

# **ESCUELA POLITÉCNICA NACIONAL**

**ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**IMPLEMENTACIÓN DE PROTOTIPOS DE SISTEMAS DE  
ALERTA BASADOS EN COMUNICACIÓN INALÁMBRICA**

**IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ALERTA  
POR DETECCIÓN DE SONIDO**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR  
EN REDES Y TELECOMUNICACIONES**

**ERICK ESTEBAN HIDALGO PORTILLO**

**DIRECTOR: FANNY PAULINA FLORES ESTÉVEZ**

**DMQ, agosto 2022**

## **CERTIFICACIONES**

Yo, ERICK ESTEBAN HIDALGO PORTILLO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



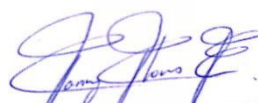
---

**ERICK ESTEBAN HIDALGO PORTILLO**

**erick.hidalgo01@epn.edu.ec**

**yover100@hotmail.com**

Certifico que el presente trabajo de integración curricular fue desarrollado por ERICK ESTEBAN HIDALGO PORTILLO, bajo mi supervisión.



---

**FANNY PAULINA FLORES ESTEVÉZ**

**DIRECTOR**

**fanny.flores@epn.edu.ec**

## DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.



---

ERICK ESTEBAN HIDALGO PORTILLO

## **DEDICATORIA**

A mi familia quienes, con paciencia, cariño y afecto estuvieron incondicionalmente brindando su apoyo en todo este proceso, que gracias a su amor me han permitido cumplir un objetivo más.

Esteban.

## **AGRADECIMIENTO**

A la Escuela Politécnica Nacional y a todas las autoridades que hacen la facultad de la Escuela de Formación de Tecnólogos (ESFOT) por permitir formar a quienes estuvieron y están formándose dentro de sus aulas, a mis ingenieros que me acompañaron y brindaron consejos, experiencia y conocimientos durante todo este proceso y a mi tutora Ing. Fanny Flores quien con paciencia y sabiduría supo brindarme consejos y darme una guía para poder desarrollar este trabajo.

A mis compañeros y amigos quienes a lo largo de mi carrera universitaria me brindaron su apoyo y amistad a Erick, Alejandro, Iván, Byron, a todos gracias por el tiempo compartido.

Esteban.

# ÍNDICE DE CONTENIDOS

CERTIFICACIONES .....	I
DECLARACIÓN DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
ÍNDICE DE CONTENIDOS .....	V
RESUMEN.....	VII
<i>ABSTRACT</i> .....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general .....	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
1.4 Marco Teórico.....	3
Señales de Audio.....	3
Wi-Fi.....	4
ESP-32 .....	5
Arduino UNO .....	5
Sensor de sonido KY-037 .....	6
2 METODOLOGÍA.....	7
3 RESULTADOS .....	9
3.1 Establecimiento de requerimientos del prototipo.....	9
3.2 Definición de componentes de <i>hardware</i> y <i>software</i> .....	10
<i>Hardware</i> .....	10
<i>Software</i> .....	13
3.3 Diseño del prototipo.....	14
Circuito electrónico .....	14
Justificación del rango de operaciones del prototipo.....	15
Diagrama de flujo.....	17

Vista en 3D del prototipo.....	18
Alimentación .....	20
3.4 Implementación del prototipo .....	20
Placa electrónica .....	20
Programación de Arduino .....	23
Creación de la aplicación para notificación en Blynk.....	24
Programación de ESP-32 .....	26
Implementación del prototipo en su <i>case</i> .....	27
3.5 Pruebas de funcionamiento .....	29
4 CONCLUSIONES .....	32
5 RECOMENDACIONES.....	33
6 REFERENCIAS BIBLIOGRÁFICAS.....	34
7 ANEXOS.....	i
ANEXO I: Certificado de Originalidad .....	i
ANEXO II: Enlaces .....	ii
ANEXO III: Códigos Fuente .....	iii
Código Arduino uno .....	iii
Código ESP-32 .....	v

## RESUMEN

A través del presente Proyecto de Integración Curricular, se presenta la implementación de un prototipo de alerta por detección de sonido, el cual permite alertar en caso de sobrepasar el nivel de ruido permitido en áreas como auditorios o bibliotecas; de ser así el caso, se enviará una notificación vía Wi-Fi.

En el documento se presenta el desarrollo de cada una de las siete secciones en las que se divide este proyecto.

En la primera sección se define el componente de desarrollo, aquí se explica los objetivos generales y específicos y también de manera breve y resumida el funcionamiento del prototipo.

En la segunda sección se explica la metodología que se desarrolló de acuerdo con los objetivos trazados.

En la tercera sección se presenta de forma específica detalles sobre el *hardware* y *software* que fueron implementados; además se incluye procedimientos para la elaboración del circuito eléctrico y la instalación del dispositivo en su *case*.

En la sección cuarta y quinta se presentan las conclusiones y recomendaciones del tema trabajado.

Para finalizar, en la sexta y séptima sección se presenta la bibliografía y anexos que ayudan a sustentar la investigación del tema.

**PALABRAS CLAVE:** sensor de sonido, nivel de sonido, ruido, Arduino, módulo Wi-Fi



## **ABSTRACT**

*Through this Curricular Integration Project, the implementation of a sound detection alert prototype is presented, which alerts in case of exceeding the allowed noise level in areas such as auditoriums or libraries, if so the case, a notification will be sent by Wi-Fi.*

*The document presents the development of each of the seven sections into which this project is divided.*

*In the first section the development component is defined, here the general and specific objectives are explained and also in a brief and summarized way the operation of the prototype.*

*The second section explains the methodology that was developed in accordance with the objectives outlined.*

*In the third section, details about the hardware and software that were implemented are specifically presented; it also includes procedures for the elaboration of the electrical circuit and the installation of the device in its case.*

*In the fourth and fifth sections, conclusions and recommendations are presented.*

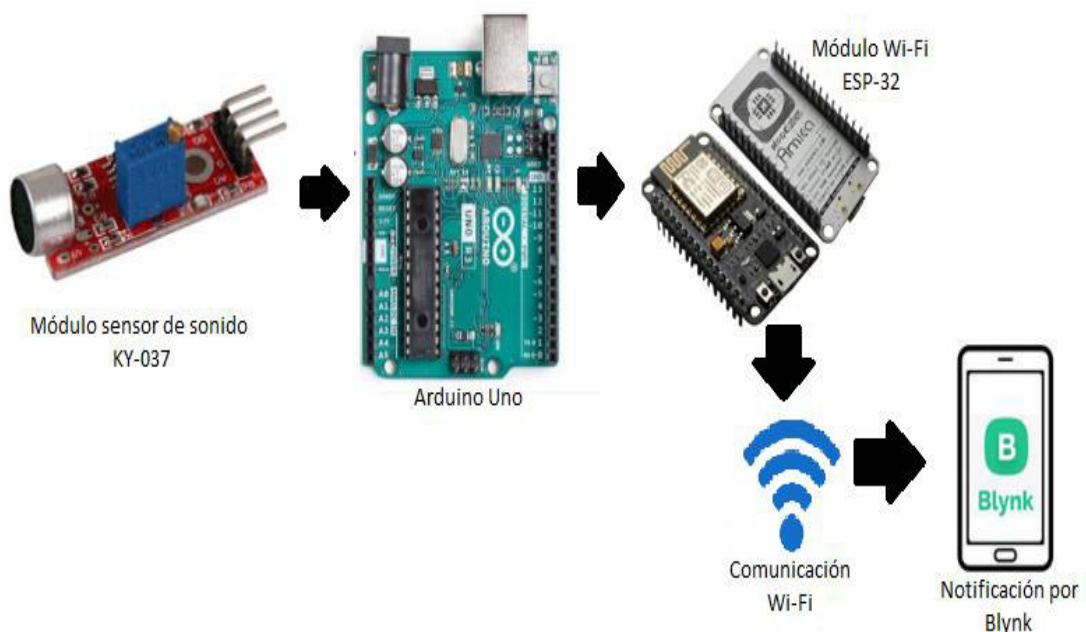
*Finally, in the sixth and seventh sections, the bibliography and annexes that help to support the research of the topic are presented.*

**KEY WORDS:** *sound sensor, sound level, noise, Arduino, Wi-Fi module*

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El prototipo consiste en un sistema de alerta por detección de sonido. El prototipo permitirá captar el nivel de sonido de un lugar establecido (bibliotecas o auditorios) y en función de dicho nivel determinar si el nivel de sonido es el permitido o es excesivo. En caso de que sea excesivo, se envía una alerta por notificación vía Wi-Fi y también en una pantalla LCD para inducir a los ocupantes a reducir los niveles de ruido en estos lugares. En la Figura 1.1 se presenta el funcionamiento general del sistema; se puede observar que el sensor de sonido capta la señal para después enviarla hacia la placa Arduino Uno. Aquí, la señal es procesada mediante su código programado, que incluye un límite de sonido, que, al ser excedido, se envía la alerta.

La señal de alerta se enviará a través de una notificación vía Wi-Fi, esto gracias al módulo ESP-32 que ayudará con la comunicación entre el sistema y la web. Esta notificación se realizará mediante la plataforma de Blynk. Todos los dispositivos que conforman el sistema estarán alimentados por una fuente de poder de 5 (V) conectada hacia una toma de corriente eléctrica.



**Figura 1.1** Diagrama de funcionamiento

## 1.1 Objetivo general

Implementar un prototipo de sistema de alerta por detección de sonido.

## 1.2 Objetivos específicos

- Establecer los requerimientos del prototipo
- Investigar los componentes de *hardware* y *software*
- Diseñar el prototipo
- Implementar el prototipo
- Realizar pruebas de funcionamiento

## 1.3 Alcance

El análisis de requerimientos abarca un estudio sobre las necesidades que se presentan con el desarrollo del prototipo. Esto incluye factores como comunicación, alimentación y protección. La investigación de los dispositivos que conforman el sistema se sujeta a una revisión de los componentes principales del prototipo (módulo Wi-Fi, sensor de sonido y placa principal de procesamiento) que se encuentran disponibles en el mercado, y en base a dicha revisión se elige cada dispositivo.

El diseño permite tener una vista en 3D del prototipo a implementar utilizando un *software* especializado de modelado en 3D para así anticipar cómo se verá el sistema. Además, se incluyen los cálculos necesarios para justificar el circuito electrónico, un diagrama de flujo que corresponde al proceso del *software* en el cual se mostrará un resumen fácil de entender el algoritmo utilizado.

La implementación considera la elaboración de una placa electrónica, así como la elección de los materiales que se usan para el *case* ya que estos deben garantizar un funcionamiento óptimo del sistema, sin perder de vista el aspecto estético. En las pruebas se simula un ambiente de biblioteca, el cual permite tener el nivel de sonido adecuado para el cual el prototipo está diseñado, captando el ruido a través del sensor de sonido y procesándolo por el microcontrolador de Arduino y a su vez enviando la notificación vía Wi-Fi cuando lo amerite.

## 1.4 Marco Teórico

### Señales de Audio

El ser humano tiene una capacidad auditiva sorprendente; puede escuchar desde los 20 (Hz) hasta los 20 (KHz) [6]. Gracias a estudios, se puede afirmar que los sonidos por encima de los 85 (dB) son peligrosos para los humanos cuando se prolonga su exposición [1]; dejando en claro este punto, se puede decir que el ruido de una conversación podría llegar a tener entre 45 (dB) a 55 (dB) ya que en un ambiente cerrado se podría tener una exposición prolongada a este tipo de ruido, como se indica en la Figura 1.2 [6].

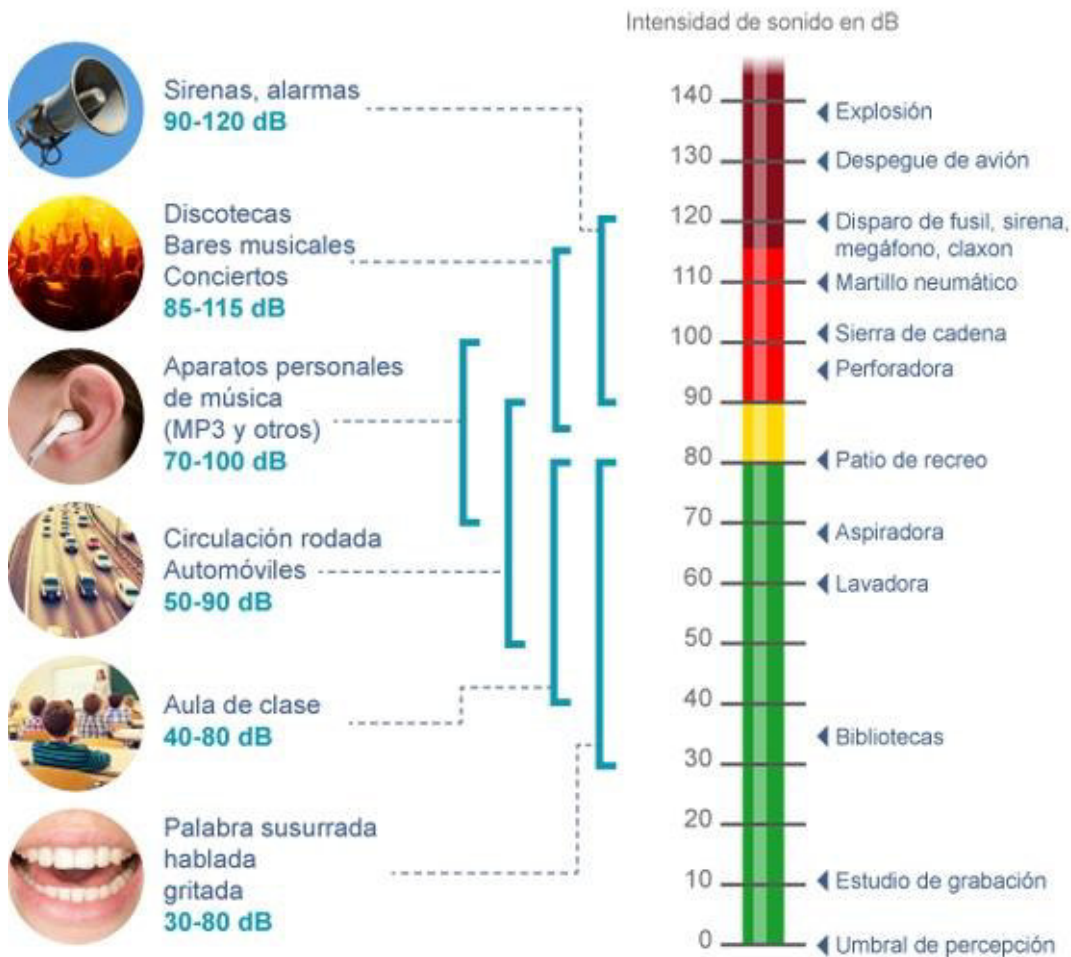


Figura 1.2 Rango auditivo humano [1]

Este proyecto se define como una alerta por medio de un sensor de sonido o ruido, de tal manera que se tenga una supervisión con un sistema que permita controlar de manera rápida el nivel de ruido en caso de que se tenga un nivel superior al sugerido. Teniendo en cuenta los datos anteriormente mostrados, se define cómo se procederá a utilizar este proyecto y en qué circunstancia y lugar; brevemente se procederá a introducir a los elementos que se utilizarán en el prototipo.

## **Wi-Fi**

Wi-Fi es una red inalámbrica que históricamente revolucionó al mundo y a su manera de conectarse a la red ya que, gracias a esto, múltiples dispositivos sin necesidad de ninguna conexión física se podían conectar a Internet. Gracias a esto, la tecnología hoy en día ha avanzado de una manera exponencial. Esto se debe a la publicación del estándar 802.11 en 1997, creado por la *Wi-Fi Alliance* que desde un principio fue el principal estándar para la conexión inalámbrica entre dispositivos y la red.

En la actualidad, existen diversos estándares que funcionan en diversas frecuencias; por ejemplo, se puede hablar del estándar 802.11a, el cual es el sucesor del estándar original. Este alcanza velocidades de 54 (Mbps), también fue el primer estándar que funcionaba en la frecuencia de 5 (GHz), siendo este un punto en contra ya que esta frecuencia es muy fácil de ser bloqueada por objetos, teniendo así un alcance limitado. Después de este, surgieron los estándares 802.11b, 802.11g y 802.11n, los cuales alcanzaron velocidades de 11, 54 y 300 (Mbps), respectivamente. Estos tenían la facilidad de ser compatibles entre sí ya que al transmitir en la frecuencia de 2.4 (GHz), la cual es una banda universal, podían funcionar y coexistir en un mismo lugar; se debe tener en cuenta que estos estándares son los más usados en la actualidad.

El estándar 802.11ac es el llamado Wi-Fi 5; se caracteriza por tener una velocidad de 1.3 (Gbps) [5]. Para finalizar, Wi-Fi 6 que corresponde al estándar 802.11ax, sigue en estudio y ofrece una velocidad máxima de 9 (Gbps) en las bandas de 2.4 y 5 (GHz); es la máxima tecnología que se ha alcanzado para el estándar 802.11.

Se espera que Wi-Fi 7 priorice la máxima capacidad, ofreciendo mayores ventajas en la transmisión de datos [6].

## ESP-32

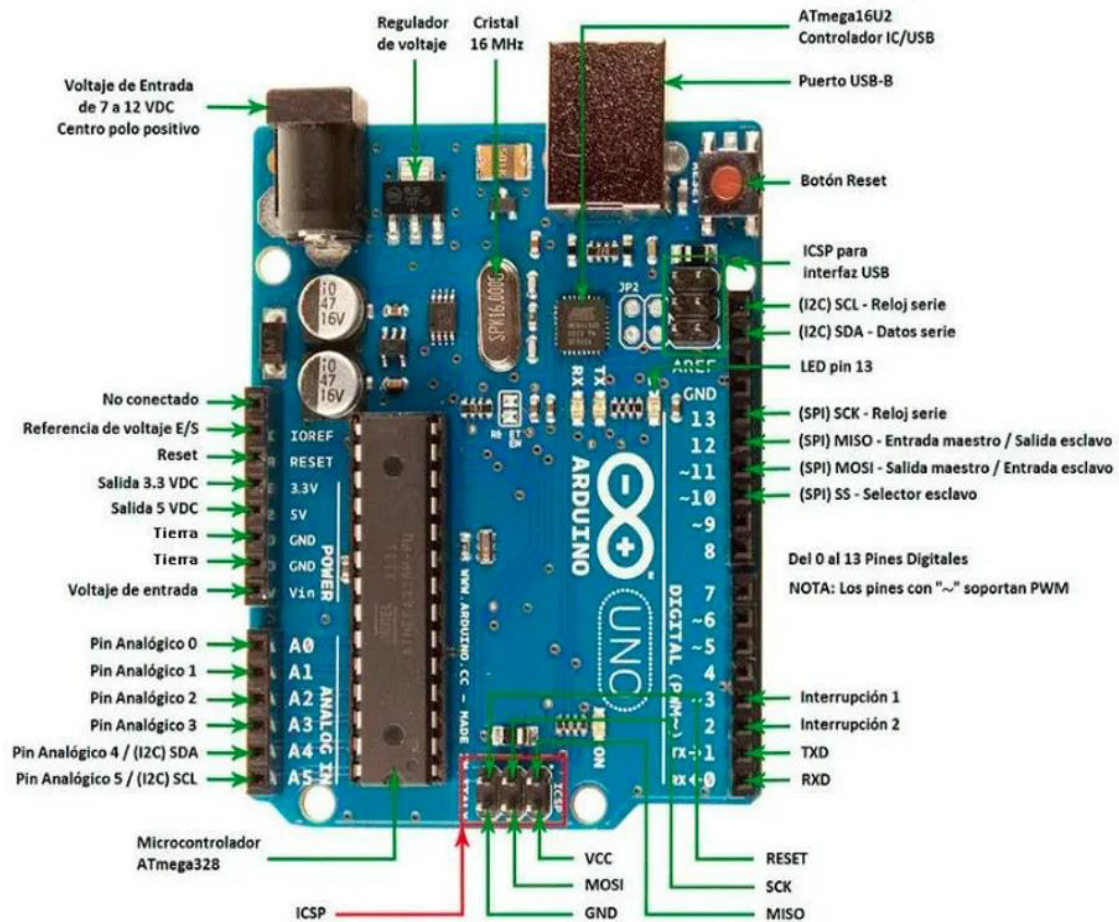
Debido a la importancia acogida que ha tenido Arduino en el desarrollo de los proyectos educativos, los diferentes tipos de módulos que existen han ido evolucionando como consecuencia de los constantes avances que se ha tenido en el área tecnológica. Uno de ellos es el ESP-32; es un módulo que está basado en la tecnología del microcontrolador ESP8266. Debido a su facilidad de uso, este dispositivo ha ido ocupando un espacio cada vez mayor en el entorno de proyectos y ha dejado atrás a su antecesor el ESP8266-01 [7]. En la Figura 1.3 se muestra el módulo ESP-32 que facilitará la comunicación para la aplicación.



**Figura 1.3** Módulo Wi-Fi [7]

## Arduino UNO

Arduino es una plataforma de trabajo muy sencilla de aprender, es un hardware libre que se basa en una placa que contiene un microcontrolador. La programación se puede desarrollar por medio de un software del mismo nombre; aquí se pueden implementar múltiples proyectos de diversas áreas ya que la flexibilidad con los dispositivos que se pueden conectar a esta es muy extensa y su uso está más orientado al entorno educativo. En la Figura 1.4 se observa las partes del Arduino Uno.



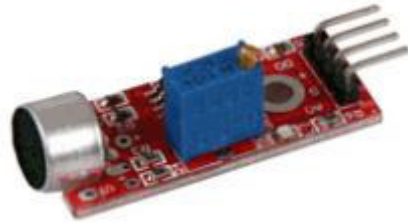
**Figura 1.4** Arduino UNO [9]

La placa electrónica Arduino Uno cuenta con un microcontrolador ATmega328P, que es el cerebro del dispositivo, el cual tiene sus entradas y salidas listas para su uso; dicho microcontrolador ejecuta, lee, envía y controla dispositivos y señales. Tiene un reloj u oscilador que está hecho de cuarzo y funciona a una frecuencia de 16 (MHz), la cual determina la velocidad en la que los programas se ejecutan. Funciona con los voltajes de 3.3 (V) y 5 (V), que son salidas externas de la tarjeta, las cuales se usan para alimentar dispositivos externos y tiene un voltaje de entrada de 12 (V); sin embargo, la tarjeta solo trabaja con 5 (V) ya que utiliza un regulador para esta tarea. Para finalizar, posee un par de entradas Tx y Rx, 14 pines digitales, de los cuales 6 sirven para Modulación por Ancho de Pulso (PWM) y 6 pines son analógicos [9].

### Sensor de sonido KY-037

El sensor tiene 3 componentes principales que se muestran en la Figura 1.5, los cuales son la unidad de sensor de la parte delantera la cual mide físicamente el área y envía una señal analógica. La segunda parte es un amplificador, el cual amplifica la señal

según el valor del potenciómetro y envía una señal a la siguiente parte, la cual está integrada por un comparador y un LED, los cuales leen la señal, determinando así si esta cae por debajo de un valor específico [3].



**Figura 1.5** Sensor de sonido [3]

## 2 METODOLOGÍA

Para el desarrollo del presente proyecto, se definieron cinco objetivos para la implementación del sistema de alerta por detección de sonido, que permite tener un control del ruido generado en ciertos lugares donde permanecer en silencio es una prioridad. La metodología utilizada se presenta en la Figura 2.1.



**Figura 2.1** Metodología utilizada



Para el desarrollo del proyecto se tomó en cuenta que el prototipo se encontrará fijo en una biblioteca o auditorio, donde generalmente se restringe el sonido al menor ruido posible. Al ser un sistema de alerta fijo, se consideró que su alimentación será a través de la energía eléctrica por tomacorriente, asegurando su funcionamiento de manera continua. Adicionalmente, para garantizar que la alerta sea recibida a tiempo por el usuario, se consideró una conexión Wi-Fi como la opción más recomendada; de este modo, el usuario será alertado de manera más rápida y podrá tomar las medidas necesarias para poder evitar el nivel de ruido no deseado. Se determinaron los componentes utilizados; para la medición de los niveles de sonido se investigaron diferentes sensores de nivel, entre ellos: el sensor de sonido KY-037 y el sensor de sonido KY-038. Por otra parte, se compararon diferentes modelos de Arduino, entre ellos: Arduino Uno, Arduino Mega2560 y Arduino Nano. También, se indagó sobre un módulo Wi-Fi que cumpla con las necesidades planteadas, por lo que se realizó la investigación de los módulos ESP8266-01 y ESP-32.

Se desarrolló un diagrama esquemático donde se aprecia la conexión de los dispositivos electrónicos, que luego se plasmó en la herramienta Proteus para su implementación. En esta etapa de diseño, se realizaron los cálculos necesarios para dimensionar adecuadamente los dispositivos electrónicos escogidos. Además, se realizó un diagrama de flujo que facilitó la programación del código. Un punto esencial del diseño se dio en la selección del lugar donde se implementaría, a partir de lo cual se definieron los rangos de operación del sensor, tomando en cuenta el momento en que el sonido llegue a un límite que represente un ruido excesivo, tal que pudiera molestar a los usuarios de dichos lugares, se debe enviar al usuario el mensaje de alerta. Para el desarrollo del circuito electrónico se utilizó el software Proteus; la herramienta Isis facilitó la realización del modelo esquemático, permitiendo la conexión de todos los dispositivos electrónicos. Luego, en la herramienta Ares se obtuvo la Placa de Circuito Impreso (PCB), donde se crean las pistas y el posicionamiento de componentes. Finalmente, la construcción del circuito se realizó en una baquelita de cobre, utilizando el método del planchado donde se limpió, lijó, taladró y soldó los componentes. Así mismo, para la elaboración de la maqueta se realizó previamente un modelado en 3D en el software SketchUp, que facilitó el desarrollo del *case* del prototipo. Finalmente, como materiales se usó acrílico y madera para la construcción del *case*.

Se realizaron las pruebas de funcionamiento correspondientes; inicialmente, a través del monitor serial se comprobó que el sensor de sonido KY-037 realizaba lecturas correctas; también, se verificó el envío de la notificación al usuario vía Wi-Fi por la plataforma Blynk y el despliegue del mensaje en la pantalla LCD.

### **3 RESULTADOS**

El prototipo realizado de un sistema de alerta por detección de sonido permite al usuario observar un mensaje de alerta mediante una pantalla LCD y también una notificación al administrador del lugar vía Wi-Fi mediante la plataforma Blynk. Para la detección del sonido, se maneja un sensor KY-037, el cual envía una señal a la placa Arduino Uno que es la encargada de procesar el dato recibido y en base a un algoritmo programado en la placa se procede a enviar un mensaje de alerta hacia el módulo ESP-32 para que envíe al usuario una notificación de que los límites de sonido se han excedido. Para el desarrollo óptimo del proyecto, se toman en cuenta cinco tópicos esenciales. En un inicio, se establecen los requerimientos principales. Posterior a esto, se elige el hardware y software necesarios; después se procede con el diseño del prototipo y en base a las anteriores fases se procede con la implementación y las pruebas de funcionamiento.

#### **3.1 Establecimiento de requerimientos del prototipo**

Teniendo en cuenta que el sistema debe ser implementado de una manera simple, de forma que al momento de utilizarlo no se dificulte la interacción entre usuario y sistema y que pueda abarcar un área correspondiente a una biblioteca o un auditorio normal, la conexión Wi-Fi sería más que suficiente. Adicionalmente, se garantiza una comunicación eficiente y amigable entre el sistema y el usuario debido a que en la actualidad este tipo de conexión inalámbrica es la más usada y también porque reduce el costo y el tiempo de la puesta en marcha.

Dado que el sistema es instalable en cualquier ambiente cerrado, y que este permanecerá ubicado de manera fija, se podrá considerar la alimentación de la red eléctrica pública sin necesidad de baterías u otros mecanismos orientados a sistemas portátiles. El sistema también estará protegido por una caja para precautelar la integridad de los circuitos y demás módulos que estén conectados a este; debido a que el sistema es considerado solo para interiores, las únicas amenazas serán la manipulación no autorizada y el polvo. El único componente que estará en contacto con el exterior será el sensor de sonido debido a que su micrófono integrado es sensible para captar el ruido y así conseguir un mejor desempeño del sistema.

Dado que el sistema estará en ambientes cerrados, se debe tomar en cuenta que para el oído humano los sonidos emitidos a partir de los 40 (dB) empiezan a considerarse

como ruido y superiores a 80 (dB) empiezan a dañar el sistema auditivo humano cuando su exposición es prolongada [6]. En auditorios o bibliotecas, el rango común de ruido está entre 30 (dB) a 79 (dB), límite que pudiera comenzar a causar molestias a la percepción del oído humano [6]. Dado que se busca alertar al público del nivel de ruido que está presente en el sitio donde se implemente el sistema, se requerirá una interfaz, la cual despliegue un mensaje de alerta en un *display*.

## 3.2 Definición de componentes de *hardware* y *software*

### ***Hardware***

En el mercado existen diversos *hardwares* y *softwares* que se pueden emplear para poner en marcha este sistema. Ya que Arduino es uno de los principales competidores en el ámbito de la creación de proyectos educativos, por el mismo hecho de que cuenta con un *kit* completo de diferentes dispositivos que son sencillos al momento de configurarlos y utilizarlos, también constituye un *software* libre que se puede usar de forma flexible ya que funciona en casi todos los sistemas operativos. Por lo tanto, se puede decir que Arduino está enfocado a aquellas personas que están empezando en programación y que tienen conocimientos básicos de electrónica.

Las placas base de Arduino en el mercado son muchas y ya que esta será la base del proyecto, es importante saber que existen diferencias entre placas y que cada una servirá para un proyecto diferente debido a sus dimensiones, voltajes, y tipo de procesamiento que ofrece. En la Tabla 3.1 se muestran las diferencias entre las placas de Arduino Nano, Uno, Mega 2560, que se encuentran en la Figura 3.1. Para la implementación del sistema se ha elegido la placa de Arduino Uno debido a que el proyecto requiere una cantidad de pines nada elevada, también por la facilidad de uso y la dimensión que tiene es muy bien adaptable a un *case* de tamaño pequeño y portable.

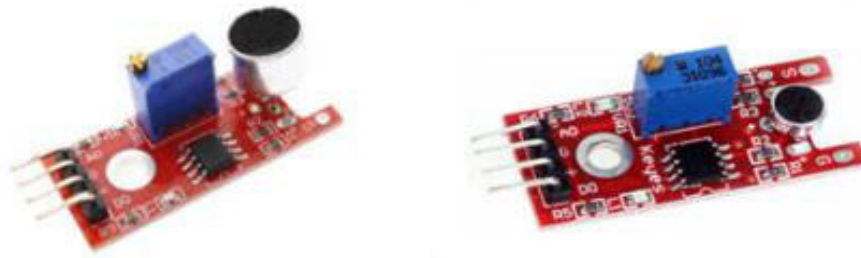


**Figura 3.1:** Arduino Nano, Uno y Mega2560 [12]

**Tabla 3.1:** Comparación placas de Arduino [12]

<b>Módulo</b>	<b>Arduino Nano</b>	<b>Arduino Uno</b>	<b>Arduino Mega2560</b>
<b>Precio</b>	\$20 - \$24.40	\$20 - \$23	\$36,60 - \$39
<b>Dimensiones</b>	0.7" x 1.9"	2.7" x 2.1"	4" x 2.1"
<b>Procesador</b>	ATMega32U4	ATMega328P	ATMega2560
<b>Velocidad del reloj</b>	16 (MHz)	16 (MHz)	16 (MHz)
<b>EEPROM (kB)</b>	1	1	4
<b>SRAM (kB)</b>	2.5	2	8
<b>Voltaje</b>	5 (V)	5 (V)	5 (V)
<b>Pines Digitales E/S</b>	20	14	54
<b>Pines Digitales E/S PWM</b>	7	6	15
<b>Pines Analógicos</b>	12	6	16
<b>Conectividad USB</b>	Micro USB	USB estándar	USB estándar
<b>Compatibilidad de escudo</b>	No	Sí	Sí
<b>Ethernet, Wi-Fi, Bluetooth</b>	No	No	No

Con respecto al tipo de sensor que se usará en este sistema, se analizan dos opciones que comparten la misma función de detectar la presencia de sonido mediante una señal; esta es captada en forma de energía y se envía hacia un receptor. Esta señal puede ser digital o analógica debido a que, si se requiere extraer los datos de la señal, se debe conectar directamente a la salida analógica; pero si se prefiere medir el nivel de ruido o sonido se debe configurar la sensibilidad de tal manera que, al momento de recibir la señal, se encienda [10]. Para identificar mejor estos sensores que se encuentran en la Figura 3.2, en la Tabla 3.2 se presentan sus características. Debido a que se necesita que el sistema sea muy sensible, resulta conveniente optar por el sensor KY-037 para tener la capacidad de captar incluso señales mínimas.

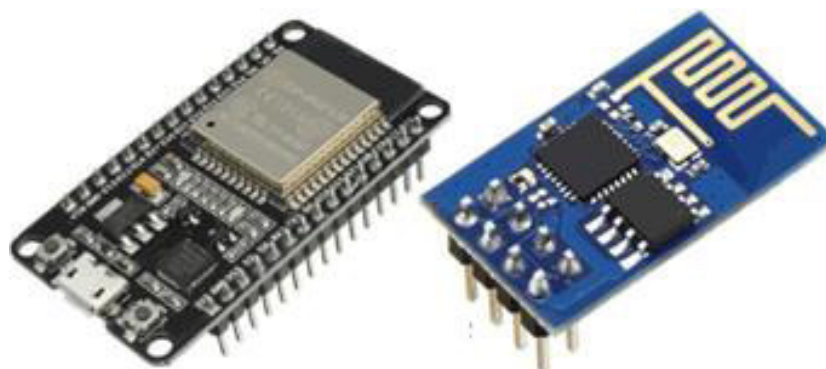


**Figura 3.2** Sensores de sonido KY-037 y KY-038 [10]

**Tabla 3.2** Tabla comparativa KY-037 y KY-038 [10]

Módulo	KY-037	KY-038
Voltaje	5 (V)	5 (V)
Sensibilidad	Mayor	Normal
Potenciómetro	Sí	Sí

Para la conectividad, se debe considerar que el lugar donde va a ser instalado el sistema es en un auditorio o en una biblioteca de tamaño normal. El ESP8266-01 es un módulo independiente que puede ser conectado a cualquier placa Arduino ya que necesita una memoria *flash* para su funcionamiento; este, como otros dispositivos, tiene su competidor que es el ESP-32. Gracias a la mayor cantidad de pines y facilidad de programación que ofrece, el ESP-32 es el elegido para recibir la orden enviada por la placa Arduino Uno una vez interpretados los datos recogidos por el sensor y mandar una señal hacia una aplicación. En la Tabla 3.3 se presentan las diferencias entre los dispositivos que se encuentran en la Figura 3.3.



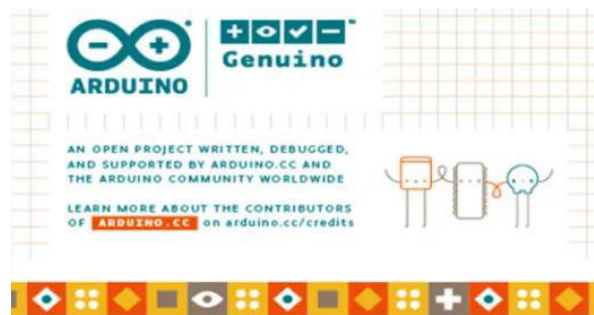
**Figura 3.3** Módulos ESP-32 y ESP8266-01 [13]

**Tabla 3.3** Tabla comparativa ESP-32 y ESP8266-01 [11]

Módulo	ESP-32	ESP8266-01
Velocidad Wi-Fi	802.11n hasta 150 (Mbps)	Hasta 72.2 (Mbps)
Protocolos Wi-Fi	802.11 b/g/n - 2.4 (GHz)	802.11 b/g/n - 2.4 (GHz)
Bluetooth	Sí	No
DAC	2 canales de 8 bits	No
ADC	SAR de 12 bits	SAR de 10 bits
Protocolos de red	IPv4, IPv6, SSL TCP/UDP/HTTP/FTP/MQTT	IPv4 TCP/UDP/HTTP/FTP/MQTT
Corriente operativa	Promedio: 80 (mA)	Valor medio: 80 (mA)
Tensión de funcionamiento	2.5 (V) – 3.6 (V)	2.5 (V) – 3.6 (V)
Precio	\$4.27 - \$10.68	\$3.20 - \$6.41

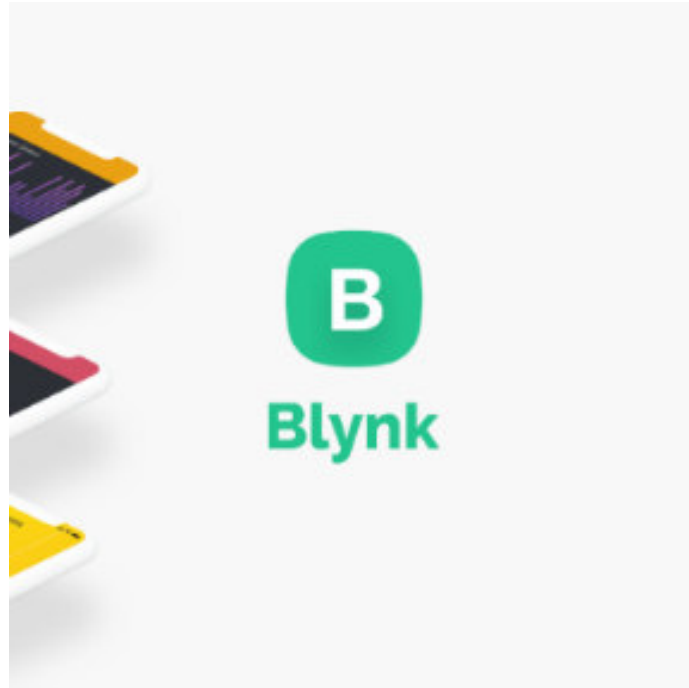
### Software

En el mundo de la programación, el poder entender de manera sencilla cómo se procesa la información a través de los diferentes dispositivos es muy importante, basta con programar algunas líneas de código para conseguir un resultado considerable. Debido a varios avances en la actualidad, se ha llegado a elegir Arduino IDE como una opción muy viable para este proyecto; este posee gran cantidad de información que ayuda a las personas que no tengan mucho conocimiento en el ámbito de la programación. Los llamados *sketches* en el mundo de Arduino son los programas que se configuran dependiendo de cómo se va a utilizar su *hardware*; aquí se puede escribir, depurar, editar y grabar lo que se requiera. También, su accesibilidad ayuda a que el *software* sea uno de los más utilizados y exitosos en proyectos educativos [14]. Tal y como se observa en la Figura 3.4, la cual es la imagen que aparecerá al iniciar el *software*, es un programa accesible y de muchos usos para todo público.



**Figura 3.4:** IDE Arduino [14]

Además, para la implementación de la alerta vía *web* y usando el módulo ESP-32 Wi-Fi, se utilizará Blynk. Esta aplicación para servicio del internet de las cosas (IoT), se popularizó gracias a su interfaz amigable que proporciona la oportunidad de crear una aplicación o notificación *web* mediante su base de datos y su servidor propio. Gracias a la garantía que ofrece Blynk, se puede enviar un mensaje de alerta a quien se encuentre conectado con el dispositivo ESP-32. En la Figura 3.5 se observa el logo representativo de la plataforma [16].

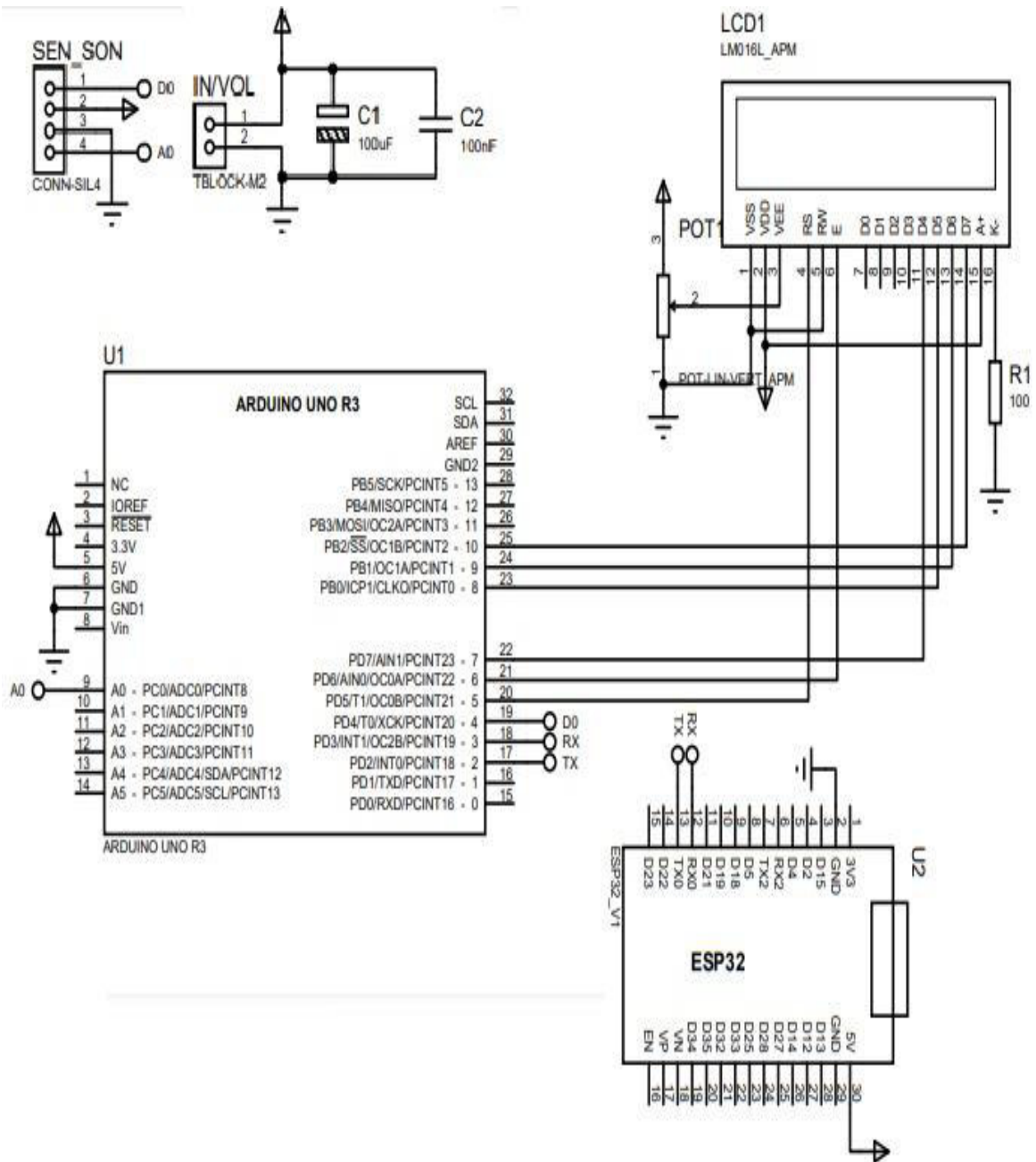


**Figura 3.5** Plataforma Blynk [16]

### **3.3 Diseño del prototipo**

#### **Circuito electrónico**

Para simular el sistema de una forma sencilla se utiliza el *software* Proteus; este cuenta con muchas herramientas que permiten una simulación de circuitos electrónicos gracias a Isis, herramienta que permite el desarrollo de diagramas esquemáticos. Entre otras herramientas, se tiene a Ares, la cual es la encargada de realizar la placa PCB. En la Figura 3.6 se presenta el circuito del prototipo.



**Figura 3.6** Circuito esquemático del diseño

### Justificación del rango de operaciones del prototipo

Gracias al funcionamiento de Arduino y a su microcontrolador integrado ATmega328P que posee un convertor analógico a digital (ADC) de 10 bits, se puede utilizar la resolución que corresponde a una notación digital. Por ejemplo, si se tiene un valor analógico de 0 (V), de forma digital se lo puede expresar como 0000000000, que si se está familiarizado con esta conversión será de un valor que corresponde a 0 decimal; caso contrario, si se tiene un valor analógico de 5 (V), se tendrá como respuesta



1111111111, esto en valor decimal representa 1023. Se puede decir que, el valor analógico que se obtiene de una placa Arduino Uno varía entre 0 a 1023 [15]. A través de la Ecuación 3.1 se define el número posible de niveles, en función del número de bits.

$$M = 2^n$$

**Ecuación 3.1** Niveles en función de bits [15]

Donde:

n: número de bits

M: número de niveles

Usando la Ecuación 3.1, se obtiene

$$M = 2^{10}$$

$$M = 1024$$

El sensor de sonido KY-037 funciona a 10 bits; como se explicó anteriormente, esta comparación se la hace con números decimales ya que al tener 0000000000 se tendrá un voltaje de 0 (V), que en número decimal corresponde a 0 y en nivel de decibeles (dB) corresponde también a un valor de 0. Sin embargo, si se aplica un sonido o una señal analógica hacia el sensor, este corresponderá a una señal de voltaje que dependerá de los decibeles que se le aplique. En la Tabla 3.4 se observa los resultados de la regla de 3 aplicada y los rangos de operación del sistema; la regla de 3 empleada se presenta mediante la Ecuación 3.2.

$$\frac{1023}{499} = \frac{80 \text{ dB}}{X} = \frac{5 \text{ V}}{X}$$

**Ecuación 3.2** Regla de tres

**Tabla 3.4** Rangos de operación del sistema [1]

Rango de Bits	Límite en decibeles	Límite en voltaje	Mensaje en pantalla LCD	Explicación
499	38.88 (dB)	2,43 (V)	“Ruido Normal”	Teniendo en cuenta el gráfico de nivel de ruido, se observa que una conversación susurrada termina en 39.99 (dB)

Continúa

505	39.36 (dB)	2.46 (V)	“Ruido Normal”	
512	40.02 (dB)	2.51 (V)	“Ruido Excesivo”	A partir de los 40 (dB) se puede decir que el ruido emitido se convierte en un ruido equiparado a un aula de clases, lo cual es lo que se busca evitar

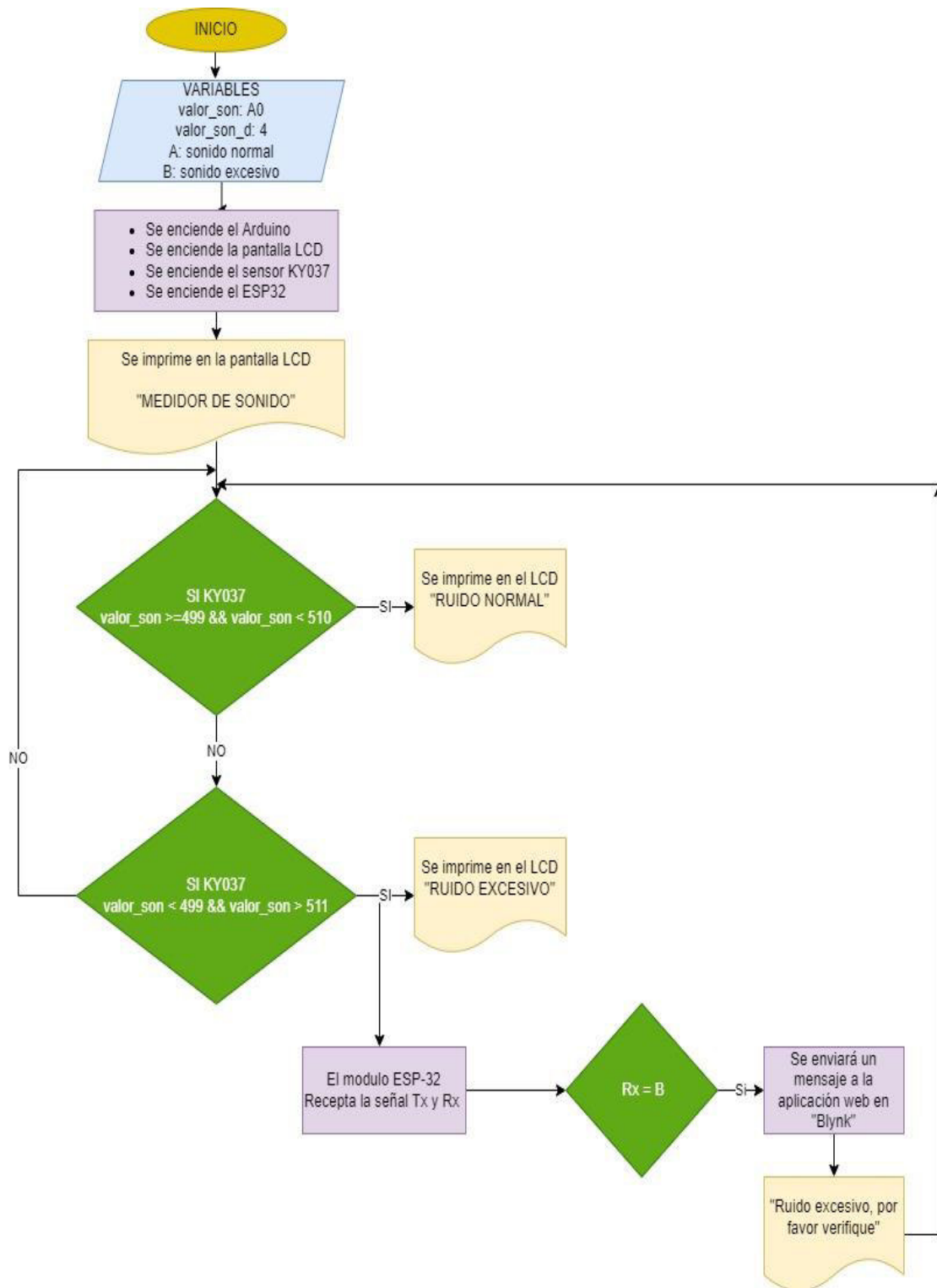
La pantalla LCD funciona con un potenciómetro que ayudará a controlar el *back light*, es decir, para que se pueda controlar el contraste de la misma; la resistencia de este potenciómetro es de 10 (k $\Omega$ ). Esta resistencia limita el voltaje sobre la pantalla LCD, lo que da como efecto lo mostrado en la Tabla 3.5.

**Tabla 3.5** Potenciómetro LCD

Resistencia	Voltaje
0 $\Omega$	5 (V)
5000 $\Omega$	2.5 (V)
10000 $\Omega$	0 (V)

### Diagrama de flujo

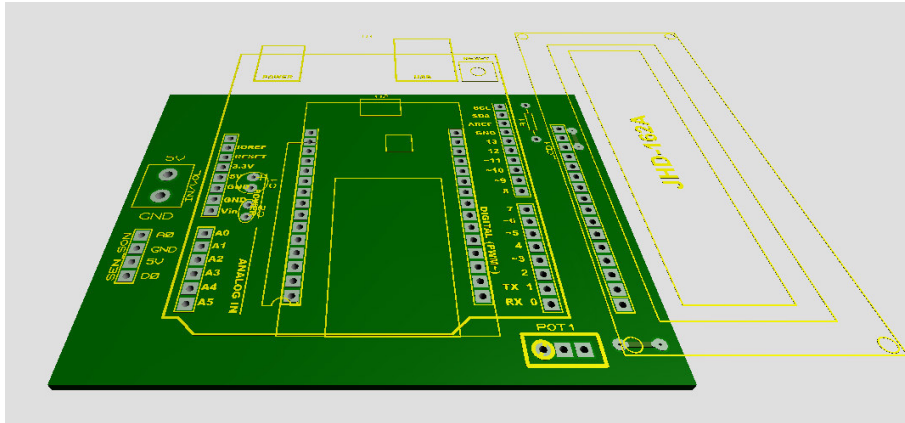
En la Figura 3.7 se presenta el diagrama de flujo del código del prototipo. Inicialmente, se realiza la declaración de las variables en Arduino y después la ejecución de estas; a la vez, se enciende la pantalla LCD y también se enciende el ESP-32 para establecer comunicación con la aplicación de Blynk en la red. Después de esto, se define un condicional que se referirá a la medición que se obtendrá del micrófono del sensor de sonido. Si este capta alguna señal mayor a un rango específico programado en Arduino, este llamará a una variable guardada para interactuar con el ESP-32 que enviará un mensaje a la red. Este llegará a la aplicación si es que se ha excedido el rango delimitado para el sonido; caso contrario, el sensor seguirá pendiente de medir los sonidos que estén a su alcance, y en la pantalla LCD se podrá leer un mensaje que diga “SONIDO NORMAL”, si es que el sonido no supera el rango programado. En caso de exceder el rango, se enviará un mensaje hacia el LCD que indique “SONIDO EXCESIVO” y adicionalmente se enviará una notificación a la aplicación.



**Figura 3.7** Diagrama de flujo del sistema

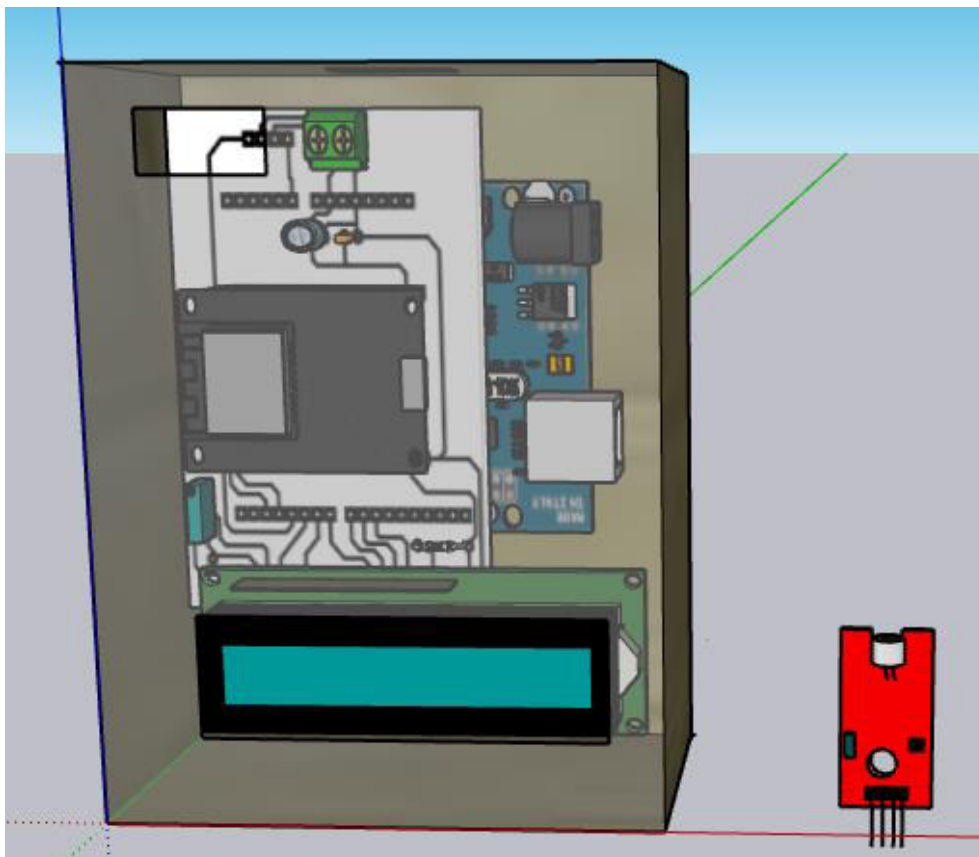
### Vista en 3D del prototipo

En la Figura 3.8 se muestra la vista en tres dimensiones del prototipo sin el *case* para que se pueda ver con facilidad los puntos de conexión de los componentes y también se da un vistazo a la placa PCB.



**Figura 3.8** Placa del prototipo en 3D

Gracias a la prueba gratuita que ofrece el programa de construcción en 3D Sketchup, se pudo desarrollar el *case* del prototipo; al ser un *software* muy versátil, se usa en diversas ramas como: ingeniería, diseño gráfico y arquitectura. Sirve tanto para crear edificios como para modelar un diseño de prototipo en varios ámbitos. En la Figura 3.9 se presenta el modelado de la maqueta que se diseñó para el prototipo [16].



**Figura 3.9** Diseño en 3D del *case*

## Alimentación

Para alimentar un Arduino Uno se puede emplear diversos métodos; uno de ellos es la alimentación por el adaptador Universal en Serie (USB), otra de las posibles formas de alimentación es la entrada de Voltaje (VIN) de 5 (V). Esta última resulta ser la mejor para poder alimentar el sistema debido a que el módulo ESP-32 y la placa Arduino Uno utilizan voltajes diferentes, siendo el de la placa 5 (V) y el voltaje de operación del módulo 3.3 (V).

Gracias a la separación de voltajes, se puede distribuir la energía que sale de la fuente de poder de 5 (V) y esta a su vez ser distribuida a los componentes gracias a la cualidad de Arduino que puede entregar 3.3 (V) y 5 (V).

## 3.4 Implementación del prototipo

### Placa electrónica

Finalizando el diseño del esquema del circuito en Isis, se procede a elaborar en la herramienta Ares el prototipo de la placa para implementarse en PCB. Como se muestra en la Figura 3.10, con esta herramienta se define el tamaño para la placa PCB que se requiera y también se generan las pistas o caminos de los circuitos que conectan los diferentes módulos del sistema.

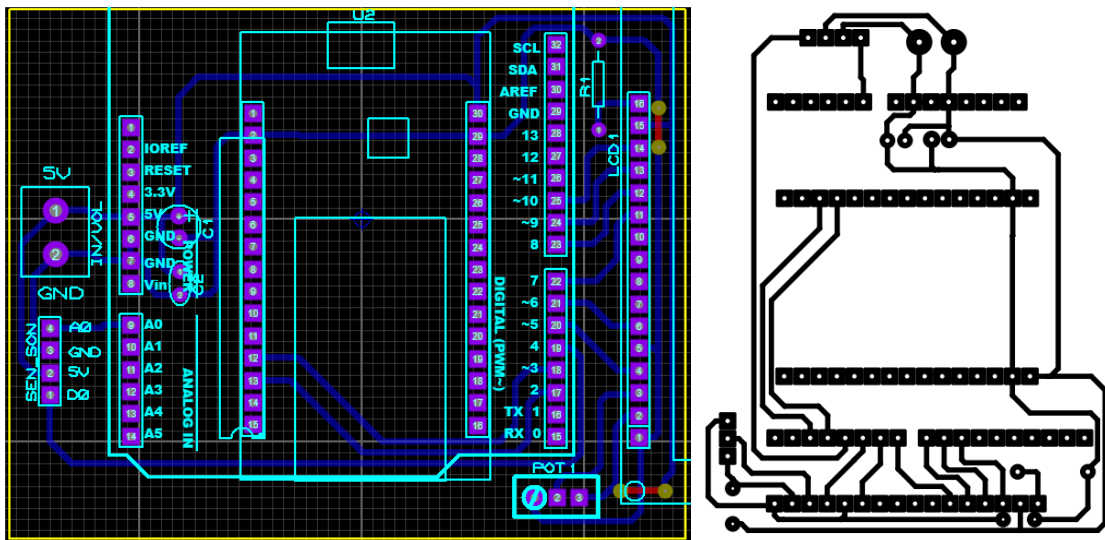
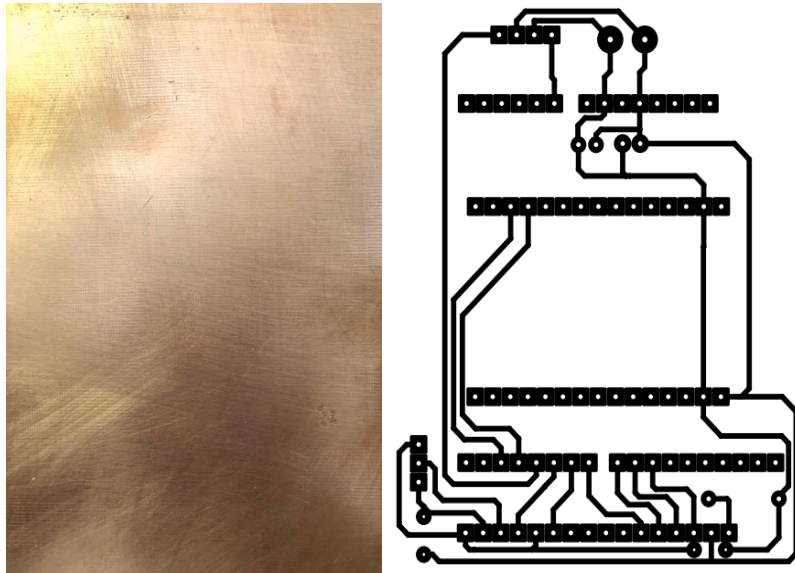


Figura 3.10 Circuito diseñado en Ares

Después de haber diseñado las pistas de manera adecuada, se procede a imprimir las mismas en láser y se procede a lijar y limpiar la placa de baquelita, como se muestra en la Figura 3.11. Posteriormente, se aplica el calor suficiente para que las pistas se transfieran de una manera correcta. La transferencia de calor se puede realizar con una plancha de manera uniforme, como se muestra en la Figura 3.12.



**Figura 3.11** Baquelita lista para el traspaso de pistas



**Figura 3.12** Transferencia de calor

Después de la transferencia de calor, se procede a aplicar un ataque de cloruro férrico, esto se observa en la Figura 3.13 que consiste en sumergir la placa sin el papel couché y con la tinta ya seca en una mezcla de agua y cloruro férrico en un recipiente plástico para que la placa pueda quemarse y así tener las pistas claras.



**Figura 3.13** Recipiente con mezcla de cloruro férrico y agua

Después, se procede a taladrar con una broca de 1.0 (mm) para poder hacer los agujeros necesarios para que los dispositivos puedan ser soldados con estaño, como se muestra en la Figura 3.14. Para el Arduino y el módulo ESP-32, se procedió a colocar espadines para facilitar su retiro, en caso de que se requiera cambios futuros en el prototipo.



**Figura 3.14** Proceso de soldadura

## Programación de Arduino

El proceso de programación se realizó en el IDE de Arduino, tanto para el módulo Arduino Uno como para el ESP-32. Cada programa incorpora los comentarios correspondientes, facilitando la comprensión de los mismos. En la Figura 3.15 se observa la librería que se usó para la programación de la LCD y las diferentes librerías para la programación de los dispositivos utilizados. También se definieron algunas variables que utilizan la entrada del sensor y de la LCD.

```
#include <LiquidCrystal.h> //libreria del LC
#include <SoftwareSerial.h> //libreria de lo

SoftwareSerial mySerial(2, 3); // RX, TX est

const int rs = 5, en = 6, d4 = 7, d5 = 8, d6

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int valor_son; // variable entera con la que
bool val_son_d; // variable booleana que sol
```

**Figura 3.15** Librerías y variables de Arduino Uno

Como se muestra en la Figura 3.16, en el *void setup* se configuró la velocidad con la que funciona el módulo ESP-32 y Arduino Uno, así como algunos parámetros de la pantalla LCD.

```
void setup() {

    Serial.begin(9600);
    mySerial.begin(115200); // velocidad de trasmision de datos
    lcd.begin(16, 2);

    pinMode(A0, INPUT); // pin de lectura analoga para el sensor
    pinMode(4, INPUT); // pin de lectura digital para el sensor

    lcd.setCursor(4,0);
    lcd.print("MEDIDOR");

    lcd.setCursor(3,1);
    lcd.print("DE SONIDO");
    delay(2000);
    lcd.clear();
}
```

**Figura 3.16** *Void setup*

Para finalizar la programación del módulo Arduino, en la Figura 3.17 se muestra un condicional que permite detectar los rangos de sonido y así enviar la alerta hacia el módulo y a la pantalla LCD.



```

void loop() {

    valor_son = analogRead(A0); //declarar v
    val_son_d = digitalRead(4);

    Serial.print("Val_anag: "); // impresior
    Serial.println(valor_son);
    Serial.print("Val_dig: "); // impresion
    Serial.println(val_son_d);

    if(valor_son <= 500 && valor_son < 510){
        Serial.println(" Ruido normal");
        mySerial.print('B'); // variable que
        lcd.setCursor(0,0);
        lcd.print(" RUIDO NORMAL "); // in
    }
    if(valor_son >= 510 && valor_son < 511){
    }

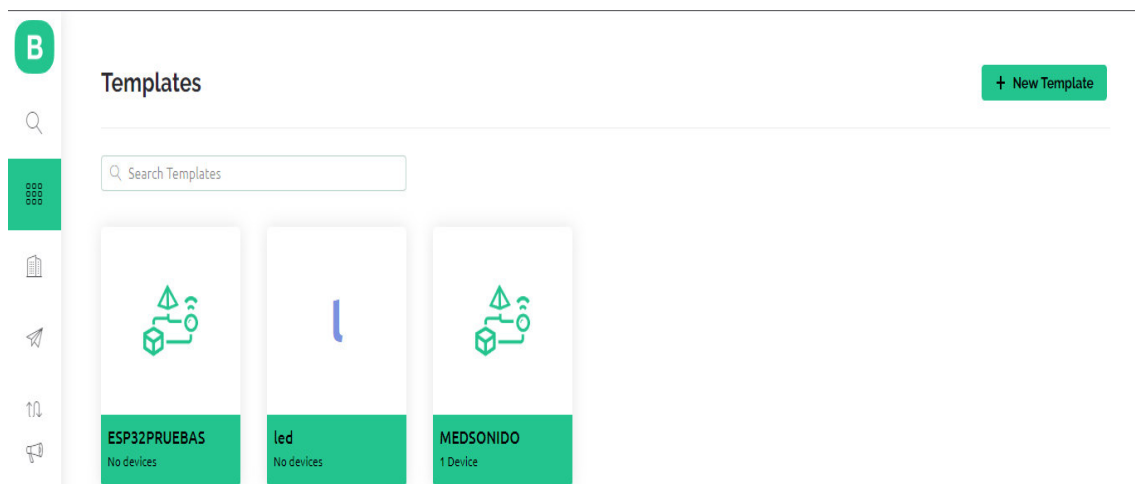
    if(valor_son > 511){ // condicional del
        Serial.println("Ruido excesivo"); //
        mySerial.print('A'); // variable que
        lcd.setCursor(0,0);
        lcd.print(" RUIDO EXCESIVO"); // imp
    }
    delay(200);
}

```

**Figura 3.17** Rangos de medición para alerta

### Creación de la aplicación para notificación en Blynk

En la Figura 3.18 se muestra la pantalla de inicio para realizar la aplicación para notificación de la plataforma de Blynk. Para iniciar sesión, solo se debe registrar con un correo electrónico y crear un *template* para el dispositivo.



**Figura 3.18** Página de inicio de Blynk

Después, se abrirá el *template* creado para el dispositivo y se obtendrá la información que se muestra en la Figura 3.19.

```
FIRMWARE CONFIGURATION

#define BLYNK_TEMPLATE_ID "TMPLAGLRtNxd"
#define BLYNK_DEVICE_NAME "MEDSONIDO"

Template ID and Device Name should be included at the top of your
main firmware
```

**Figura 3.19** Nombre e ID *template* del dispositivo

Como se muestra en la Figura 3.20 y Figura 3.21 se ingresa a la pestaña de *Events* y se selecciona el *template*; aquí, se activará la opción de *warning* seguido de la pestaña *notification* y *Enable notification*.

### Edit Event

General Notifications

EVENT NAME: ALARMASON      EVENT CODE: alarmason

TYPE: Info **Warning** Critical Content

DESCRIPTION (OPTIONAL): Event description (optional)

0 / 300

Cancel

**Figura 3.20** Creación de notificación

## Edit Event

General **Notifications**

Enable notifications

Default recipients

E-MAIL TO

PUSH NOTIFICATIONS TO

SMS TO

Notifications limit

**Figura 3.21** Activación de notificación

### Programación de ESP-32

Para el módulo ESP-32, primero se procede a crear un dispositivo en la plataforma Blynk y también se procede a descargar la aplicación en un dispositivo móvil. En el código de programación se obtendrá de la plataforma un *token* automático y un ID independiente del dispositivo al que esté conectado. En la Figura 3.22 se muestran las librerías para la conexión Wi-Fi y de la plataforma para la alerta.

```
#include <WiFi.h> //libreria Wi-Fi
#include <WiFiClient.h> //libreria que usa Wi-Fi como cliente
#include <BlynkSimpleEsp32.h> //libreria para conectarse a ESP32

#define BLYNK_TEMPLATE_ID "TMPLAGLRtNxd" //Template de la aplicacion
#define BLYNK_DEVICE_NAME "MEDSONIDO" // nombre del dispositivo virtual creado en BLYNK
#define BLYNK_AUTH_TOKEN "5v0oRJN4SAlFsRDFyrot_5xZd7cYHQrc"// ID del Token creado automaticamente en BLYNK

char auth[] = BLYNK_AUTH_TOKEN; // variable tipo caracter que llama al Token de la APP
char ssid[] = " "; // nombre de la red a la que se conectara
char pass[] = " "; // contraseña de la red a la que se conectara

char valor; //variable que tomara desde Arduino para mostrar si el valor se a excedido del propuesto
```

**Figura 3.22** Librerías del módulo ESP-32

Para finalizar la programación del módulo, se procede a ingresar los comandos como se muestra en la Figura 3.23, que permiten enviar la transmisión hacia Blynk cuando el prototipo detecte un sonido que sobrepase el límite.

```

void setup(){
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80); //autenticacion, nombre de l:
}

void loop(){

  delay(2000);
  //Blynk.logEvent("SONIDOALTO", "El Sonido es demasiado Alto");

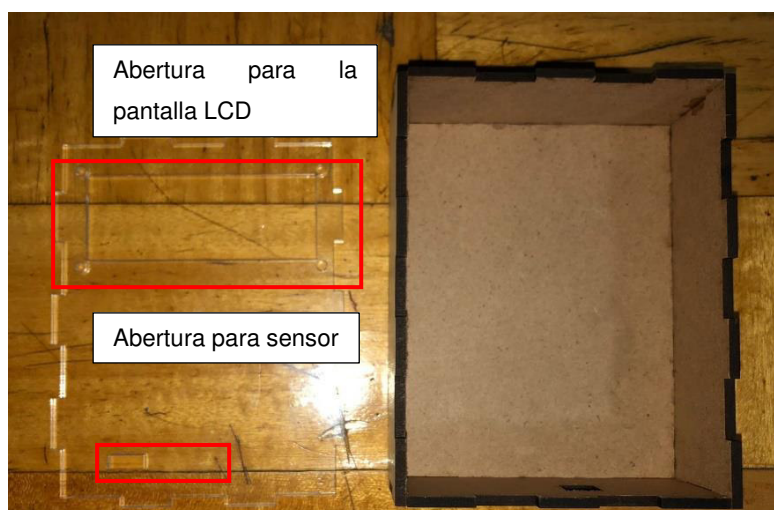
  while(Serial.available() > 0){
    valor = Serial.read();
    //Serial.println(valor);
  }
  if(valor == 'A'){
    Serial.println("El ruido es excesivo");
    Blynk.logEvent("alarmason", "El Ruido es Excesivo, por favor verificar");
    //Blynk.run();
    delay(5000);
  }
  else if(valor == 'B'){
    Serial.println("El ruido es normal");
  }
  Blynk.run();
}

```

**Figura 3.23** Void setup y void loop del ESP-32

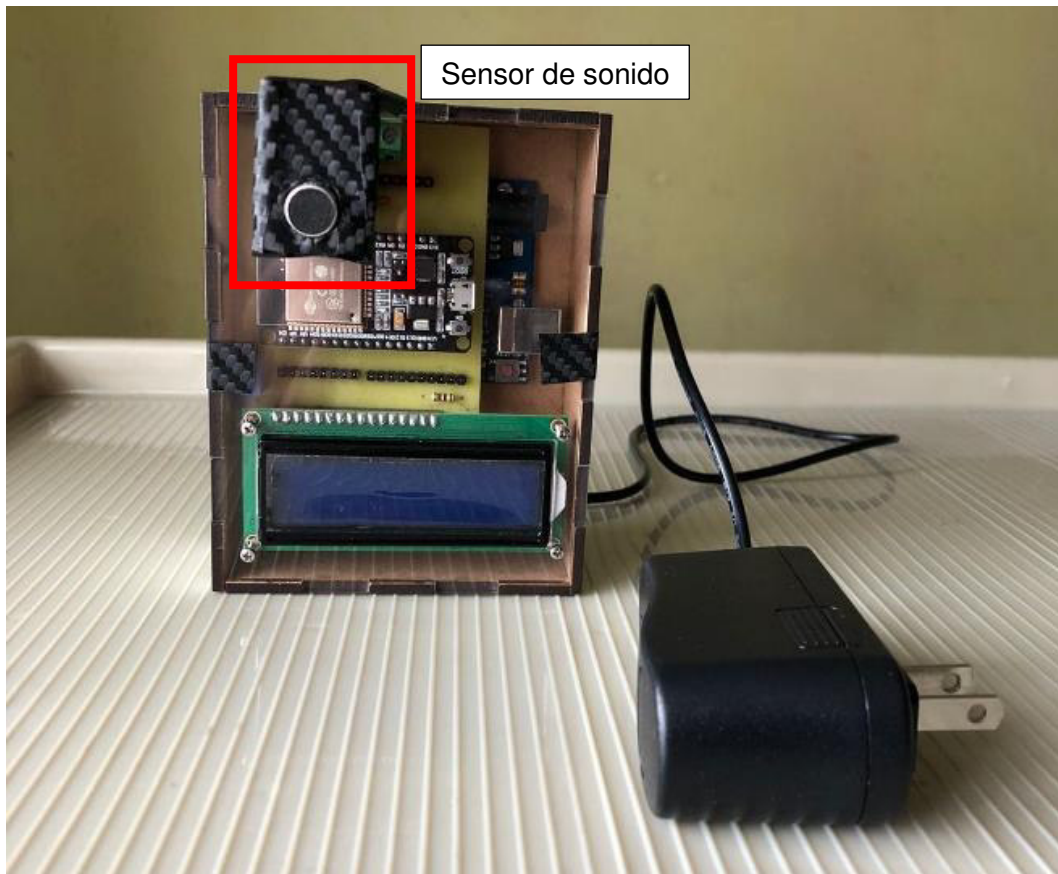
### Implementación del prototipo en su case

Para la creación del *case*, como se puede ver en la Figura 3.24, los materiales usados fueron madera de 3 (mm) y acrílico transparente de 3 (mm), los cuales se distribuyeron en 6 partes formando así un prisma rectangular el cual contendrá en su interior el circuito. En la cara frontal del prisma se tiene dos aberturas, una de 0.5x1 (cm) para la salida del sensor y otra de 7x2.5 (cm) para la salida de la pantalla LCD de 16x2. Se debe tomar en cuenta que la cara frontal estará hecha con acrílico para así poder observar el funcionamiento de los diferentes dispositivos conectados a la placa PCB.



**Figura 3.24** Case del prototipo

En función de que el prototipo no ocupa un espacio considerable, se planteó que el único dispositivo que tendrá contacto directo con el exterior será el sensor de sonido, ya que está conectado hacia el exterior y fuera del *case* del prototipo. Como se puede observar en la Figura 3.25, el sensor quedará afuera para poder captar de manera inmediata el ruido y así también la fuente de energía del prototipo se conectará a dos borneras y esta irá a una toma de corriente.



**Figura 3.25** Prototipo armado

### **Costes de la implementación del prototipo**

En la Tabla 3.6 se muestra el precio en dólares americanos (\$) de cada uno de los elementos utilizados para la creación del prototipo.

**Tabla 3.6** Precio de cada elemento del prototipo

<b>Elementos</b>	<b>Precio (\$)</b>
Arduino Uno	\$23,00
ESP-32	\$13,00
Sensor de sonido KY0-37	\$5,00
Pantalla LCD 16x2	\$6,25

Continúa

Cargador 5 (V), 2 (A)	\$3,50
Borneras	\$0,75
Capacitores 10uF y 100uF	\$2,50
Baquelita 20x20 (cm)	\$4,00
Ácido férrico	\$1,75
Case de madera y acrílico	\$14,00
Total	\$73,75

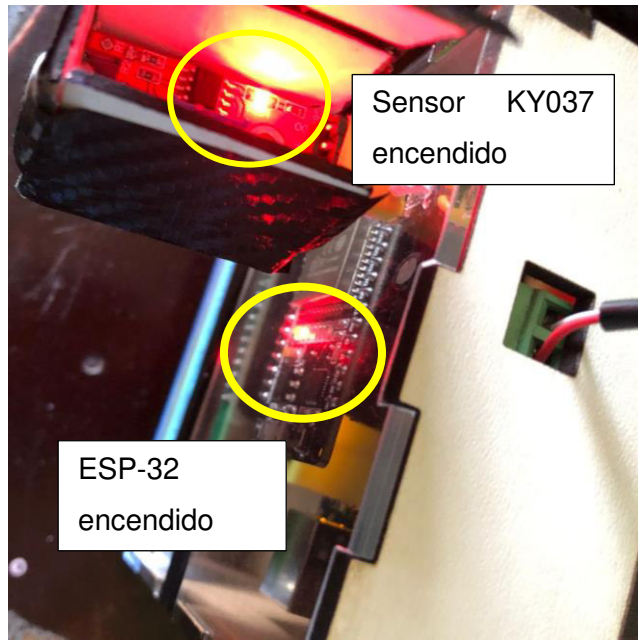
### 3.5 Pruebas de funcionamiento

Cuando la programación y calibración están listas, la placa funcional y el *case* armado, se procede a realizar las pruebas de correcto funcionamiento. Como se puede observar en la Figura 3.26 cuando el prototipo se enciende, se mostrará un mensaje de “MEDIDOR DE SONIDO” y procederá a medir el ruido.



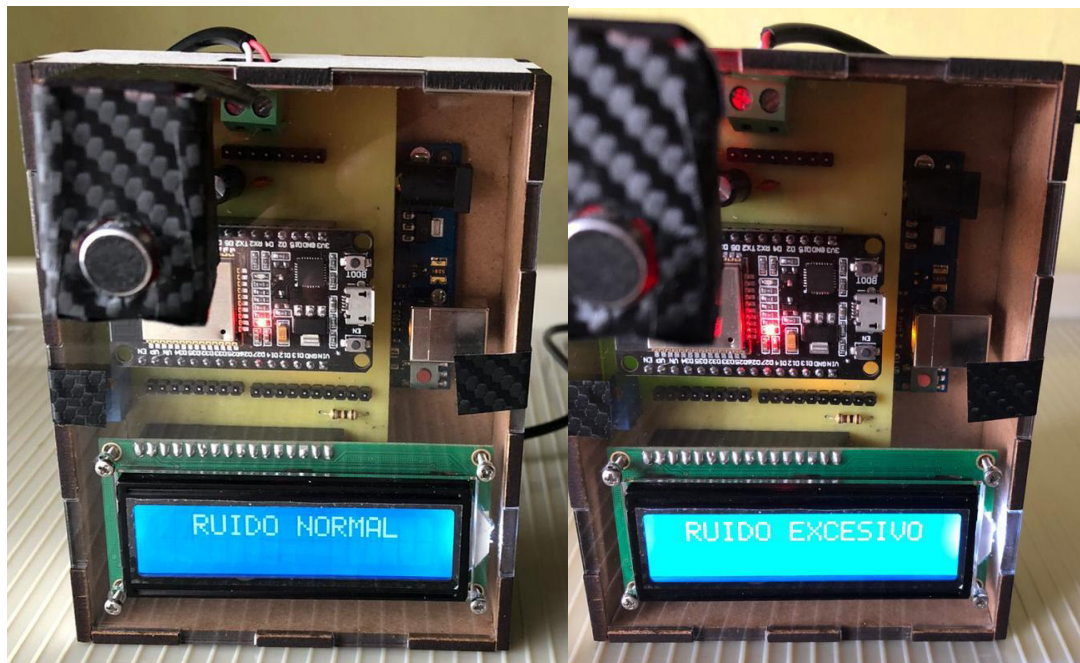
**Figura 3.26** Prototipo encendido

En la Figura 3.27 se puede observar tanto el sensor como el módulo ESP-32, que están funcionando con normalidad.



**Figura 3.27** Sensor y ESP-32 encendidos

El sensor procede a medir el sonido, de tal manera que, cuando se sobrepase el nivel de sonido programado, se mostrará el mensaje de “RUIDO EXCESIVO” y cuando el sonido sea normal este mostrará el mensaje de “RUIDO NORMAL”; esto se puede observar en la Figura 3.28.



**Figura 3.28** Ruido detectado

Cuando el nivel de ruido sea excesivo, este enviará un mensaje a la aplicación de Blynk, el cual aparecerá como notificación y esta indicará que el ruido es excesivo y que se tome las medidas correspondientes; por ejemplo, que el encargado de la biblioteca solicite a los presentes hacer silencio. Cabe recalcar que la aplicación es orientada a quienes estén a cargo del lugar donde se medirá el sonido. Estas notificaciones se pueden observar en la Figura 3.29.



**Figura 3.29** Notificación de Blynk

En la Tabla 3.7 se muestran los resultados de las pruebas realizadas al dispositivo

**Tabla 3.7** Parámetros de prueba

<b>Parámetro de prueba</b>	<b>Funcionamiento correcto</b>	<b>Funcionamiento incorrecto</b>
<b>Encendido del prototipo</b>	X	
<b>Mensaje en pantalla al encenderse</b>	X	
<b>Mensaje en pantalla de ruido normal</b>	X	
<b>Mensaje en pantalla de ruido excesivo</b>	X	
<b>Notificación desde la aplicación</b>	X	

En el Anexo II.I se incluye el enlace y código QR a través de los cuales se puede verificar el funcionamiento del sistema implementado.



## 4 CONCLUSIONES

- Teniendo en cuenta que el sonido captado por el sensor del sistema debe encontrarse en un rango superior a 39 (dB), se ha considerado programarlo y configurarlo manualmente para que se pueda tener la mayor sensibilidad posible al momento de captar esta señal por el micrófono analógico que tiene el módulo sensor de sonido, de tal manera que se pueda tener una respuesta casi inmediata de un aviso por aplicación y pantalla. La calibración consiste en programar los niveles analógicos de bits para ser transformados en una regla de tres y obtener el valor en decibles; de igual manera, el potenciómetro manual con el que cuenta el sensor de sonido se tiene que modificar para calibrar su sensibilidad.
- Gracias a los módulos que son de un tamaño realmente adaptable para un *case* pequeño, se puede tener una fuente de energía fija, la corriente eléctrica que se proporciona por una conexión de 110 (V) se podrá usar con el sistema ya que se tendrá un transformador que permite pasar este voltaje a lo requerido por la placa Arduino y por los módulos. Adicionalmente, se considera que el sistema permanecerá fijo en una biblioteca o auditorio.
- La facilidad con la que el módulo Wi-Fi ESP-32 se puede programar es una gran ventaja; esta ventaja se debe a que todos los pines del módulo son programables para las necesidades del sistema, así no se tendrá problema al momento de preocuparse por la falta de pines. La conexión Tx y Rx de la placa Arduino también facilita la comunicación con los componentes del sistema y el módulo Wi-Fi ESP-32.
- Para el control de la pantalla LCD es necesaria la configuración de una resistencia de 330 ( $\Omega$ ) para protección del LED que tiene integrada la pantalla y también de un potenciómetro de 1 (k $\Omega$ ) para ajustar el *backlight* de la pantalla.
- Ya que Arduino IDE es especialmente para aquellos quienes tienen un reciente ingreso al mundo de la programación, su entorno con quienes no están familiarizados con lenguajes de programación avanzados, se vuelve amigable porque solo se requiere tener conocimientos básicos para poder desarrollar un programa en específico.
- Gracias a que la aplicación de Blynk brinda muchas facilidades a la hora de crear tanto una notificación vía *web* como una aplicación, esta misma se

encarga de brindar las librerías con las que se puede programar en el entorno de Arduino IDE. En el caso de Wi-Fi, se puede encontrar las librerías del cliente y del dispositivo en la *web*, esto instalando en el administrador de librerías en el mismo entorno de Arduino.

- La conexión Wi-Fi depende del ancho de banda y velocidad a la que el módulo ESP-32 esté conectado ya que, si se tiene una buena velocidad, se podrá conectar más rápido hacia la *web* y así también enviar la notificación hacia la aplicación. Por otro lado, la aplicación depende de la velocidad y de la saturación de la base de datos de la plataforma de Blynk.

## 5 RECOMENDACIONES

- Se recomienda que la calibración del sensor de sonido se realice en el lugar donde se implementará el sistema de detección; así, se ajustará a la necesidad del lugar.
- Para tener un mejor funcionamiento del sistema, es necesario comprar una suscripción a la plataforma de Blynk, ya que se tendrá la ventaja de poder tener notificaciones ilimitadas al correo y a la aplicación. La versión gratuita es suficiente para desarrollar el prototipo, considerando que solo se puede enviar 100 notificaciones diarias.
- El prototipo desarrollado se adapta adecuadamente al entorno de una biblioteca, pues en dicho lugar siempre se requiere de un ambiente en absoluto silencio. Para el desarrollo de un prototipo para auditorios, se recomienda añadir un interruptor de modo activo e inactivo, de modo que cuando por el propio uso del lugar se haga ruido, el sistema no envíe alertas, pero cuando se requiera silencio, se pueda cambiar al estado activo.
- Si se deseara tener un sistema que abarque un lugar mucho más amplio, se recomienda usar un vúmetro para así tener mayor precisión en los lugares que se desea controlar ya que al tener un espacio más amplio, el prototipo no será suficiente para abarcar el espacio total.
- El sensor de sonido debe tener una conexión directa o casi directa con la placa ya que, al momento de utilizar cables para realizar su conexión, podría ingresar ruido al sistema evitando que su funcionamiento sea el óptimo.

- Gracias a la cantidad de pines que se tiene en el ESP-32, dicho módulo se podría convertir en la placa principal del prototipo ya que cuenta con un microprocesador que puede ser programado en el entorno de Arduino IDE. Esto, si en un futuro se quiere abaratar costos, pero se tendrá que tomar en cuenta que la opción de conectar directamente voltajes superiores a los 3.3 (V) quedará anulada ya que el dispositivo tiene pines de entrada y salida de 3.3 (V) y los demás elementos del prototipo funcionan con 5 (V).

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Pujol, R., 2022. *Viaje al Centro de la audición*. [online] Cochlea.org. Available at: <<http://www.cochlea.org/es/sonidos/campo-auditivo-humano>> [Accessed 11 May 2022].
- [2] Arduino Project Hub. 2022. *How to Use KY-037 Sound Detection Sensor with Arduino*. [online] Available at: <<https://create.arduino.cc/projecthub/electropeak/how-to-use-ky-037-sound-detection-sensor-with-arduino-a757a7>> [Accessed 11 May 2022].
- [3] DatasheetCafe. 2022. *KY-037 Datasheet - Sound Sensor Module with Arduino*. [online] Available at: <<http://www.datasheetcafe.com/ky-037-datasheet-microphone-sensor/>> [Accessed 11 May 2022].
- [4] Estudio, A., 2022. *¿Cuáles son las ventajas que nos da Arduino?*. [online] Cerebro Curioso. Available at: <<https://cerebrocurioso.com.ar/cuales-son-las-ventajas-que-nos-da-arduino/>> [Accessed 11 May 2022].
- [5] SoftwareLab. 2022. *¿Qué es WiF, qué significa y para qué sirve? // SoftwareLab*. [online] Available at: <<https://softwarelab.org/es/que-es-wifi-que-significa-y-para-que-sirve/>> [Accessed 27 June 2022]
- [6] Qué es el Wifi 6: definición, c., 2022. *Wifi 6: definición, cobertura y eficie*. [online] LPS Ingenieria, empresa consultoría y dirección de proyectos de ingeniería y Telecomunicaciones. Available at:

- <<https://lpsingenieria.com/que-es-wifi-6-cobertura-eficiencia-moviles-que-lo-consideran/>> [Accessed 27 June 2022].
- [7] Descubrearduino.com. 2022. NodeMCU una plataforma para IOT de código abierto. [online] Available at: <<https://descubrearduino.com/nodemcu/>> [Accessed 1 June 2022].
- [8] Widex.cl. 2022. [online] Available at: <<https://www.widex.cl/es-cl/blog/rango-auditivohumano#:~:text=Mientras%20que%20a%20a%20C000,a%20partir%20de%200%20dB.>> [Accessed 11 May 2022].
- [9] Arduino.cl - Compra tu Arduino en Línea. 2022. *Principales partes de un Arduino | Arduino.cl - Compra tu Arduino en Línea.* [online] Available at: <<https://arduino.cl/principales-partes-de-un-arduino/>> [Accessed 18 May 2022]
- [10] Aguirre, C., 2022. *Uso de los Sensores de sonido KY-038/KY-037 para controlar el encendido de un foco - UNIT Electronics.* [online] UNIT Electronics. Available at: <<https://blog.uelectronics.com/tarjetas-desarrollo/uso-de-los-sensores-de-sonido-ky-038-ky-037-para-controlar-el-encendido-de-un-foco/#:~:text=INTRODUCCI%C3%93N,sensibilidad%20que%20tiene%20cada%20uno.>> [Accessed 25 May 2022].
- [11] Descubrearduino.com. 2022. ESP32 vs ESP8266 ¿Cuáles son las diferencias entre ambos módulos?. [online] Available at: <<https://descubrearduino.com/esp32-vs-esp8266/#:~:text=Veamos%20cu%C3%A1l%20es%20la%20diferencia,de%20n%C3%BAcleo%20%C3%BAnico%20de%20160MHz.>> [Accessed 25 May 2022].
- [12] Gudiño, M., 2022. [online] Available at: <<https://www.arrow.com/es-mx/research-and-events/articles/arduino-uno-vs-mega-vs-micro>> [Accessed 25 May 2022].

- [13] Volt, C., 2022. *¡ESP8266 Y ESP32 HACKEADOS! – RogerBit*. [online] Rogerbit.com. Available at: <<https://rogerbit.com/wprb/2019/09/esp8266-y-esp32-hackeados/>> [Accessed 25 May 2022].
- [14] Arduino.cl - Compra tu Arduino en Línea. 2022. Software de Arduino | Arduino.cl - Compra tu Arduino en Línea. [online] Available at: <<https://arduino.cl/programacion/>> [Accessed 2 June 2022].
- [15] Aprendiendo Arduino. 2022. Conversor Analógico Digital – Aprendiendo Arduino. [online] Available at: <<https://aprendiendoarduino.wordpress.com/tag/conversor-analogico-digital/#:~:text=Arduino%20Uno%20tiene%20una%20resoluci%C3%B3n,valores%20entre%200%20y%201023.>> [Accessed 9 June 2022].
- [16] 2022. [online] Available at: <<https://www.capterra.ec/software/179337/blynk>> [Accessed 15 June 2022].
- [17] aula21 | Formación para la Industria. 2022. Qué es un Autómata Programable o PLC y cómo funciona | Aula21. [online] Available at: <<https://www.cursosaula21.com/que-es-un-automata-programable-o-plc-y-como-funciona/#:~:text=C%C3%B3mo%20funciona%20un%20aut%C3%B3mata%20programable,salidas%20bas%C3%A1ndose%20en%20par%C3%A1metros%20programados.>> [Accessed 9 June 2022].
- [18] Domestika. 2022. ¿Qué es SketchUp y cómo empezar a usarlo? | Blog | Domestika. [online] Available at: <<https://www.domestika.org/es/blog/5619-que-es-sketchup-y-como-empezar-a-usarlo>> [Accessed 15 June 2022].

## 7 ANEXOS

### ANEXO I: Certificado de Originalidad

#### CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 20 de Agosto de 2022

De mi consideración:

Yo, FANNY PAULINA FLORES ESTÉVEZ, en calidad de Director del Trabajo de Integración Curricular titulado IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE ALERTA POR SONIDO elaborado por el estudiante ERICK ESTEBAN HIDALGO PORTILLO de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

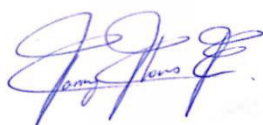
El documento escrito tiene un índice de similitud del 12%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

[https://epnecuador-my.sharepoint.com/:b:/g/personal/fanny\\_flores\\_epn\\_edu\\_ec/EYuM5W8vRNIFIO9G6Q916UAB0QMXLanq\\_UGTIkTm0FkXjw?e=woHNLs](https://epnecuador-my.sharepoint.com/:b:/g/personal/fanny_flores_epn_edu_ec/EYuM5W8vRNIFIO9G6Q916UAB0QMXLanq_UGTIkTm0FkXjw?e=woHNLs)

Atentamente,



FANNY PAULINA FLORES ESTÉVEZ

Docente

Escuela de Formación de Tecnólogos

## ANEXO II: Enlaces



**Anexo II.I** Código QR de la implementación y pruebas de funcionamiento

El acceso del video se encuentra en el siguiente enlace: <https://youtu.be/yRaARgNrAJc>

## ANEXO III: Códigos Fuente

### Código Arduino Uno

```
#include <LiquidCrystal.h> //libreria del LCD

#include <SoftwareSerial.h> //libreria de los puertos seriales virtuales

SoftwareSerial mySerial(2, 3); // RX, TX estas son variabls virtuales creadas para no
interferir con las RX y TX propias del Arduino

const int rs = 5, en = 6, d4 = 7, d5 = 8, d6 = 9, d7 = 10; //variables constantes que se
utiliza para controlar la pantalla LCD

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int valor_son; // variable entera con la que el sensor de sonido capta la señal

bool val_son_d; // variable booleana que solo se refiere a 0 o 1

void setup() {

Serial.begin(9600);

mySerial.begin(115200); // velocidad de trasmision de datos

lcd.begin(16, 2);

pinMode(A0, INPUT); // pin de lectura analoga para el sensor

pinMode(4, INPUT); // pin de lectura digital para el sensor

lcd.setCursor(4,0);

lcd.print("ALARMA");

lcd.setCursor(3,1);

lcd.print("DE SONIDO");

delay(2000);

lcd.clear();

}

void loop() {

valor_son = analogRead(A0); //declarar variables analogicas y digitales
```



```

val_son_d = digitalRead(4);

Serial.print("Val_anag: "); // impresion de en la pantalla de serial del valor analogico

Serial.println(valor_son);

Serial.print("Val_dig: "); // impresion de en la pantalla de serial del valor digital

Serial.println(val_son_d);

if(valor_son <= 500 && valor_son < 510){    // condicional del rango de sonido que
puede ser variable

Serial.println("  Ruido normal"); // impresion de mensaje si el rango de sonido no es
excedido

mySerial.print('B'); // variable que se envia al ESP32

lcd.setCursor(0,0);

lcd.print(" RUIDO NORMAL "); // impresion de mensaje en el LCD

}

if(valor_son >= 510 && valor_son < 511){    // condicional del rango de sonido que
puede ser variable (Tolerancia)

}

if(valor_son > 511){ // condicional del rango de sonido que puede ser variable

Serial.println("Ruido excesivo"); // impresion de mensaje si el rango de sonido es
excedido

mySerial.print('A'); // variable que se envia al ESP32

lcd.setCursor(0,0);

lcd.print(" RUIDO EXCESIVO"); // impresion de mensaje en el LCD

}

delay(200);

}

```

## Código ESP-32

```
#define BLYNK_PRINT Serial //libreria que permite imprimir en BLYNK

#include <WiFi.h> //libreria Wi-Fi

#include <WiFiClient.h> //libreria que usa Wi-Fi como cliente

#include <BlynkSimpleEsp32.h> //libreria para conectarse a ESP32

#define BLYNK_TEMPLATE_ID "TMPLAGLRtNxd" //Template de la aplicacion

#define BLYNK_DEVICE_NAME "MEDSONIDO" // nombre del dispositivo virtual
creado en BLYNK

#define BLYNK_AUTH_TOKEN "5v0oRJN4SAIFsRDFyrot_5xZd7cYHQrc"// ID del
Token creado automaticamente en BLYNK

char auth[] = BLYNK_AUTH_TOKEN; // variable tipo caracter que llama al Token de la
APP

char ssid[] = "NETLIFE-uiordportillot1"; // nombre de la red a la que se conectara

char pass[] = "1713597860"; // contraseña de la red a la que se conectara

char valor; //variable que tomara desde Arduino para mostrar si el valor se a excedido
del propuesto

void setup(){

Serial.begin(115200);

Blynk.begin(auth, ssid, pass, "blynk.cloud", 80); //autenticacion, nombre de la red,
contraseña, llamda al cloud de blynk por el puerto 80

}

void loop(){

delay(2000);

while(Serial.available() > 0){

valor = Serial.read();

}

if(valor == 'A'){

Serial.println("El ruido es excesivo");
```

```
Blynk.logEvent("alarmason", "El Ruido es Excesivo, por favor verificar");  
  
delay(5000);  
  
}  
  
else if(valor == 'B'){  
  
Serial.println("El ruido es normal");  
  
}  
  
Blynk.run();  
  
}
```