tiempo.

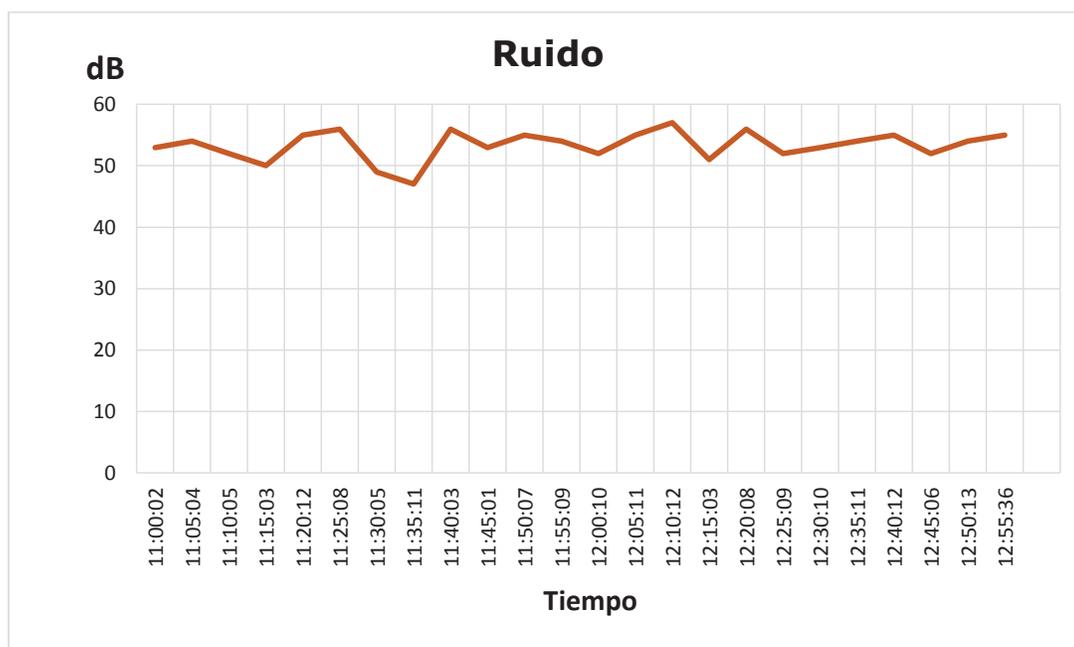


Figura 3.32 Ruido vs Tiempo en la sesión 1, en medio de los edificios Electrónica – Química y Eléctrica

Se observa en la figura 3.29 que el ruido en este sitio oscila entre 50 dB y 60 dB. Lo cual indica que en este sitio durante el tiempo de medición presenta un promedio

de ruido elevado, sin llegar a ser crítico, siendo las personas la principal fuente generadora de ruido. Además a las 11 horas con 20 minutos se presenta un evento atípico generando un pico en la curva.

3.5.2 ANÁLISIS DE RESULTADOS DE LA MEDICIÓN DE RUIDO EN EL ÁREA RECREATIVA DEL EDIFICIO DE INGENIERÍA EN SISTEMAS

3.5.2.1 Análisis de la sesión 1

A continuación se presentan las características del sitio al momento de realizar las mediciones.

- Cielo parcialmente nublado.
- No hay presencia de vehículos.
- Presencia abundante de personas.

En la figura 3.33 se muestra un comportamiento del ruido a medida que transcurre el tiempo.

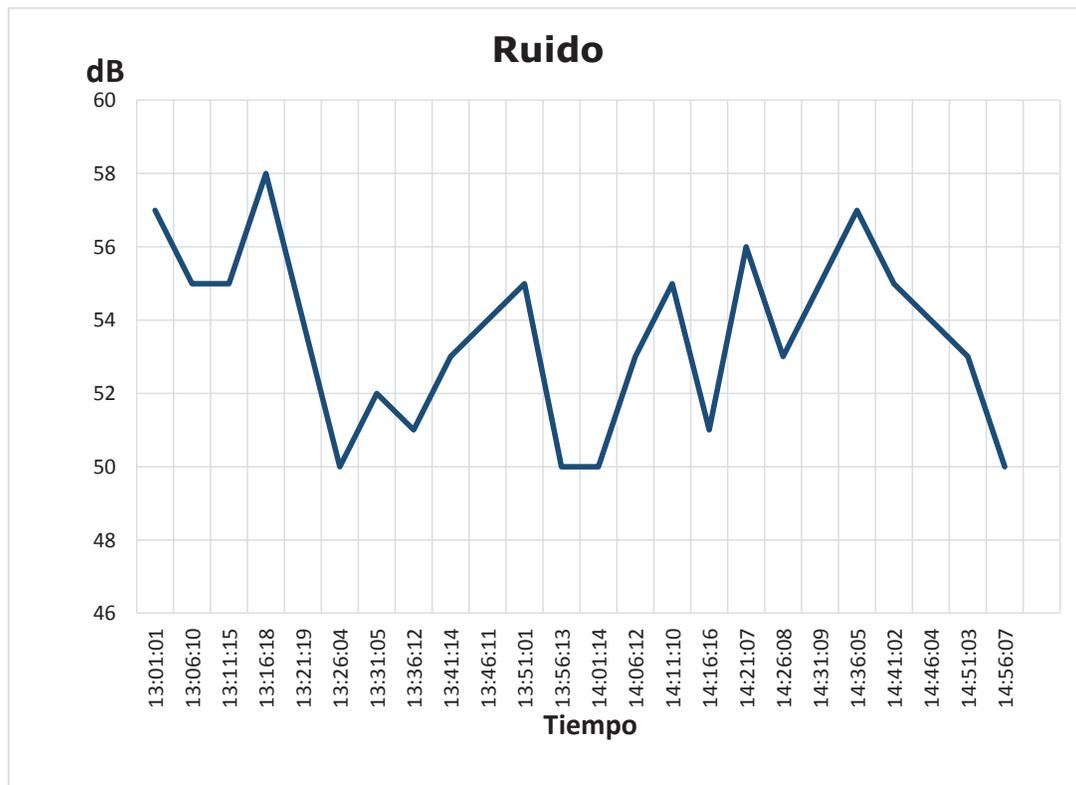


Figura 3.33 Ruido vs Tiempo en la sesión 1, en el área recreativa del edificio de Ingeniería en Sistemas

Se observa en la figura 3.33 que el ruido en este sitio oscila entre 50 dB y 65 dB. Lo cual indica que en este sitio durante el tiempo de medición presenta un promedio de ruido elevado, sin llegar a ser crítico, siendo las personas que habitan este lugar la principal fuente generadora de ruido.

3.5.2.2 Sesión 2

A continuación se presentan las características del sitio al momento de realizar las mediciones.

- Cielo nublado.
- No hay presencia de vehículos.
- Presencia mínima de personas.

En la figura 3.34 se muestra un comportamiento del ruido a medida que transcurre el tiempo.

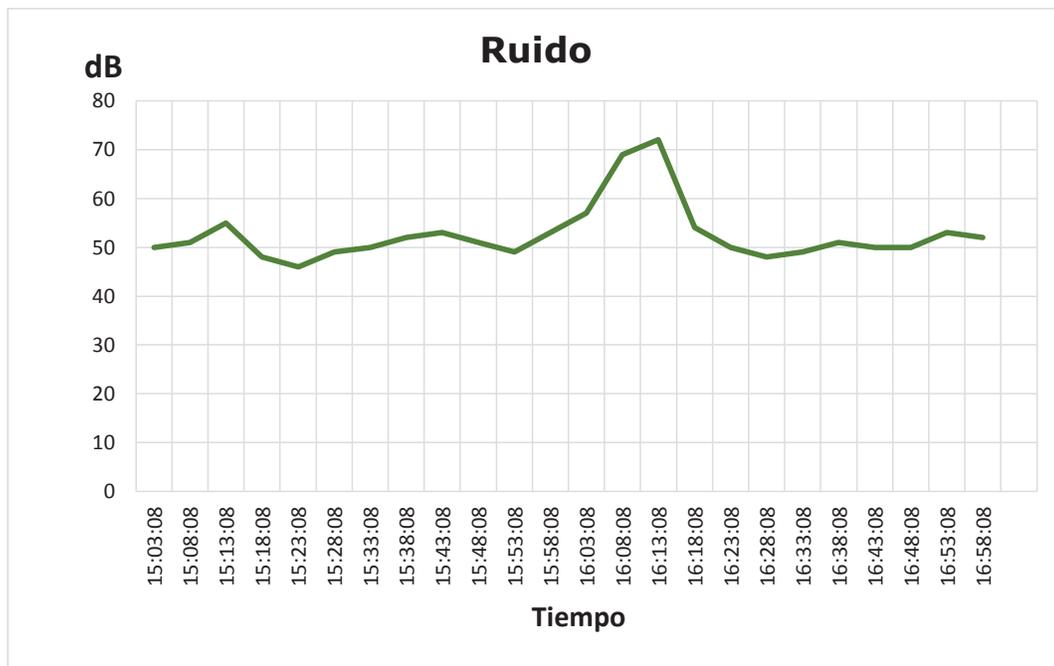


Figura 3.34 Ruido vs Tiempo en la sesión 2, en el área recreativa del edificio de Ingeniería en Sistemas

Se observa en la figura 3.34 que el ruido en este sitio oscila entre 50 dB y 65 dB. Lo cual indica que en este sitio durante el tiempo de medición presenta un promedio de ruido elevado, sin llegar a ser crítico, siendo las personas que habitan este lugar

la principal fuente generadora de ruido. Además a las 16 horas con 13 minutos el paso de un avión generando un pico en la curva.

3.5.3 ANÁLISIS DE RESULTADOS DE LA MEDICIÓN DE RUIDO EN EL HALL DEL EDIFICIO DE MEDIOS EXTERNOS

3.5.3.1 Análisis de la sesión 1

A continuación se presentan las características del sitio al momento de realizar las mediciones.

- Techo de concreto.
- No hay presencia de vehículos.
- Presencia abundante de personas.

En la figura 3.35 se muestra un comportamiento del ruido a medida que transcurre el tiempo.

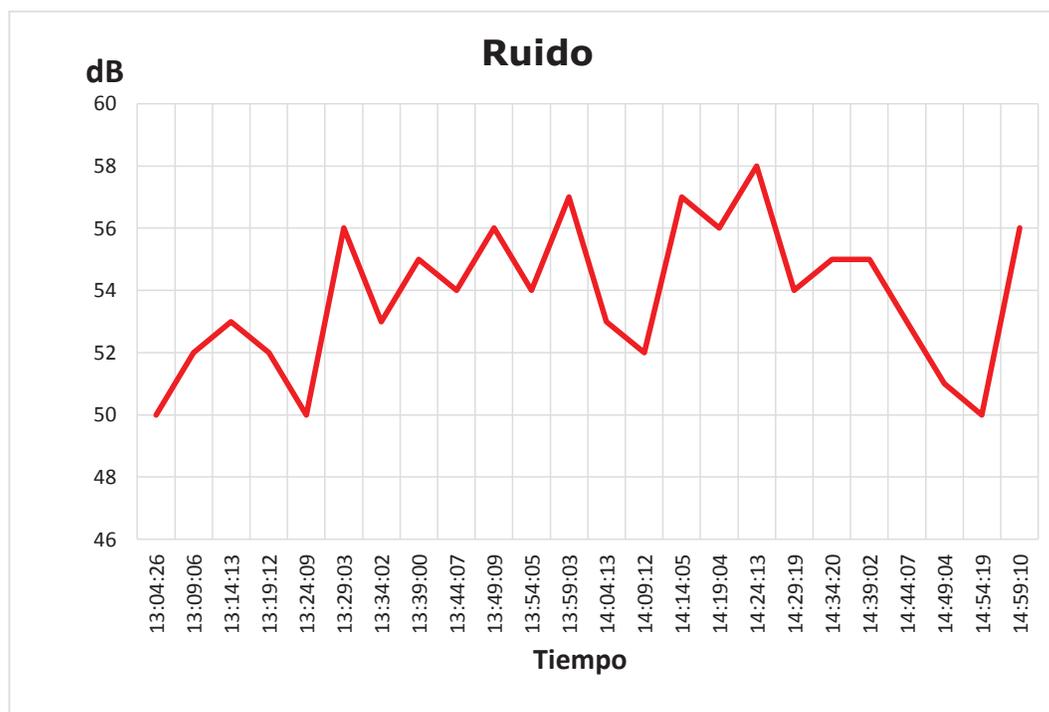


Figura 3.35 Ruido vs Tiempo en la sesión 1, en el hall del edificio de Medios Externos

Se observa en la figura 3.35 que el ruido en este sitio oscila entre 50 dB y 65 dB aproximadamente. Lo cual indica que en este sitio durante el tiempo de medición

presenta un promedio de ruido elevado, sin llegar a ser crítico, siendo las personas que transitan este lugar la principal fuente generadora de ruido.

3.5.3.2 Análisis de la sesión 1

A continuación se presentan las características del sitio al momento de realizar las mediciones.

- Techo de concreto.
- No hay presencia de vehículos.
- Presencia abundante de personas.

En la figura 3.36 se muestra un comportamiento del ruido a medida que transcurre el tiempo.

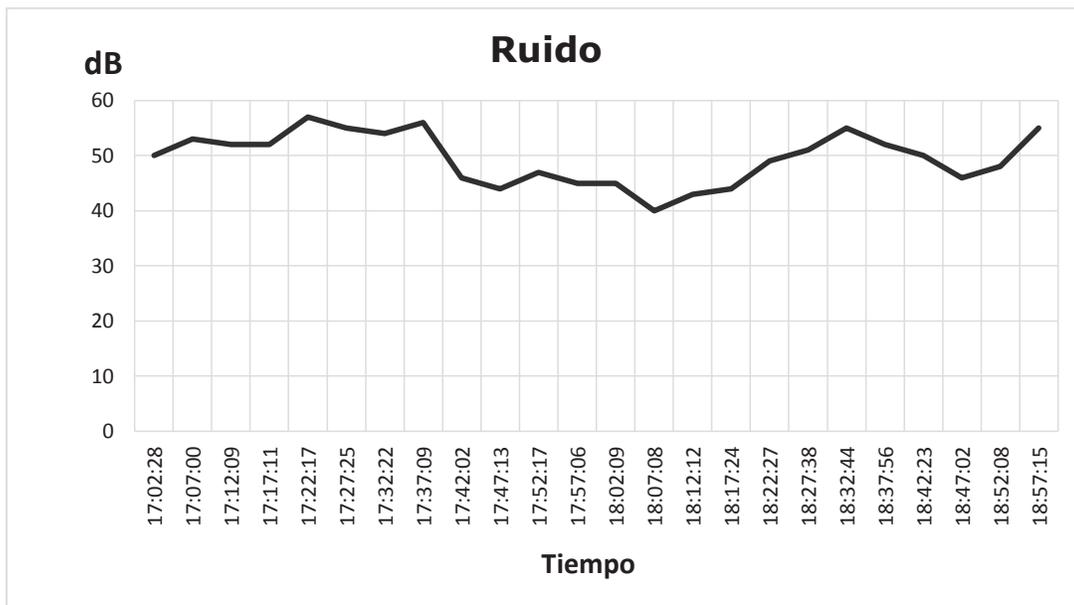


Figura 3.36 Ruido vs Tiempo en la sesión 2, en el hall del edificio de Medios Externos

Se observa en la figura 3.36 que el ruido en este sitio oscila entre 40 dB y 62 dB aproximadamente. Lo cual indica que en este sitio durante el tiempo de medición presenta un promedio de ruido adecuado, siendo las personas que transitan este lugar la principal fuente generadora de ruido. En horario que se realizó medición transitaban muy pocas personas en el lugar.

3.6 COSTO REFERENCIAL DEL SISTEMA PROTOTIPO

El costo referencial del sistema prototipo se presenta en la tabla 3.1.

SISTEMA PROTOTIPO			
SUBSISTEMA TRANSMISOR			
Equipo	Cantidad	Costo Unitario (USD)	Costo Total
SBC Raspberry Pi B+	1	65	65
Micrófono USB	2	10	20
Modulo GPS	1	35	35
Plan móvil de Internet	1	19	19
Memoria micro SD 16 GB	1	18	18
Cable USB	1	5	5
Batería de celular	1	20	20
Cables conectores de pines	4	0,15	0,60
Horas de trabajo	200	5	1000
SUBSISTEMA RECEPTOR			
Computador	1	400	400
Plan fijo internet	1	20	20
Horas de trabajo	225	5	1125
SUBTOTALES			
Subsistema Transmisor			1182,6
Subsistema Receptor			1545
		Total	2727.6 \$

Tabla 3.10 Costo referencial del sistema prototipo

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las conclusiones y recomendaciones, obtenidas de la realización del presente trabajo.

4.1 CONCLUSIONES

- En el presente sistema prototipo se diseña e implementa un sistema inalámbrico de medición de ruido en el ambiente, utilizando tecnologías *open source*.
- El sistema prototipo permite realizar mediciones en tiempo real del ruido con su respectiva información de geolocalización. El sistema permite enviar y almacenar los datos a un servidor para procesarlos y presentarlos a través de una interfaz web. Los resultados se presentan mediante un mapa de calor y en una tabla de texto, para ello se utiliza la interfaz de Google Maps.
- En el análisis de requerimientos se determinaron los parámetros que debe cumplir el sistema prototipo, para lo cual el sistema se dividió en dos subsistemas. El subsistema transmisor se encarga de realizar las mediciones de ruido y geolocalización y enviar el resultado de las mediciones al subsistema receptor a través de Internet. El subsistema receptor por su parte se encarga de recibir, almacenar y presentar los datos enviados por el subsistema transmisor.
- En las pruebas de funcionamiento se evidenció que los micrófonos USB poseen una sensibilidad de +/- 4 dB que al tratarse del fenómeno de ruido por cada dB la intensidad del ruido se duplica o reduce a la mitad, por tal razón fue importante realizar un adecuado proceso de calibración del sistema prototipo, para conseguir mediciones acertadas.
- En las pruebas de funcionamiento se evidenció que el modulo GPS provee información de las coordenadas del sitio de forma acertada en ambientes de tipo *outdoor*; pero al no tener una línea de vista en ambientes de tipo *indoor*

el modulo GPS tiene inconvenientes al momento de conectarse al sistema GPS.

- Las mediciones fueron realizadas en dos sesiones para cada sitio por dos horas consecutivas de acuerdo a lo planteado en el plan de trabajo, lo cual indica que los resultados obtenidos en estas mediciones son únicamente una referencia del comportamiento del ruido en dichos lugares.
- El análisis de resultados muestra que el ruido promedio no supera los límites de 55 dB mencionados en la ordenanza Municipal N 146, para establecimientos educativos; sin embargo el valor promedio esta alrededor de 50 dB, siendo un valor cercano al límite permitido. También el análisis de resultados indica que la principal fuente generadora de ruido es la cantidad de personas que transitan en los sitios donde se realizó las mediciones.
- La contaminación sonora afecta la conducta y salud de las personas. El sistema prototipo se presenta como un instrumento de medición que permite recolectar información del ruido y mediante un análisis de resultados determinar las acciones a realizar y así controlar el nivel ruido presentado.

4.2 RECOMENDACIONES

- Se recomienda ampliar el estudio del presente trabajo para realizar las mediciones en ambientes de tipo *indoor* a través de recolección de datos de geolocalización mediante redes Wi-Fi.
- El subsistema transmisor debe desplazarse por el sector donde se realizan las mediciones, por tal razón se recomienda contratar el plan de Internet móvil con la operadora celular que ofrezca la mayor área de cobertura.
- Para la selección de los sensores de ruido se recomienda considerar los niveles de ruido a medir. Así, si el nivel de intensidad de ruido a medir es

bajo (20 dB), la sensibilidad podría ser 60 dB. Si la intensidad de ruido a medir es alta (90 dB), la sensibilidad podría ser 4 dB.

- Para determinar el nivel del ruido en un lugar se recomienda tener tiempos de medición prolongados y periódicos. El tiempo y los periodos de medición se determinan mediante las características que presenten dicho lugar.
- La carga de la batería utilizada para el subsistema transmisor se recomienda que dure al menos seis horas consecutivas para tener tiempos de medición adecuados con el fin de tener un correcto análisis de resultados. Se debe mencionar que la Raspberry Pi al tener una entrada de 5V y un puerto micro USB se puede conectar inclusive al cargador de un automóvil y así prolongar el tiempo de funcionamiento del subsistema transmisor.
- Se recomienda utilizar una tarjeta micro SD en la Raspberry Pi con una capacidad de 16 GB, esto permitirá la actualización del software, instalación de nuevos paquetes y aplicaciones sin afectar el desempeño del sistema operativo.
- Para realizar las mediciones de ruido se recomienda ubicar al subsistema transmisor junto a la fuente generadora de ruido, ya que el ruido se atenúa de acuerdo a la distancia. La dirección en que se ubique el subsistema transmisor no es relevante puesto que los micrófonos son omnidireccionales.

REFERENCIAS BIBLIOGRÁFICAS

- [1] APKGPLAY, «Sound Meter,» 21 04 2016. [En línea]. Disponible: <https://www.apkgplay.com/es/sonometro-sound-meter>.
- [2] FCEIA, «FCEIA,» [En línea]. Disponible: <http://www.fceia.unr.edu.ar/acustica/comite/queesrui.htm>. [Último acceso: 25 10 2016].
- [3] F. Miyara, «Naturaleza del sonido,» [En línea]. Disponible: <http://www.fceia.unr.edu.ar/acustica/comite/sonido.htm>. [Último acceso: 24 10 2016].
- [4] «FisicaModerna8,» 24 10 2016. [En línea]. Disponible: <https://fisicamoderna8.wordpress.com/acerca-de/tareas/tarea-2/>. [Último acceso: 24 10 2016].
- [5] S. M. C. I. Cajamarca Yunga Pablo Vinicio, Elaboración de un mapa de ruido del Campus Politécnico José Rubén Orellana Ricaurte, Quito: Tesis , 2013.
- [6] C. d. P. O.M.S, «Organización Mundial de la Salud,» Centro de Prensa, 02 27 2015. [En línea]. Disponible: <http://www.who.int/mediacentre/news/releases/2015/ear-care/es/>. [Último acceso: 24 10 2016].
- [7] Junta de Andalucía y O.S.M.A.N, «Ruido Y Salud,» [En línea]. Disponible: https://www.diba.cat/c/document_library/get_file?uuid=72b1d2fd-c5e5-4751-b071-8822dfdfdded&groupId=7294824. [Último acceso: 24 10 2016].
- [8] D. A. Q. Chimarro, Diseño e Implementación de un prototipo que permita cuantificar el nivel de presión sonora con indicación visual y audible, Quito: Tesis, 2008.
- [9] M. d. D. M. d. Quito, «MDMQ Ordenanzas,» Código Municipal para el Distrito Metropolitano de Quito, 5 7 2004. [En línea]. Disponible: http://www7.quito.gob.ec/mdmq_ordenanzas/Ordenanzas/ORDENANZAS%20A%C3%91OS%20ANTERIORES/ORDM-123%20-%20RUIDO%20-%20MEDIO%20AMBIENTE.pdf. [Último acceso: 25 10 2016].
- [10] A. E. p. I. S. y. S. e. e. Trabajo, «FACTS,» [En línea]. Disponible: http://www.insht.es/InshtWeb/Contenidos/Documentacion/TextosOnline/Rev_INSHT/2005/36/FactS_58.pdf. [Último acceso: 25 10 2016].
- [11] C. Bell, Beginning Sensor Network with Arduino and Raspberry Pi, tia, Technology in Action.

- [12] A. Uno, «Arduino Uno,» [En línea]. Disponible: <https://www.arduino.cc/en/Main/ArduinoBoardUno>. [Último acceso: 25 10 2016].
- [13] J. V. Acero Delgado y C. D. Reyes Larco, Sistema prototipo para el monitorero inalámbrico de gases contaminantes del aire desarrollado bajo plataformas de hardware y software libre, Quito, 2016.
- [14] R. Pi, «Raspberry Pi 1 Modelo B+,» 2 2015. [En línea]. Disponible: <https://www.raspberrypi.org/products/model-b-plus/>. [Último acceso: 25 10 2016].
- [15] Stackchange, «Raspberry Pi,» 10 2016. [En línea]. Disponible: <http://raspberrypi.stackexchange.com/questions/27765/raspberry-pi-2-has-a-unified-rca-and-audio-jack-what-cable-should-i-use-to-conn>. [Último acceso: 25 10 2016].
- [16] A. Rodríguez, «Sensor de Sonido,» 6 3 2014. [En línea]. Disponible: <https://prezi.com/ucealvf3m09u/sensores-del-sonido/>. [Último acceso: 25 10 2016].
- [17] P. Correira, de *Guía Práctica del GPS*, Barcelona, MARCOMBO, S.A, 2002.
- [18] C. robots, «ChiOSZ robots,» 11 05 2014. [En línea]. Disponible: <https://chioszrobots.com/2014/05/11/ublox-neo-6m-gps-module-aircraft-flight-controller-arduino-mwc-imu-apm2/>. [Último acceso: 25 10 2016].
- [19] W. Stallings, *Comunicaciones y Redes de Computadores*, Madrid: Pearson Prentice Hall, 2004, p. 780.
- [20] N. W. Group, «IETF,» 6 1999. [En línea]. Disponible: <https://www.ietf.org/rfc/rfc2616.txt>. [Último acceso: 25 10 2016].
- [21] VetrineInrete, «VetrineInrete,» [En línea]. Disponible: <http://www.vetrineinrete.net/art1/0/10857/11592/Microfono-da-tavolo-usb-flessibile-per-pc-notebook-mac-.html>. [Último acceso: 14 11 2016].
- [22] D. robots, «ChiOSZ robots,» [En línea]. Disponible: <https://chioszrobots.com/2014/05/11/ublox-neo-6m-gps-module-aircraft-flight-controller-arduino-mwc-imu-apm2/>. [Último acceso: 14 11 2016].
- [23] T. P. Project, «The Pi4J Project,» 26 07 2016. [En línea]. Disponible: <http://pi4j.com/pins/model-2b-rev1.html>. [Último acceso: 14 11 2016].
- [24] A. d. G. Maps, «Capa del mapa de calor,» [En línea]. Disponible: <https://developers.google.com/maps/documentation/javascript/heatmaplayer?hl=es>. [Último acceso: 15 11 2016].
- [25] Huawei, «Consumer Huawei,» [En línea]. Disponible: <http://consumer.huawei.com/py/mobile-broadband/mobile-wifi/tech-specs/e3531s-py.htm>. [Último acceso: 14 11 2016].

- [26] UNOCERO, «UNOCERO,» [En línea]. Disponible: <https://www.unocero.com/2012/08/02/raspbian-sistema-operativo-gratuito-para-la-raspberry-pi/>. [Último acceso: 14 11 2016].
- [27] P. M. a. D. Hows, Learn Raspberry Pi with Linux, tia, Technology in Action.
- [28] P. S. Fundation, «Python,» 23 09 2016. [En línea]. Disponible: <https://docs.python.org/3/tutorial/introduction.html#lists>. [Último acceso: 14 11 2016].
- [29] M. Python, «Monty Python's OFFICIAL WEBSTE,» 2017. [En línea]. Disponible: <http://www.montypython.com/>. [Último acceso: 12 01 2017].
- [30] W. Donat, Learn Raspberry Pi proگرامing with Python, tia, Technology in Action.
- [31] Instructables, «Instructables,» 2016. [En línea]. Disponible: <http://www.instructables.com/id/Raspberry-Pi-the-Neo-6M-GPS/>. [Último acceso: 14 11 2016].
- [32] HP, «PC Notebook de entretenimeinto HP Pavilion dm4-1280la,» 2016. [En línea]. Disponible: <http://support.hp.com/mx-es/product/HP-Pavilion-dm4-Entertainment-Notebook-PC-series/4346357/model/5052242>. [Último acceso: 14 11 2016].
- [33] Oracle, «VirtualBox,» [En línea]. Disponible: <https://www.virtualbox.org/>. [Último acceso: 14 11 2016].
- [34] T. U. o. s. Debian, «Debian,» 17 09 2016. [En línea]. Disponible: <https://www.debian.org/index.es.html>. [Último acceso: 14 11 2016].
- [35] Oracle, «MySQL,» 2016. [En línea]. Disponible: <https://www.mysql.com/products/enterprise/monitor.html>. [Último acceso: 14 11 2016].
- [36] ORACLE, «Integrated Cloud Applications & Plataform Services,» 2017. [En línea]. Disponible: <https://www.oracle.com/es/index.html>. [Último acceso: 12 01 2017].
- [37] P. Copyright, «PHP,» 2016. [En línea]. Disponible: <http://php.net/manual/es/intro-what-is.php>. [Último acceso: 14 11 2016].
- [38] T. w. l. w. d. site, «w3schools,» 2016. [En línea]. Disponible: <http://www.w3schools.com/html/>. [Último acceso: 14 11 2016].
- [39] A. d. G. Maps, «Maps JavaScript API,» [En línea]. Disponible: <https://developers.google.com/maps/documentation/javascript/heatmaplayer?hl=es-419>. [Último acceso: 14 11 2016].

- [40] CNET, «Genius SP-S110 speakers - ForPC - wired Series,» [En línea]. Disponible: <https://www.cnet.com/products/genius-sp-s110-speakers-for-pc-wired-series/specs/>. [Último acceso: 25 01 2017].
- [41] I. CONFIDENTIAL, «Single Chip PC Audio System,» *DATASHEET*, vol. V 0.995, n° 1, p. 39, 2009.
- [42] F. d. C. S.A, «Falabella.com,» [En línea]. Disponible: <http://www.falabella.com.co/falabella-co/product/2004296/Parlante-para--PC-SP-S110-Negro>. [Último acceso: 16 01 2017].

ANEXOS

ANEXO I. Archivo de texto que contiene el código del *script* Gpsread.Py

ANEXO II. Archivo de texto que contiene el código del *script* Audiodevices.Py

ANEXO III. Archivo de texto que contiene el código del *script* Mysqlhandler.Py

ANEXO IV. Archivo de texto que contiene el código del *script* Sonometrogps.Py

ANEXO V. Especificaciones técnicas de la Single Board Computer Raspberry Pi Modelo B + (Datasheet)

ANEXO VI. Ordenanza Municipal N 146, El Consejo Del Distrito Metropolitano De Quito

Nota: Los archivos de los anexos mencionados se encuentran almacenados en el CD adjunto.