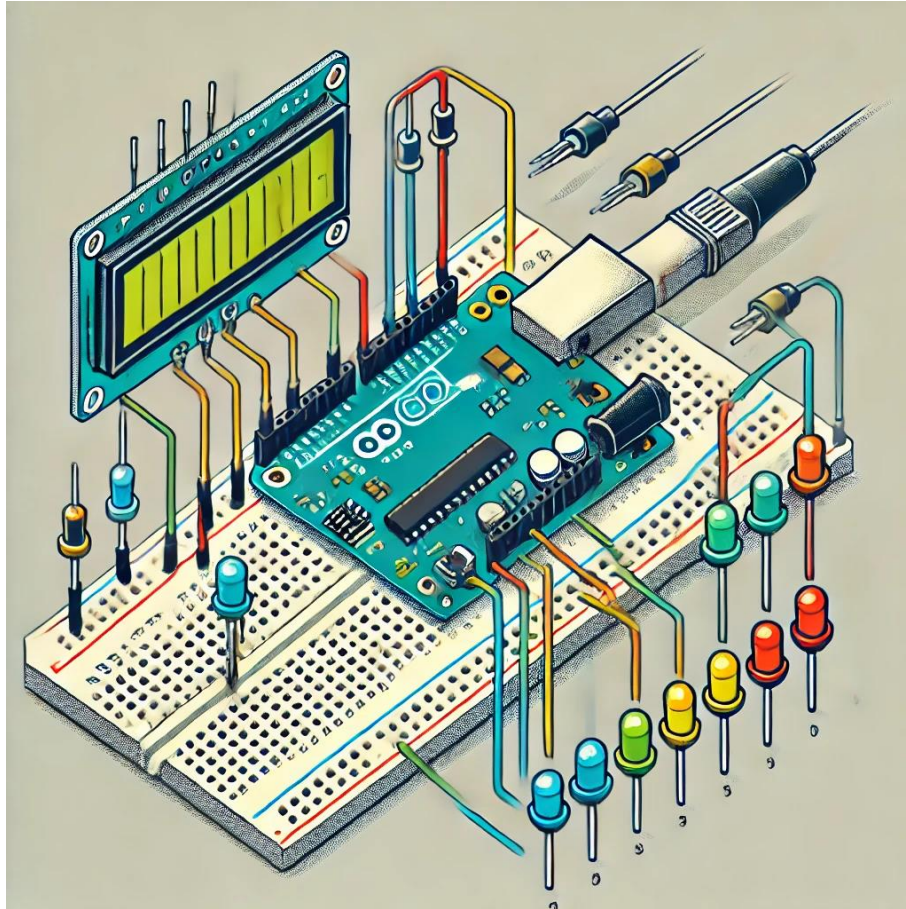


# Sistema de Alerta de Nivel de Ruido para Ambientes Sensibles



Integrantes:

Abigail Caro.

Manuel Medina.

Constanza Martínez.

## INTRODUCCION

Este proyecto tiene como objetivo el desarrollo de un **Sistema de Alerta de Nivel de Ruido para Ambientes Sensibles**. Utilizando **el IDE de Arduino**, se creó un dispositivo que mide el nivel de ruido ambiente mediante un **sensor de sonido**, y presenta los datos de forma visual a través de una **pantalla OLED**. Además, el sistema emite alertas con **LEDs de colores** (verde, amarillo, rojo) que indican el nivel de ruido (bajo, moderado, alto). El módulo **WiFi** integrado permite monitorear los niveles de ruido de manera remota.

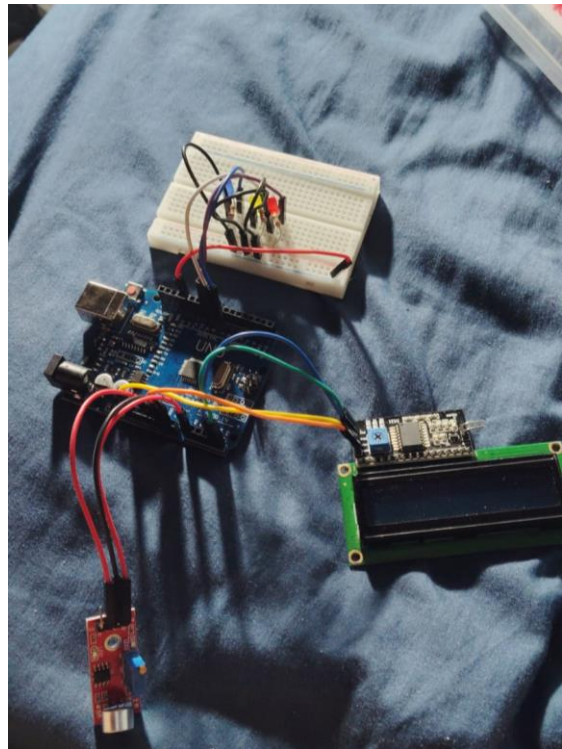
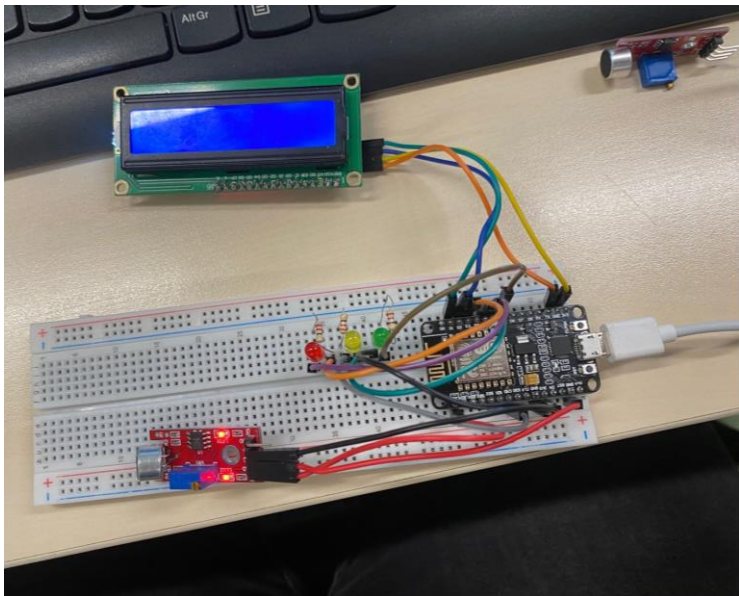
### ➤ **Objetivo**

El sistema está diseñado para ambientes donde el control del ruido es crucial, como hospitales, bibliotecas, o guarderías. A través de sus alertas visuales y la pantalla OLED, el sistema ayuda a las personas a mantener los niveles de ruido dentro de un rango aceptable, mejorando así la comodidad y la calidad del ambiente.

## Descripción Técnica del Proyecto

### Componentes Utilizados

- **Sensor de Sonido:** Permite medir los niveles de ruido ambiente en decibelios (dB).
- **Pantalla OLED:** Muestra los datos del nivel de ruido en tiempo real, facilitando la interpretación visual.
- **LEDs (Verde, Amarillo, Rojo):** Indican el nivel de ruido de forma intuitiva:
  - **Verde:** Ruido bajo.
  - **Amarillo:** Ruido moderado.
  - **Rojo:** Ruido alto.
- **Módulo WiFi (ESP32/ESP8266):** Transmite los datos de ruido a una red o servidor para monitoreo remoto.
- **Microcontrolador (Arduino/ESP32):** Gestiona la lectura del sensor, controla la pantalla OLED y los LEDs, y maneja la comunicación WiFi.



## Implementación Técnica

El dispositivo está programado para realizar las siguientes acciones:

1. **Lectura continua del nivel de ruido** a través del sensor conectado al microcontrolador.
2. **Visualización de los datos** en la pantalla OLED, mostrando los niveles de ruido en tiempo real.
3. **Activación de los LEDs** según el nivel de ruido:
  - Verde si el nivel es bajo.
  - Amarillo si el nivel es moderado.
  - Rojo si el nivel es alto.
4. **Transmisión de datos a través de WiFi**, lo que permite monitoreo remoto de los niveles de ruido.

## Implementación del Código en el IDE de Arduino

### Resumen del Código del Proyecto

El código está diseñado para el **Sistema de Alerta de Nivel de Ruido para Ambientes Sensibles**, utilizando el **IDE de Arduino**. El sistema emplea un **sensor de sonido** para medir el nivel de ruido ambiente y una **pantalla OLED** para mostrar los resultados. Además, los **LEDs** proporcionan una señal visual sobre el estado del ruido.

### Componentes y Configuración:

1. **Pines y Hardware:**
  - **LEDs** de colores (verde, amarillo, rojo) conectados a los pines digitales 5, 4, y 3, respectivamente.
  - **Sensor de sonido** conectado al pin analógico A0.
  - **Pantalla OLED** para mostrar los datos, conectada a través del bus I2C.
2. **Conexión WiFi:**
  - El código permite la conexión a una red WiFi mediante un módulo ESP8266, lo que permite monitorear remotamente el nivel de ruido.

### Funcionamiento del Código:

## 1. Inicialización:

- Se establecen los pines para los LEDs como salidas.
- Se intenta establecer la conexión WiFi con las credenciales predefinidas.
- La pantalla OLED se inicializa para mostrar los resultados.

## 2. Lectura y Visualización:

- En el ciclo continuo (loop()), el código lee el valor del sensor de sonido a través de analogRead(A0).
- El valor del sonido se muestra en la pantalla OLED en tiempo real.
- Según el nivel de ruido detectado:
  - **Verde:** Ruido bajo.
  - **Amarillo:** Ruido moderado.
  - **Rojo:** Ruido alto, lo que indica que se debe reducir el volumen.

## Estructura del Código:

- **Configuración inicial:** Definición de pines y parámetros, conexión a WiFi, inicialización de la pantalla.
- **Lectura de sensor y control de LEDs:** Determinación del nivel de ruido y encendido de LEDs según el umbral.
- **Visualización en pantalla OLED:** Muestra el nivel de ruido en formato de texto y gráfico.

```
sketch_nov25a | Arduino IDE 2.0.3
File Edit Sketch Tools Help

sketch_nov25a.ino
1 #include <Wire.h>
2 #include <Adafruit_SSD1306.h> // Librería para OLED
3 #include <Adafruit_GFX.h>
4
5 #define SCREEN_WIDTH 128
6 #define SCREEN_HEIGHT 64
7 #define OLED_RESET -1
8 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
9
10 // Definir los pines de los LEDs utilizando GPIO directamente
11 #define LED_RED 14 // LED Rojo en D5, GPIO14
12 #define LED_YELLOW 2 // LED Amarillo en D4, GPIO2
13 #define LED_GREEN 0 // LED Verde en D3, GPIO0
14 #define MICROPHONE A0 // Micrófono conectado a A0
15
16 void setup() {
17   Serial.begin(115200);
18
19   // Inicialización de la pantalla OLED
20   Wire.begin(5, 4); // SDA en GPIO5 y SCL en GPIO4 (usados en muchos modelos de ESP8266)
21   if (!display.begin(0x3C, OLED_RESET)) {
22     Serial.println(F("No se puede inicializar el OLED"));
23     while (true);
24   }
25   display.display();
26   delay(2000); // Espera para mostrar el logo de inicio
27
28   // Configura los pines de los LEDs
29   pinMode(LED_RED, OUTPUT);
30   pinMode(LED_YELLOW, OUTPUT);
31   pinMode(LED_GREEN, OUTPUT);
32
33   // Configura el pin del micrófono
34   pinMode(MICROPHONE, INPUT);
35 }
36
37 void loop() {
38   // Lee el valor del micrófono
39   int micValue = analogRead(MICROPHONE);
40
41   // Muestra el valor del micrófono en la pantalla OLED
42   display.clearDisplay();
43   display.setTextSize(1);
44   display.setTextColor(SSD1306_WHITE);
45   display.setCursor(0, 0);
46
47   while (true);
48 }
49
50 // Controla los LEDs dependiendo del valor del micrófono
51 if (micValue > 600) { // Si el micrófono detecta un sonido fuerte
52   digitalWrite(LED_RED, HIGH); // Enciende LED rojo
53   digitalWrite(LED_YELLOW, LOW); // Apaga LED amarillo
54   digitalWrite(LED_GREEN, LOW); // Apaga LED verde
55 } else if (micValue > 400) { // Si el sonido es moderado
56   digitalWrite(LED_RED, LOW); // Apaga LED rojo
57   digitalWrite(LED_YELLOW, HIGH); // Enciende LED amarillo
58   digitalWrite(LED_GREEN, LOW); // Apaga LED verde
59 } else { // Si el sonido es bajo
60   digitalWrite(LED_RED, LOW); // Apaga LED rojo
61   digitalWrite(LED_YELLOW, LOW); // Apaga LED amarillo
62   digitalWrite(LED_GREEN, HIGH); // Enciende LED verde
63 }
64
65 delay(100); // Espera para evitar lecturas rápidas
66 }
```

## Conclusión

El **Sistema de Alerta de Nivel de Ruido para Ambientes Sensibles** proporciona una solución efectiva para monitorear y controlar los niveles de ruido en entornos donde es crucial mantener un ambiente silencioso. El proyecto permite visualizar los niveles de ruido en tiempo real y recibir alertas visuales sobre el estado del sonido. La conectividad WiFi asegura que los datos sean accesibles de manera remota, lo que facilita la supervisión continua.

---

## Próximos Pasos

Para futuras mejoras, se recomienda integrar más características como:

- Enviar notificaciones a dispositivos móviles.
- Registrar los niveles de ruido durante largos periodos y generar reportes.
- Implementar controles adicionales para regular los niveles de sonido de forma automática.

