Guilherme Henrique Bernardi

**DOCUMENTO TÉCNICO  
Sensor de Temperatura para Ambientes Internos**

LIMEIRA – SP  
2025

Sumário

[**1.** **INTRODUÇÃO** 1](#_Toc200112384)

[**2.** **OBJETIVOS E HISTÓRIA DA PIZZARIA** 1](#_Toc200112385)

[*2.1.* *ONDE TUDO COMEÇOU?* 1](#_Toc200112386)

[**3.** **LOGOTIPO DA PIZZARIA** 2](#_Toc200112387)

[**4.** **WIREFRAME E ESBOÇO DO LAYOUT VISUAL DA PÁGINA** 3](#_Toc200112388)

[**5.** **LEVANTAMENTO DE REQUISITOS (FUNCIONAIS E NÃO FUNCIONAIS)** 6](#_Toc200112389)

[*5.1.* *FUNCIONAIS* 6](#_Toc200112390)

[*5.2.* *NÃO FUNCIONAIS* 6](#_Toc200112391)

[**8.** **FLUXOGRAMA** 7](#_Toc200112392)

[**9.** **DIAGRAMA CASO DE USO (UML)** 8](#_Toc200112393)

[**10.** **DIAGRAMA DE CLASSES** 10](#_Toc200112394)

[**11.** **DIAGRAMA DE SEQUÊNCIA** 11](#_Toc200112395)

[**12.** **DIAGRAMA DE ESTADO** 12](#_Toc200112396)

[**13.** **METODOLOGIAS ÁGEIS** 13](#_Toc200112397)

[**14.** **CONCLUSÃO** 14](#_Toc200112398)

# **FINALIDADE DA APLICAÇÃO**

O sistema foi desenvolvido para **monitorar e controlar a temperatura de ambientes internos**, garantindo conforto e segurança térmica.  
Ele atua como um **sistema inteligente de alerta e automação**, que identifica quando o local está muito quente ou frio e reage automaticamente com **sinais visuais, sonoros e comandos remotos.**

Além disso, o usuário pode **acompanhar os dados em tempo real por meio de um aplicativo ou página web**, tornando o controle mais acessível e moderno.

# **O QUE SERÁ MONITORADO/CONTROLADO?**

**Temperatura (°C)** e **umidade relativa (%)**, medidas continuamente pelo sensor **DHT11**.

**LED vermelho**: indica temperatura alta.

**LED verde**: indica temperatura baixa.

**Buzzer**: emite som de alerta quando a temperatura ultrapassa o limite configurado.

**Display LCD**: exibe em tempo real os valores de temperatura e umidade.

**Controle remoto**: o sistema permite que o usuário **acione ventiladores ou aquecedores remotamente**, através de comandos enviados pelo aplicativo ou página web conectada ao Arduino.

# **3. COMO O USUÁRIO INTERAGE COM O SISTEMA?**

O sistema é **automático**, mas também **interativo**:

1. No ambiente físico, o usuário acompanha as informações no **LCD**, observa os **LEDs** e ouve o **buzzer**.
2. No ambiente digital, o usuário pode acessar os dados de temperatura e umidade **via aplicativo ou página web**.
3. De forma remota, ele pode **enviar comandos** para **ligar ou desligar ventiladores e aquecedores**, ajustando o conforto térmico sem precisar estar no local.

Essa integração garante **monitoramento e controle total**, tanto local quanto remoto.

# **4. ESTRUTURA DO SISTEMA**

* **Arduino UNO**
* **Protoboard**
* **LCD 16x2**
* **DHT11**
* **Buzzer**
* **Potenciômetro**
* **LED Verde**
* **LED Vermelho**
* **Resistores**
* **Jumpers**

# **5. CÓDIGO PARA O ARDUINO**

#include "DHT.h"

#include <LiquidCrystal.h>

#define DHTPIN 2

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

const int ledPinVerde = 7;

const int ledPinVermelho   = 8;

const int buzzerPin = 9;

LiquidCrystal lcd(12, 11, 5, 4, 3, 6);

float limiteAlto  = 30.0;

float limiteBaixo = 18.0;

unsigned long lastRead = 0;

const unsigned long readInterval = 2000;

void setup() {

  Serial.begin(9600);

  dht.begin();

  pinMode(ledPinVerde, OUTPUT);

  pinMode(ledPinVermelho, OUTPUT);

  pinMode(buzzerPin, OUTPUT);

  lcd.begin(16, 2);

  lcd.print("Monitor Temp");

  delay(1500);

  lcd.clear();

}

void loop() {

  if (Serial.available() > 0) {

    char command = Serial.read();

    switch (command) {

      case 'A':

        digitalWrite(ledPinVerde, HIGH);

        break;

      case 'a':

        digitalWrite(ledPinVerde, LOW);

        break;

      case 'B':

        digitalWrite(ledPinVermelho, HIGH);

        break;

      case 'b':

        digitalWrite(ledPinVermelho, LOW);

        break;

      case 'T':

        sendTempHumidOnce();

        break;

      default:

        break;

    }

  }

  if (millis() - lastRead >= readInterval) {

    lastRead = millis();

    readAndHandleSensor();

  }

}

void readAndHandleSensor() {

  float t = dht.readTemperature();

  float h = dht.readHumidity();

  if (isnan(t) || isnan(h)) {

    lcd.clear();

    lcd.print("Erro sensor!");

    Serial.println("ERRO");

    digitalWrite(ledPinVerde, LOW);

    digitalWrite(ledPinVermelho, LOW);

    noTone(buzzerPin);

    return;

  }

  lcd.setCursor(0, 0);

  lcd.print("Temp:");

  lcd.print(t, 1);

  lcd.print("C   ");

  lcd.setCursor(0, 1);

  lcd.print("Umid:");

  lcd.print(h, 1);

  lcd.print("%    ");

  Serial.print(t, 2);

  Serial.print(";");

  Serial.println(h, 2);

  if (t > limiteAlto) {

    digitalWrite(ledPinVermelho, HIGH);

    digitalWrite(ledPinVerde, LOW);

    tone(buzzerPin, 2000);

  } else if (t < limiteBaixo) {

    digitalWrite(ledPinVerde, HIGH);

    digitalWrite(ledPinVermelho, LOW);

    noTone(buzzerPin);

  } else {

    digitalWrite(ledPinVerde, LOW);

    digitalWrite(ledPinVermelho, LOW);

    noTone(buzzerPin);

  }

}

void sendTempHumidOnce() {

  float t = dht.readTemperature();

  float h = dht.readHumidity();

  if (isnan(t) || isnan(h)) {

    Serial.println("ERRO");

    return;

  }

  Serial.print(t, 2);

  Serial.print(";");

  Serial.println(h, 2);

}

# **6. IMAGENS**

# Imagem de vídeo game O conteúdo gerado por IA pode estar incorreto.

# Uma imagem contendo relógio, medidor O conteúdo gerado por IA pode estar incorreto.Uma imagem contendo luz, computador O conteúdo gerado por IA pode estar incorreto.