LIGA ACADÊMICA DE SISTEMAS EMBARCADOS
CENTRO DE INFORMÁTICA · UFPE

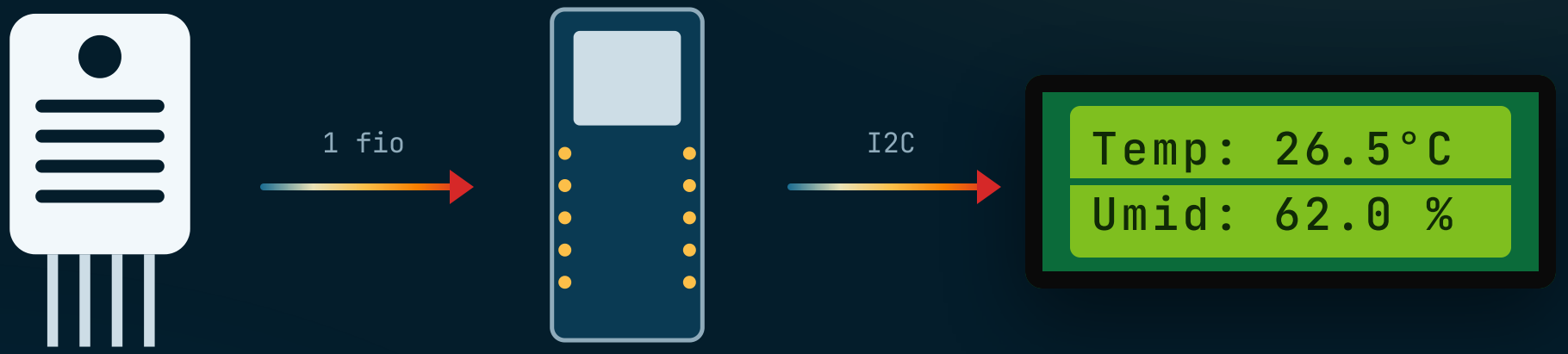
FORMAÇÃO EM ESP32 · MÓDULO 2

Estação de **temperatura e umidade**

DHT22 + LCD 16×2 + ESP32-S3, simulados no Wokwi.

LUCAS RAYAN GUERRA
CIÊNCIA EMBARCADA · ESPDOCS
LINKEDIN.COM/IN/LUCASRGUERRA

Medir, processar, mostrar

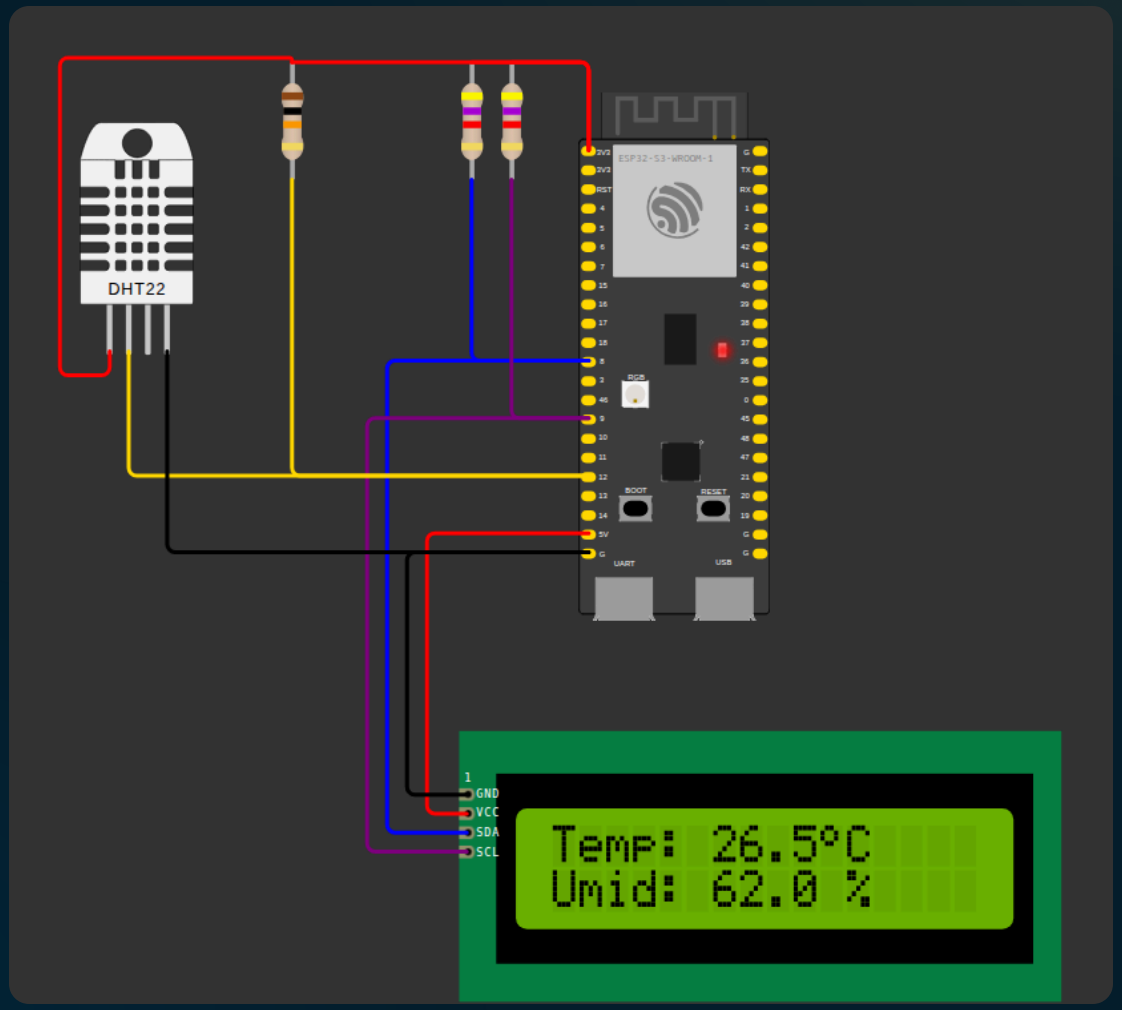


DHT22
MEDE

ESP32-S3
PROCESSA

LCD 16x2
MOSTRA

Isto, rodando no seu navegador



PARTE UM

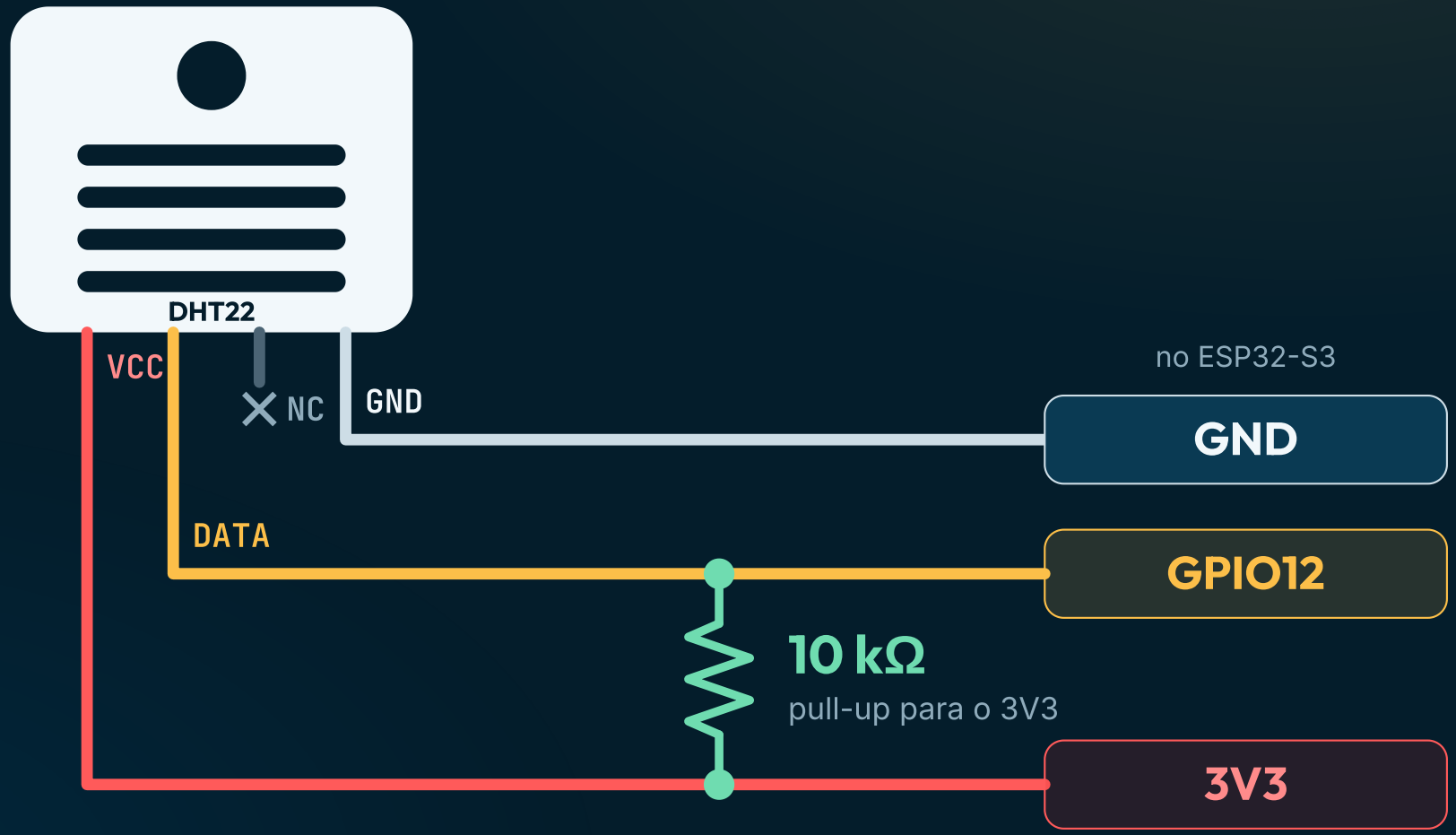
O sensor DHT22

Dois sensores numa caixinha, um fio de dados.

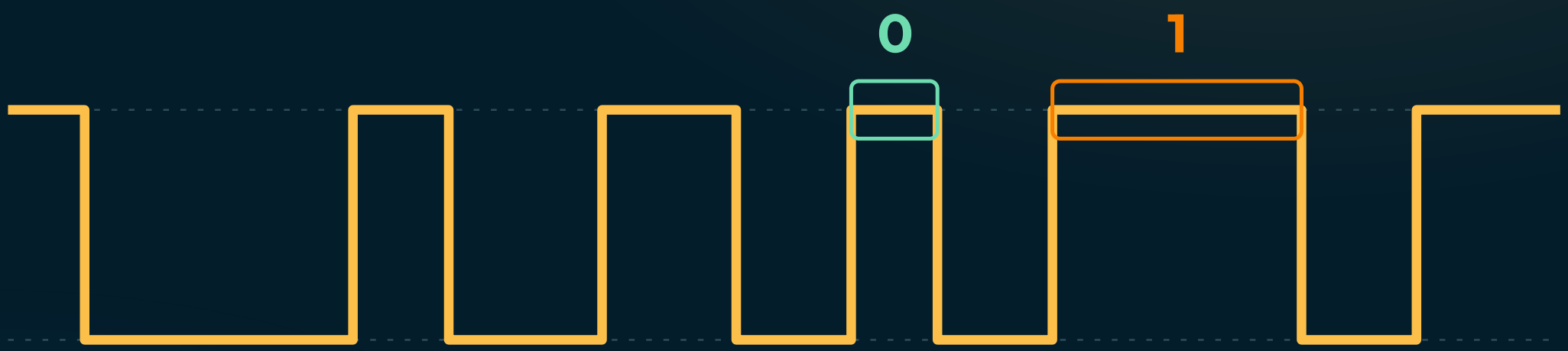
Por que o DHT22, e não o DHT11 do módulo 1

	DHT11	DHT22
PRECISÃO	±2 °C	**±0,5 °C**
FAIXA	0 a 50 °C	**–40 a 80 °C**
RESOLUÇÃO	1 °C	**0,1 °C**
LEITURAS	1 por segundo	1 a cada 2 s

Quatro pinos, um sem uso



Uma medição = 40 bits num fio só



16 bits de umidade · 16 de temperatura · 8 de verificação

Ninguém empurra a linha para cima. **Quem puxa é o resistor.**

Sem o pull-up de 10 kΩ, a linha sobe devagar demais, os tempos saem errados e a leitura vira "Falha no sensor".

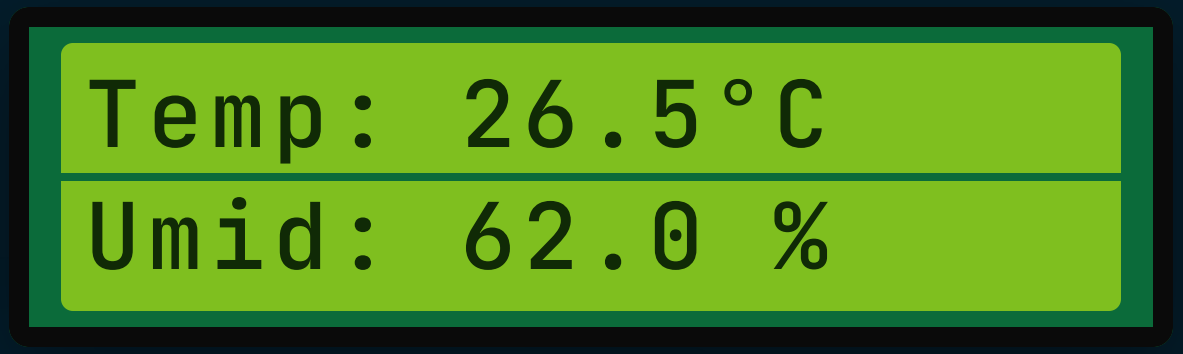
PARTE DOIS

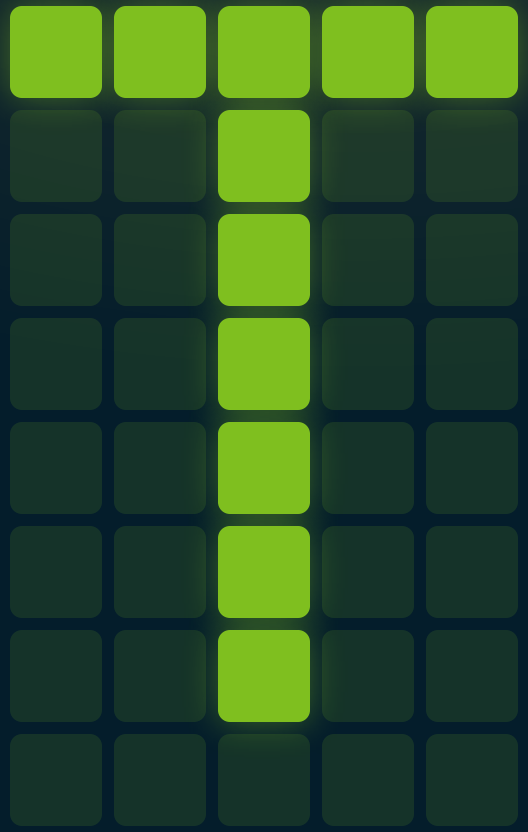
O display e o I2C

32 caracteres, 4 fios, 1 endereço.



16 colunas × 2 linhas × (5 × 8 pontos)





Cada caractere é uma grade de 5 × 8. E você pode desenhar até 8 caracteres seus.

A plaquinha atrás do display

16

fios sem a plaquinha

4

fios com o PCF8574

VCC · GND · SDA · SCL

O trimpot azul da plaquinha ajusta o contraste. Tela vazia? Gire ele primeiro.

Dois fios, vários dispositivos

"0x27, é com você."

só responde quem tem o endereço chamado



O I2C também precisa de pull-up

DHT22	SDA	SCL
10 kΩ	4,7 kΩ	4,7 kΩ

Os três resistores vão para o **3V3**. Nunca para o 5V: os pinos do ESP32-S3 não toleram 5 V.

PARTE TRÊS

O simulador Wokwi

Um laboratório de eletrônica no navegador. Sem instalar nada, sem conta.

Um projeto = três arquivos

`sketch.ino`

o programa

`diagram.json`

peças e fios

`libraries.txt`

bibliotecas

Os três estão em `formacoes/2/wokwi/`, prontos para colar.

Quatro passos

1

Abra

`wokwi.com/projects/new/esp32-s3`

2

Cole

`sketch.ino` + `diagram.json`

3

Adicione

`DHT sensor library`
`LiquidCrystal I2C`



Simule

`botão verde`

PARTE QUATRO

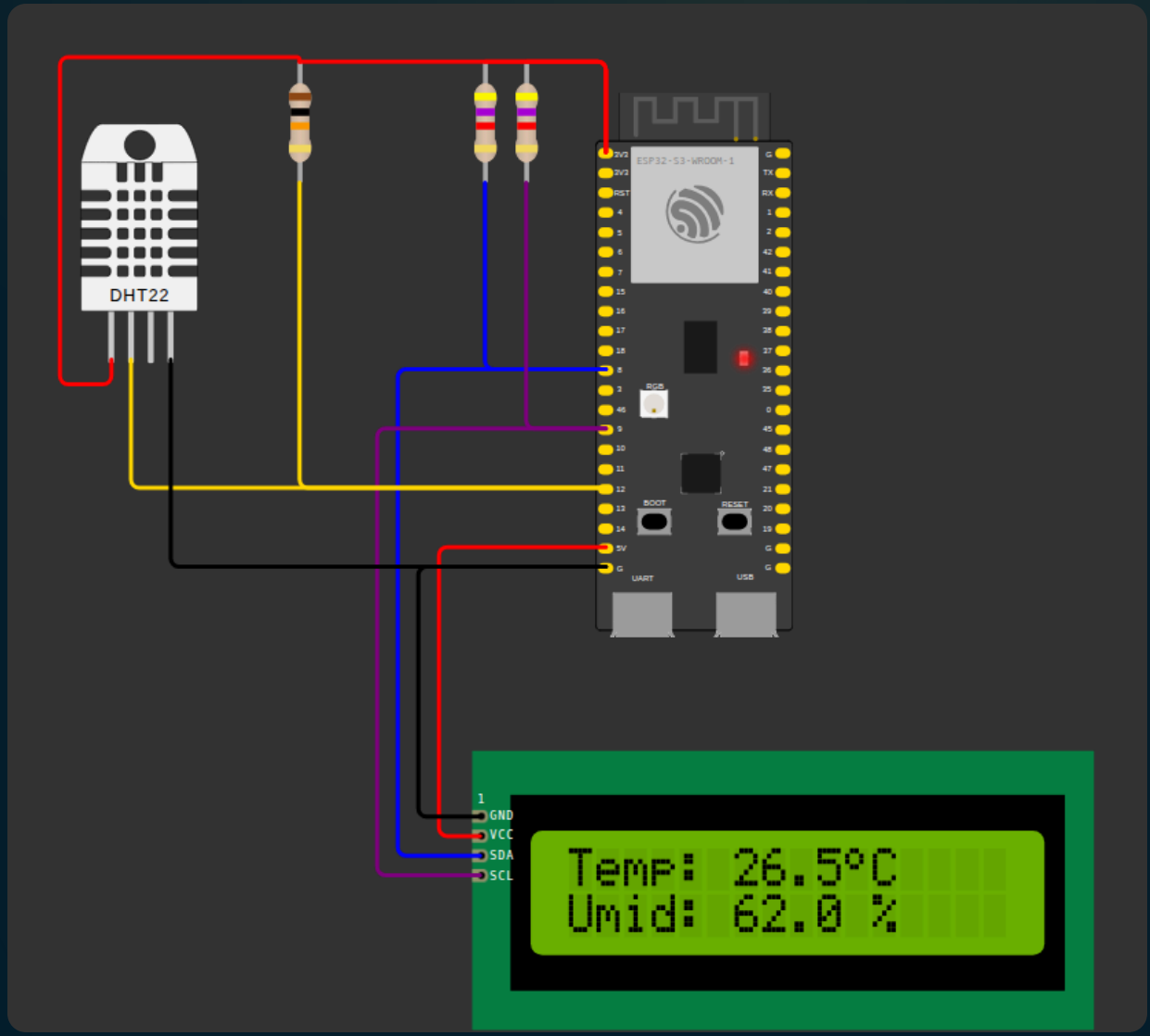
Montagem

Tudo na coluna esquerda da placa.



O circuito

3V3	DHT22 e os 3 resistores
GPIO12	dados do DHT22
GPIO8	SDA do LCD
GPIO9	SCL do LCD
5V	VCC do LCD
GND	sensor e LCD



PARTE CINCO

O programa

Configurar, iniciar, repetir.



Configuração

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

#define PINO_DHT  12        // fio amarelo
#define TIPO_DHT  DHT22
#define PINO_SDA  8         // I2C padrão do ESP32-S3
#define PINO_SCL  9

DHT dht(PINO_DHT, TIPO_DHT);
LiquidCrystal_I2C lcd(0x27, 16, 2);   // endereço, colunas, linhas
```

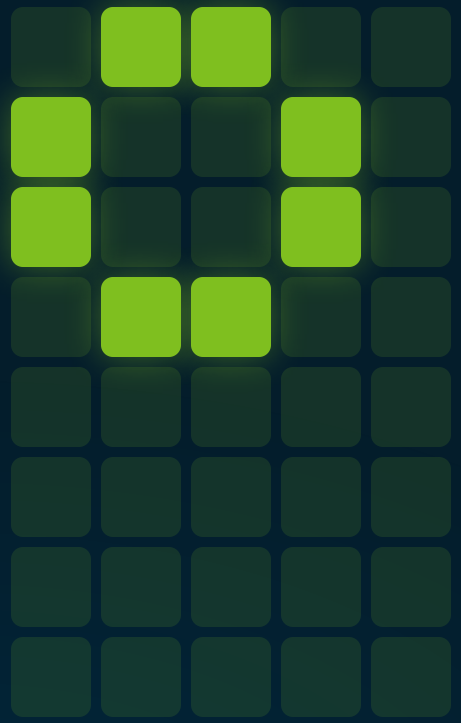
Mudou um fio?

Mude uma linha. Tudo que depende da montagem está no topo.

0x27

O endereço I2C do display.

Desenhando o símbolo de grau



```
0b01100
0b10010
0b10010
0b01100
0
0
0
0
```

```
byte grau[8] = { ... };
lcd.createChar(0, grau);
lcd.write(0);
```

Bit **1** = ponto aceso.

setup(): uma vez, ao ligar

```ino
void setup() {
  Serial.begin(115200);
  Wire.begin(PINO_SDA, PINO_SCL);
  dht.begin();

  lcd.init();
  lcd.backlight();
  lcd.createChar(0, grau);
  lcd.setCursor(0, 0);
  lcd.print("LASER  ESP32-S3");
  lcd.setCursor(0, 1);
  lcd.print("Iniciando...");
  delay(2000);            // o DHT22 precisa de ~2 s para a primeira leitura
  lcd.clear();
}
```

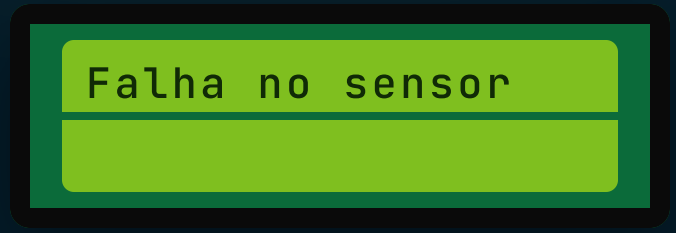


A mensagem cobre os **2 s** que o DHT22 leva para a primeira leitura.

loop(): ler, e desconfiar da leitura

```
void loop() {
  float umidade     = dht.readHumidity();
  float temperatura = dht.readTemperature();

  if (isnan(umidade) || isnan(temperatura)) {
    lcd.setCursor(0, 0);
    lcd.print("Falha no sensor ");
    Serial.println("Falha na leitura do DHT22");
  } else {
```



isnan(): a leitura falhou. Mostre o erro, não um número inventado.

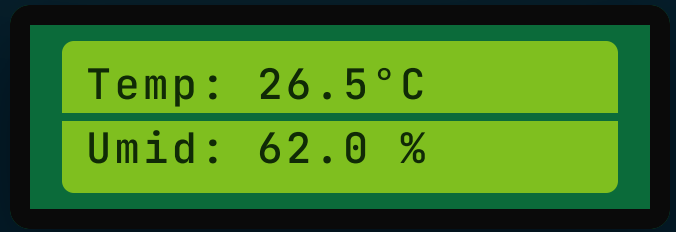
loop(): mostrar, e esperar 2 segundos

```
lcd.setCursor(0, 0);
lcd.print("Temp: ");
lcd.print(temperatura, 1);
lcd.write(0);            // o caractere de grau criado acima
lcd.print("C   ");

lcd.setCursor(0, 1);
lcd.print("Umid: ");
lcd.print(umidade, 1);
lcd.print(" %   ");

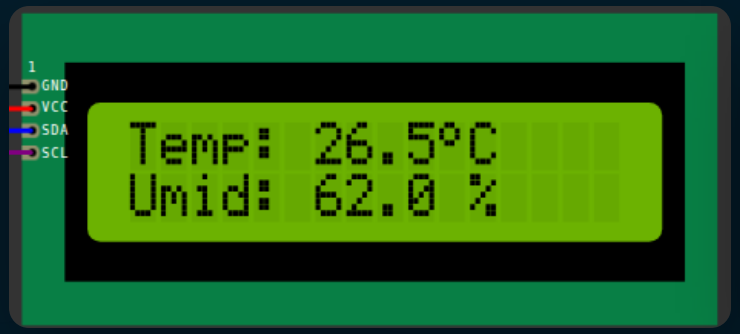
// ... envia o mesmo texto pelo Serial ...

delay(2000);   // 1 leitura a cada 2 s
```

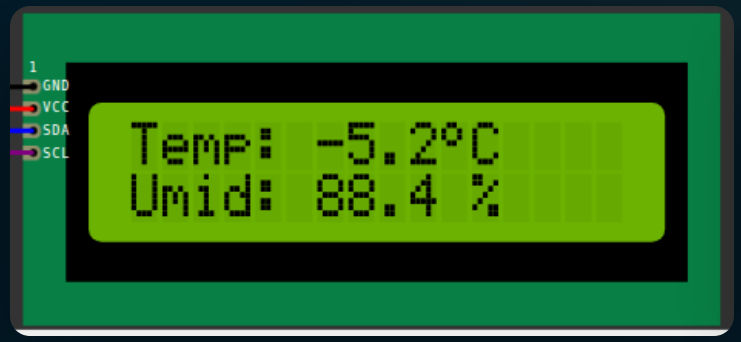


Os espaços em **"C "** apagam as sobras da leitura anterior.

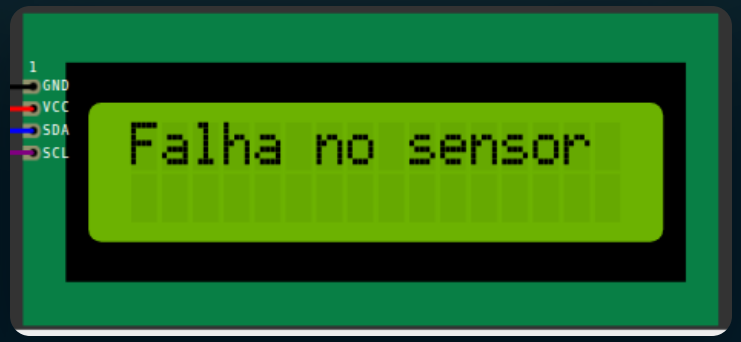
Três testes, três telas



Sala
26,5 °C · 62 %



Frio
−5,2 °C · 88,4 %



Fio solto
erro detectado

Na simulação, clique no DHT22 e mova os controles de temperatura e umidade.

PARTE SEIS

Da tela para a bancada

O que o Wokwi perdoa e o hardware não.

A plaquinha I2C traz pull-ups para 5 V

≈ 4 V

no GPIO, com a plaquinha em 5 V,
mesmo com os 4,7 kΩ para 3V3

✗ fora da especificação

3,3 V

Conversor de nível BSS138
ou remover os pull-ups da plaquinha

✓ seguro

Teste: multímetro entre SDA e VCC da plaquinha. Entre 2 e 10 kΩ, há um pull-up.

Tela vazia? Descubra o endereço

```
for (byte endereco = 1; endereco < 127; endereco++) {
  Wire.beginTransmission(endereco);
  if (Wire.endTransmission() == 0) {
    Serial.printf("Dispositivo em 0x%02X\n", endereco);
    encontrados++;
  }
}
```

```
Procurando dispositivos I2C...
Dispositivo em 0x27
```

Saída real no Wokwi.
Na sua placa pode aparecer **0x3F**.

Quando não funciona



"Falha no sensor"

Pull-up de 10 kΩ, fio no GPIO12, VCC e GND.



Aceso e vazio

Contraste, endereço, ou SDA e SCL trocados.



Linha de blocos

A energia chegou, o I2C não.



Tudo apagado

Falta o 5V ou o `lcd.backlight()`.

Desafios

Mín e máx

Guarde a menor e a maior temperatura desde que a placa ligou.

Alerta

Acenda o LED RGB do GPIO38 acima de 30 °C.

Sensação

Mostre o índice de calor com `computeHeatIndex()`.

Sem delay

Reescreva o `loop()` usando `millis()`.

MÓDULO 2 · ENCERRAMENTO

Obrigado.
Perguntas?

Apostila, código e arquivos do Wokwi em
espdocs.cienciaembarcada.com.br/formacoes/2.

LUCAS RAYAN GUERRA
CONTATO@LUCASRGUERRA.DEV.BR
LINKEDIN.COM/IN/LUCASRGUERRA
ESPDOCS.CIENCIAEMBARCADA.COM.BR