



# **TRABALHO FINAL MINI ESTAÇÃO METEOROLÓGICA**

**PROGRAMAÇÃO PROCEDIMENTAL**

# SUMÁRIO

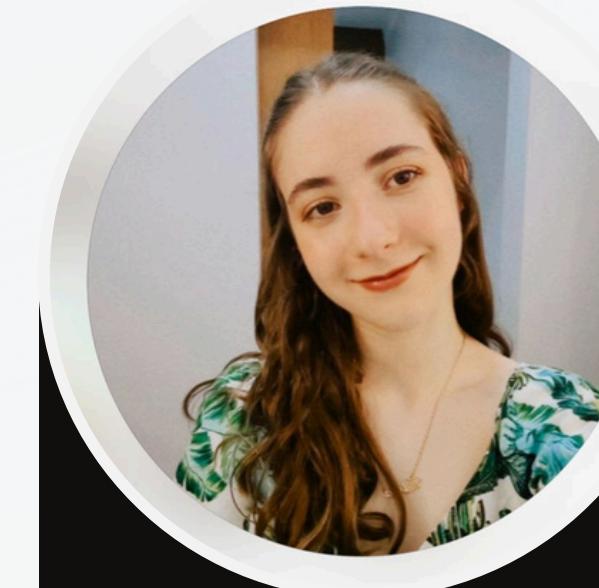
- 01** INTEGRANTES DO GRUPO
- 02** ITENS UTILIZADOS NO PROJETO
- 03** PRECISÃO DOS DADOS MEDIDOS  
PELO SENSOR DHT11
- 04** ESQUEMA DE LIGAÇÃO DA ESTAÇÃO
- 05** COMO FOI FEITA A MONTAGEM
- 06** CÓDIGO
- 07** MINI ESTAÇÃO METEOROLÓGICA  
FUNCIONANDO

# INTEGRANTES



Camila  
Alves A.  
Oliveira

12321EBIO21



Lívia Santelli  
Pegoraro

12321EBIO10



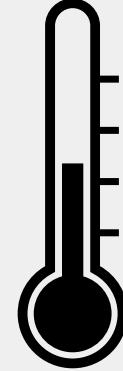
Maria Luisa  
Gonçalves  
Ferreira

12321EBIO13

# ITENS UTILIZADOS NO PROJETO

- NodeMCU ESP32 Iot com WiFi e Bluetooth
  - 38 pinos
- Sensor de temperatura, pressão e umidade DHT11
- 7 jumper para protoboard macho-macho 10cm
- Módulo display Oled 0.96 I2C Ssd1306 Lcd
- 2 protoboard 400 pontos para montagem do projeto
- Resistor de 10000ohms

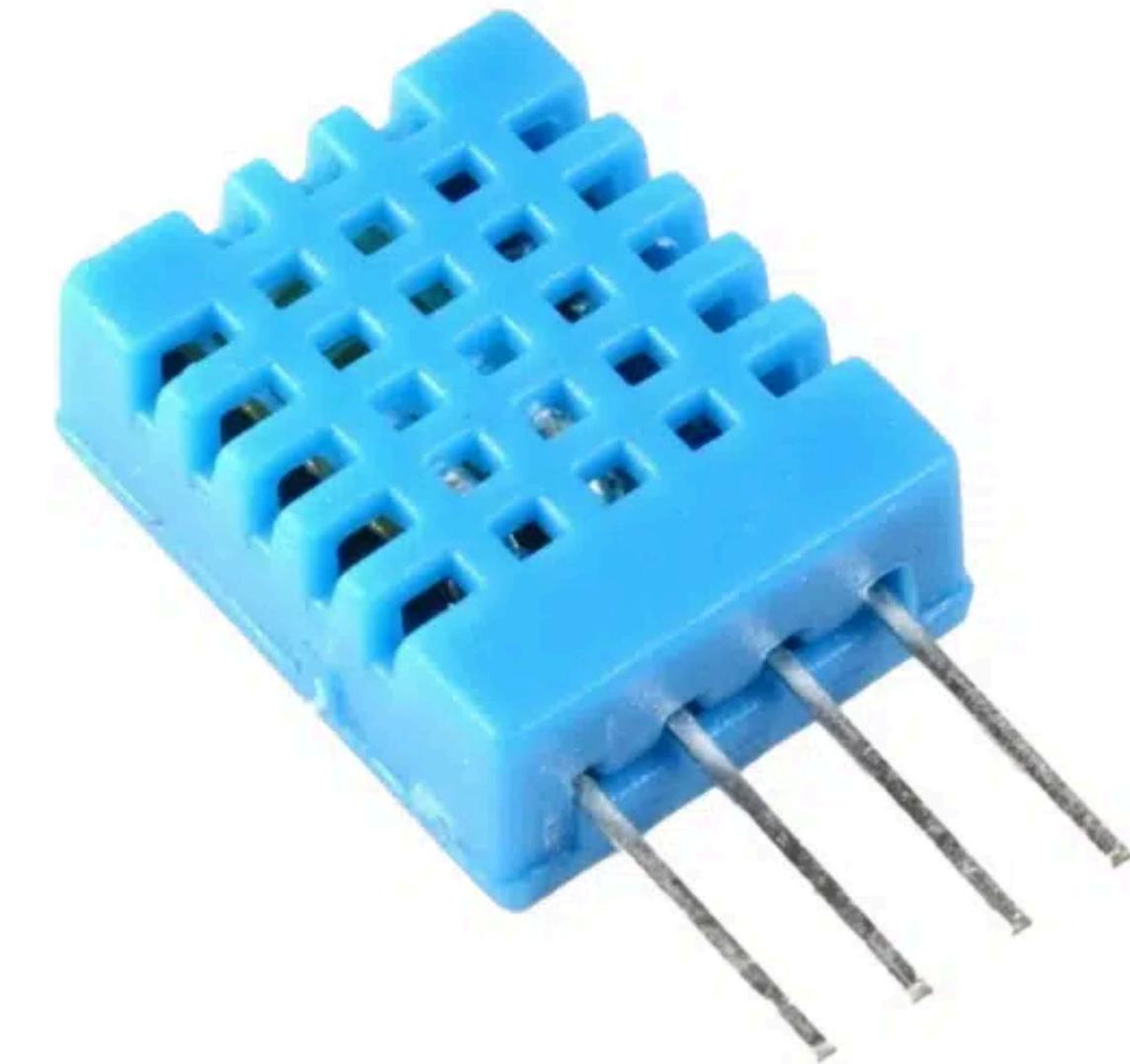
# PRECISÃO DOS DADOS MEDIDOS PELO SENSOR DHT11



- Faixa de medição de temperatura: 0° a 50°C
- Precisão de medição de temperatura:  $\pm 2,0^{\circ}\text{C}$



- Faixa de medição de umidade: 20 a 90% UR
- Precisão de umidade de medição:  $\pm 5,0\%$  UR

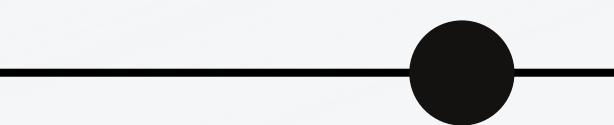


# Montagem



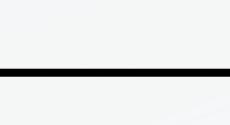
**ESP E SENSOR**

Liga-se o 3.3 da  
ESP no VCC do  
sensor. Faz-se a  
ligação dos GND  
da ESP na do  
sensor



**RESISTOR E OUTROS**

Liga-se o D16 da  
ESP em uma das  
extremidades do  
resistor e a outra  
extremidade  
liga-se ao data  
do sensor



**ESP E O DISPLAY**

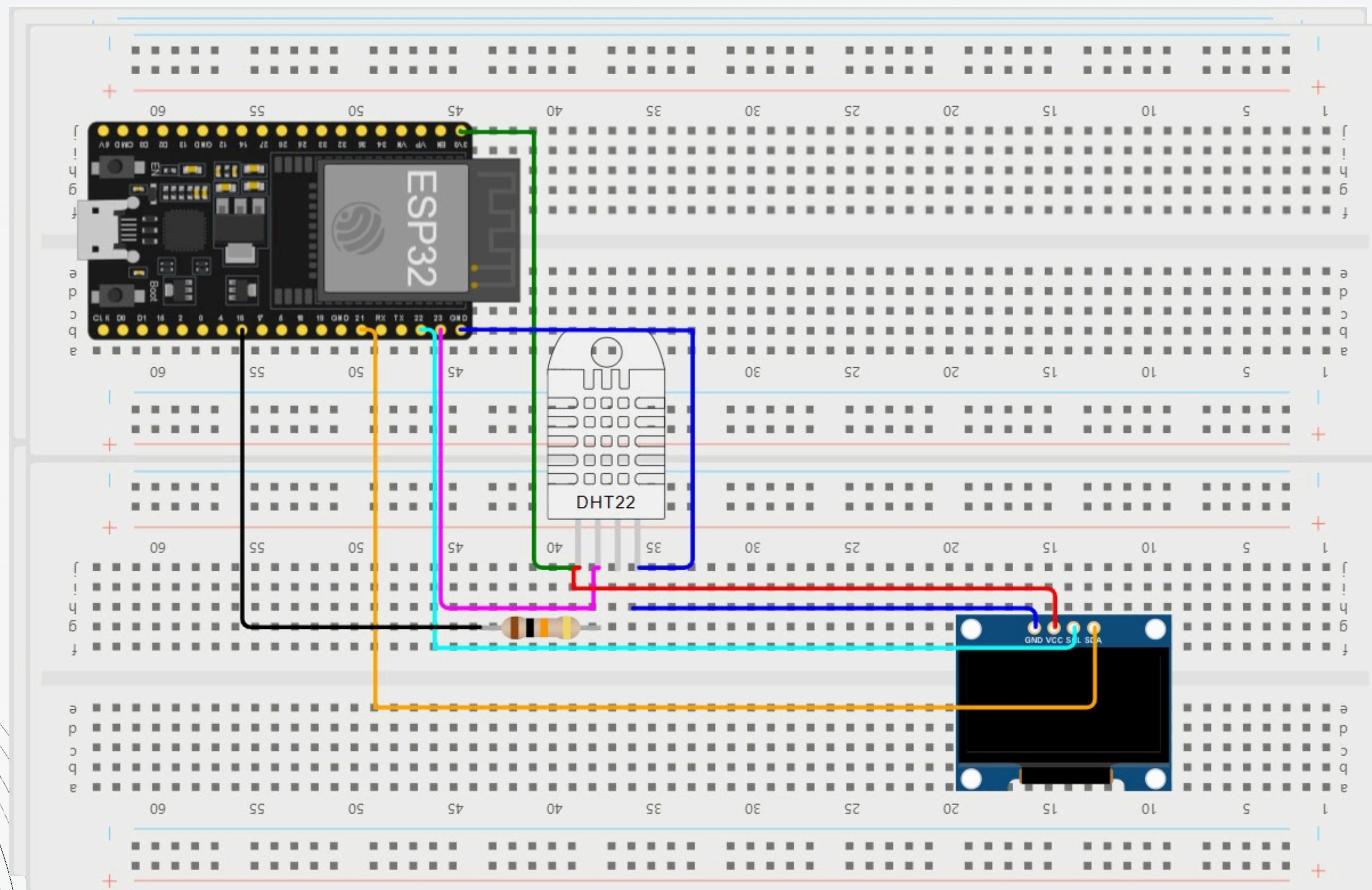
Conecta-se o  
D21 da ESP com  
o SDA do display  
e o D22 da ESP  
no SCL do  
display



**DISPLAY**

Liga-se o VCC  
do display ao do  
sensor. E  
conecta-se o  
GND do sensor e  
do display

# ESQUEMA DE LIGAÇÃO DA ESTAÇÃO



# CÓDIGO

ESP\_CODE.ino

```
1 #include <Wire.h> // display
2 #include <Adafruit_GFX.h> // display
3 #include <Adafruit_SSD1306.h> // display
4 #include "DHT.h" // sensor
5 #include <WiFi.h>
6 #include <ESPAsyncWebServer.h>
7
8 const char* ssid = "iPhone de Malu";
9 const char* password = "male1306";
10
11 #define SCREEN_WIDTH 128
12 #define SCREEN_HEIGHT 64
13 #define DHTPIN 16
14 #define DHTTYPE DHT11
15 #define SCREEN_ADDRESS 0x3C // endereço do display
16
17 const unsigned char termometro[] PROGMEM = {
18     0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x07, 0xf0, 0x00, 0x00, 0x00, 0x0f, 0xf0,
19     0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00, 0x00, 0x1e, 0x78, 0x00, 0x00, 0x00, 0x00,
20     0x3c, 0x3c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00,
21     0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x1c,
22     0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x00,
23     0x38, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00,
24     0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x1c,
25     0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x38, 0x1c,
26     0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x38, 0x1c, 0x00,
27     0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x00, 0x39, 0x9c,
28     0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x00, 0x39, 0x9c,
29     0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00,
30     0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x00, 0x39, 0x9c,
31     0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x00, 0x39, 0x9c,
32     0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00,
33     0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x00, 0x39, 0x9c,
```

...

# CÓDIGO

```
ESP_CODE.ino ...  
17 const unsigned char termometro[] PROGMEM = {  
18     0x00, 0x00, 0x03, 0xc0, 0x00, 0x00, 0x00, 0x07, 0xf0, 0x00, 0x00, 0x00, 0x0f, 0xf0,  
19     0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00, 0x00, 0x1e, 0x78, 0x00, 0x00, 0x00, 0x00,  
20     0x3c, 0x3c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00,  
21     0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c,  
22     0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c,  
23     0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00,  
24     0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c,  
25     0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c,  
26     0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x1c, 0x00, 0x00,  
27     0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c,  
28     0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x00,  
29     0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00,  
30     0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c,  
31     0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x00,  
32     0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00,  
33     0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x39, 0x9c,  
34     0x00, 0x00, 0x00, 0x39, 0x9c, 0x00, 0x00, 0x00, 0x79, 0x9e, 0x00, 0x00, 0x00, 0x00,  
35     0xf1, 0x8f, 0x00, 0x00, 0x00, 0xf3, 0xcf, 0x00, 0x00, 0x00, 0x00, 0xe7, 0xe7, 0x00, 0x00,  
36     0x00, 0x01, 0xef, 0xf7, 0x80, 0x00, 0x00, 0x01, 0xcf, 0xf3, 0x80, 0x00, 0x00, 0x01, 0xcf, 0xf3,  
37     0x80, 0x00, 0x00, 0x01, 0xcf, 0xf3, 0x80, 0x00, 0x01, 0xcf, 0xf3, 0x80, 0x00, 0x00, 0x01,  
38     0xcf, 0xf3, 0x80, 0x00, 0x00, 0x01, 0xcf, 0xf3, 0x80, 0x00, 0x00, 0x01, 0xef, 0xf7, 0x80, 0x00,  
39     0x00, 0x00, 0xe7, 0xe7, 0x00, 0x00, 0x00, 0xf3, 0xcf, 0x00, 0x00, 0x00, 0x00, 0xf0, 0x0f,  
40     0x00, 0x00, 0x00, 0x7c, 0x3e, 0x00, 0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00, 0x00, 0x00,  
41     0x1f, 0xf8, 0x00, 0x00, 0x0f, 0xf0, 0x00, 0x00, 0x00, 0x03, 0xe0, 0x00, 0x00, 0x00  
42 };  
43  
44 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT); // objeto do display  
45 DHT dht(DHTPIN, DHTTYPE); // objeto do sensor  
46  
47 AsyncWebServer server(80);  
48  
49 float temperature = 0.0;
```

# CÓDIGO

ESP\_CODE.ino

```
43
44 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT); // objeto do display
45 DHT dht(DHTPIN, DHTTYPE); // objeto do sensor
46
47 AsyncWebServer server(80);
48
49 float temperature = 0.0;
50 float humidity = 0.0;
51
52 int period = 10000;
53 unsigned long lastTime = 0;
54
55 void setup() {
56     Serial.begin(115200);
57
58     WiFi.begin(ssid, password);
59
60     while (WiFi.status() != WL_CONNECTED) {
61         delay(500);
62         Serial.println("Conectando ao WiFi...");
63     }
64
65     Serial.print("Conectado em: ");
66     Serial.println(WiFi.localIP());
67
68     if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
69         Serial.println("Erro ao inicializar display");
70     }
71
72     server.on("/", HTTP_GET, [] (AsyncWebRequest* request) {
73         String html = "<meta charset=UTF-8><meta content=25 http-equiv=refresh><meta content=\"width=device-width,initial-scale=1\"name=viewport><style>body<request->send(200, \"text/html\", html);
```

# CÓDIGO

ESP\_CODE.ino

```
64
65
66
67
68 <v>
69
70
71
72 <v>
73 <style><body{margin:0;padding:0;background-color:#121212;color:#fff;height:calc(var(--vh,1vh) * 100);display:flex;justify-content:center;align-items:center;}></style>
74
75
76
77
78
79
80
81
82 <v>
83
84
85
86 <v>
87
88
89
90
91
92
93
94
95
96
97
```

# CÓDIGO

ESP\_CODE.ino

```
b4
65
66
67
68
69
70
71
72
73 n-items:center;font-size:1.8rem}.c{width:75%;max-width:400px;height:60%;max-height:500px;border:1px solid red;border-radius:8px;background-color:#202020;
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
```

# CÓDIGO

ESP\_CODE.ino

```
64  
65  
66  
67  
68  
69  
70  
71  
72  
73 .documentElement.style.setProperty("--vh\","` ${v}px` );</script><script crossorigin=anonymous src=https://kit.fontawesome.com/85de8204c5.js></script>";  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96
```

# CÓDIGO

ESP\_CODE.ino

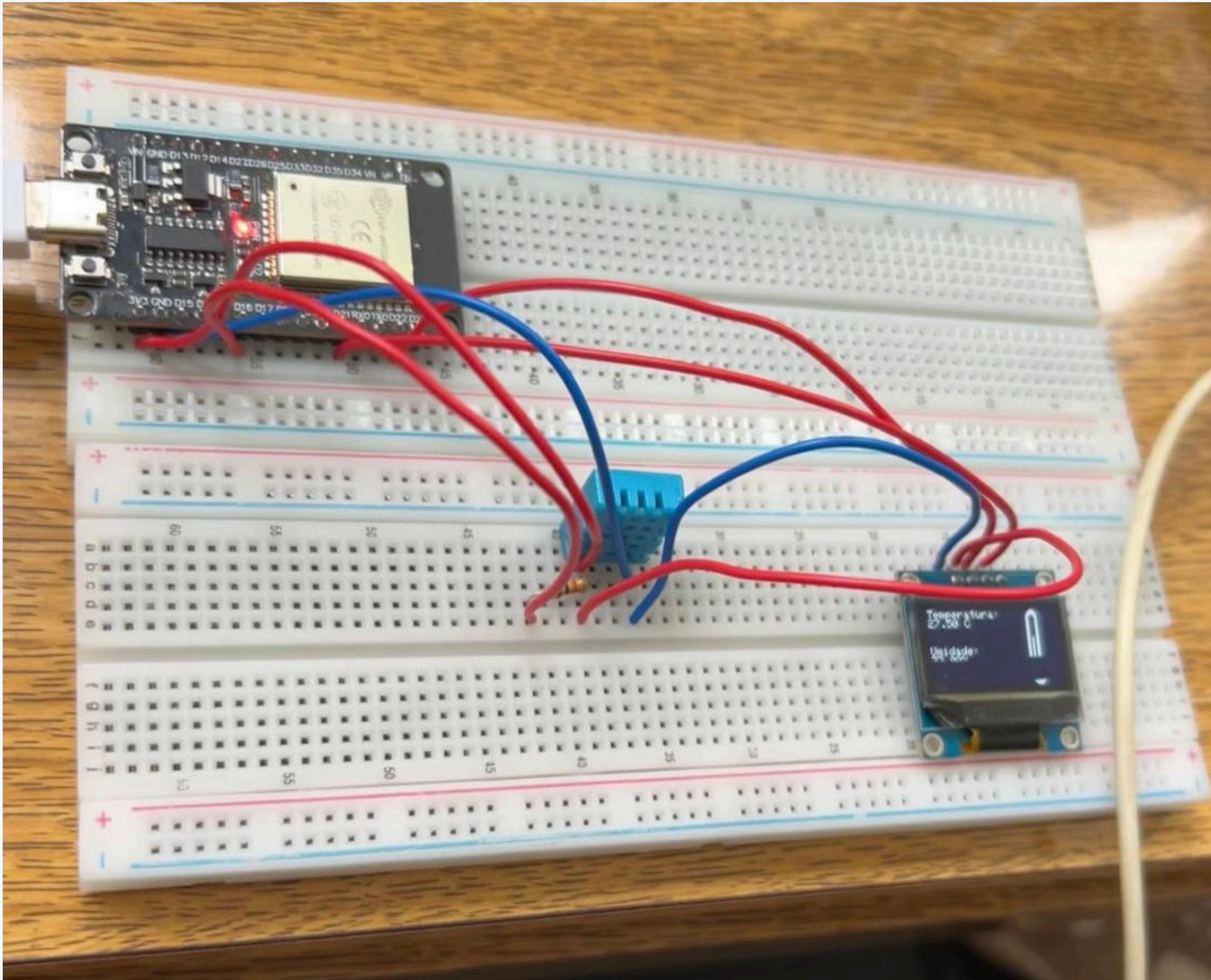
```
72     server.on("/", HTTP_GET, [](AsyncWebRequest* request) {
73         String html = "<meta charset=UTF-8><meta content=25 http-equiv=refresh><meta content=\"width=device-width,initial-scale=1\"name=viewport><style>body{background-color: black; color: white; font-size: 14px; margin: 0; padding: 0;}</style><div style='display: flex; align-items: center; justify-content: center; height: 100vh;'><div style='text-align: center;'><h1>Temperatura e Umidade</h1><h2>Termômetro</h2><img alt='Termômetro' src='termometro.png' style='width: 100px; height: auto;'><table border='1' style='margin-top: 20px; border-collapse: collapse; width: 100%; text-align: center;'><thead><tr><th>Medição</th><th>Valor</th></tr></thead><tbody><tr><td>Temperatura</td><td>" + String(temperature) + "</td></tr><tr><td>Umidade</td><td>" + String(humidity) + "%</td></tr></tbody></table></div></div>";
74         request->send(200, "text/html", html);
75     });
76
77     server.begin();
78
79     dht.begin();
80 }
81
82 void loop() {
83     humidity = dht.readHumidity();
84     temperature = dht.readTemperature();
85
86     if (isnan(humidity) || isnan(temperature))
87         return;
88
89     display.clearDisplay();
90     display.setTextSize(1);
91     display.setTextColor(WHITE);
92     display.setCursor(0, 0);
93
94     display.println("Temperatura:");
95     display.print(temperature);
96     display.println(" C");
97
98     display.println('\n');
99
100    display.println("Umidade:");
101    display.print(humidity);
102    display.println("%");
103
104    display.drawBitmap(80, 0, termometro, 48, 64, 1);
```

...

# CÓDIGO

```
ESP_CODE.ino ...  
88     display.setTextColor(WHITE);  
89  
90     display.setCursor(0, 0);  
91  
92     display.println("Temperatura:");  
93     display.print(temperature);  
94     display.println(" C");  
95  
96     display.println('\n');  
97  
98     display.println("Umidade:");  
99     display.print(humidity);  
100    display.println("%");  
101  
102    display.drawBitmap(80, 0, termometro, 48, 64, 1);  
103  
104  
105    display.display(); // função que passa tudo que foi feito para o display  
106  
107  
108  
109    if (millis() - lastTime > period) {  
110        Serial.print("Temperatura:");  
111        Serial.print(temperature);  
112        Serial.println(" C");  
113  
114        Serial.print("Umidade:");  
115        Serial.print(humidity);  
116        Serial.println("%");  
117  
118        Serial.println();  
119  
120        lastTime = millis();  
121    }  
122 }
```

# MINI ESTAÇÃO METEOROLÓGICA



# OBRIGADA PELA ATENÇÃO

*Camila Alves Arantes Oliveira - 12321EBIO21*  
*camila.arantes@ufu.br*

*Lívia Santelli Pegoraro - 12321EBIO10*  
*livia.pegoraro@ufu.br*

*Maria Luisa Gonçaves Ferreira - 12321EBIO13*  
*maria.ferreira6@ufu.br*



# REFERÊNCIAS BIBLIOGRÁFICAS

HERO, Maker. Sensor de Umidade e Temperatura DHT11. 2024. Disponível em:

- <https://www.makerhero.com/produto/sensor-de-umidade-e-temperatura-dht11/#:~:text=Precis%C3%A3o%20nas%20medi%C3%A7%C3%B5es%20de%20umidade,C%20na%20medi%C3%A7%C3%A3o%20da%20temperatura..> Acesso em: 20 out. 2024.

WOKWI projects. Disponível em:

- <https://wokwi.com/projects/412455066343429121>. Acesso em: 20 out. 2024.

STRAUB, Matheus Gebert. Estação Meteorológica Simples com BME280 e ESP32 com IP Fixo. Disponível em: <https://www.usinainfo.com.br/blog/estacao-meteorologica-simples-com-bme280-e-esp32-com-ip-fixo/>. Acesso em: 20 out. 2024.