

Roteiro de Prática

SE-P02 - Capacitores Carga e Descarga

2026-02

Participação Especial !



@calabotics

Objetivos

Identificar um capacitor eletrolítico

- Capacitância (μF) - identificação do valor;
- Tensão máxima de operação;
- Polaridade;

Carga e descarga de um capacitor

- Observar o funcionamento do capacitor no armazenamento de energia;
- Comportamento da tensão ao longo do tempo;
- Processos de carga e descarga;

Circuito RC

- Associação entre resistor e capacitor;
- Constante de tempo ($\tau = R \times C$) e tempo estimado de carga ($\tau = 5RC$);
- Relação entre resistência, capacitância e velocidade de carga/descarga.

Itens necessários



Os itens não estão em escala.

Itens necessários



Capacitor eletrolítico
100 μ F x 25V (2pçs)

Os itens não estão em escala.

Itens necessários



Potenciômetro Linear
10 K Ω

Os itens não estão em escala.

Itens necessários

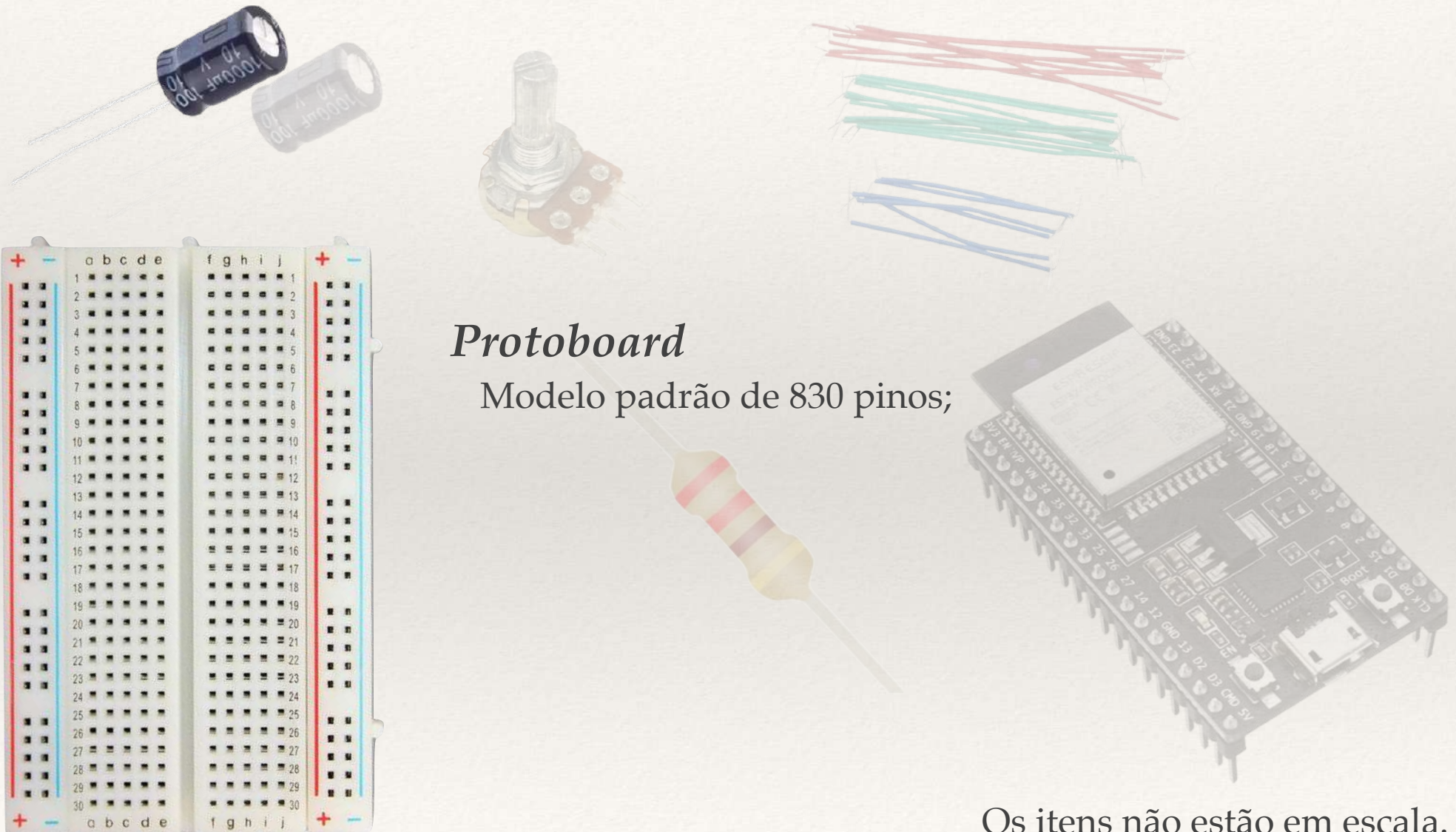


Jumpers rígidos

Cores diversas;
Pré-instalados;

Os itens não estão em escala.

Itens necessários

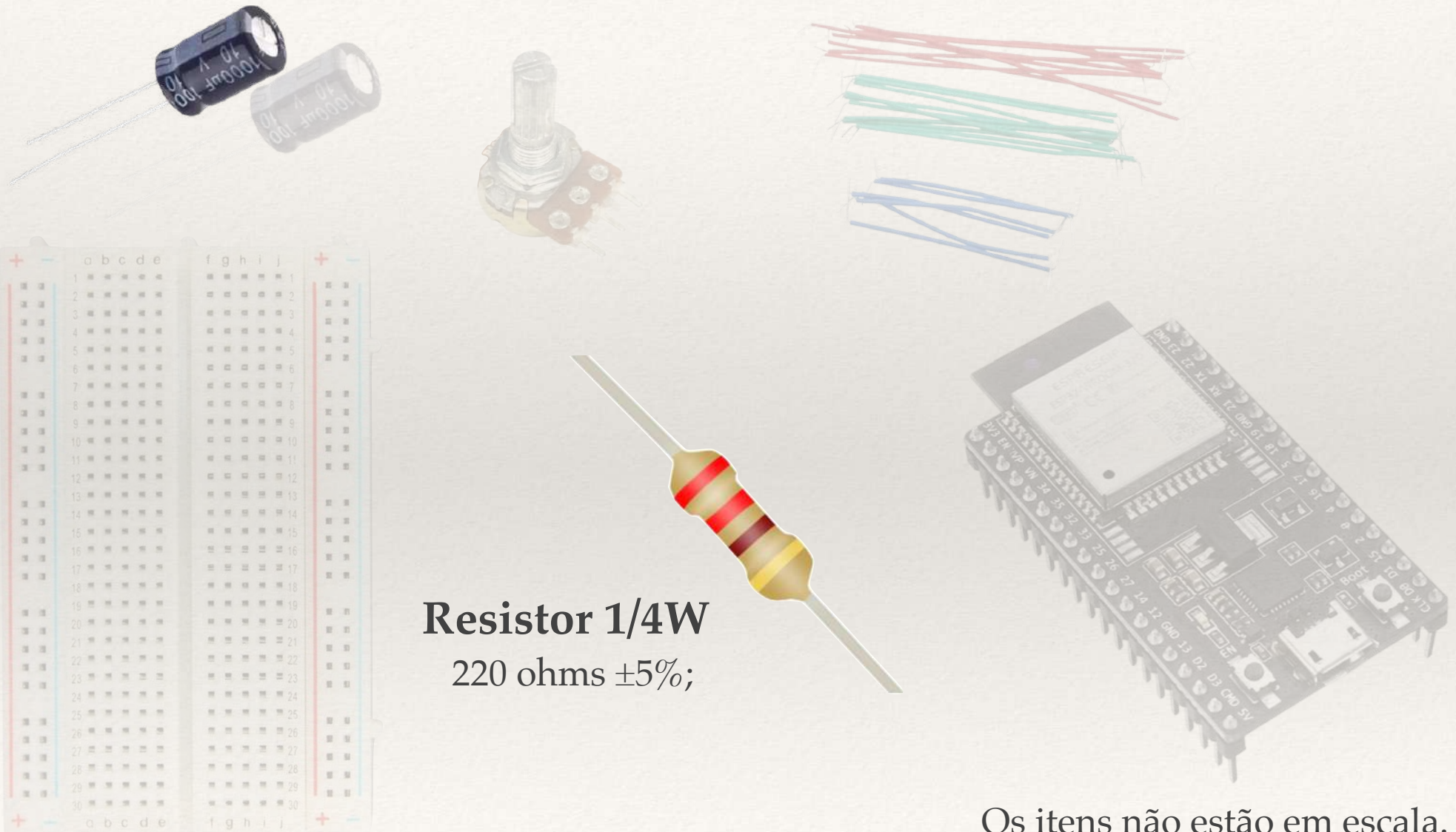


Protoboard

Modelo padrão de 830 pinos;

Os itens não estão em escala.

Itens necessários



Resistor 1/4W
220 ohms $\pm 5\%$;

Os itens não estão em escala.

Itens necessários

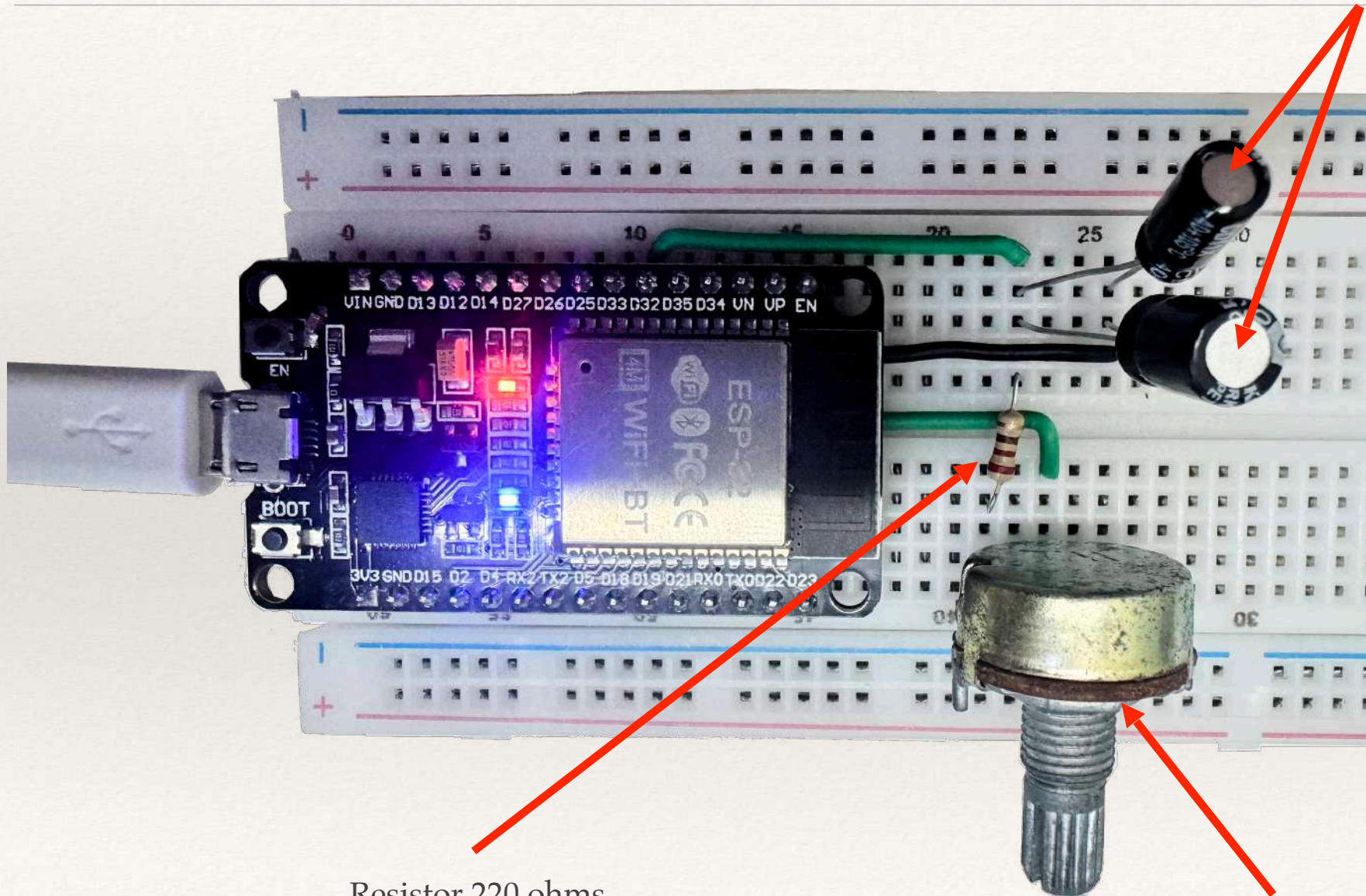


Placa de Protótipo
ESP32 Dev Module;
32 pinos;

Os itens não estão em escala.

Circuito

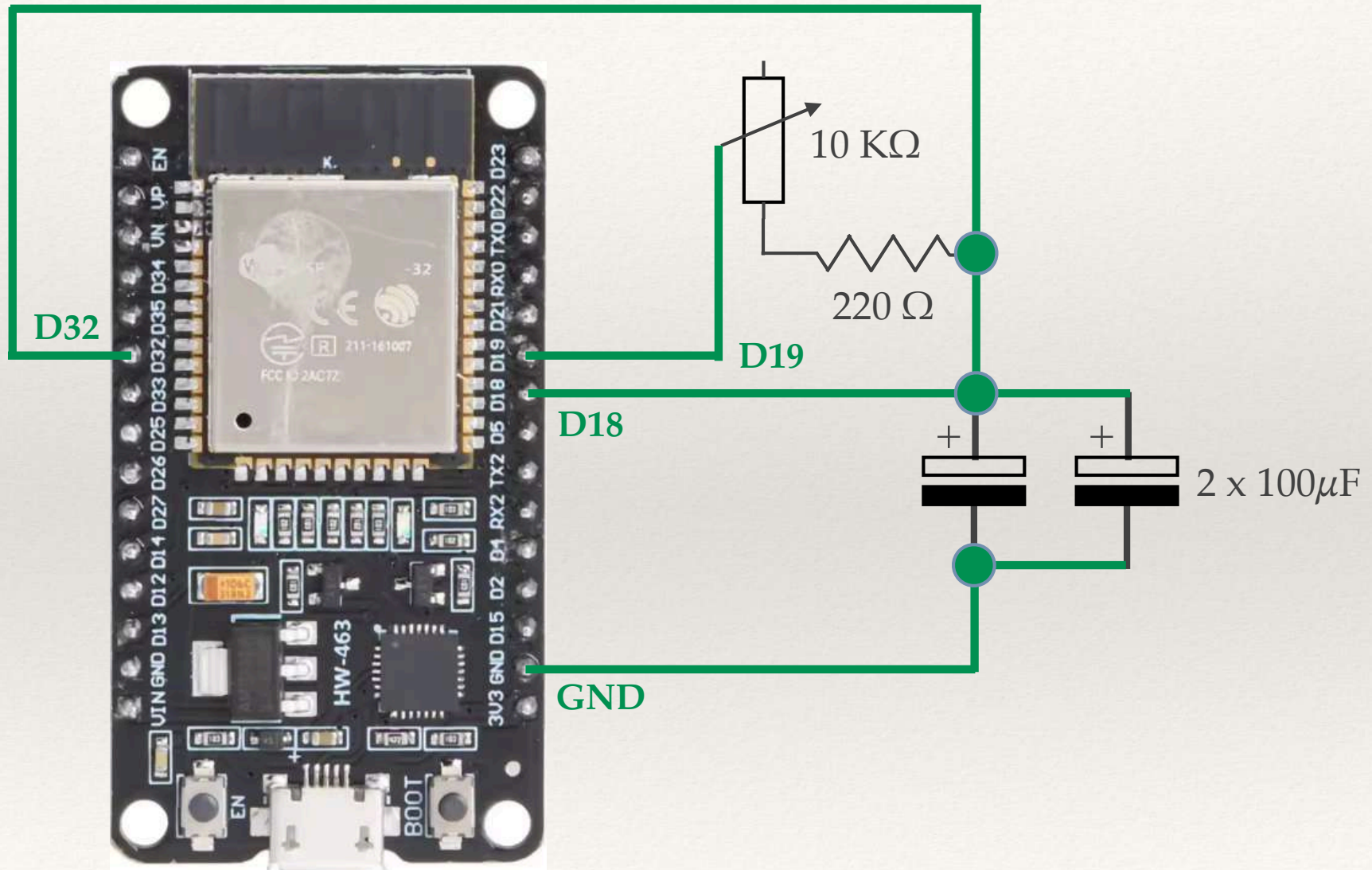
2 Capacitores 100 μ F



Resistor 220 ohms

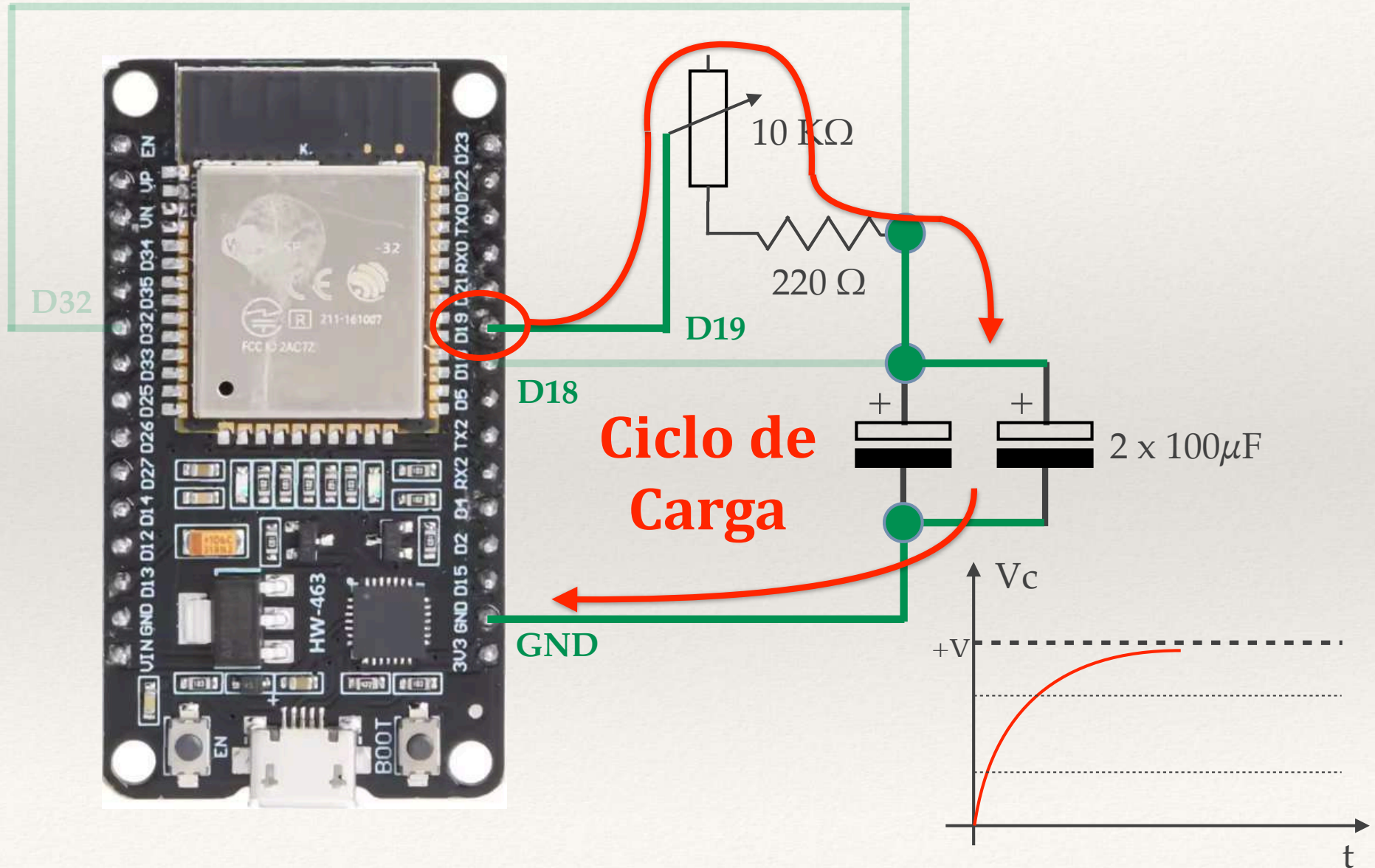
Potenciômetro 10 K Ω

Circuito



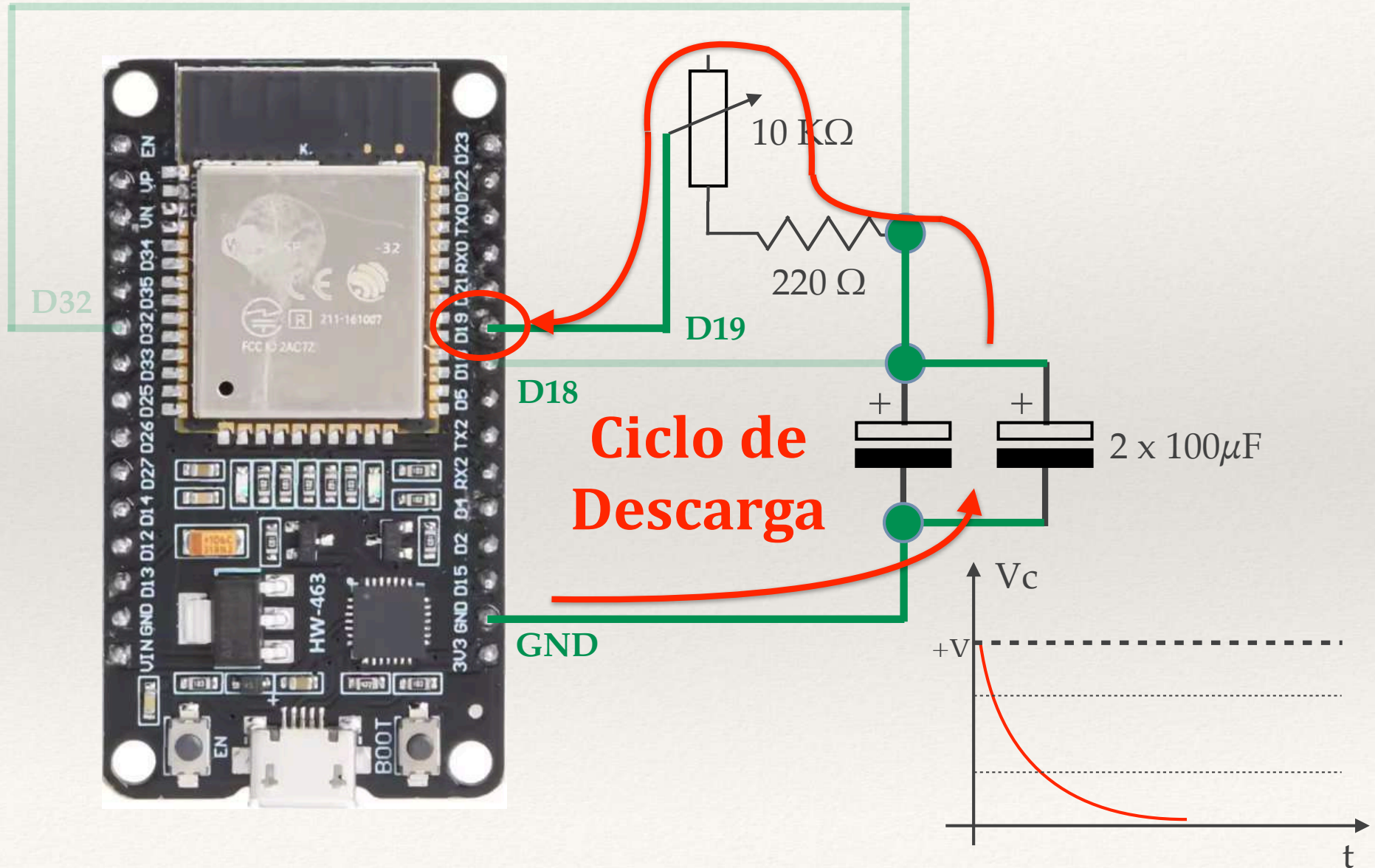
Circuito

D19: 0 → 1

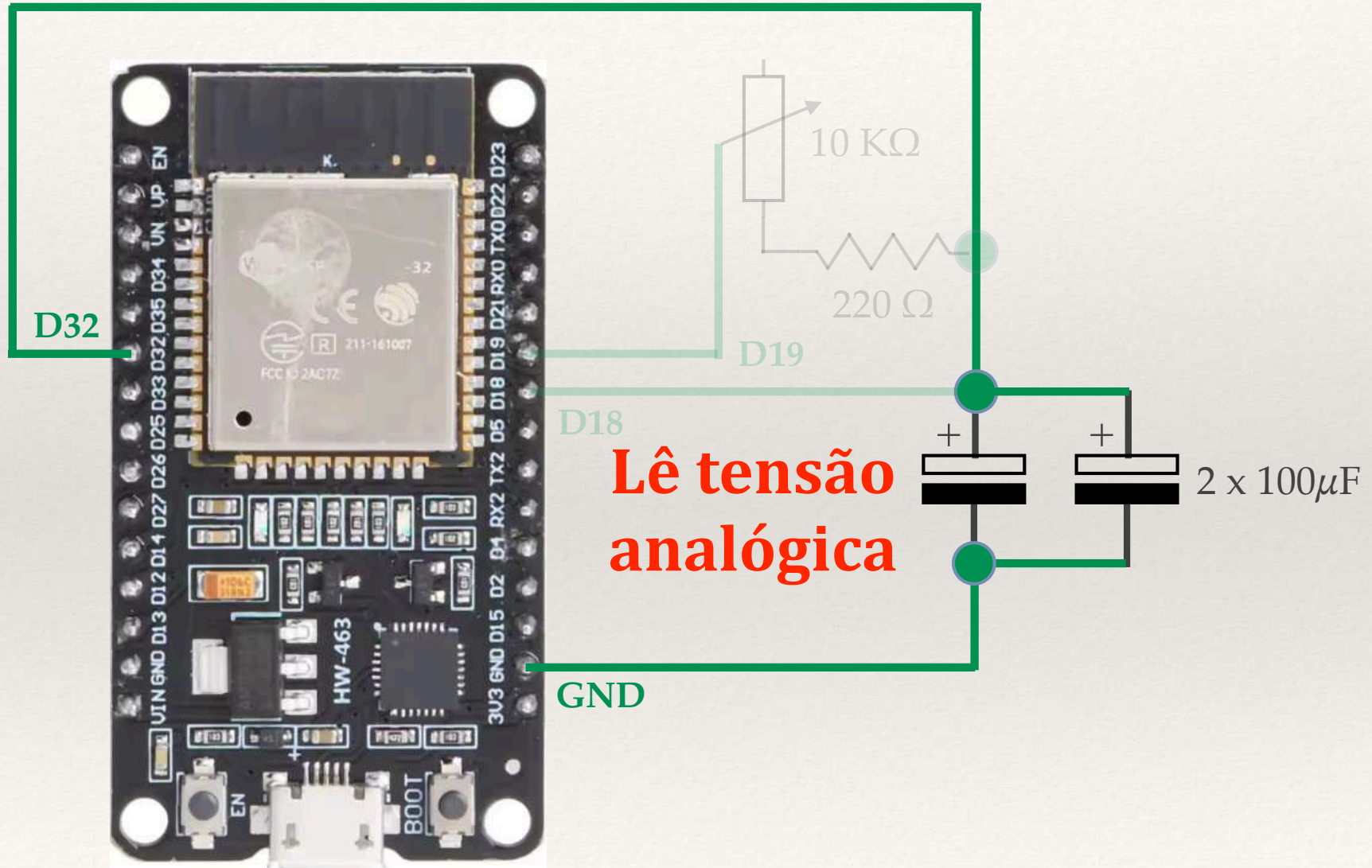


Circuito

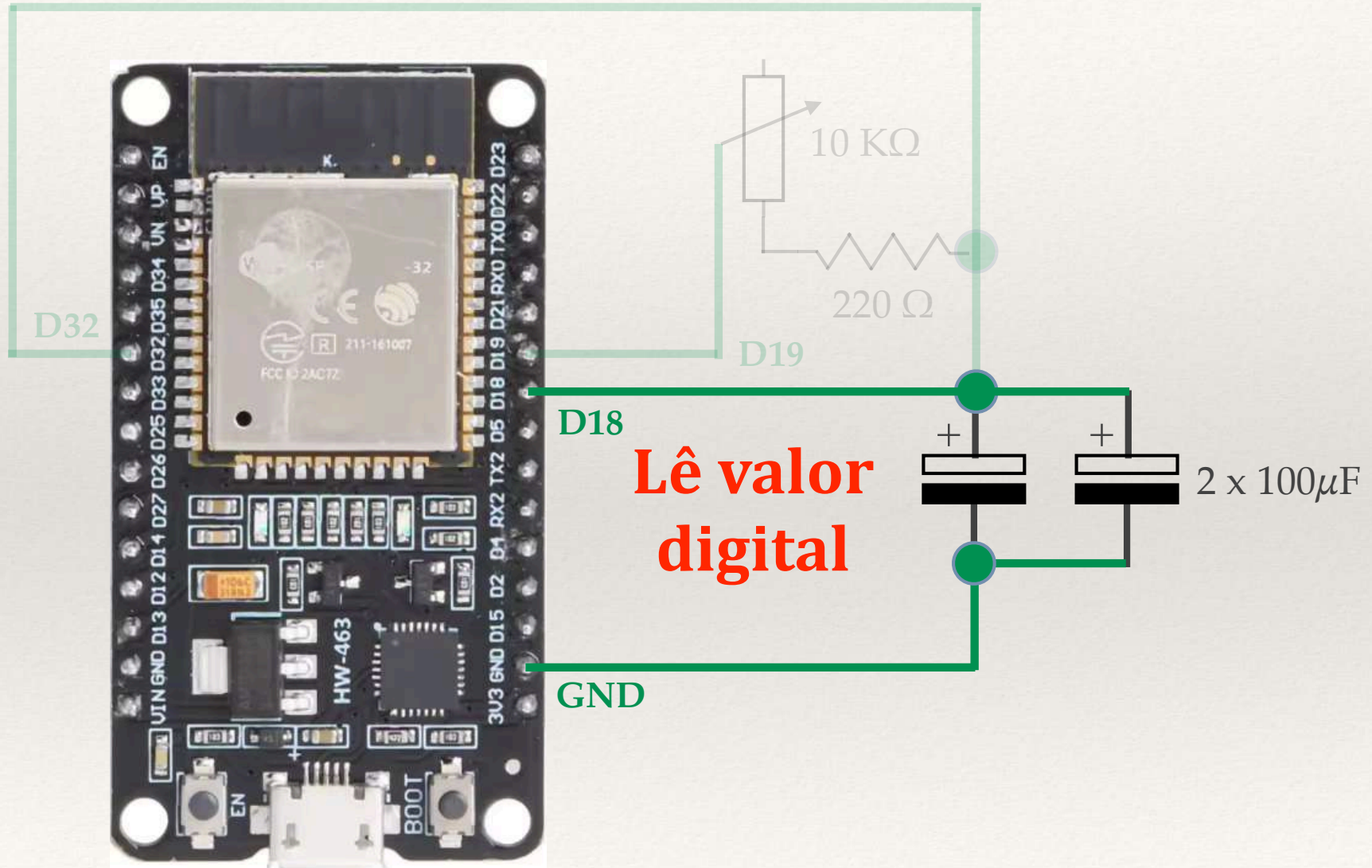
D19: 1 $\rightarrow$ 0



Circuito



Circuito



Código

```
CargaEDescargav02 | Arduino 1.8.19

// Alterações na versão 02
//
// HARDWARE
// Alterado o LED de saída para o 02 (LED interno da placa) para reduzir o número de componentes (não é mais necessário o LED azul, nem o resistor)
// Acrescentado novo capacitor em paralelo com o já existente, aumentando a capacitância para 200 uF.
// Retirado resistor em série, reduzindo R para 220 ohms.
//
// SOFTWARE
// Alterei escala no método PlotarDados para indicar 3300mV como topo de escala no plotter
// Mudei o laço de carga e descarga para plotar tudo o que for possível dentro do período definido.
// Acrescentei a plotagem da saída para o LED.
//

#define pinoControle 19          // Pino que comanda a carga (digital)
#define pinoLED 2              // Pino que controla o LED (2 = interno)
#define pinoLeituraAnalogica 32 // Pino que lê a tensão do capacitor para plotagem
#define pinoLeituraDigital 18   // Pino que lê digitalmente a tensão do capacitor (que controla o LED)
#define periodo 1000            // Período em milissegundos (80 para ver a falha na leitura digital)
#define intervaloAmostragem 10  // Tempo entre amostragens em ms (quanto menor, maior a largura do pulso no plotter)

void setup() {
    pinMode(pinoControle, OUTPUT); // Pino que alimenta/descarrega o capacitor
    pinMode(pinoLED, OUTPUT);      // Pino que acende/apaga o LED
    pinMode(pinoLeituraAnalogica, INPUT); // Pino que lê saída do capacitor em formato analógico
    pinMode(pinoLeituraDigital, INPUT);  // Pino que lê saída do capacitor em formato digital

    Serial.begin(115200);
}
```

CargaEDescargav02

```

void loop() {

  // Carga
  long int inicio = millis();           // Marca início da carga
  digitalWrite(pinoControle, HIGH);     // Liga o capacitor ao 3,3V para carregar
  int leitura = digitalRead(pinoLeituraDigital); // Lê digitalmente a tensão do capacitor
  digitalWrite(pinoLED, leitura);       // O LED espelha o estado digital do capacitor
  while( millis() < inicio + periodo ) {
    plotarDados();                     // Envia uma amostra para o plotter serial
    delay(intervaloAmostragem);       // Espera o tempo entre amostragens
  }

  // Descarga
  inicio = millis();                   // Marca início da carga
  digitalWrite(pinoControle, LOW);     // Liga o capacitor ao GND para descarregar
  leitura = digitalRead(pinoLeituraDigital); // Lê digitalmente a tensão do capacitor
  digitalWrite(pinoLED, leitura);       // O LED espelha o estado digital do capacitor
  while( millis() < inicio + periodo ) {
    plotarDados();                     // Envia uma amostra para o plotter serial
    delay(intervaloAmostragem);       // Espera o tempo entre amostragens
  }
}

// Plotar com os limites travados
void plotarDados() {

  // Lê a tensão do capacitor e ajusta a escala para 3300mV
  // Originalmente lemos 0 a 4095 no ADC de 12 bits do ESP32
  float medicao = analogRead(pinoLeituraAnalogica) * 3.3 / 4.1;
  int tensao = int (medicao);
  int leitura = digitalRead(pinoLeituraDigital) * 3300; // Lê digitalmente a tensão do capacitor

  // Plota o gráfico no Plotter Serial
  // Usa a técnica do artigo de "Internet e Coisas" para mostrar eixos superior e inferior
  Serial.print(-100);                // Plota Linha inferior fixa (-0,1V)
  Serial.print(",");                  // Separador entre os valores do plotter
  Serial.print(3400);                 // Plota Linha superior fixa (3,4V)
  Serial.print(",");                  // Separador entre os valores do plotter
  Serial.print(leitura);              // Plota Valor lido digitalmente
  Serial.print(",");                  // Separador entre os valores do plotter
  Serial.println(tensao);             // Plota a curva carga/descarga
}

```

Hora da Prática!

1º Passo - Abra o Arduino IDE



Hora da Prática!

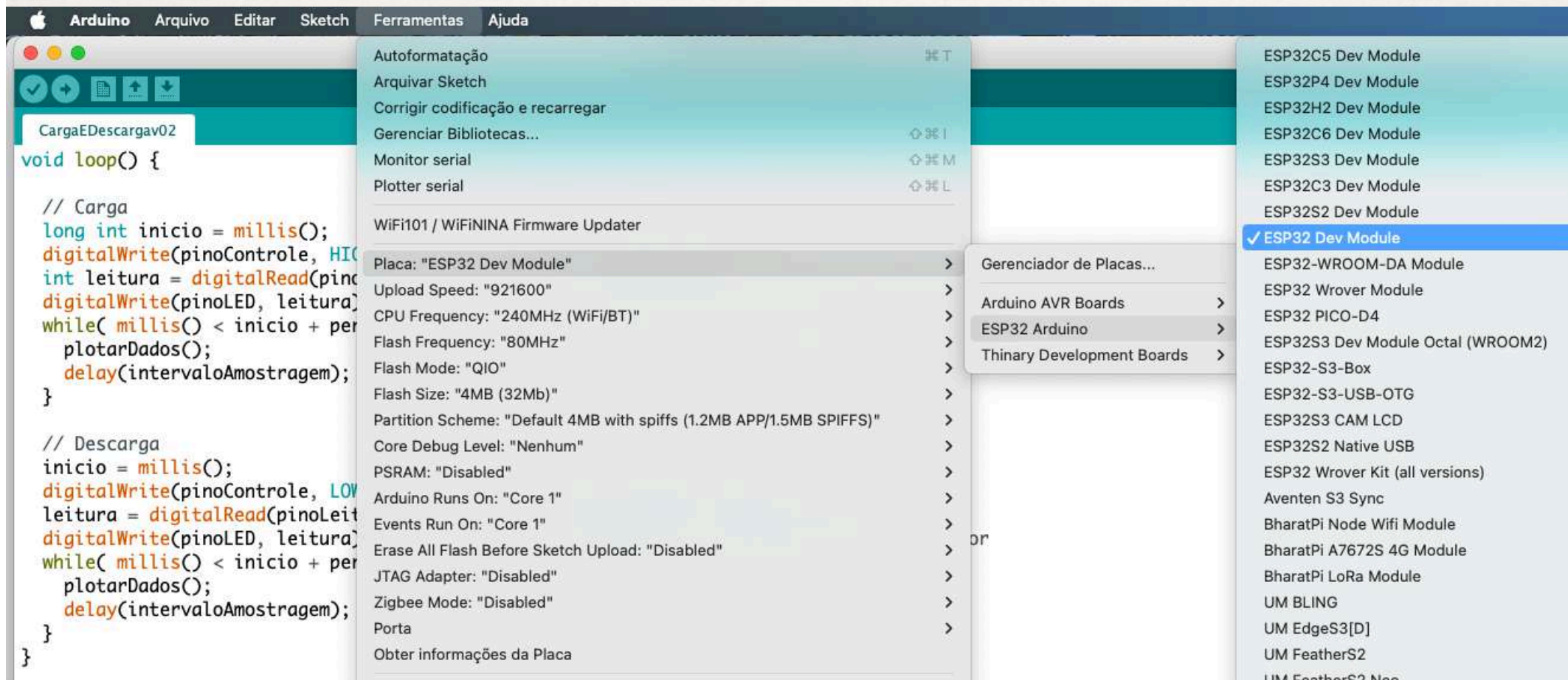
2º Passo - Conecte o ESP à interface USB do computador.



ATENÇÃO: Cuidado para não ter feito nenhuma modificação no circuito !

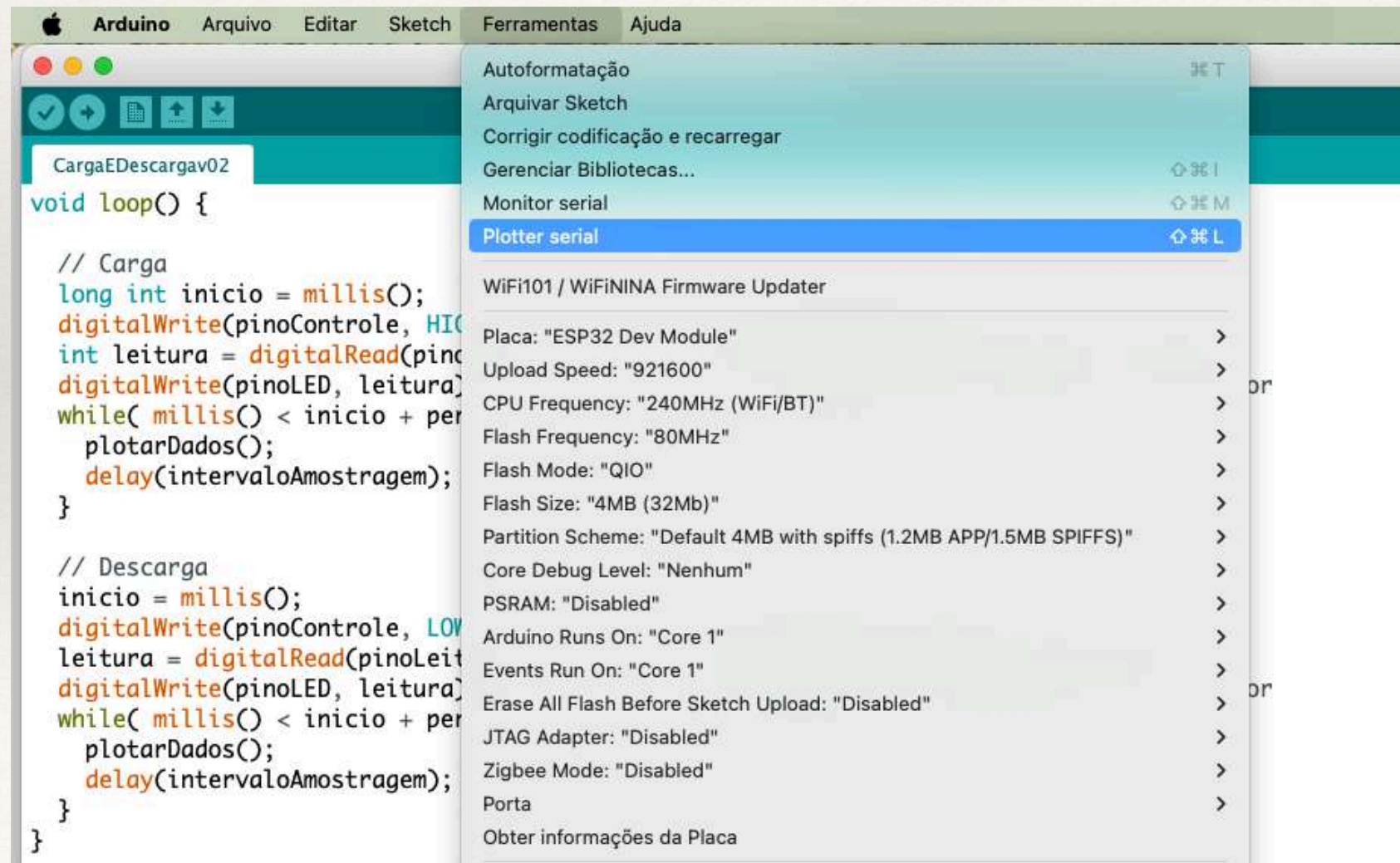
Hora da Prática!

3º Passo - Com o ESP32 conectado, escolha a placa correta, e depois a porta.



Hora da Prática!

4º Passo - Abra a janela do Plotter Serial



F I M