



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA DE INGENIERÍA ELECTRÓNICA

**DESARROLLO DE UN DISPOSITIVO IoT EN CLOUD PARA LA MEDICIÓN
DE LA CONTAMINACIÓN AUDITIVA EN AMBIENTES CERRADOS**

Trabajo de titulación previo a la obtención del
Título de Ingeniero Electrónico

AUTORES: Ana Lucia Carlosama Pupiales
José Luis Alanuca Puga

TUTOR: Eduardo Rodrigo Torres Santos

Quito-Ecuador

2023

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Ana Lucia Carlosama Pupiales, con documento de identificación No. 1003421417 y José Luis Alanuca Puga, con documento de identificación No. 1723030076 manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 8 de agosto del año 2023

Atentamente,



Ana Lucia Carlosama Pupiales

1003421417



José Luis Alanuca Puga

1723030076

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Ana Lucia Carlosama Pupiales, con documento de identificación No. 1003421417 y, José Luis Alanuca Puga con documento de identificación No. 1723030076 expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del proyecto técnico: “Desarrollo de un dispositivo IoT en cloud para la medición de la contaminación auditiva en ambientes cerrados”, el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 8 de agosto del año 2023

Atentamente,



Ana Lucia Carlosama Pupiales
1003421417



José Luis Alanuca Puga
1723030076

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Eduardo Rodrigo Torres Santos con documento de identificación No. 1713654604 docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UN DISPOSITIVO IoT EN CLOUD PARA LA MEDICIÓN DE LA CONTAMINACIÓN AUDITIVA EN AMBIENTES CERRADOS**, realizado por Ana Lucia Carlosama Pupiales con documento de identificación No. 1003421417 y por José Luis Alanuca Puga con documento de identificación No. 1723030076, obteniendo como resultado final el trabajo de titulación bajo la opción proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 8 de agosto del año 2023

Atentamente,



Ing, Eduardo Rodrigo Torres Santos, MSc

1713654604

DEDICATORIA

Quiero dedicar este trabajo a toda mi familia. Principalmente, a mi esposo Cristian y a mis hijas Helen y Carolina que me apoyaron en cada etapa buena y mala de mi vida universitaria. Gracias por enseñarme a afrontar las dificultades a ser la persona que soy hoy, mi perseverancia y mi empeño. Todo esto con su amor infinito y sin pedir nada a cambio.

También quiero dedicarles este trabajo a mis padres por su confianza que depositaron en mí y por ayudarme en tiempos que ya me estaba rindiendo y por sus consejos y por todo su fuerza y amor. Nunca dejaré de estar agradecido por esto.

Ana Lucia Carlosama Pupiales

Agradezco a Dios por darme su mano en todo este camino y por haberme otorgado una familia maravillosa, quienes me han dado el mejor ejemplo de superación, humildad y perseverancia. Tengo la certeza de que sin su apoyo no hubiera logrado cumplir este sueño. Gracias querida familia por anhelar lo mejor para mi vida, gracias por cada consejo y por cada una de sus palabras que me guiaron durante esta meta.

A mi novia, por siempre creer en mí y darme fuerza en los momentos difíciles, aquellos donde todo era oscuridad, pero ella fue luz.

A mis estimados docentes por empaparame de conocimientos, pero también por compartir conmigo su lado humano.

José Luis Alanuca Puga

AGRADECIMIENTOS

Agradezco a Dios y a mis padres María y Francisco por siempre apoyarme para conseguir mis metas personales y académicas. También son los que me han brindado apoyo emocional para poder cumplir en los estudios y nunca abandonarlos.

A mi tutor Ing. Eduardo Torres por su dedicación y paciencia, sin sus palabras y correcciones precisas no hubiese podido lograr llegar a esta instancia tan anhelada.

A los docentes por repartir sus conocimientos de manera profesional, gracias por su paciencia, perseverancia y tolerancia.

Ana Lucia Carlosama Pupiales

Quiero dar gracias a Dios, quien con su bendición ha estado presente en mi vida, por brindarme la salud y la fortaleza necesaria para cumplir todos mis anhelos y manifestarse en cada uno de ellos. De manera especial a mi tutor de tesis y a todos mis docentes, por darme esa guía, no solo en la elaboración de este trabajo, sino a lo largo de toda mi carrera universitaria y haberme brindado el apoyo para mi desarrollo humano y profesional, y a todos los que creyeron desde un principio en mí.

José Luis Alanuca Puga

Agradecemos en el presente trabajo a la Universidad Politécnica Salesiana y a todos los que la conforman, por ser una guía durante toda la carrera y abrirnos las puertas para estudiar Ingeniería Electrónica. A nuestro tutor el Ingeniero Eduardo Torres por su tiempo y conocimiento invertido para la realización de este proyecto de titulación.

Los Autores

ÍNDICE

| | |
|---|-----|
| CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN | II |
| CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA | III |
| CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN | IV |
| DEDICATORIA | V |
| AGRADECIMIENTOS | VI |
| RESUMEN | XI |
| ABSTRACT | XII |
| CAPÍTULO I | 14 |
| ANTECEDENTES | 14 |
| 1.1 Planteamiento del problema..... | 14 |
| 1.2 Justificación | 14 |
| 1.3 Objetivos | 15 |
| 1.3.1 Objetivo General..... | 15 |
| 1.3.2 Objetivos Específicos | 15 |
| 1.4 Metodología | 15 |
| CAPÍTULO II..... | 16 |
| 2.1 Contaminación Acústica..... | 16 |
| 2.2 Ruido..... | 17 |
| 2.2.1 Clasificación del ruido..... | 17 |
| 2.2.2 Efectos del ruido | 17 |
| 2.2.3 Fuentes de ruido | 18 |
| 2.3 Salud y ruido | 18 |
| 2.4 Software Node-Red..... | 19 |
| 2.5 Node MCU ESP8266..... | 19 |
| 2.6 Conexión entre Bróker y Node-red..... | 20 |
| 2.7 Protocolo MQTT..... | 21 |
| 2.8 Sensor de sonido KY038 | 21 |
| 2.9 Medidor de sonido AS804 | 22 |
| 2.10 Nivel de ruido | 23 |
| 2.11 Niveles permisibles de ruido | 24 |
| CAPÍTULO III | 26 |
| 3.1 Diseño del dispositivo IoT | 26 |
| 3.2 Bloque Controlador ESP8266..... | 27 |
| 3.3 Comunicación Node Mcu ESP8266 y Node-Red | 27 |
| 3.4 Instalación de Node-Red..... | 28 |
| 3.5 Implementación de la comunicación MQTT en la WNCS..... | 31 |
| 3.6 Creación de parámetros para MQTT. | 34 |
| 3.7 Alertas por Telegram | 35 |
| 3.8 Configuración de Bot Telegram | 37 |
| 3.9 Diseño de la placa | 40 |
| 3.10 Diseño del prototipo | 41 |
| 3.11 Diagrama de flujo..... | 42 |
| CAPÍTULO IV | 44 |
| 4.1 Nodos usados en la interfaz Node-red | 44 |
| 4.2 Características del sonómetro AS804 | 46 |
| 4.3 Análisis de resultados | 47 |
| 4.4 Análisis económico del proyecto | 52 |

| | |
|-----------------------|----|
| CONCLUSIONES..... | 55 |
| RECOMENDACIONES | 56 |
| ANEXOS | 59 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 2.1 Node MCU ESP8266..... | 20 |
| Figura 2.2 Conexión de bróker y Node-red..... | 21 |
| Figura 2.3 Funcionamiento del protocolo MQTT..... | 21 |
| Figura 2.4 Sensor KY038..... | 22 |
| Figura 2.5 Sensor AS804..... | 23 |
| Figura 2.6 Salud y nivel de ruido..... | 24 |
| Figura 3.1 Diagrama de Bloques para la comunicación..... | 26 |
| Figura 3.2 Gestor de tarjetas en el IDE de Arduino..... | 27 |
| Figura 3.3 Código para conectividad ESP8266..... | 28 |
| Figura 3.4 Página oficial de NodeJS..... | 29 |
| Figura 3.5 Configuración como servicio de Windows..... | 29 |
| Figura 3.6 Instalación de librerías de Open ssl..... | 30 |
| Figura 3.7 Aplicación Services Windows..... | 30 |
| Figura 3.8 Activación del servicio..... | 31 |
| Figura 3.9 Comprobación de puerto..... | 31 |
| Figura 3.10 Acceso al software NODE-RED..... | 32 |
| Figura 3.11 Instalación de librerías NODE-RED para dashboard..... | 32 |
| Figura 3.12 Nodos para la programación de interfaces gráficas..... | 33 |
| Figura 3.13 Configuración de nodos MQTT..... | 33 |
| Figura 3.14 Configuración de nodos CHART..... | 34 |
| Figura 3.15 Configuración de nodos GAUGE..... | 34 |
| Figura 3.16 Nodos de comunicación con TELEGRAM..... | 35 |
| Figura 3.17 Configuración de nodo SWITCH..... | 36 |
| Figura 3.18 Programación de nodo FUNCIÓN..... | 36 |
| Figura 3.19 Configuración de BotFather..... | 37 |
| Figura 3.20 Opciones para la configuración de BotFather..... | 38 |
| Figura 3.21 Nombre para despliegue de chat..... | 38 |
| Figura 3.22 Código API generado..... | 39 |
| Figura 3.23 Nodo Telegram..... | 39 |
| Figura 3.24 Chat generado para envío de notificaciones..... | 40 |
| Figura 3.25 Modelo de placa en 3D (Software Proteus)..... | 41 |
| Figura 3.26 Montaje de ESP8266 en la placa..... | 41 |
| Figura 3.27 Diagrama de flujo para adquisición de datos..... | 42 |
| Figura 3.28 Diagrama de flujo para adquisición de datos..... | 43 |
| Figura 4.1 Nodos para recepción y envío de datos..... | 44 |
| Figura 4.2 Ventana para ejecutar Node-red..... | 45 |
| Figura 4.3 Tipos de nodos para la programación..... | 45 |
| Figura 4.4 Ventana de configuración para nodos..... | 46 |
| Figura 4.5 Nodo función..... | 46 |
| Figura 4.6 Comando ping de entre el servidor y el Gateway..... | 48 |
| Figura 4.7 Interfaz gráfica Node-red medición por la mañana..... | 49 |
| Figura 4.8 Interfaz gráfica Node-red medición en la tarde..... | 50 |
| Figura 4.9 Comportamiento de los sensores de sonido..... | 50 |
| Figura 4.10 Notificación de mensaje de Telegram entregada..... | 51 |
| Figura 4.11 Mensajes recibidos en Telegram..... | 52 |

INDICE DE TABLAS

| | |
|---|----|
| Tabla 2.1 Características de funcionamiento del controlador ESP8266 | 20 |
| Tabla 2.2 Características de funcionamiento..... | 22 |
| Tabla 2.3 Características de funcionamiento del sensor AS804. | 23 |
| Tabla 2.4 Niveles de ruido según su ubicación. | 24 |
| Tabla 2.5 Niveles de ruido según su ubicación. | 25 |
| Tabla 4.1 Comparación de la medición realizada por los dispositivos. | 47 |
| Tabla 4.2 Costos para el desarrollo del proyecto..... | 52 |

RESUMEN

El presente proyecto hace referencia al diseño de un dispositivo inalámbrico para la medición del ruido en un espacio cerrado el cual utilizara un sensor de sonido KY038 para la recolección de datos de forma analógica. La medición del ruido es una técnica ambiental de mucha importancia pues existen lugares que tienen disposiciones reglamentarias sobre los peligros de un medio con ruido, existen parámetros para la medición que deben ser tomados en cuenta para la calibración del sensor KY038. El sensor recoge la señal y la amplifica para su posterior lectura con la tarjeta embebida ESP8266 que se comunica con el servidor MQTT de manera inalámbrica pues maneja la tecnología Wifi, además mediante el software Node-RED se propone realizar la programación mediante nodos que serán implementados para generar las alertas y comunicar los elementos que integran el sistema inalámbrico.

ABSTRACT

This project refers to the design of a wireless device for measuring noise in an enclosed space which uses an KY038 sound sensor to collect data in analogue form. The measurement of noise is a very important environmental technique as there are places that have regulatory provisions on the dangers of a noisy environment, there are parameters for the measurement that must be taken into account for the calibration of the KY038 sensor. The sensor collects the signal and amplifies it for subsequent reading with the ESP8266 embedded card that communicates with the MQTT server wirelessly as it uses Wifi technology. In addition, using the Node-RED software, it is proposed to carry out the programming using nodes that will be implemented to generate alerts and communicate the elements that make up the wireless system.

INTRODUCCIÓN

El presente proyecto muestra el desarrollo de un dispositivo IoT para el diseño e implementación de un dispositivo que permita la medición de ruido en ambientes cerrados el cual enviara una alerta si el ruido sobrepasa los niveles establecidos. A continuación, se describe los puntos a tratar para el desarrollo del proyecto:

En el capítulo 1, se muestran los ítems acordes al desarrollo del proyecto, así como sus generalidades y metodologías aplicadas.

En el capítulo 2, se indica parte de la fundamentación teórica en la que esta basado el proyecto según los elementos que lo componen

En el capítulo 3, se indica el diseño tanto de la red inalámbrica como la placa donde van a ir conectado el sensor de sonido.

En el capítulo 4, se describe las pruebas de funcionamiento realizadas para la comunicación del prototipo con la red inalámbrica y la alerta generada según los parámetros establecidos para la programación y recolección de datos en el puerto análogo del controlador ESP8266.

En el siguiente capítulo 5 se muestran las conclusiones y recomendaciones posterior al conocimiento adquirido durante todo el desarrollo del proyecto poniendo bases para posibles desarrollos que tengan que ver con la investigación realizada sobre los dispositivos IoT.

CAPÍTULO I

ANTECEDENTES

1.1 Planteamiento del problema

Día a día todas las personas son expuestas constantemente a ruido sonoro, por lo que el ruido se encuentra presente en todas partes, por ejemplo: en aulas de clases, laboratorios, habitaciones, salas de cine, etc. Estas actividades requieren de acciones preventivas para garantizar protección contra los efectos nocivos que generan los factores ambientales que inciden sobre la población. Los efectos negativos causados por la prolongada exposición al ruido sonoro son preocupantes para la sociedad, porque pueden dañar el oído, y causar molestias cuando este interfiere con información dada por personas que hablan de un tema o sonidos que vienen de algún aparato reproductor como televisiones, radios, reproductores hacia la persona.

1.2 Justificación

Considerando lo establecido por la Organización Mundial de la Salud se tiene que hacer conciencia en la población y advertir a las personas sobre la necesidad de adoptar las medidas necesarias para la protección en ambientes ruidosos. Cabe resaltar que la prolongada exposición a ruido intenso es una de las mayores causas de daños del oído, así como casos en que se provoca estrés, ansiedad e incluso ataques de pánico, así como otras enfermedades auditivas y de tipo psicológico; las causas del ruido pueden ser tratadas utilizando protección o disminuyendo las causas externas de su origen.

Por estos motivos es necesario el desarrollo del dispositivo IoT que permita medir la intensidad del ruido para la medición en tiempo real de la intensidad auditiva la cual se está expuesto a lo largo de todo el día, y también a la creación de alarmas para alertar a grupos de personas cuando existen altas intensidades de ruido que están fuera de los límites, logrando de esta manera que el usuario evite o elimine las fuentes de ruidos. El desarrollo de este dispositivo busca la utilización e innovación de la perspectiva tecnológica usándola no solo para un fin económico sino también para fines sociales, como lo es en este caso, ayudar a la salud auditiva de las personas.

1.3 Objetivos

1.3.1 Objetivo General

- Desarrollar un dispositivo IoT en cloud para la medición de la contaminación auditiva que existen en espacios cerrados.

1.3.2 Objetivos Específicos

- Analizar los requerimientos para la medición de niveles de ruido para el desarrollo del dispositivo.
- Desarrollar un dispositivo para la medición de niveles de ruido en IoT y cloud para el cumplimiento de los parámetros establecidos en el análisis de requerimientos.
- Realizar pruebas de funcionamiento para la comprobación de la comunicación entre el medidor de ruido auditivo y el dispositivo móvil.
- Analizar los costos de implementación del dispositivo de medición de ruido auditivo para la comprobación de su viabilidad económica.

1.4 Metodología

Metodología analítica: A través de este método se analizará las variables físicas que permitan realizar la búsqueda de los sensores y dispositivos requeridos para la realización del proyecto.

Metodología deductiva: de acuerdo con la información obtenida se diseñará un dispositivo IoT que permitirá enviar alertas a un dispositivo móvil.

Metodología experimental: Mediante la realización de pruebas se podrá verificar el correcto funcionamiento del dispositivo de medición de ruido auditivo para determinar los diferentes niveles de intensidad con respecto al ruido en ambientes cerrados.

CAPÍTULO II

FUNDAMENTACIÓN TEÓRICA

El presente capítulo describe mediante la fundamentación teórica y la investigación tanto los elementos que componen la red inalámbrica como los conceptos que sustentan la información recopilada para la construcción del proyecto.

2.1 Contaminación Acústica

El ruido ambiental o contaminación acústica se considera un factor importante en el impacto ambiental pues en las grandes ciudades el ruido me está presente día a día afectando a la población que se encuentra en el lugar los niveles sonoros por lo general superan en gran medida a los criterios permitidos y ya normas internacionales de tolerancia acústica afectando directamente el nivel de vida y trayendo consecuencias no deseadas. Según la Agencia de Protección Ambiental de Estados Unidos (EPA), el nivel seguro para no sufrir afectaciones auditivas permanentes, no debe exceder un valor promedio de 70[dB] durante 24 horas, o de 75[dB], 20 durante 8 horas, las viviendas deben garantizar un aislamiento acústico que permita en el interior, como mínimo, un nivel de 45[dB] (Niranjan, 2018). Cabe indicar que los principales efectos del ruido se han considerado como auditivos y extra auditivos; los efectos auditivos están en correlación a la pérdida de la capacidad auditiva de las personas expuestas (el daño auditivo no sólo depende de su nivel, sino de su duración, aunque se acepta que un medio ambiente sonoro por debajo de 70[dB] no es dañino para la salud auditiva); los no auditivos son los que pueden generar estrés por perturbar el sueño, por ser trastornadores de las actividades humanas cotidianas o por efectos en el comportamiento humano (MacHeso, 2021).

Actualmente la contaminación ambiental por medio del ruido es un grave problema en la sociedad ya que produce repercusiones de manera fisiológica física y social esto se demuestra en múltiples de estudios e informes que se han hecho a lo largo del tiempo por expertos. Son muchos los sectores de la población que empiezan a reconocer públicamente que la contaminación acústica es tan peligrosa para la salud como la contaminación atmosférica (Negishi, 2018).

2.2 Ruido

El ruido es toda señal no deseada que es transmitida sin ser aceptada esto depende mucho del tipo de percepción que tenga la persona, sin embargo, desde el punto de vista ocupacional el ruido tiene características no deseadas que desencadenan daños en la salud se puede decir que es el resultado de muchos tipos de perturbaciones en una banda de frecuencia. Fisiológicamente se considera que el sonido es cualquier sonido desagradable o molesto (Shi, 2018).

2.2.1 Clasificación del ruido

El ruido está clasificado en dos secciones según la intensidad y el periodo y la frecuencia.

Según la intensidad y el periodo el ruido puede ser fluctuante esto quiere decir que puede estar variando de manera periódica en el tiempo también existe el ruido impulsivo que es muy fuerte pero breve en el tiempo. Según la frecuencia el ruido puede ser blanco, su principal característica es una distribución uniforme en el espectro audible con frecuencias constantes. El ruido rosa tiene una intensidad que decae aproximadamente 3[dB], por octava y el ruido marrón que existe de manera natural compuesto por ondas graves y medias. (Mitrpanont, 2018).

2.2.2 Efectos del ruido

Al exponer a una persona al ruido se pueden ocasionar múltiples consecuencias entre las cuales están la interferencia de la comunicación, estrés y perturbación del sueño. De los peligros a la salud causados por el ruido, el más notable suele ser la pérdida auditiva que ha sido científicamente observada, medida, y establecida con un efecto de los impactos sonoros excesivos (Mitrpanont, 2018).

La pérdida de la audición se puede dar de manera temporal o permanente pues el ruido provoca un desgaste en las personas además de dificultades para dormir, unos estudios muestran que la perturbación del sueño llega a niveles que sobrepasan los límites de la serenidad, existen diversos efectos negativos que se pueden dar si no se controla de

manera adecuada. Para otras personas aún más susceptibles, los ruidos podrían ser un factor agravante en enfermedades cardiacas y en otras enfermedades(Chen & Lin, 2017).

El ruido actúa como distracción ya afecta al estado del individuo, este puede modificar el estado de alerta y aumentar o disminuir la eficiencia de la persona de tal forma que las consecuencias pueden ser irreversibles ya sea en la infancia juventud o adultez el ruido también ocasiona dificultad en el aprendizaje y mala salud, en la edad avanzada las personas suelen tener dificultad en conciliar el sueño y en descansar (Swett, 2020).

2.2.3 Fuentes de ruido

Industria. El ruido actúa como distracción ya afecta al estado del individuo, este puede modificar el estado de alerta y aumentar o disminuir la eficiencia de la persona de tal forma que las consecuencias pueden ser irreversibles ya sea en la infancia juventud o adultez el ruido también ocasiona dificultad en el aprendizaje y mala salud (Zinkevich, 2021).

Es producida por los motores a combustión y la fricción que existe entre el vehículo y el aire ya sea en motores pequeños o grandes causando ruidos molestos y contaminación ambiental

Construcción de obras civiles. - En las construcciones el ruido es permanente pues es producido por la maquinaria que existe según el trabajo que estén realizando y que tan cerca se encuentran los individuos de la fuente ruidosa.

2.3 Salud y ruido

Con respecto a la salud el ruido generalmente es desagradable, en el medio ambiente es molesto y fuente de enfermedad, sí hace referencia a la contaminación auditiva cuando el ruido sobrepasa ciertos niveles de intensidad.

Enfermedades fisiológicas: estas enfermedades son producidas por el exceso de ruido generando alteraciones en el sueño estrés falta de rendimiento y en muchos casos

agresividad (Jean, 2018).

Enfermedades psíquicas: Producidas por exceso de ruido, se pueden citar el estrés, las alteraciones del sueño, disminución de la atención, depresión, falta de rendimiento o agresividad. Enfermedades sociológicas: alteraciones en la comunicación, el rendimiento, etc.

La contaminación auditiva es la causante de enfermedades por lógicas y nerviosas La contaminación acústica también provoca algunos trastornos de la salud como el insomnio, dolores de cabeza, ataques al corazón entre otros (Jean, 2018).

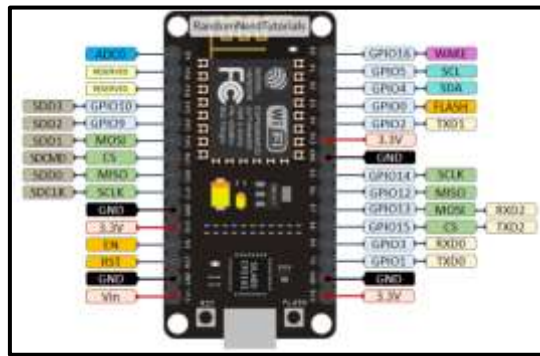
2.4 Software Node-Red

Esta herramienta permite al usuario un entorno de programación de forma interactiva por medio del uso de nodos tanto de publicación como suscripción. Mediante el tablero gráfico se controla los dispositivos y se monitorea el estado de los mismos dentro del mismo dominio, los nodos son subrutinas con entradas y salidas que pueden recibir mensajes por MQTT o por su significado en inglés Message Queuing Telemetry Transport. (Ono, 2019).

2.5 Node MCU ESP8266

El controlador ESP8266 tiene la capacidad de conectarse a una red inalámbrica pues posee varios modos de conexión y conectividad entre estos el Wi-Fi que permiten realizar múltiples aplicaciones pues operan de la misma forma que un Arduino sin embargo es necesario implementar librerías para su correcto funcionamiento. Con esto los clientes pueden disfrutar de conectividad en la nube, operación de bajo consumo de energía y soporte para seguridad Wi-Fi (Delgado, 2021).

Figura 0.1 Node MCU ESP8266.



Fuente: (Valencia & Delgado, 2021)

Tabla 0.1 Características de funcionamiento del controlador ESP8266

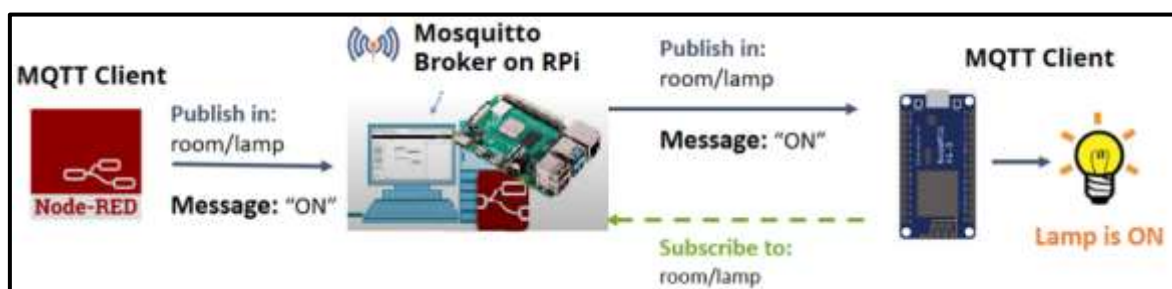
| | |
|---------------------------|--------------------------------|
| Voltaje de funcionamiento | 3,3[V] |
| consumo de corriente | 10 [uA]-170 [uA] |
| tipo de procesador | Tensilica L106 32bit |
| velocidad del procesador | 80-160[Mhz] |
| GPIOs | 17 |
| Analógico y Digital | 1 entrada 10bit (1024 valores) |
| Soporte 802,11 | b/g/n/d/e/i/k/r |
| tipo de conexión | Wifi |

Fuente: (Valencia & Delgado, 2021)

2.6 Conexión entre Bróker y Node-red

Es un software que tiene la función de herramienta en la cual permite la conexión entre múltiples dispositivos de hardware con API y servicios de conectividad en línea. El software no de red se enfoca en dar soluciones por medio del internet de las cosas realizando programación y enlazando dispositivos mediante nodos, esta conexión se realiza por medio de una interfaz gráfica que está instalada en él broker, permitiendo un manejo fácil al usuario y generando varios bloques de programación que permiten el almacenamiento, envío y procesamiento de datos. Por lo tanto, hace que el desarrollo sea más rápido en programas de IoT, tal como se requiere en este proyecto (Alsmirat, 2019).

Figura 0.2 Conexión de bróker y Node-red

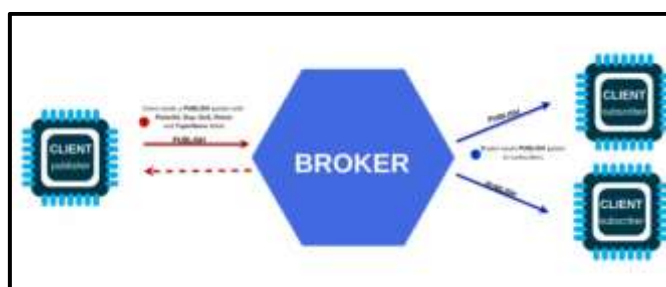


Fuente: (Barrientos ., 2020)

2.7 Protocolo MQTT

El protocolo MQTT posee características específicas para soluciones IoT a través de una comunicación inalámbrica que usa un bajo ancho de banda y un enlace a la red mediante suscripción cliente-servidor. En el caso de MQTT las conexiones se mantienen abiertas y se “reutiliza” en cada comunicación. Es una diferencia, por ejemplo, a una petición HTTP 1.0 donde cada transmisión se realiza a través de conexión (Cheng, 2017).

Figura 0.3 Funcionamiento del protocolo MQTT.

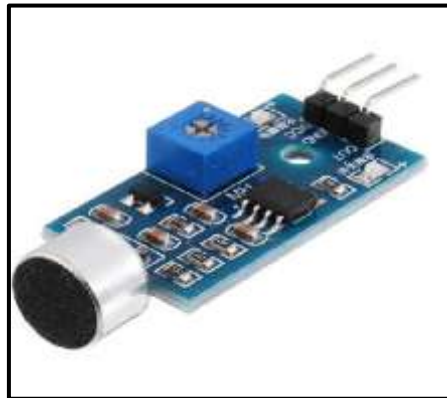


Fuente: (Valencia & Delgado, 2021)

2.8 Sensor de sonido KY038

El sensor de sonido es una tarjeta de pequeñas dimensiones para aplicaciones donde se necesite sensar la intensidad del sonido o ejecutar alguna instrucción posterior a detectar un ruido. El módulo cuenta con ganancia ajustable para el micrófono para así tener una mejor lectura además cuenta con tres pines para conectarlo con tu Arduino, microcontrolador o cualquier tarjeta de desarrollo (Macheso, 2021).

Figura 0.4 Sensor KY038



Fuente: Recuperado de: (Barrientos ., 2020)

Analógica (A0): Lleva toda la información que está detectando el micrófono.

Las salidas digitales son utilizadas cuando se requiere un encendido o apagado mediante la programación de sus puertos además es posible ajustar el umbral para realizar diversas aplicaciones.

El sensor que se utiliza posee una unidad de muestreo en la parte principal del módulo este permite enviar la señal de forma analógica hacia el controlador además posee un amplificador de señal de acuerdo a un valor de Resistencia en él potenciómetro enviará la señal analógica del módulo. El tercer componente es un comparador que apaga la salida digital y el LED si la señal cae por debajo de un valor específico (Zinkevich, 2021).

Tabla 0.2 Características de funcionamiento.

| | |
|--------------------------------|----------------|
| Voltaje de trabajo: | 3[V] y 5[V]. |
| Tipo de señal del codificador: | Análoga |
| Temperatura de funcionamiento: | 0° C ~ + 70° C |

Fuente: (Zinkevich, 2021).

2.9 Medidor de sonido AS804

El medidor digital de sonido se ha diseñado para realizar la medición en varios lugares y diferentes entornos con un diseño compacto con las siguientes características que se indican en la tabla 2.3. Este sensor tiene una excelente sensibilidad y precisión con una

pantalla retroiluminada que permite una lectura aun en ambientes de oscuridad su lectura se hace en decibelios y es utilizado en varias pruebas de sonido ya sea de forma industrial o en el hogar.

Figura 0.5 Sensor AS804



Fuente: Recuperado de: (Valencia & Delgado, 2021)

Tabla 0.3 Características de funcionamiento del sensor AS804.

| Medidor de sonido AS804 | |
|-------------------------|----------------------|
| Rango de Medición | 30-130 [dB] |
| Resolución | 0,1 [dB] |
| Precisión | 1,5 [dB] |
| Rango de frecuencia | 31,5 [Hz] a 8,5 [Hz] |

Fuente: Recuperado de: (Barrientos ., 2020)

2.10 Nivel de ruido

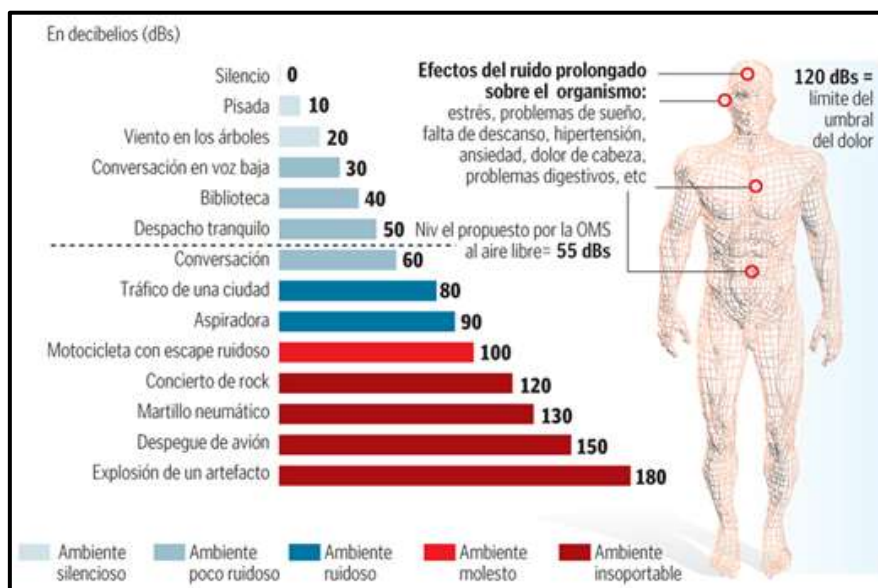
El oído es un órgano complejo que puede dañarse si se somete a niveles de ruido extremos. Las células ciliadas, que se encuentran dentro de nuestro oído, son las encargadas de convertir el sonido en impulsos nerviosos para que el cerebro la intérprete. Un aumento de volumen recae sobre las mismas y provoca deterioro con el tiempo, perdiendo sus facultades y la capacidad de restituirse (Jean, 2018)..

A partir de 100 [dB], el sonido empieza a ser molesto. Al tener 120 decibeles prácticamente el sonido es insoportable y agota de forma rápida, el oído reacciona cuando él sonido tiene cierta intensidad creando una protección natural, sin embargo, al estar expuesto por 8 horas al día la sensación es igual de fuerte por el tiempo de exposición a fuentes de sonido con mucha intensidad. En conclusión, se puede afirmar que las personas

pueden llegar aguantar sonidos de 110 [dB] por minutos (Zinkevich, 2021).

Las personas pueden ser expuestas a sonidos fuertes durante tiempos cortos con una intensidad superior a 85 decibeles, sin embargo, es recomendable el uso de protecciones auditivas para salvaguardar la salud. Los protectores auditivos, entre otras cosas, amortiguan la intensidad del sonido a niveles más bajos (Cheng, 2019).

Figura 0.6 Salud y nivel de ruido.



Fuente: (Valencia & Delgado, 2021) Recuperado de: <https://naylampmechatronics.com/>

2.11 Niveles permisibles de ruido

Como es de conocimiento existen varios niveles de ruido que se pueden permitir dependiendo de la zona en la que se esté haciendo la medición pues es necesario tomar en consideración los controles ambientales para zonas comerciales, industriales, residenciales o zonas hospitalarias como se indica en la tabla 2.4.

Tabla 0.4 Niveles de ruido según su ubicación.

| TIPO DE ZONA SEGÚN USO | NIVEL DE PRESIÓN SONORA EQUIVALENTE [dB] | |
|-------------------------------|--|------------------|
| | DE 06h00 A 20h00 | DE 20h00 A 06h00 |
| Zona hospitalaria y educativa | 45 | 35 |
| Zona Residencial | 50 | 40 |

| | | |
|------------------------|----|----|
| Zona Residencial mixta | 55 | 45 |
| Zona Comercial | 60 | 50 |
| Zona Comercial mixta | 65 | 55 |
| Zona Industrial | 70 | 65 |

Fuente: (Delgado, 2021)

Un factor a considerar es el ruido que producen los automóviles pues son fuente de contaminación sonora de igual manera se establecen niveles máximos y mínimos tolerables los cuales se presentan en la tabla 2.5.

Tabla 0.5 Niveles de ruido según su ubicación.

| CATEGORÍA DE VEHÍCULO | DESCRIPCIÓN | NPS MAXIMO [dB] |
|-----------------------|--|-----------------|
| Motocicletas: | De hasta 200 centímetros cúbicos. | 80 |
| | Entre 200 y 500 [c.c.] | 85 |
| | Mayores a 500 [c.c.] | 86 |
| Vehículos: | Transporte de personas, nueve asientos, incluido el conductor. | 80 |
| | Transporte de personas, nueve asientos, incluido el conductor, y peso no mayor a 3.5 toneladas. | 81 |
| | Transporte de personas, nueve asientos, incluido el conductor, y peso mayor a 3.5 toneladas. | 82 |
| | Transporte de personas, nueve asientos, incluido el conductor, peso mayor a 3,5 toneladas, y potencia de motor mayor a 200 [hp]. | 85 |
| Vehículos de Carga: | Peso máximo hasta 3.5 toneladas | 81 |
| | Peso máximo de 3.5 toneladas hasta 12.0 toneladas | 86 |
| | Peso máximo mayor a 12.0 toneladas | 88 |

Fuente: (Delgado, 2021)

CAPÍTULO III

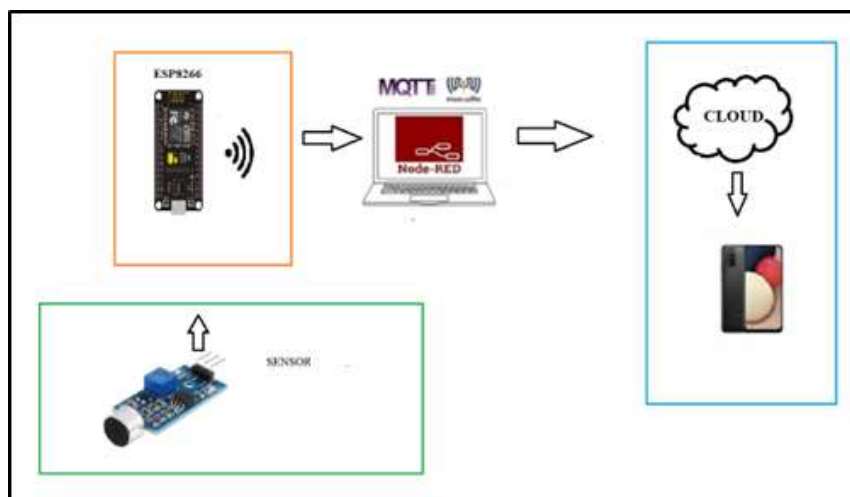
DISEÑO DE LA RED INALÁMBRICA

En este capítulo se describe a detalle el desarrollo y diseño de un dispositivo capaz de captar el sonido y clasificarlo según su nivel en un ambiente cerrado generando así una alerta si existe contaminación auditiva.

3.1 Diseño del dispositivo IoT

El dispositivo IoT está conformado por un controlador ESP8266 el cual es un módulo con la capacidad de conectarse a una red inalámbrica para un envío y recepción de datos a este se suma el sensor de sonido KY038 de alta sensibilidad que es capaz de captar los niveles de sonido en un rango de frecuencia de 30 a 15000 [Hz]. Estos dos elementos son los encargados de la recepción de datos y el acondicionamiento de la señal análoga para posteriormente asociarlos a un servidor MQTT que será el canal de comunicación entre la placa y la red inalámbrica capaz de generar una alerta si existe ruido en el lugar donde este ubicado el dispositivo. A continuación, en la Figura 3.1 se muestra el diagrama de bloques para la comunicación de los elementos.

Figura 0.1 Diagrama de Bloques para la comunicación.



Elaborado por: Ana Carlosama y José Alanuca.

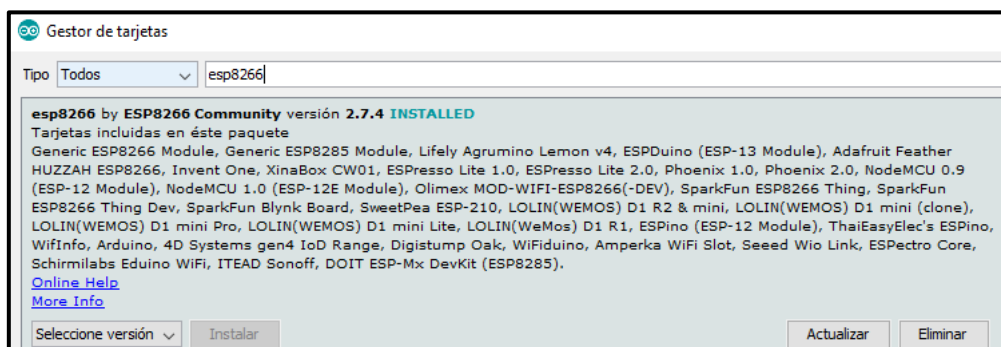
3.2 Bloque Controlador ESP8266

Para el desarrollo del proyecto se utiliza un controlador ESP8266 el cual permite enlazar el sensor de sonido con el servidor y la aplicación móvil mediante el uso de la red inalámbrica con tecnología Wi-Fi ya que se puede hacer un intercambio de datos de forma bidireccional utilizando las características del controlador y su pin analógico para posteriormente administrar los datos en el software.

3.3 Comunicación Node Mcu ESP8266 y Node-Red

Para conectar el software Node Red a la tarjeta controladora es necesario instalar varias librerías y drivers en el IDE de Arduino, así como la placa utilizar dependiendo de la versión cómo se observa en la figura 3.2. Es necesario ingresar al gestor de tarjetas para descargar el ESP8266 de esta manera el software puede reconocer a la tarjeta.

Figura 0.2 Gestor de tarjetas en el IDE de Arduino.



Elaborado por: Ana Carlosama y José Alanuca.

Para la programación es necesario instalar librerías que ya fueron diseñadas con anterioridad una de estas permite acceder al protocolo MQTT dentro de la red IoT piden que la búsqueda del dispositivo por medio de una IP generada, dentro de la programación se debe establecer el nombre de la red a la que se va a acceder, así como la contraseña y la IP del servidor.

En caso de haber algún tipo de interferencia se crea una subrutina la cual va a recolectar las veces que sea necesaria para enlazar el dispositivo al bróker para el proyecto es necesario suscribirse a un módulo de función con el topic “proyecto/medición/sonido”
Cómo se puede observar en la figura 3.3

Figura 0.3 Código para conectividad ESP8266.

```
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("ESP82661")) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.publish("esp2", "Enviando el primer mensaje");
      // ... and resubscribe
      client.subscribe("proyecto/medicion/sonido");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

Elaborado por: Ana Carlosama y José Alanuca.

3.4 Instalación de Node-Red

El software Node Red es una herramienta que trabaja junto con librerías destinadas a realizar un trabajo de manera más simple ofreciendo varios servicios para la comunicación de la red con tarjetas controladoras y dispositivos inteligentes uno de los pasos más importantes para su instalación empieza con la descarga de Node JS como se indica en la figura 3.4

Figura 0.4 Página de Node JS

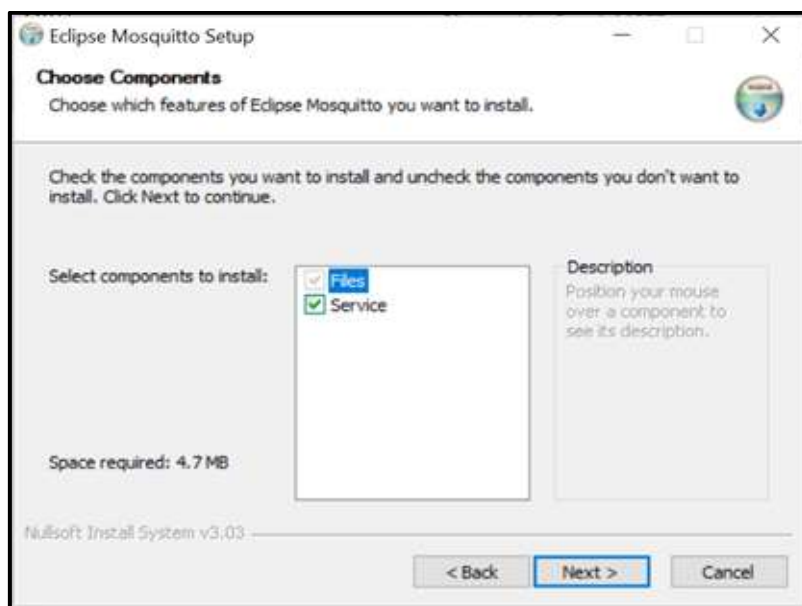


Elaborado por: Ana Carlosama y José Alanuca.

Luego de realizar la instalación se procede a la descarga de paquetes npm que son asociados al software estos permiten ejecutar el CMD por medio del comando node-red, de esta manera empezará con la instalación de nodos e interfaces y la identificación del equipo dentro de la red asignada por lo general está nombrada en la ubicación IP 127.0.0.1 en el puerto local 1880.

Es necesario utilizar los nodos para la comunicación por lo cual el servidor MOSQUITTO debe ser instalado con el protocolo MQTT siendo instalado como se indica en la figura 3.5

Figura 0.5 Configuración como servicio de Windows



Elaborado por: Ana Carlosama y José Alanuca.

El servidor necesita ciertas librerías a modo de dependencias dependiendo del equipo en el que se va instalar como se muestra en la figura 3.6 la arquitectura del equipo es fundamental para elegir la librería.

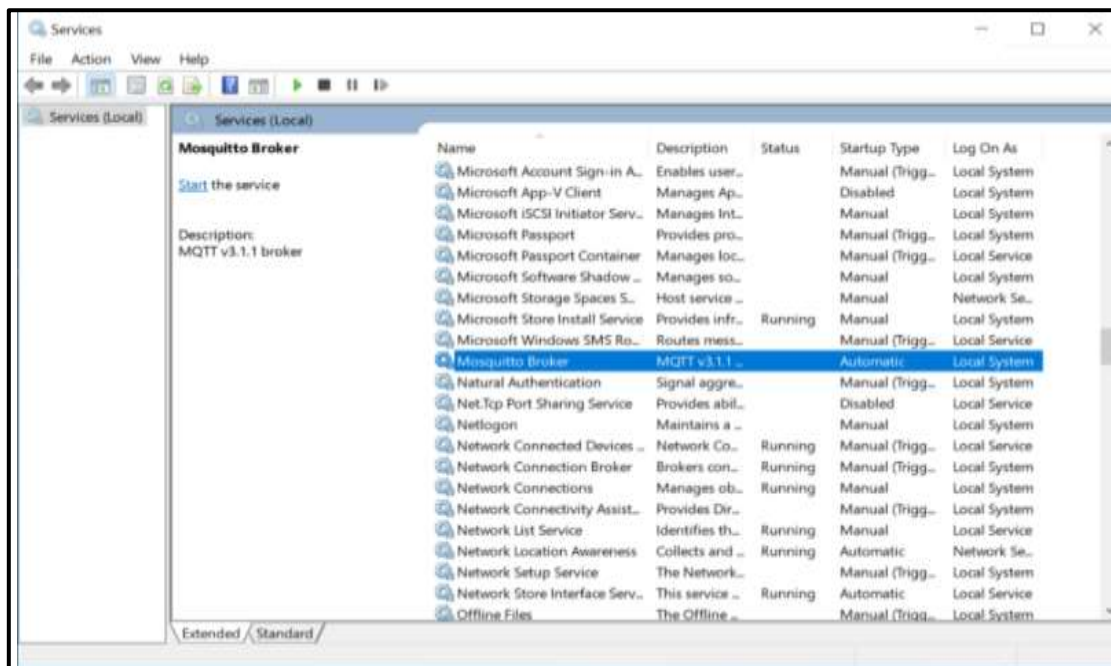
Figura 0.6 Instalación de librerías de Open ssl

| Nombre | Tamaño | Fecha de modificación |
|------------------------|---------|-----------------------|
| [directorio principal] | | |
| read5.man | 293 B | 5/2/15 0:00:00 |
| pthreadsGC2.dll | 117 kB | 27/5/12 0:00:00 |
| pthreadsGCE2.dll | 119 kB | 27/5/12 0:00:00 |
| pthreadsVC2.dll | 54.5 kB | 27/5/12 0:00:00 |
| pthreadsVCE2.dll | 60.5 kB | 27/5/12 0:00:00 |
| pthreadsVSE2.dll | 56.0 kB | 27/5/12 0:00:00 |
| sha512.man | 866 B | 5/2/15 0:00:00 |

Elaborado por: Ana Carlosama y José Alanuca.

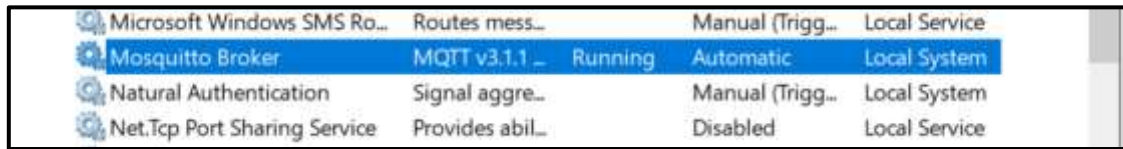
Para que el servicio empiece es necesario activar el Bróker desde la aplicación de Services de Windows como se indica en las Figuras 3.7 y 3.8.

Figura 0.7 Aplicación Services Windows



Elaborado por: Ana Carlosama y José Alanuca.

Figura 0.8 Activación del servicio

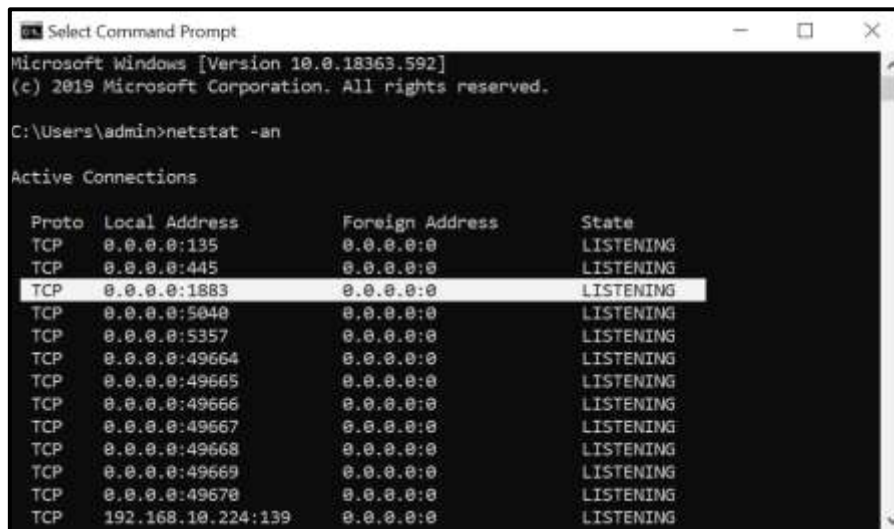


| Service Name | Path | Status | Startup Type | Log On As |
|------------------------------|------------------------|----------------|------------------|---------------------|
| Microsoft Windows SMS Ro... | Routes mess... | Stopped | Manual (Trigg... | Local Service |
| Mosquitto Broker | MQTT v3.1.1 ... | Running | Automatic | Local System |
| Natural Authentication | Signal aggre... | Stopped | Manual (Trigg... | Local System |
| Net.Tcp Port Sharing Service | Provides abil... | Stopped | Disabled | Local Service |

Elaborado por: Ana Carlosama y José Alanuca.

Si se desea comprobar el estado de la conexión o se puede utilizar el comando “netstat -an” para despegar las características de escucha en el puerto asignado en este caso el TCP 1883 como se indica en la figura 3.9.

Figura 0.9 Comprobación de puerto



```
Microsoft Windows [Version 10.0.18363.592]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\admin>netstat -an

Active Connections

Proto Local Address           Foreign Address         State
TCP   0.0.0.0:135              0.0.0.0:0               LISTENING
TCP   0.0.0.0:445              0.0.0.0:0               LISTENING
TCP   0.0.0.0:1883            0.0.0.0:0               LISTENING
TCP   0.0.0.0:5040             0.0.0.0:0               LISTENING
TCP   0.0.0.0:5357             0.0.0.0:0               LISTENING
TCP   0.0.0.0:49664            0.0.0.0:0               LISTENING
TCP   0.0.0.0:49665            0.0.0.0:0               LISTENING
TCP   0.0.0.0:49666            0.0.0.0:0               LISTENING
TCP   0.0.0.0:49667            0.0.0.0:0               LISTENING
TCP   0.0.0.0:49668            0.0.0.0:0               LISTENING
TCP   0.0.0.0:49669            0.0.0.0:0               LISTENING
TCP   0.0.0.0:49670            0.0.0.0:0               LISTENING
TCP   192.168.10.224:139      0.0.0.0:0               LISTENING
```

Elaborado por: Ana Carlosama y José Alanuca.

3.5 Comunicación MQTT en la WNCS

Si se desea integrar el protocolo a MQTT en la red es necesario codificar en el editor del programa mediante los nodos al enlazar múltiples dispositivos mediante el puerto 1880 como se observa en la figura 3.10 se debe tomar en cuenta que el servidor debe arrancar en la dirección especificada.

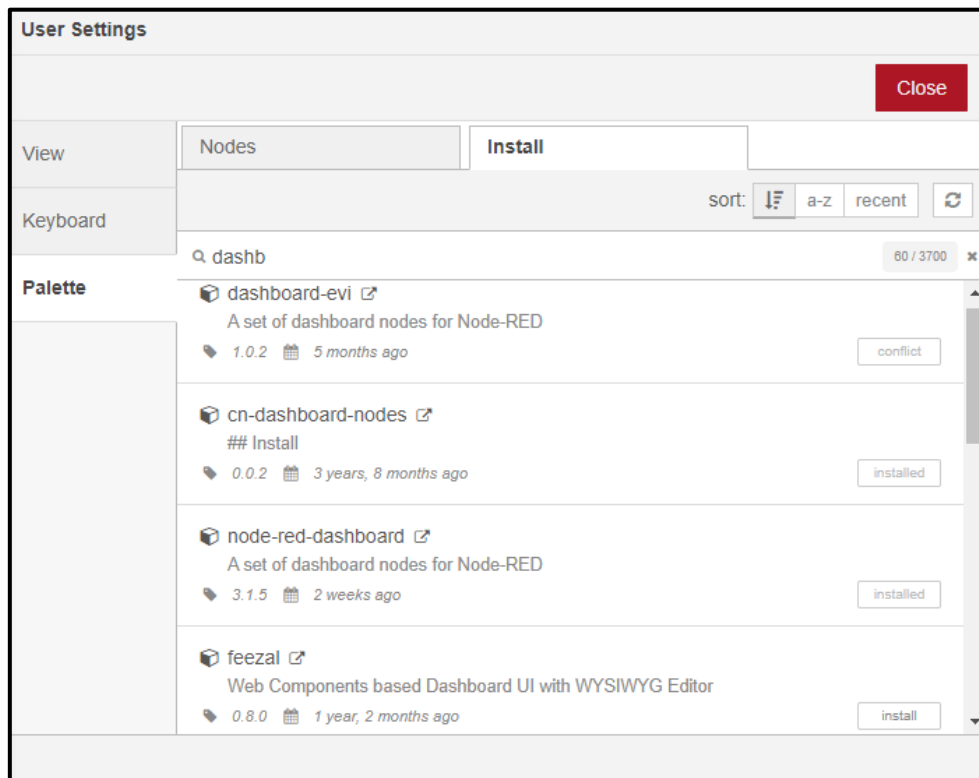
Figura 0.10 Acceso al software NODE-RED

```
C:\Users\Usuario>node-red
11 Feb 01:19:04 - [info]
Welcome to Node-RED
*****
11 Feb 01:19:04 - [info] Node-RED version: v1.1.3
11 Feb 01:19:04 - [info] Node.js version: v12.18.3
11 Feb 01:19:04 - [info] Windows_NT 10.0.18362 x64 LE
11 Feb 01:19:05 - [info] Loading palette nodes
11 Feb 01:19:06 - [warn] [node-red-contrib-telegrambot] Node module cannot be loaded on this version. Requires: >=1.3.7
11 Feb 01:19:06 - [info] Dashboard version 2.29.0 started at /ui
11 Feb 01:19:06 - [info] Settings file : \Users\Usuario\.node-red\settings.js
11 Feb 01:19:06 - [info] Context store : 'default' [module=memory]
11 Feb 01:19:06 - [info] User directory : \Users\Usuario\.node-red
11 Feb 01:19:06 - [warn] Projects disabled : editorTheme.projects.enabled=false
11 Feb 01:19:06 - [info] Flows file : \Users\Usuario\.node-red\flows_DESKTOP-VP97RWN.json
11 Feb 01:19:06 - [info] Server now running at http://127.0.0.1:1880/
11 Feb 01:19:06 - [warn]
```

Elaborado por: Ana Carlosama y José Alanuca.

Al iniciar el servicio y realizar la programación gráfica puede observar diferentes tipos de nodos uno de los principales es el nodo Dashboard el cual permite observar en tiempo real de forma gráfica los datos recibidos desde el sensor, de esta manera se puede obtener una gráfica con valores en tiempo real siempre y cuando se complete las configuraciones del nodo cómo se indica en la figura 3.11.

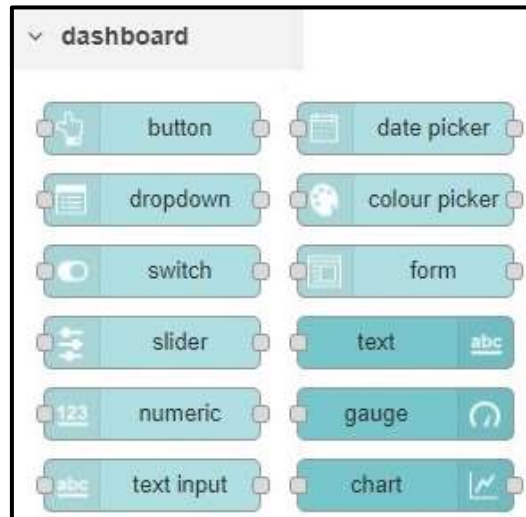
Figura 0.11 Librerías NODE-RED para Dashboard



Elaborado por: Ana Carlosama y José Alanuca.

Luego de instalar la librería se puede observar varios nodos que aparecen en la interfaz de desarrollo estos permiten realizar múltiples acciones de manera gráfica para mostrar al usuario de una manera más interactiva cómo se puede observar en la figura 3.12.

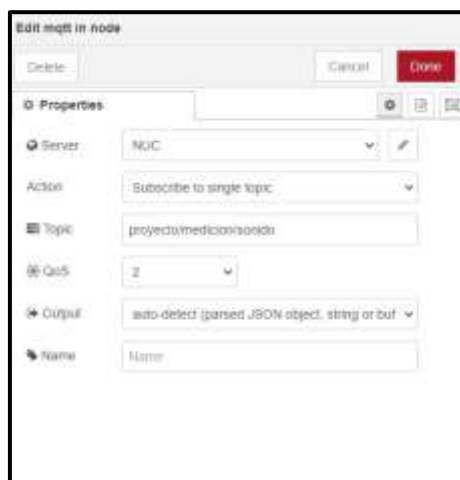
Figura 0.12 Nodos para la programación de interfaces gráficas.



Elaborado por: Ana Carlosama y José Alanuca.

Para realizar un intercambio de datos en la red es necesario enlazar los dispositivos mediante los nodos de entrada y salida MQTT estos a la vez van a identificar los datos y los dispositivos mediante el uso de topic para seleccionar una entrada y salida de datos cómo se observa en la figura 3.13

Figura 0.13 Configuración de nodos MQTT



Elaborado por: Ana Carlosama y José Alanuca.

3.6 Creación de parámetros para MQTT.

Al realizar un monitoreo se deben configurar varios nodos dentro del Dashboard para obtener un nivel apropiado del sonido medido en tiempo real además es posible visualizar el comportamiento de la variable que se está midiendo mediante el nodo CHART como se muestra en la figura 3.14

Figura 0.14 Configuración de nodos CHART



Elaborado por: Ana Carlosama y José Alanuca.

Para identificar un valor sujeto a una variable física y que se ha medido en tiempo real es necesario utilizar otro nodo indicador GAUGE el mismo que puede medir valores máximos y mínimos separados entre colores como se indica en la figura 3.15

Figura 0.15 Configuración de nodos GAUGE



Elaborado por: Ana Carlosama y José Alanuca.

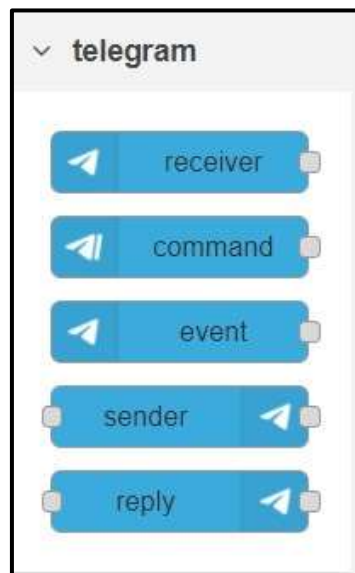
Al agregar estos nodos los dispositivos conectados pueden generar una interfaz muy fácil de manejar para el usuario donde puede observar el comportamiento de la variable a través del tiempo.

3.7 Alertas por Telegram

El uso de la plataforma Telegram permite enviar alertas a las personas que se encuentren en el lugar o fuera de él con alarmas establecidas mediante mensajes creados para los usuarios finales. De esta forma es posible prevenir a las personas de posibles fallos en una red o elementos nocivos en el ambiente.

Para la configuración de estas alarmas con la plataforma Telegram es necesario instalar una librería denominada “node-red-contrib-telegrambot” Mismo que permite crear nodos para una conexión directa con el servidor de Telegram como se indica en la figura 3.16.

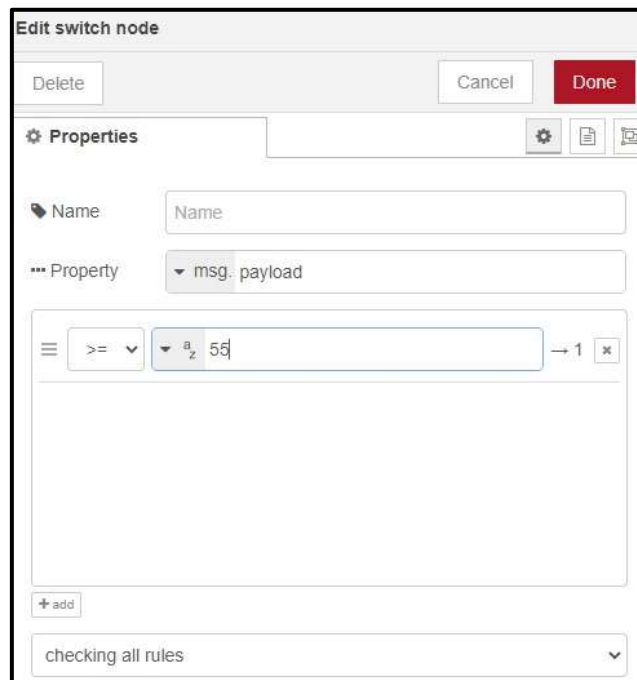
Figura 0.16 Nodos de comunicación con TELEGRAM.



Elaborado por: Ana Carlosama y José Alanuca.

Posterior a la instalación de la librería es necesario crear elementos de conexión llamados boots directamente en la aplicación de Telegram de esta manera se puede generar una alerta con un aviso a manera de texto para esto se fija un valor predeterminado el cual debe ser seteado en el nodo como se observa en la figura 3.17

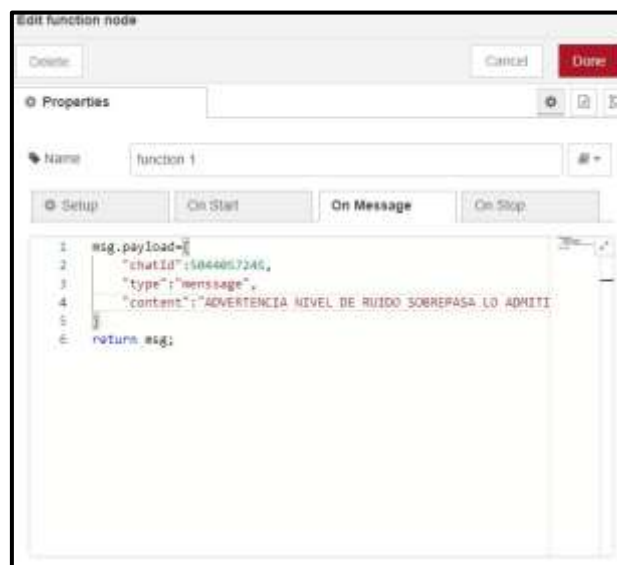
Figura 0.17 Configuración de nodo SWITCH



Elaborado por: Ana Carlosama y José Alanuca.

Al crear el mensaje para el usuario o el grupo de usuarios se debe crear la advertencia en el nodo Gracias a la programación en java de este modo el mensaje queda establecido” ADVERTENCIA EL NIVEL DE RUIDO SOBREPASA LO ESTABLECIDO” cómo se indica en la figura 3.18

Figura 0.18 Programación de nodo FUNCIÓN



Elaborado por: Ana Carlosama y José Alanuca.

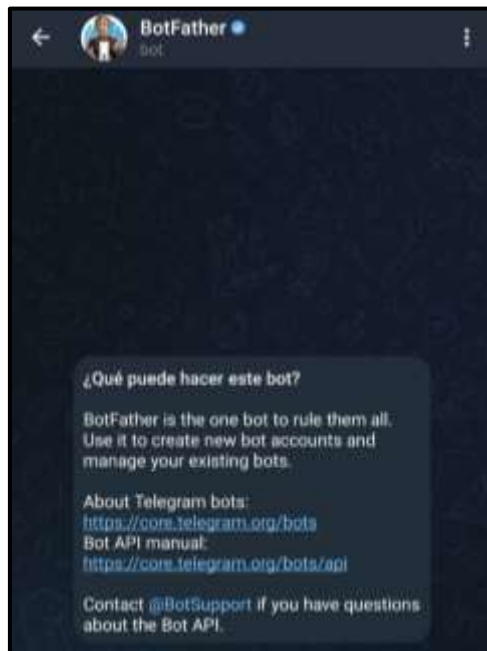
Al configurar las características de los nodos, se podrá enviar advertencias a la plataforma de Telegram a una o varias personas que se encuentren en el lugar.

3.8 Configuración de Bot Telegram

Se propone la integración de mensajería Telegram con el software Node-red para lo cual es necesario configurar un llamado Bot para el envío y recepción de los mensajes con el fin de obtener el enlace entre los dos terminales.

Se inicia con la creación del Bot por medio del comando BotFather este desplegara una pantalla como se indica en la Figura 3.19.

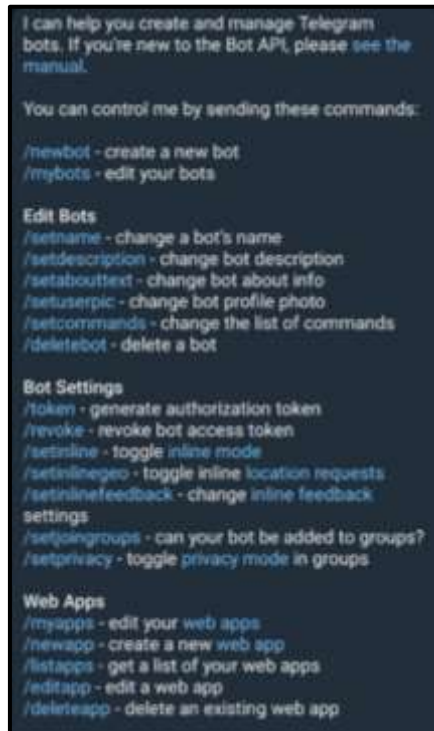
Figura 0.19 Configuración de BotFather.



Elaborado por: Ana Carlosama y José Alanuca.

Se procede a iniciar con la asignación del Bot posterior a esto se despliega las opciones que se pueden dar para la configuración de la mensajería enviando el comando /newbot.

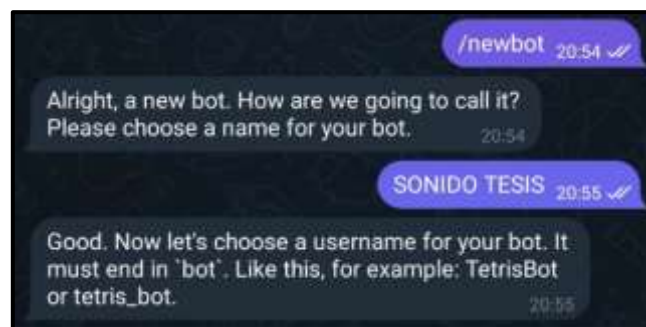
Figura 0.20 Opciones para la configuración de BotFather.



Elaborado por: Ana Carlosama y José Alanuca.

Es necesario crear un nombre de usuario que no exista previamente en este caso utilizo el nombre Tesis_sonido.bot.

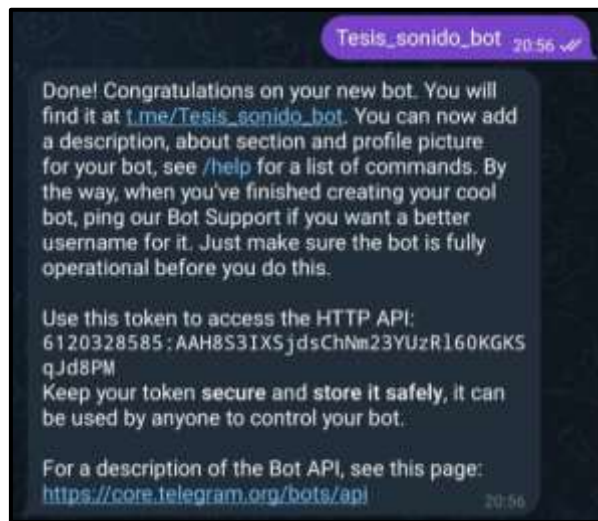
Figura 0.21 Nombre para despliegue de chat



Elaborado por: Ana Carlosama y José Alanuca.

Una vez creado el chat se genera un código API que se ingresara en Node-red para el envío y recepción de datos además de un usuario clave como se indica en la Figura 3.22

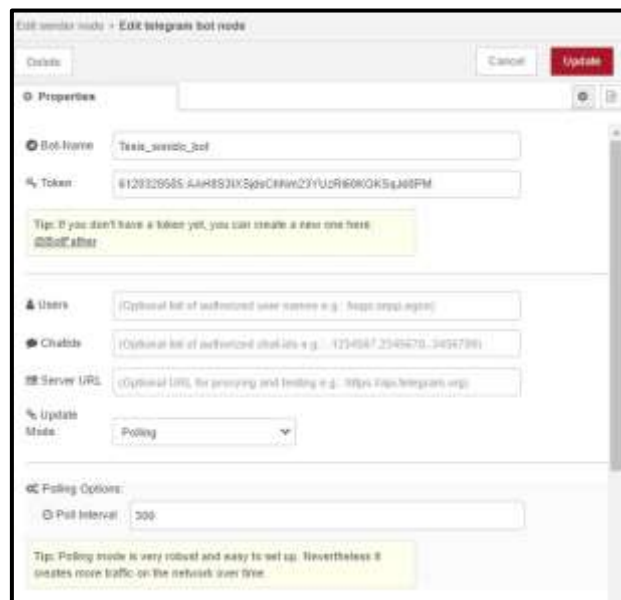
Figura 0.22 Código API generado.



Elaborado por: Ana Carlosama y José Alanuca.

Existe un nodo en el software node-red que permite el enlace con Telegram en este se debe llenar el Bot-Name y el Token que fueron generados previamente como se indica en la Figura 3.23.

Figura 0.23 Nodo Telegram.



Elaborado por: Ana Carlosama y José Alanuca.

De esta manera se crea un chat en el cual se puede interactuar con el software y por ende con el sensor de sonido teniendo un apartado para el envío de datos como se indica en la Figura 3.24.

Figura 0.24 Chat generado para envío de notificaciones.



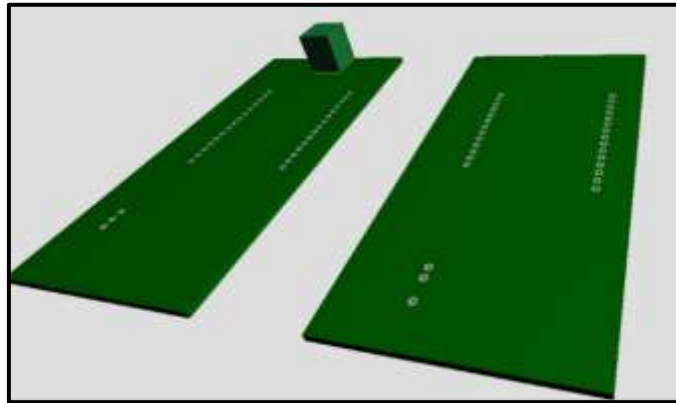
Elaborado por: Ana Carlosama y José Alanuca.

3.9 Diseño de la placa

Es importante realizar un diseño de la placa para el montaje del controlador mediante el uso del software Proteus, hoy esta placa está provista por pistas que conectan el sensor y el voltaje de alimentación además de pines de lectura por medio de un puerto análogo.

En la figura 3.25 se observa el modelo que se utiliza para colocar el controlador visto en una impresión 3 d para posteriormente materializarlo mediante el uso de la baquelita.

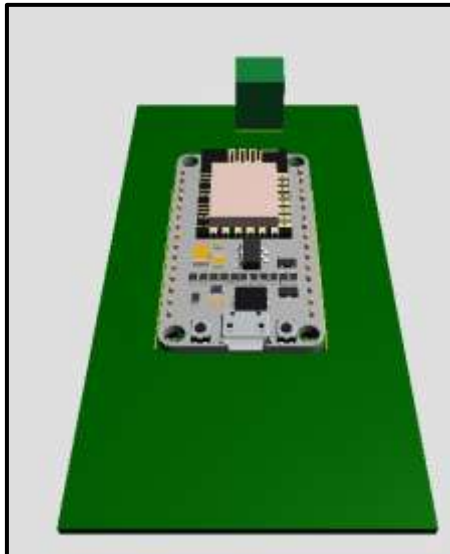
Figura 0.25 Modelo en 3D (Software Proteus).



Elaborado por: Ana Carlosama y José Alanuca.

En la Figura 3.26 se muestra el montaje completo del ESP8266 desde una vista superior, se debe verificar que las pistas tengan continuidad pues muchas veces pueden existir saltos o cortes.

Figura 0.26 Montaje de ESP8266 en la placa.



Elaborado por: Ana Carlosama y José Alanuca.

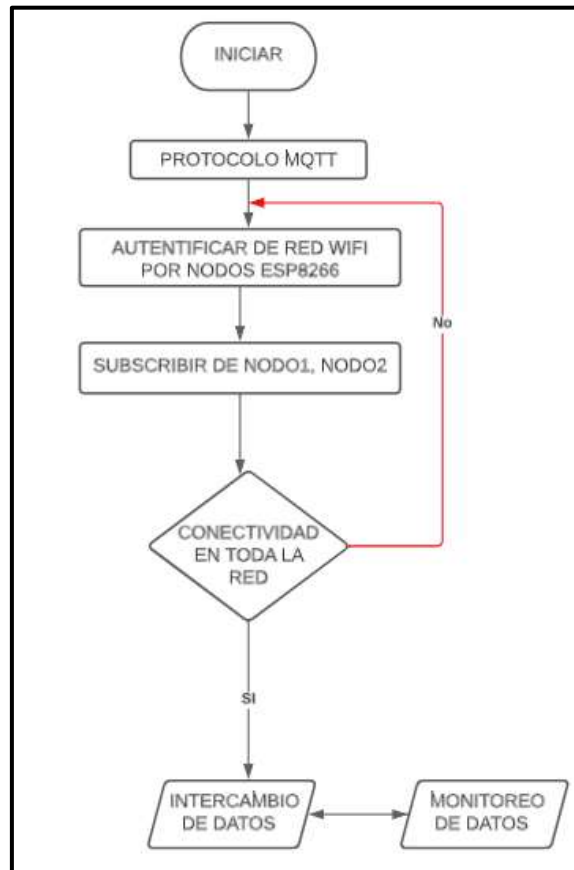
3.10 Diseño del prototipo

Se realiza un diseño para almacenar tanto la tarjeta controladora ESP8266, así como el sensor LM393 procurando que no sufran ningún daño con las dimensiones que se indican en el Anexo 3 realizadas en el Software AutoCAD 3D que tiene como principal función ser el contenedor del dispositivo electrónico.

3.11 Diagrama de flujo

En la figura 3.27 se muestra el diagrama de flujo utilizado para la adquisición de información del sensor hasta el software Node red Para el monitoreo de la variable analógica que es capaz de intercambiar datos de forma inalámbrica.

Figura 0.27 Diagrama de flujo para adquisición de datos.

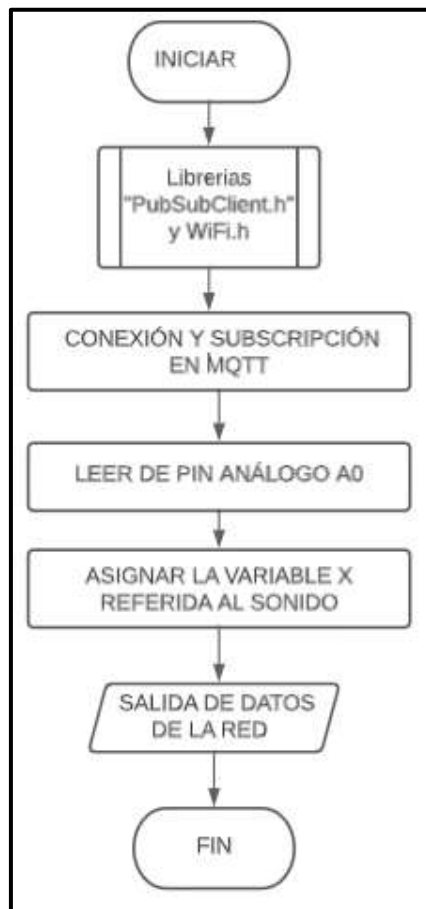


Elaborado por: Ana Carlosama y José Alanuca.

Con respecto a la lectura del sensor es necesario instalar librerías como Wifi.h y PubSubClient.h Las cuales permiten suscribirse al servidor y deben ser instaladas en el ID de Arduino para una compilación y carga sin errores además de acondicionar el valor analógico según se requiera.

El protocolo MQTT permite una comunicación bidireccional desde el servidor hasta los sensores realizados mediante la suscripción de los nodos y todos los dispositivos que lo conforman.

Figura 0.28 Diagrama de flujo para adquisición de datos.



Elaborado por: Ana Carlosama y José Alanuca.

CAPÍTULO IV

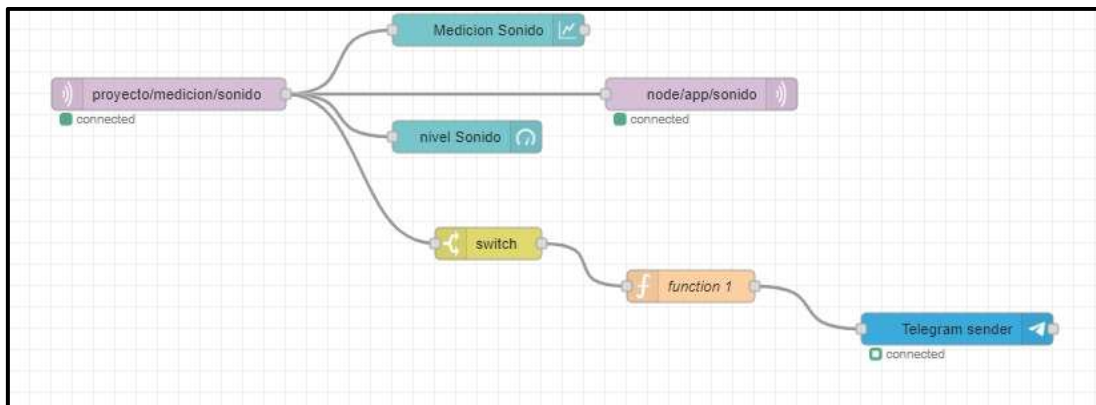
IMPLEMENTACIÓN Y PRUEBAS

Este capítulo detalla los procesos realizados para la implementación de prototipo de un medidor de ruido mediante el uso y programación del software NODE-RED y la programación realizada en el ide de Arduino para la recolección de datos brindados por el sensor KY038.

4.1 Nodos usados en la interfaz Node-red

Para tener un correcto envío y recepción de datos en la red inalámbrica se necesita usar una programación de nodos propia de uno de red los cuales envían datos mediante el protocolo MQTT y son mostrados en las Dashboard mediante el uso de nodos gráfico. cómo se indica en la figura 4.1 la programación de los nodos está conectados entre sí para el envío y flujo de datos desde el módulo ESP8266 hasta los nodos chart y gauge además de los nodos de Telegram.

Figura 0.1 Nodos para recepción y envío de datos.



Elaborado por: Ana Carlosama y José Alanuca.

Para la conexión de los nodos se utiliza el protocolo MQTT el mismo que permite escuchar a través de mensajes enviados a la red. El nombre que se utiliza en el topic es el siguiente: “sonido/esp8266/node”. En la Figura 4.2 se indica como poner en marcha el servidor MQTT para que el software Node-red pueda ser ejecutado en el ordenador para el uso de los nodos y la respectiva programación.

Figura 0.2 Ventana para ejecutar Node-red

```
Welcome to Node-RED
=====
25 May 20:46:53 - [info] Node-RED version: v3.0.2
25 May 20:46:53 - [info] Node.js version: v18.16.0
25 May 20:46:53 - [info] Windows_NT 10.0.19045 x64 LE
25 May 20:46:57 - [info] Loading palette nodes
25 May 20:47:03 - [info] Dashboard version 3.4.0 started at /ui
25 May 20:47:03 - [info] Settings file : C:\Users\javie\.node-red\settings.js
25 May 20:47:03 - [info] Context store : 'default' [module=memory]
25 May 20:47:03 - [info] User directory : \Users\javie\.node-red
25 May 20:47:03 - [warn] Projects disabled : editorTheme.projects.enabled=false
25 May 20:47:03 - [info] Flows file : \Users\javie\.node-red\flows.json
25 May 20:47:03 - [info] Server now running at http://127.0.0.1:1880/
```

Elaborado por: Ana Carlosama y José Alanuca.

En la Figura 4.3 se indica los nodos que pueden ser utilizados en la programación dependiendo del caso o la necesidad de programación el nodo inject es utilizado para la entrada y salida de datos al leer una variable enviada desde el controlador por medio del ESP8266 estos datos pueden ser vistos en la Dashboard del ordenador mediante el nodo chart o gauge que sirven para observar de manera grafica los valores obtenidos.

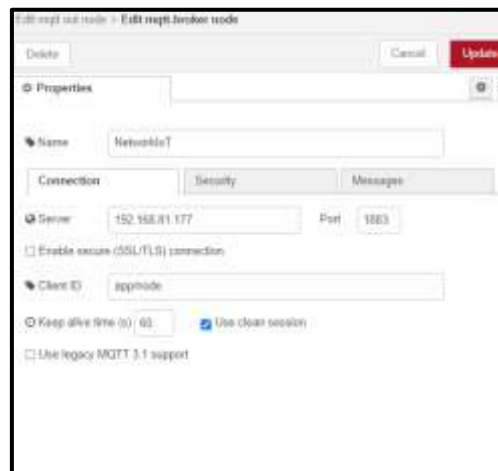
Figura 0.3 Tipos de nodos para la programación.



Elaborado por: Ana Carlosama y José Alanuca.

En la Figura 4.4 se muestra la configuración que se debe realizar en el nodo, se debe tener en cuenta que los nodos deben compartir la IP del servidor para el envío y recepción de datos, así como un nombre o ID en específico para saber a cuál se está dirigiendo los datos así también se designa el puerto de escucha que para este caso es el 1883.

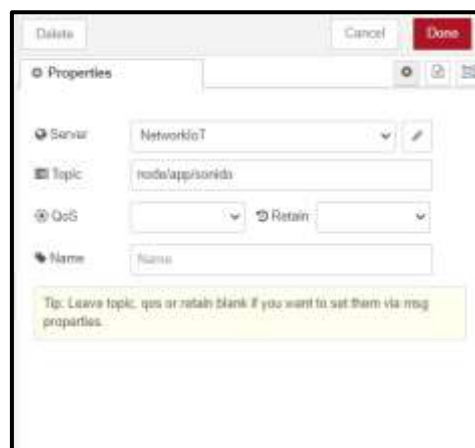
Figura 0.4 Ventana de configuración para nodos.



Elaborado por: Ana Carlosama y José Alanuca.

Los nodos deben tener el nombre del servidor y el topic por el cual van a ser designados los datos tanto de entrada como salida que también es descrito en la programación utilizada en el IDE de Arduino.

Figura 0.5 Nodo función.



Elaborado por: Ana Carlosama y José Alanuca.

4.2 Características del sonómetro AS804

Es necesario conocer otros dispositivos que realizan un trabajo similar por lo que el sonómetro más accesible en el mercado es el AS804 este puede medir niveles de ruido con un rango de 30 [dB] a 130 [dB] lo que permite medir sonidos altos y bajos con una precisión de -1.5 [dB], en el Anexo 9 se indican más características asociadas al funcionamiento del dispositivo.

4.3 Análisis de resultados

Se realiza una comparación entre el dispositivo diseñado a partir de un sensor KY-038 y un sonómetro AS804 realizando múltiples mediciones durante un día laborable en el lapso de 8:00 am a 17:00 pm para saber cuál es la exactitud del dispositivo creado y el margen de error que este maneja tomando como punto de partida las mediciones del dispositivo adquirido, estas mediciones se realizan en el local ubicado en Conocoto con los datos que indica la tabla 4.1.

Tabla 0.1 Comparación de la medición realizada por los dispositivos.

| | Hora Aproximada | Sensor AS804[dB] | Sensor KY038 [dB] | Error [%] |
|----|--------------------|---------------------|----------------------|--------------|
| 1 | 8:00 | 53,9 | 52,6 | 2,41 |
| 2 | 8:20 | 62,1 | 59,1 | 4,83 |
| 3 | 8:40 | 55,3 | 54,6 | 1,27 |
| 4 | 9:00 | 66,4 | 64,3 | 3,16 |
| 5 | 9:20 | 60,2 | 58,7 | 2,49 |
| 6 | 9:40 | 56,7 | 55,8 | 1,59 |
| 7 | 10:00 | 67,2 | 66,5 | 1,04 |
| 8 | 10:20 | 53,1 | 51,4 | 3,20 |
| 9 | 10:40 | 50,4 | 49,6 | 1,59 |
| 10 | 11:00 | 67,3 | 65,3 | 2,97 |
| 11 | 11:20 | 59,3 | 57,5 | 3,04 |
| 12 | 11:40 | 68,2 | 67,1 | 1,61 |
| 13 | 12:00 | 59,4 | 57,8 | 2,69 |
| 14 | 12:20 | 68,3 | 66,4 | 2,78 |
| 15 | 12:40 | 64,7 | 62,9 | 2,78 |
| 16 | 13:00 | 66,9 | 65,7 | 1,79 |
| 17 | 13:20 | 53,2 | 51,8 | 2,63 |
| 18 | 13:40 | 59,3 | 57,6 | 2,87 |
| 19 | 14:00 | 62,5 | 60,4 | 3,36 |
| 20 | 14:20 | 56,3 | 54,8 | 2,66 |
| 21 | 14:40 | 57,6 | 55,1 | 4,34 |
| 22 | 15:00 | 65,4 | 64,2 | 1,83 |
| 23 | 15:20 | 67,4 | 65,6 | 2,67 |
| 24 | 15:40 | 63,2 | 60,2 | 4,75 |
| 25 | 16:00 | 63,5 | 62,5 | 1,57 |
| 26 | 16:20 | 60,2 | 58,3 | 3,16 |
| 27 | 16:40 | 65,3 | 63,9 | 2,14 |
| 28 | 17:00 | 64,5 | 62,8 | 2,64 |

Elaborado por: Ana Carlosama y José Alanuca.

Se toman un total de 28 datos en el día pensado en el horario laboral de las personas que están en el lugar con lapsos de tiempo de 20 minutos por medición, se debe tener en cuenta que las mediciones se realizan de manera instantánea y siempre están sujetas a variaciones por diversos factores. Luego de tomar los datos se puede observar las diferentes mediciones en varios instantes de tiempo en el día notando que el promedio en decibeles que recibe el lugar es de 61.41[dB] aproximadamente medidos con el sonómetro AS804, las mediciones con el prototipo son similares sin embargo tienen un margen de error en cada medición por lo es necesario saber el error en promedio según la tabla con la siguiente fórmula.

$$\bar{x} = \frac{\sum x}{n}$$
$$\bar{x} = 3.46\%$$

Dónde:

n= número de intentos.

x= dato medido en cada intento

\bar{x} = media aritmética

Con estas mediciones el error relativo es de $\pm 3.46\%$ aproximadamente en los datos analizados teniendo en cuenta un factor importante como la latencia de la red pues los datos son enviados de un lugar a otro pasando por la red local del sitio donde se realiza la medición, en la figura 4.6 se indica el tiempo que demora en recibir los datos la red inalámbrica al hacer ping desde el servidor al Gateway de la red a diferencia del sonómetro que toma sus datos de forma directa.

Figura 0.6 Comando ping de entre el servidor y el Gateway.

```
Respuesta desde 192.168.1.55: bytes=32 tiempo=97ms TTL=128
Respuesta desde 192.168.1.55: bytes=32 tiempo=5ms TTL=128
Respuesta desde 192.168.1.55: bytes=32 tiempo=5ms TTL=128
Respuesta desde 192.168.1.55: bytes=32 tiempo=15ms TTL=128
```

Elaborado por: Ana Carlosama y José Alanuca.

Continuando con el análisis se procede hacer diferentes mediciones en el local comercial ubicado en el sector de Conocoto el mismo que tiene un considerable flujo de automóviles

que generan ruido, se parte de la premisa que el nivel de ruido no debe exceder los 55 [dB] según la OMS, se realizan las mediciones correspondientes a diferentes horas del día como se indica en la Figura 4.7.

Figura 0.7 Interfaz gráfica Node-red medición por la mañana.



Elaborado por: Ana Carlosama y José Alanuca.

Como se observa en la interfaz de Node-red los diferentes valores muestran una lectura similar en decibeles y una gráfica de los valores medidos en tiempo real, al realizar la medición en la mañana se muestra que los valores están dentro de lo establecido sin embargo al tener el valor seteado en 55 [dB] se enviara la notificación como se indica en la Figura 4.10.

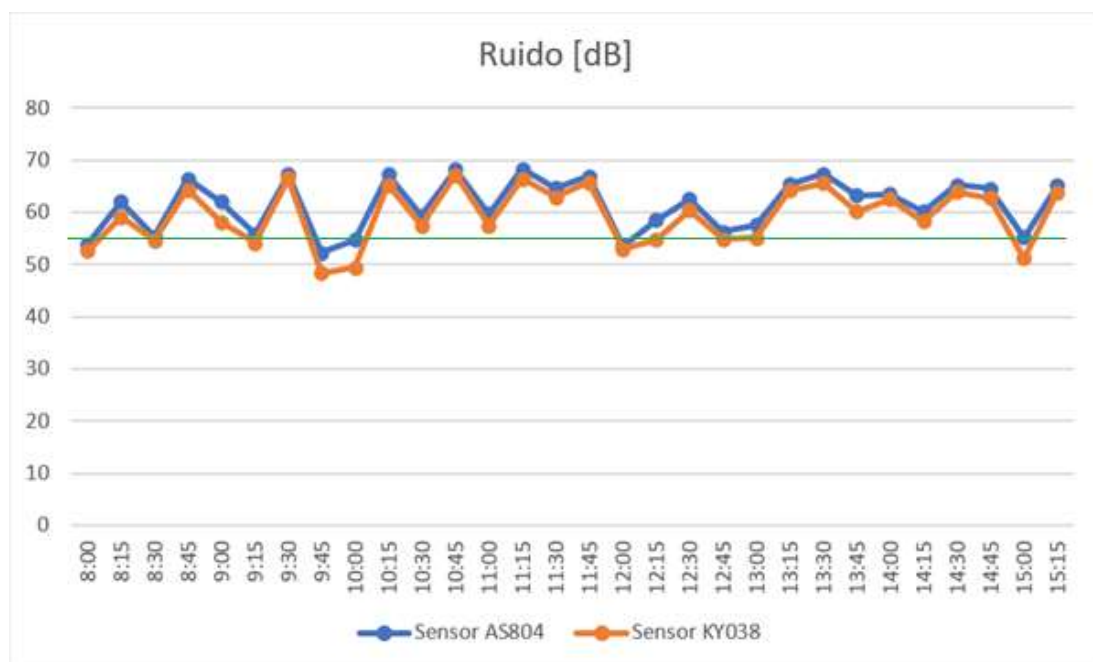
Figura 0.8 Interfaz gráfica Node-red medición en la tarde.



Elaborado por: Ana Carlosama y José Alanuca.

La medición realizada por la tarde muestra un aumento en el nivel de decibeles, al ser un lugar con mucha afluencia de vehículos se constató que casi todo el día los valores están oscilando en un rango de 50 a 70 [dB].

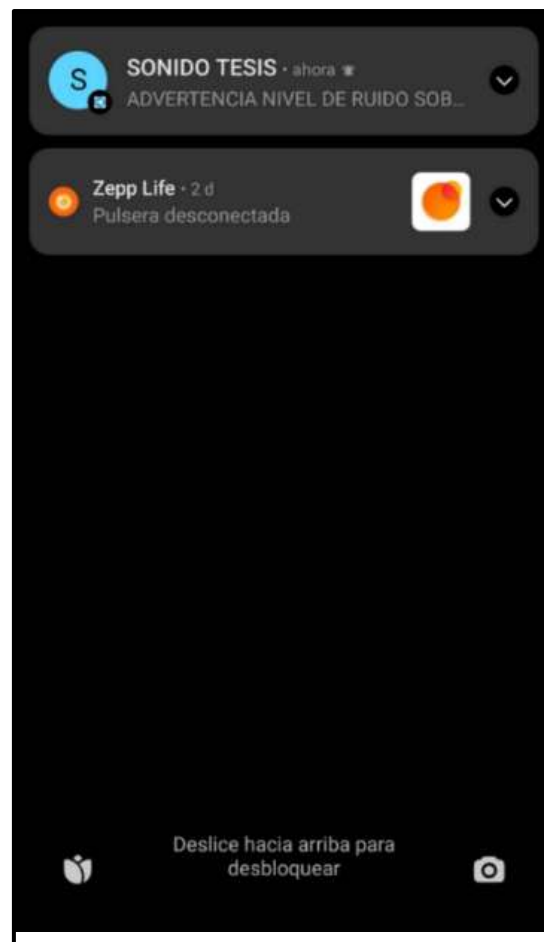
Figura 0.9 Comportamiento de los sensores de sonido.



Elaborado por: Ana Carlosama y José Alanuca.

En la figura 4.9 se indica el comportamiento de los sensores durante el día asemejándose en sus lecturas, al estar sobre el rango de los 55 [dB] se va tener un constante envío de las alertas para tratar de minimizar el ruido.

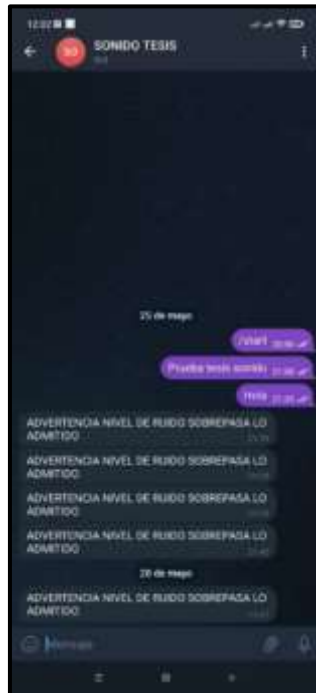
Figura 0.10 Notificación de mensaje de Telegram entregada



Elaborado por: Ana Carlosama y José Alanuca.

En la Figura 4.11 se muestra los mensajes enviados a la interface del smartphone donde se da la advertencia que los niveles de ruido están excediendo lo normal para lo cual debe cumplir con un proceso de envío y recepción de datos desde el sensor hasta la plataforma Telegram.

Figura 0.11 Mensajes recibidos en Telegram



Elaborado por: Ana Carlosama y José Alanuca.

4.4 Análisis económico del proyecto

Para la implementación del prototipo se deben tomar en cuenta el costo de los materiales para su fabricación, así como el sistema inalámbrico implementado para un control y monitoreo de la variable de control.

Tabla 0.2 Costos de materiales para el proyecto.

| DESCRIPCIÓN | COSTOS |
|---------------------------|-----------|
| sensor KY034 | \$ 12,00 |
| ESP8266 | \$ 10,00 |
| Placa | \$ 8,00 |
| Access Point | \$ 64,00 |
| Protección para circuitos | \$ 20,00 |
| Batería 5V | \$ 4,00 |
| Total | \$ 118,00 |

Elaborado por: Ana Carlosama y José Alanuca.

En la tabla 4.3 se indican los costos generados por los diseños y configuración de la red inalámbrica y el tiempo invertido en su desarrollo.

Tabla 4.3 Costos para el desarrollo de la red inalámbrica.

| DESCRIPCIÓN | TIEMPO | C.UNIT | COSTOS |
|------------------------------------|-----------|---------|-----------|
| Diseño y configuración de nodos | 140 horas | \$ 1.75 | \$ 245,00 |
| Personal especializado | 40 horas | \$ 1.75 | \$ 70,00 |
| Pruebas de conectividad | 20 horas | \$ 1.75 | \$ 35,00 |
| Capacitación al personal del lugar | 8 horas | \$ 1.75 | \$ 14,00 |
| | | TOTAL | \$ 364,00 |

Elaborado por: Ana Carlosama y José Alanuca.

Para lograr montar la red inalámbrica es necesario hacer uso de varios elementos así como personal calificado que cumplen con la función de técnicos para la implementación del sistema, al ser un prototipo los costos son altos pues son hechos de forma individual por lo que el valor de implementación del sistema solo en materiales es de 118 dólares adicional a esto se suma el valor de construcción y desarrollo que es de 364 dólares por concepto de configuración y diseño, para la elaboración del proyecto por lo tanto el valor total es de 482 dólares por lo que en la actualidad no sería viable ya que existen medidores de sonido con costos desde los 30 dólares, sin embargo se toma en cuenta que el medidor de sonido realizado puede monitorear el ruido de manera prolongada en una interfaz gráfica con las alertas respectivas siendo este una mejor opción dependiendo del uso que se le quiera dar.

Es de conocimiento que el uso de este dispositivo alerta sobre los niveles de ruido en el lugar siendo un incentivo al cuidado de la salud para las personas que laboran en el lugar, se toma un valor base de 200 dólares asignado a cada persona que trabaja en el lugar para gastos médicos, si el ruido trae problemas en la salud de las 3 personas que laboran se tendrá un costo negativo de 600 dólares que se pueden evitar al tomar acciones según el uso del dispositivo, si la implementación del proyecto tiene un costo de 482 dólares se puede observar el benéfico que resulta según la ecuación

$$ROI = \frac{\text{Ingreso} - \text{Inversion}}{\text{Inversion}}$$
$$ROI = \frac{600 - 482}{482} * 100 = 24.48\%$$

Donde: ROI es el retorno de inversión

El proyecto es capaz de generar un ahorro de casi el 25% del costo destinado a problemas de salud por concepto de daños auditivos.

4.5 Relación costo beneficio

Si se destina un valor de 600 dólares al año por gastos de salud de los empleados se evalúa el costo beneficio en 4 años donde la tasa interna de oportunidad es del 25%, cuando el costo beneficio es igual a 1 se puede decir que es indiferente el proyecto, si el costo beneficio es mayor a uno el proyecto es aceptable y si el costo beneficio es menor a 1 se puede rechazar el proyecto, se realiza el cálculo según la tabla 4.4

Tabla 4.4 Costos Beneficio para el proyecto.

| Taza interna de oportunidad | Activo en 4 años | Beneficio | Costo | B/C |
|-----------------------------|------------------|------------|-------|------|
| 25% | \$2400 | \$1.416,96 | \$480 | 2,95 |

Elaborado por: Ana Carlosama y José Alanuca.

Con respecto a la relación costo beneficio el proyecto es aceptable ya que supera el valor de 1 según el análisis realizado.

CONCLUSIONES

- El dispositivo inalámbrico mide los niveles de ruido en [dB] y proporciona un monitoreo constante en tiempo real a través de una red local instalada en el lugar tomando como datos la lectura análoga del sensor KY038.
- El sistema inalámbrico tiene un servidor MQTT que administra los datos enviados por el sensor además mediante la programación en Node Red es capaz de generar una interfaz gráfica en el ordenador siendo este quien envía las alertas a el teléfono cuando se excede los 55 [dB].
- Al realizar las mediciones correspondientes con el sonómetro AS804 y el dispositivo inalámbrico se observa un margen de error relativo de un 3.46% en promedio con respecto a los 30 datos tomados en el día además las lecturas en el lugar muestran un rango de 50 [dB] a 70 [dB] estos valores se asemejan a el tráfico de una ciudad.
- El análisis económico indica que el dispositivo tiene un costo de 482 dólares por concepto de materiales, diseño e implementación, si se toma en cuenta que existen dispositivos similares en el mercado que tienen un costo menor se puede decir que no es factible sin embargo el medidor de sonido realizado cumple con varios requerimientos como una medición prolongada, una interfaz de monitoreo y alertas generadas a través de mensajes de texto en cualquier lugar.
- El desarrollo de una red inalámbrica local tiene como valor agregado el uso de mensajería en la nube como un modo de alerta o aviso en este caso cuando sobrepasa el nivel de ruido esto es importante pues se puede llegar a todas las personas ya sea que estén en el lugar o no.

RECOMENDACIONES

- Es necesario conocer los niveles admisibles de ruido en un ser humano pues al estar expuesto a niveles altos por un tiempo prolongado puede traer problemas a la salud por lo que es necesario disminuir en lo posible.
- Es recomendable revisar la posición y distancia tanto del sonómetro como del dispositivo para obtener mediciones reales ya que pueden existir objetos que bloqueen la medición del sonido.
- La respuesta que da el dispositivo está sujeta a la velocidad de la red por lo que se recomienda tener una red lo más cercana posible y estable caso contrario habrá retardos en la medición.
- Para la toma de mediciones es necesario escoger valores representativos y el promedio de varias mediciones identificando los puntos máximos y mínimos para establecer si el ruido sobrepasa los niveles normales.

REFERENCIAS BIBLIOGRÁFICAS

- Alsmirat, M., Jaraweh, Y., Institute of Electrical and Electronics Engineers. Spain Section, Universidad Politécnica de Valencia, Universidad de Granada, Technische Universität Graz, Staffordshire University, Jordan University of Science & Technology, & Institute of Electrical and Electronics Engineers. (n.d.). *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IoTSMS) : Granada, Spain, October 22-25, 2019.*
- Chen, H. W., & Lin, F. J. (2014). Converging MQTT Resources in ETSI Standards Based M2M Platform. *2014 IEEE International Conference on Internet of Things (IThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, 292–295. <https://doi.org/10.1109/iThings.2014.52>
- Institute of Electrical and Electronics Engineers, & IEEE Sensors Council. (n.d.). *IEEE Sensors 2017 : October 29-November 1, 2017, Glasgow, Scotland, UK, Scottish Event Campus (SEC) : 2017 conference proceedings.*
- MacHeso, P., Manda, T. D., Chisale, S., Dzupire, N., Mlatho, J., & Mukanyiligira, D. (2021a). Design of ESP8266 Smart Home Using MQTT and Node-RED. *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*, 502–505. <https://doi.org/10.1109/ICAIS50930.2021.9396027>
- MacHeso, P., Manda, T. D., Chisale, S., Dzupire, N., Mlatho, J., & Mukanyiligira, D. (2021b). Design of ESP8266 Smart Home Using MQTT and Node-RED. *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*, 502–505. <https://doi.org/10.1109/ICAIS50930.2021.9396027>
- Mitrpanont, J. L., Sawangphol, Wudhichart., & Institute of Electrical and Electronics Engineers. (n.d.). *Proceeding of 2018 Seventh ICT International Student Project Conference (ICT-ISPC) : July 11-13, 2018, Faculty of ICT, Mahidol University, Nakhon Pathom, Thailand.*
- Negishi, Y., & Kawaguchi, N. (n.d.). *Instant Learning Sound Sensor: Flexible Environmental Sound Recognition System.*
http://www.xbow.com/Products/Wireless_Sensor_Networks.htm
- Niranjan, S. K., Institute of Electrical and Electronics Engineers, Institute of Electrical and Electronics Engineers. Bangalore Section, & IEEE Computational Intelligence Society. Bangalore Chapter. (n.d.). *Proceedings of the Second International Conference on Green Computing and Internet of Things (ICGCIoT 2018) : 16-18 August 2018, Karnataka,*

India.

- Ono, N., Saito, A., & Ando, S. (n.d.). *Design and Experiments of Bio-mimicry Sound Source Localization Sensor with Gimbal-Supported Circular Diaphragm.*
- Ranganathan Engineering College, organizer., & Institute of Electrical and Electronics Engineers. (n.d.). *Proceedings of the International Conference on Inventive Communication and Computational Technologies : ICICCT 2018 : 20-21, April 2018.*
- Shi, J. (n.d.). *Determining Direction of A Noise Source in Water with Four-sensor Vector Sound Intensity Arrays of Arbitrary Geometry.*
- Swett, D. W. (2020). Experimental Characterization of a Piezoelectric Leaf-Cell Sensor for Simultaneous Fluid Density and Sound Speed Measurement. *IEEE Sensors Letters*, 4(5).
<https://doi.org/10.1109/LENS.2020.2982616>
- Valencia, W. G., & Delgado, C. E. (2021). *DISEÑO E IMPLEMENTACIÓN DE PROTOTIPO IOT PARA EL MONITOREO REMOTO DE LA CALIDAD DEL AGUA PARA LA CRIANZA DE TILAPIAS EN ESTANQUES.* UNIVERSIDAD POLITÉCNICA SALESIANA.
- Zinkevich, A. v. (2021). ESP8266 microcontroller application in wireless synchronization tasks. *Proceedings - 2021 International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2021*, 670–674.
<https://doi.org/10.1109/ICIEAM51226.2021.9446411>

ANEXOS

Anexo 1 Código para el IDE de Arduino

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Update these with values suitable for your network.
const char* ssid = "Bryan's Galaxy A51";
const char* password = "qwgq2017";
const char* mqtt_server = "192.168.61.177";

WiFiClient espClient;
PubSubClient client(espClient);

long lastMsg = 0;
char msg[50];
char msg3[50];
char msg4[50];

const char* msg1="1";
const char* msg2="0";

int value = 0;

const int led=14;//d5
const int led1=13;//d6

const int analogsonido = A0; //inicializar analogico

int sensorValue=0 ;
double escalamiento=0;

void setup() {
  pinMode(BUILTIN_LED, OUTPUT); // Initialize the BUILTIN_LED pin as an output
  pinMode(led,OUTPUT);
  pinMode(led1,OUTPUT);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
;
}

void setup_wifi() {
```

```

delay(10);
// We start by connecting to a WiFi network
Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.println();
  // Switch on the LED if an 1 was received as first character
  if ((char)payload[0] == '1') {
    digitalWrite(BUILTIN_LED, LOW); // Turn the LED on (Note that LOW is the voltage level
    // but actually the LED is on; this is because
    // it is active low on the ESP-01)
    digitalWrite(led, HIGH);
    Serial.print(" encendido");
  }
  else {
    digitalWrite(BUILTIN_LED, HIGH); // Turn the LED off by making the voltage HIGH

```

```

digitalWrite(led, LOW);
Serial.print(" APAGADO");
}
}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("ESP82661")) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.publish("esp2", "Enviando el primer mensaje");
      // ... and resubscribe
      client.subscribe("app/node/led1");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  long now = millis();
  if (now - lastMsg > 500) {
    lastMsg = now;

```

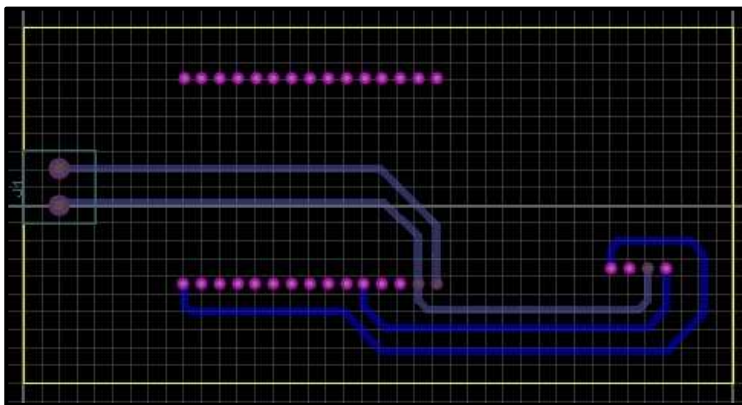
```

++value;
// sprintf (msg, 75, "hello #%ld", value);
//Serial.print("Publish message: ");
Serial.println(msg);

//client.publish("casa/despacho/temperatura", msg);
sensorValue = analogRead(analogsonido);
//sensorValue = 666;
sprintf (msg4, 75, "%lf", sensorValue );
Serial.print("Publish message: ");
Serial.println(sensorValue);
// **** CAMBIAR AQUI POR EL TOPIC ***
client.publish("sonido/esp8266/node", msg4);
}

```

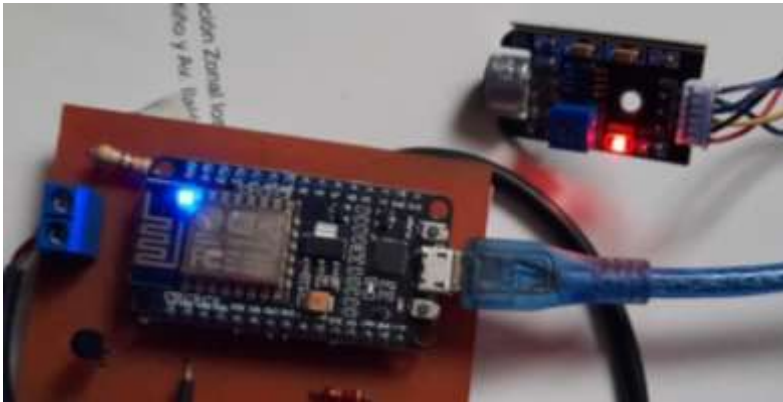
Anexo 2 Pistas creadas para la placa.



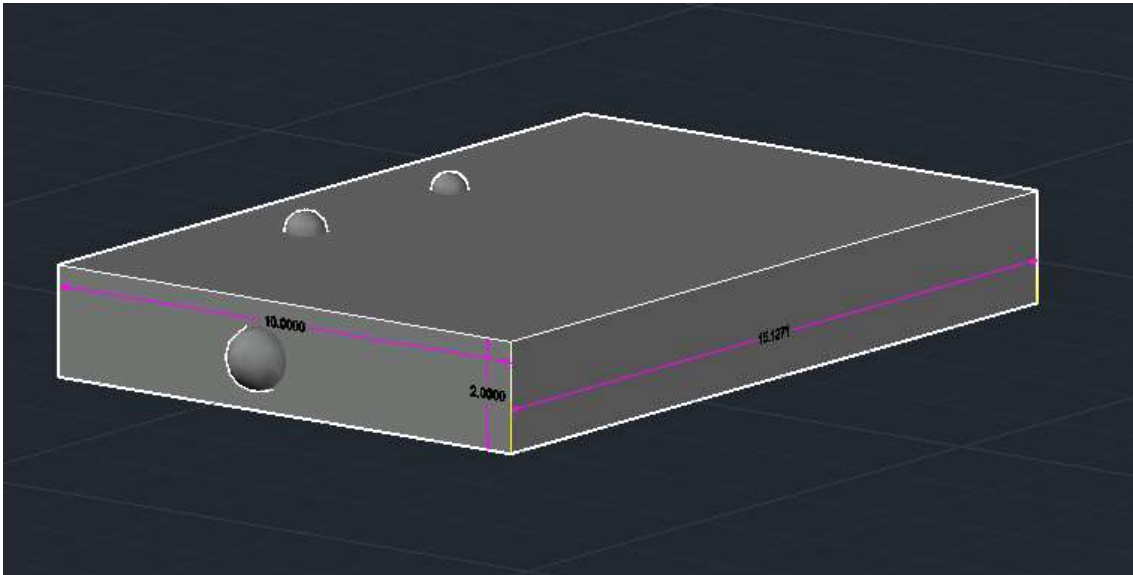
Anexo 3 ESP8266 colocado en la placa.



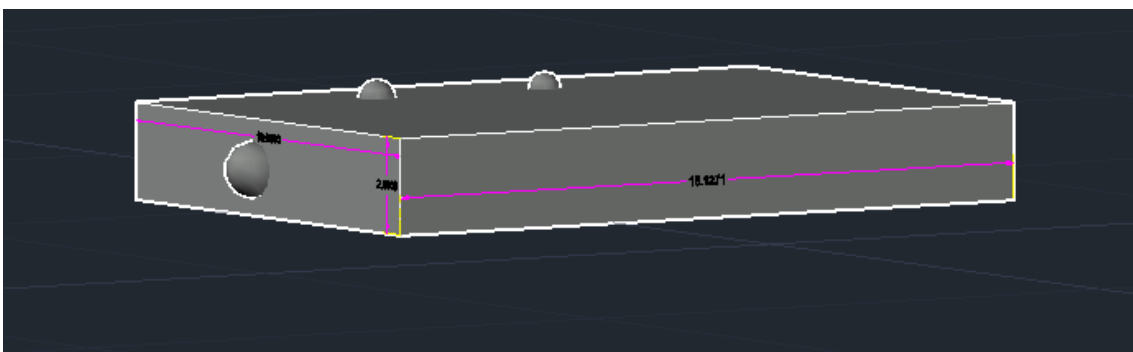
Anexo 4 ESP8266 y sensor KY038.



Anexo 5 Vista superior del prototipo



Anexo 6 Vista frontal del prototipo.



Anexo 7 Comparación de mediciones realizadas 1.



Anexo 8 Comparación de mediciones realizadas 2.

