

Monitoramento Ambiental e Controle Remoto em Salas de Aula Inteligentes

SENAI Roberto Mange

■ Objetivo Geral

Desenvolver um sistema de monitoramento ambiental com ESP32, capaz de registrar temperatura e umidade em tempo real e acionar um LED de forma automatizada ou remota, promovendo conforto térmico e eficiência energética.

■ Objetivos Específicos

- Monitorar temperatura e umidade em tempo real com sensor DHT11 ou DHT22.
- Publicar os dados em uma planilha do Google Sheets automaticamente.
- Visualizar os dados por meio de gráficos atualizados na nuvem.
- Acionar um LED em resposta a condições ambientais (ex: calor excessivo).
- Montar e testar o circuito com protoboard, resistores e jumpers.
- Aplicar os conceitos de Cidades Inteligentes no ambiente escolar.

■ Benefícios do Projeto

- Conforto térmico: detecta ambientes quentes e permite ações (abrir janelas, ligar ventiladores).
- Eficiência energética: evita uso excessivo de ar-condicionado ou ventiladores.
- Segurança: alertas visuais com LED para condições fora do ideal.
- Tomada de decisão baseada em dados (data-driven).
- Experiência prática com tecnologias reais de IoT.

■ Diagrama do Circuito

Este é o circuito básico com ESP32, sensor DHT11/DHT22 e um LED com resistor conectado à protoboard. O LED está ligado ao GPIO 2 com resistor de 220Ω , e o sensor DHT está ligado ao GPIO 4.

- VCC do DHT → 3.3V do ESP32
- GND do DHT → GND do ESP32
- DATA do DHT → GPIO 4 do ESP32
- Anodo do LED → GPIO 2 do ESP32 (com resistor de 220Ω em série)
- Catodo do LED → GND do ESP32

■ Código Arduino (ESP32)

```
#include #include #include #define DHTPIN 4 #define DHTTYPE DHT11 DHT dht(DHTPIN,
DHTTYPE); const char* ssid = "SUA_REDE_WIFI"; const char* password = "SUA_SENHA_WIFI";
String nome = "Ana Luisa"; const char* serverName =
"https://script.google.com/macros/s/SEU_LINK_AQUI/exec"; void setup() { Serial.begin(115200);
WiFi.begin(ssid, password); dht.begin(); pinMode(2, OUTPUT); // LED no GPIO 2 while
(WiFi.status() != WL_CONNECTED) { delay(1000); Serial.println("Conectando ao WiFi..."); }
Serial.println("WiFi conectado!"); } void loop() { float temperatura = dht.readTemperature(); float
umidade = dht.readHumidity(); if (!isnan(temperatura) && !isnan(umidade)) { if (temperatura > 30) {
digitalWrite(2, HIGH); } else { digitalWrite(2, LOW); } if (WiFi.status() == WL_CONNECTED) {
HTTPClient http; String url = String(serverName) + "?temperatura=" + temperatura + "&umidade="
+ umidade + "&nome=" + nome; http.begin(url); int httpResponseCode = http.GET(); http.end(); } }
delay(10000); }
```

■ Script Google Sheets (Apps Script)

```
function doGet(e) { var sheet =  
SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Dados"); if (!sheet) { sheet =  
SpreadsheetApp.getActiveSpreadsheet().insertSheet("Dados"); sheet.appendRow(["Data", "Hora",  
"Temperatura", "Umidade", "Nome"]); } var temperatura = e.parameter.temperatura; var umidade =  
e.parameter.umidade; var nome = e.parameter.nome; var data = Utilities.formatDate(new Date(),  
"GMT-3", "dd/MM/yyyy"); var hora = Utilities.formatDate(new Date(), "GMT-3", "HH:mm:ss");  
sheet.appendRow([data, hora, temperatura, umidade, nome]); return  
ContentService.createTextOutput("Dados recebidos com sucesso"); }
```