

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Endereço I2C e dimensões do LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Pinos do sistema
const int TMP = A0;
const int LED_AZUL = 5;
const int LED_VERM = 6;
const int BUZZER = 7;

unsigned long lastBlink = 0;
bool estadoLedVermelho = false;

void setup() {
  Serial.begin(9600);

  // Inicializa LCD
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("AiOT_Firefighter");
  delay(2000); // Mostra por 2 segundos a mensagem inicial
  lcd.clear(); // Limpa a tela e deixa estática depois disso

  // Configura pinos
  pinMode(LED_AZUL, OUTPUT);
  pinMode(LED_VERM, OUTPUT);
  pinMode(BUZZER, OUTPUT);

  digitalWrite(LED_AZUL, LOW);
  digitalWrite(LED_VERM, LOW);
  noTone(BUZZER);
}

void loop() {
  float temperatura = lerTemperatura(TMP);

  Serial.print("Temperatura: ");
  Serial.print(temperatura);
  Serial.println(" °C");

  // Atualiza temperatura no LCD
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(temperatura, 1); // uma casa decimal
  lcd.print(" °C "); // limpa resquício visual

```

```

// Atualiza status no LCD e dispositivos
if (temperatura < 60) {
    lcd.setCursor(0, 1);
    lcd.print("Status: Seguro ");

    digitalWrite(LED_AZUL, HIGH);
    digitalWrite(LED_VERM, LOW);
    noTone(BUZZER);
} else {
    lcd.setCursor(0, 1);
    lcd.print("Status: ALERTA ");

    digitalWrite(LED_AZUL, LOW);

    // Piscar LED vermelho a cada 500 ms
    if (millis() - lastBlink > 500) {
        lastBlink = millis();
        estadoLedVermelho = !estadoLedVermelho;
        digitalWrite(LED_VERM, estadoLedVermelho);
    }

    // Buzzer proporcional à temperatura
    int freq = map((int)temperatura, 60, 100, 1000, 3000);
    freq = constrain(freq, 1000, 3000);
    tone(BUZZER, freq);
}

delay(200);
}

float lerTemperatura(int pino) {
    int leitura = analogRead(pino);
    float tensao = leitura * (5.0 / 1023.0);
    float temperatura = (tensao - 0.5) * 100.0;
    return temperatura;
}

```