Cábula VisuMicro · Java no Maker Pi Pico

Programa · Saídas · Entradas · Fluxo · Objetos · Rede · Ajuda

Holger Sebastião · visucode.de
Versão 1.0 · 2026-11-12

Como trabalhar — VisuMicro

1. Abre `visumicro.visucode.de/?lang=pt` (navegador Chromium: Helium, Chrome, Edge).

2. Escreve o programa como texto ou com **Blocos** (`Ctrl`+`Shift`+`B`).

3. Iniciar simulação: a placa virtual mostra LEDs, sons e sensores. Os valores dos sensores ajustas lá com o rato.

4. Ligar a placa: liga um Pico real por USB. Na primeira vez, **Instalar o firmware**.

5. Transferir: o programa fica no Pico e funciona sem computador.

> O que escreves com `System.out.println` aparece na **consola** por baixo do editor.

O Maker Pi Pico — Cytron, RP2040

`LED`	GP25	LED verde do Pico
`BOTAO1-3`	GP20–22	três botões na placa
`BUZZER`	GP18	piezo, ao lado o interruptor de silêncio
`RGB_LED`	GP28	LED de cores (NeoPixel)
`0-15`	GP0–15	LED de estado azul em cada pino

ENTRADAS GROVE

`GROVE1-5`	GP1, 3, 5, 7, 9	fio amarelo = sinal
`GROVE6`	**GP27 (A1)**	**única entrada para sensores analógicos**

Todos os nomes estão na classe `MakerPiPico`: `new Led(MakerPiPico.LED)`.

Sensores analógicos — só na GROVE6

`getValor()`	0 a 1023
`getPercentagem()`	0 a 100

```
SensorLuz luz = new SensorLuz(MakerPiPico.GROVE6);
luz.estaEscuro()    // abaixo de 30 %
SensorSom s = new SensorSom(MakerPiPico.GROVE6);
s.haBarulho()       // acima de 50 %
BotaoRotativo rot =
    new BotaoRotativo(MakerPiPico.GROVE6);
rot.getAngulo()     // 0 a 300 graus
```

Estrutura do programa

```
public class Pisca {
  // atributos: os componentes
  private Led led = new Led(MakerPiPico.LED);

  public static void main(String[] args) {
    Pisca b = new Pisca();
    b.iniciar();
  }

  public void iniciar() {
    while (true) {      // repete sempre
      led.alternar();
      delay(500);       // esperar 500 ms
    }
  }
}
```

Cada componente é um **objeto**: `new Classe(pino)`. Depois chamas os seus **métodos**: `led.ligar();`

LED — Led, LedRegulavel

```
Led led = new Led(MakerPiPico.LED);
led.ligar();          led.desligar();
led.alternar();       // ligado ↔ desligado
led.piscar(200);  // 200 ms ligado, 200 ms desligado
if (led.estaLigado()) { … }

LedRegulavel d = new LedRegulavel(2);
d.setLuminosidade(128); // 0 a 255
```

Buzzer e sons — Buzzer, Som

```
Buzzer b = new Buzzer(MakerPiPico.BUZZER);
b.apitar();               // apito curto
b.apitar(440, 300);       // 440 Hz, 300 ms
b.ligarSom(Som.frequencia("C5"));  // até desligarSom()
b.desligarSom();
```

`Som.frequencia("A4")` → 440 (lá). Nomes de `C3` a `C8`, com `#` ou `b`: `"F#4"`, `"Bb3"`. C = dó, D = ré, E = mi …

> ! Sem som? Põe o **interruptor de silêncio** ao lado do buzzer em som.

Botões — Botao

```
Botao t = new Botao(MakerPiPico.BOTAO1);
t.estaPremido() // true enquanto premido
t.foiPremido()  // true uma vez por cada toque
```

`estaPremido` para "enquanto", `foiPremido` para "uma vez por toque" (sem ressaltos).

Padrão: reagir a um botão

```
while (true) {
  if (botao.foiPremido()) {
    led.alternar();
  }
  delay(10);   // o Pico continua a responder
}
```

> ! Em cada ciclo infinito, um `delay(10)` curto; senão o Pico responde mal.

Tempo — delay, millis

`delay(500);`	esperar 500 ms (1000 ms = 1 s)
`millis()`	ms desde o início, como `long`

DUAS COISAS AO MESMO TEMPO

```
long ultimo = millis();
while (true) {
  if (millis() - ultimo >= 500) {
    led.alternar();          // pisca …
    ultimo = millis();
  }
  if (botao.foiPremido()) {
    buzzer.apitar();         // … e está atento
  }
  delay(10);
}
```

LED RGB — LedRgb

```
LedRgb rgb = new LedRgb(MakerPiPico.RGB_LED);
// vermelho, verde, azul: 0-255
rgb.setCor(255, 120, 0);
rgb.desligar();
```

Cábula VisuMicro · Java no Maker Pi Pico

Saídas · Entradas · Objetos · Rede · Ajuda

Holger Sebastião · visucode.de
Versão 1.0 · 2026-11-12

Mover e ligar — Servo, Ventoinha, Rele

```
Servo servo = new Servo(MakerPiPico.GROVE1);
servo.setAngulo(90);        // 0 a 180 graus

Ventoinha v = new Ventoinha(MakerPiPico.GROVE2);
v.setVelocidade(60);        // 0 a 100 %

Rele r = new Rele(MakerPiPico.GROVE3);
r.ligar();  r.desligar();
```

Ecrã 16×2 — Lcd, I²C

```
Lcd lcd = new Lcd(MakerPiPico.GROVE1);
lcd.limpar();
lcd.escrever(0, "Temperatura:");  // linha 0
lcd.setPosicao(0, 1);  // coluna, linha
lcd.escrever(21.5);
lcd.escrever(" graus");
```

16 colunas (0–15), 2 linhas (0–1). `escrever(linha, texto)` apaga o resto da linha.

Mais sensores

```
SensorMovimento pir =
    new SensorMovimento(MakerPiPico.GROVE2);
pir.temMovimento()

SensorUltrassons us =
    new SensorUltrassons(MakerPiPico.GROVE3);
us.getDistancia()      // 2 a 400 cm
us.estaMaisPertoQue(20)

SensorClima c = new SensorClima(MakerPiPico.GROVE4);
c.getTemperatura()     // °C, double
c.getHumidade()        // %, double

Co2Sensor co2 = new Co2Sensor(MakerPiPico.GROVE5);
co2.getCo2()  // ppm, ao ar livre ≈ 420
co2.estaArViciado()    // acima de 1000 ppm
```

O sensor de clima mede no máximo uma vez por segundo, o sensor de CO2 a cada 2–5 s.

Cartões RFID — LeitorRfid

```
LeitorRfid leitor = new LeitorRfid(MakerPiPico.GROVE1);
String id = leitor.esperarCartao(); // espera
if (id.equals("04:A2:3F:1B")) { … }
leitor.temCartao()
leitor.getNumeroCartao() // null sem cartão
```

Só na `GROVE1`, `GROVE3` ou `GROVE5`. Compara textos com `equals`, não com `==`.

Wi-Fi e servidor web — Pico W, Pico 2 W

```
WiFi wifi = new WiFi();
ServidorWeb servidor = new ServidorWeb();
wifi.iniciar("segredo123");  // palavra-passe: 8-63
System.out.println(wifi.getName());  // VisuMicro-3F2A

while (true) {
  Pedido p = servidor.proximoPedido();
  if (p != null) {
    if (p.getCaminho().equals("/ligar")) {
      led.ligar();
      p.responder("<h1>LED ligado</h1>");
    } else {
      p.responderNaoEncontrado();
    }
  }
  delay(10);
}
```

O Pico abre a sua própria rede, sem internet. Liga o telemóvel e abre `http://192.168.42.1` no browser.

`p.getParameter("cor")` lê `/cor?cor=vermelho`. Cada pedido precisa de uma resposta.

Polimorfismo — um tipo, muitos componentes

```
Comutavel[] todos = { led, rele, ventoinha };
for (Comutavel c : todos) {
  c.desligar();  // cada objeto à sua maneira
}
```

As classes por trás

```
Componente (pino)
├─ Atuador  implements Comutavel
│   ├─ Led ── LedRegulavel
│   ├─ Rele
│   └─ Ventoinha
├─ SensorDigital ── Botao, SensorMovimento
├─ SensorAnalogico ── SensorLuz, SensorSom,
│                     BotaoRotativo
└─ Buzzer, LedRgb, Servo, Lcd, …
```

`Comutavel` é uma **interface** com `ligar()`, `desligar()`, `estaLigado()`.

Derivar uma classe própria — extends

```
public class LuzAviso extends Led {
  public LuzAviso(int pino) {
    super(pino);  // construtor de Led
  }
  public void avisar(int vezes) {
    for (int i = 0; i < vezes; i++) {
      piscar(100);           // método herdado
    }
  }
}
```

A biblioteca é Java a sério. Podes vê-la no editor e derivar dela.

Quando não funciona

sensor de luz, som ou rotativo falha	sensores analógicos só na `GROVE6`
módulo Grove não faz nada	entrada no código = entrada na placa?
sem som	interruptor de silêncio ao lado do buzzer
placa não encontrada	cabo de dados, não de carregamento; navegador Chromium (Helium, Chrome, Edge); firmware instalado?