

```

#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>

// Definindo os pinos
const int micPin = 13; // Pino digital conectado à saída do sensor de som
const int ledPins[] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11}; // Pinos digitais
conectados aos LEDs da barra
const int numLeds = sizeof(ledPins) / sizeof(ledPins[0]); // Número de LEDs na
barra
LiquidCrystal_I2C lcd(0x3F, 16, 2);

// Variável para contar o número de vezes que o som é detectado acima dos
limiares
int soundCount = 0;

// Variável para contar quantas vezes a barra de LED chegou a 10.000 variações
int tenThousandCount = 0;

void setup() {
    // Configurando os pinos dos LEDs como saída
    for (int i = 0; i < numLeds; i++) {
        pinMode(ledPins[i], OUTPUT);
        digitalWrite(ledPins[i], LOW); // Inicialmente apaga todos os LEDs
    }

    pinMode(micPin, INPUT);

    lcd.begin(16, 2); // Inicializa o display LCD
    lcd.init();       // Inicializa o display LCD
    lcd.backlight();  // Liga o backlight do display LCD
    lcd.leftToRight(); // Define a direção da escrita da esquerda para a
direita
    lcd.setCursor(2, 0); // Define a posição do cursor
    lcd.print("Ola, me chamo"); // Escreve uma mensagem no display LCD
    lcd.setCursor(1, 1); // Define a posição do cursor
    lcd.print("SILENCE WARDEN"); // Escreve uma mensagem no display LCD
    delay(5000);

    // Inicializando a comunicação serial
    Serial.begin(9600);
}

```

```

void displayLCD(int soundCount, int tenThousandCount) {
    // Função para exibir uma mensagem no display LCD
    lcd.clear();                // Limpa o display LCD
    lcd.setCursor(5, 0);        // Define a posição do cursor
    lcd.print("STATUS:");       // Escreve uma mensagem no display LCD
    lcd.setCursor(1, 1);        // Define a posição do cursor

    lcd.print("R:");            // SC para Sound Count
    lcd.print(soundCount);       // Escreve a contagem de sons no display LCD
    lcd.print("    C:");         // TC para Ten Thousand Count
    lcd.print(tenThousandCount); // Escreve a contagem de 10 mil no display LCD
}

```

```

void updateLedBar(int soundCount) {
    // Função para atualizar a barra de LED
    int ledsToLight = (soundCount / 1000); // Acende um LED a cada 1000
    variações
    for (int i = 0; i < numLeds; i++) {
        if (i < ledsToLight) {
            digitalWrite(ledPins[i], HIGH);
        } else {
            digitalWrite(ledPins[i], LOW);
        }
    }
}

```

```

void loop() {
    // Lendo o valor do microfone
    int soundLevel = digitalRead(micPin);

    // Verificando se o valor lido é HIGH (acima do limiar)
    if (soundLevel == HIGH) {
        // Incrementando a contagem de sons detectados
        soundCount++;

        // Verificando se atingiu 10.000 variações
        if (soundCount >= 10000) {
            // Incrementando a contagem de 10.000
            tenThousandCount++;

            // Resetando a contagem de sons
            soundCount = 0;
        }
    }
}

```

```
}

// Imprimindo a contagem de sons detectados na serial
Serial.print("Sound Count: ");
Serial.print(soundCount);
Serial.print(" | Ten Thousand Count: ");
Serial.println(tenThousandCount);

// Atualizando o display LCD
displayLCD(soundCount, tenThousandCount);

// Atualizando a barra de LED
updateLedBar(soundCount);

// Pequena pausa para evitar sobrecarga no loop
delay(50);
}
}
```