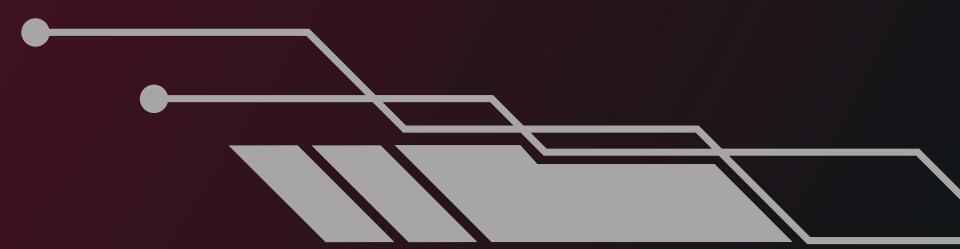




# TRAINING





# RETROSPECTIVA

- O QUE É ROBÓTICA?
- ROBÓTICA VS AUTOMAÇÃO
- MICROCONTROLADORES
- O QUE É ARDUINO ?

- PRINCIPAIS VANTAGENS
- CARACTERÍSTICAS DA PLACA
- LÓGICA DE PROGRAMAÇÃO
- TINKERCAD



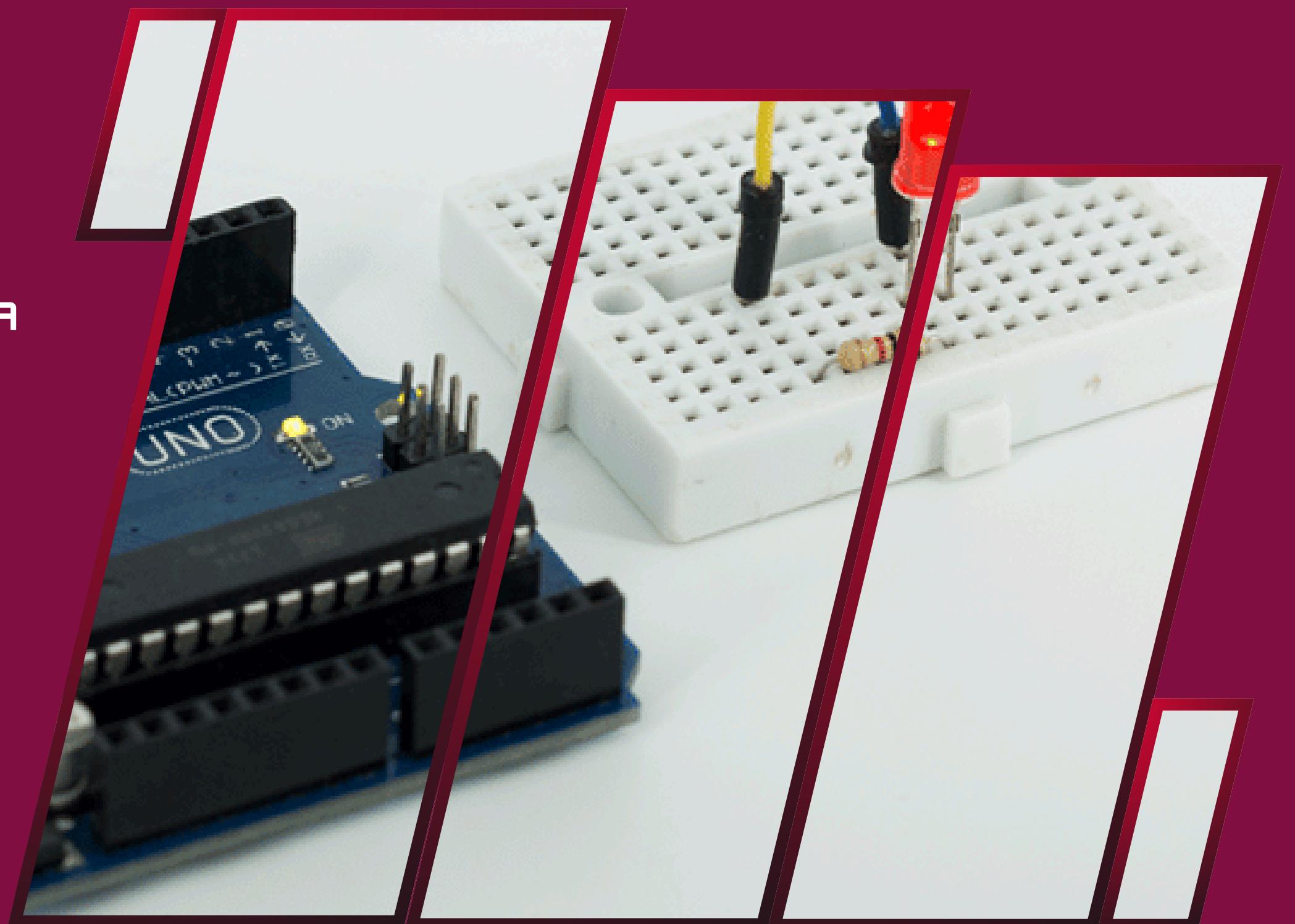
UNICAMP



FACULDADE DE TECNOLOGIA

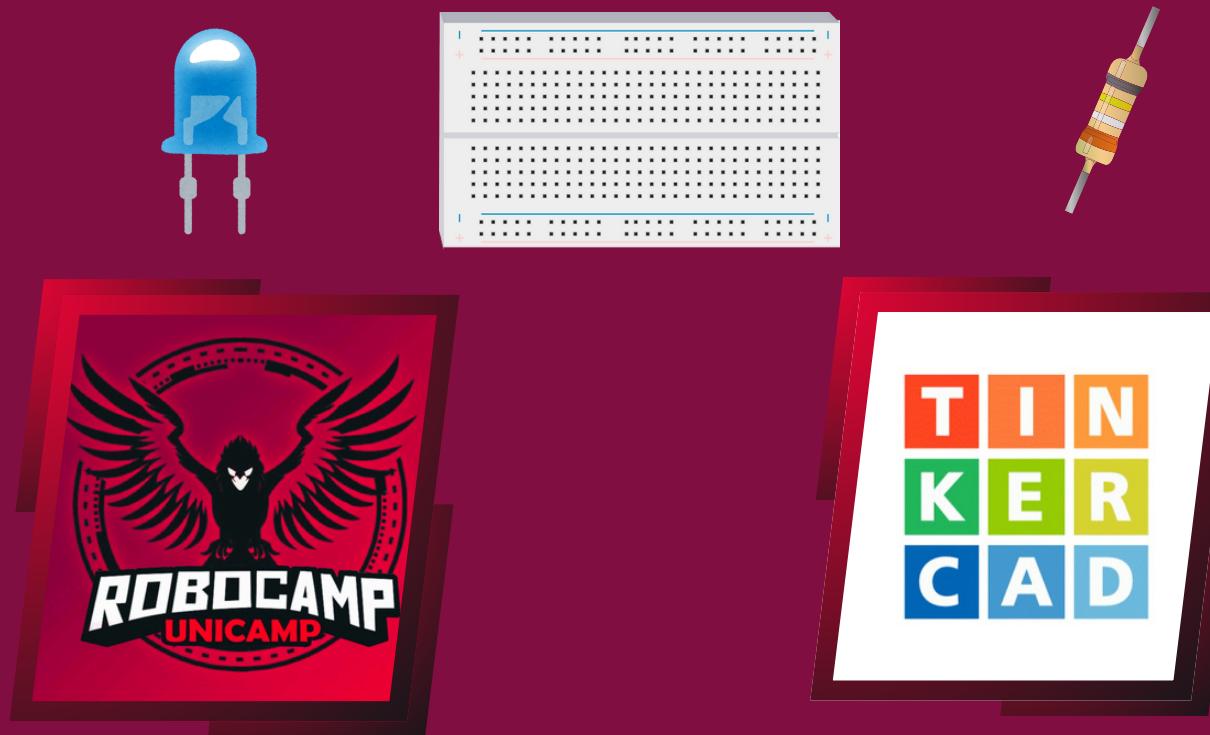
# PROTOTIPAGEM DE CIRCUITOS ELETRÔNICOS

- O QUE É UMA PROTOBOARD?
- COMO USAR UMA PROTOBOARD
- CORRENTE, TENSÃO E RESISTÊNCIA
- PRIMEIRA LEI DE OHM
- USO DE CONSTANTES E VARIÁVEIS
- LAÇOS CONDICIONAIS



# EXERCÍCIOS DE FIXAÇÃO

MONTAGEM NA PROTOBOARD

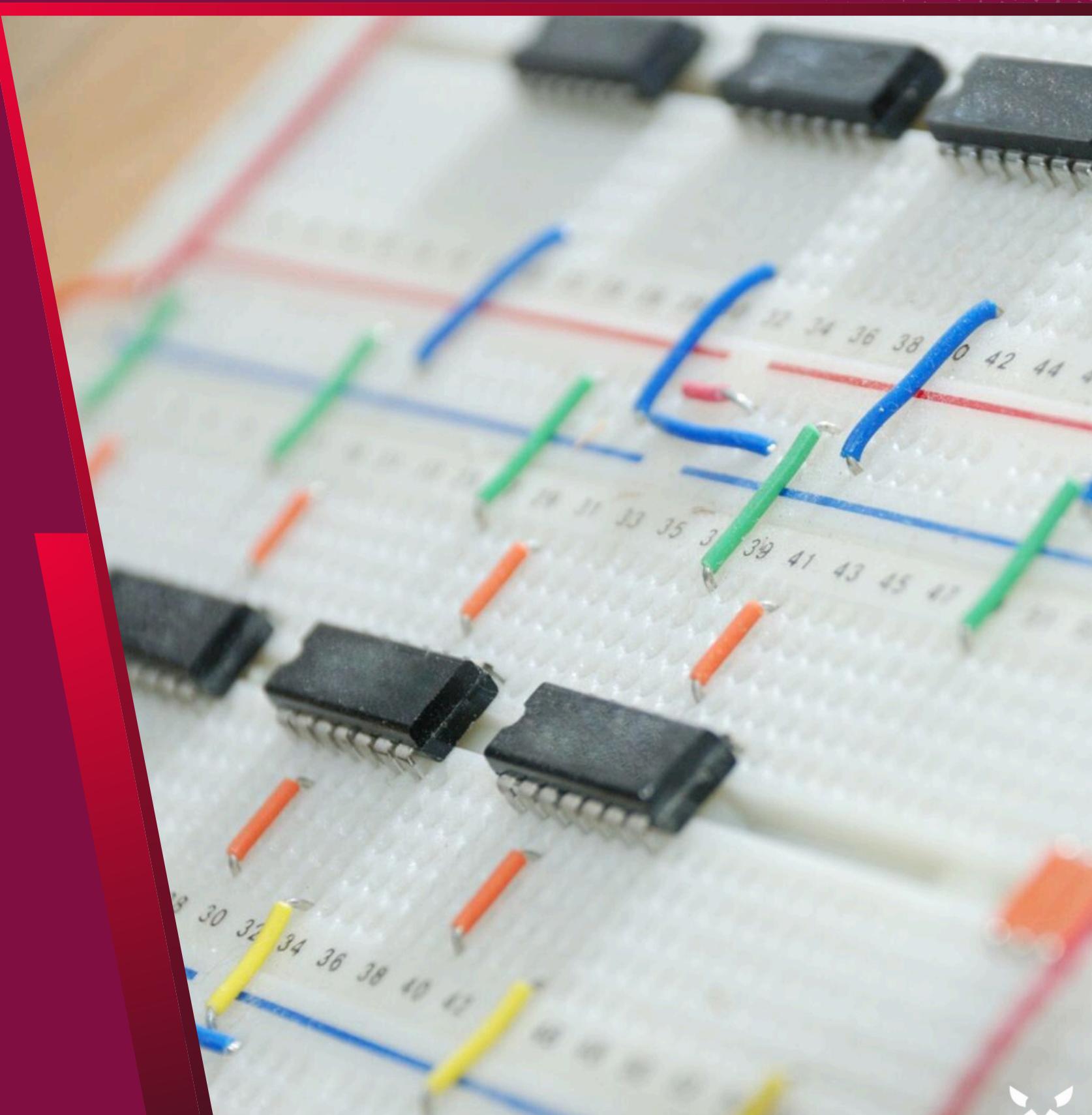


SIRENE LED

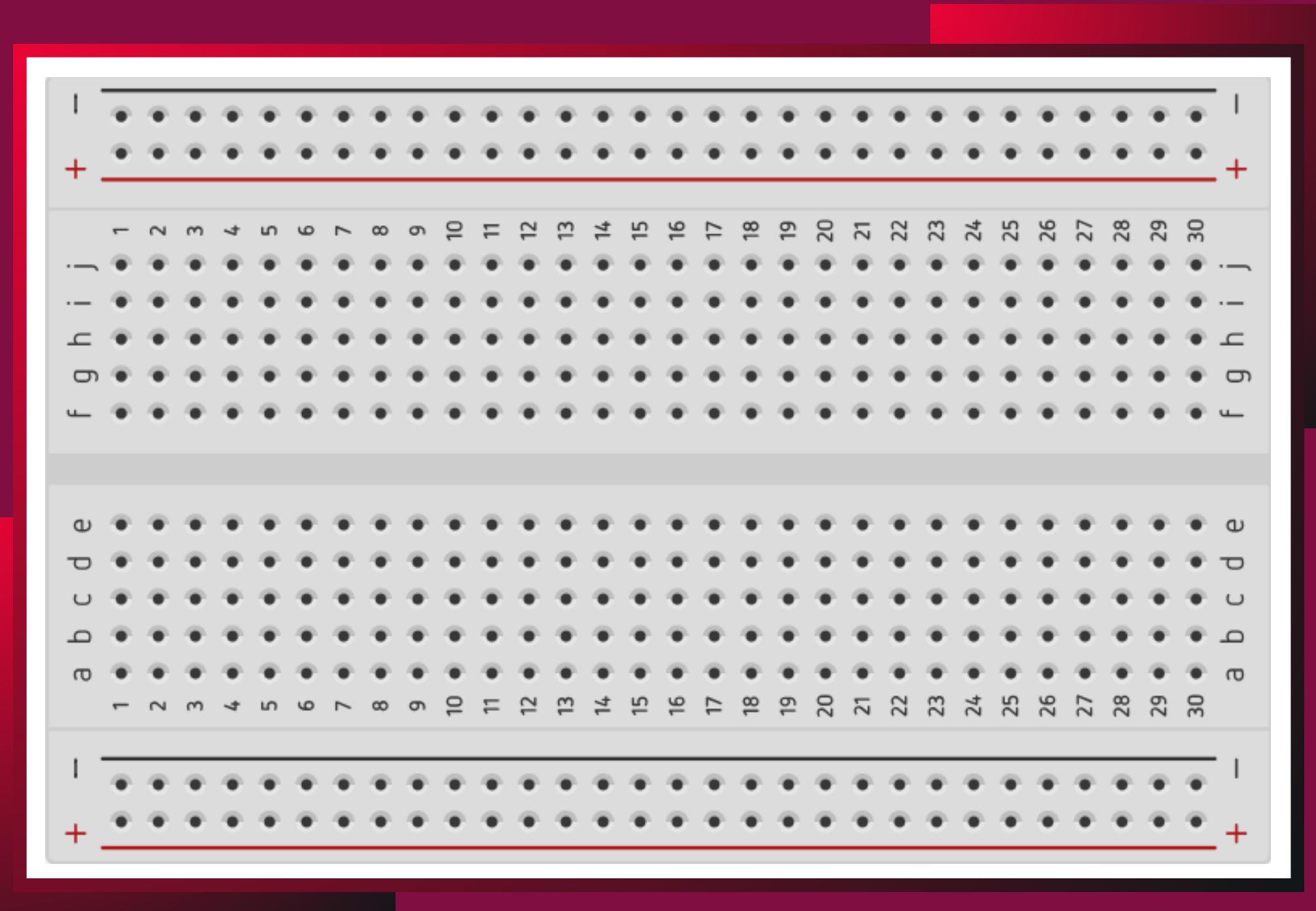


# O QUE É UMA PROTOBOARD?

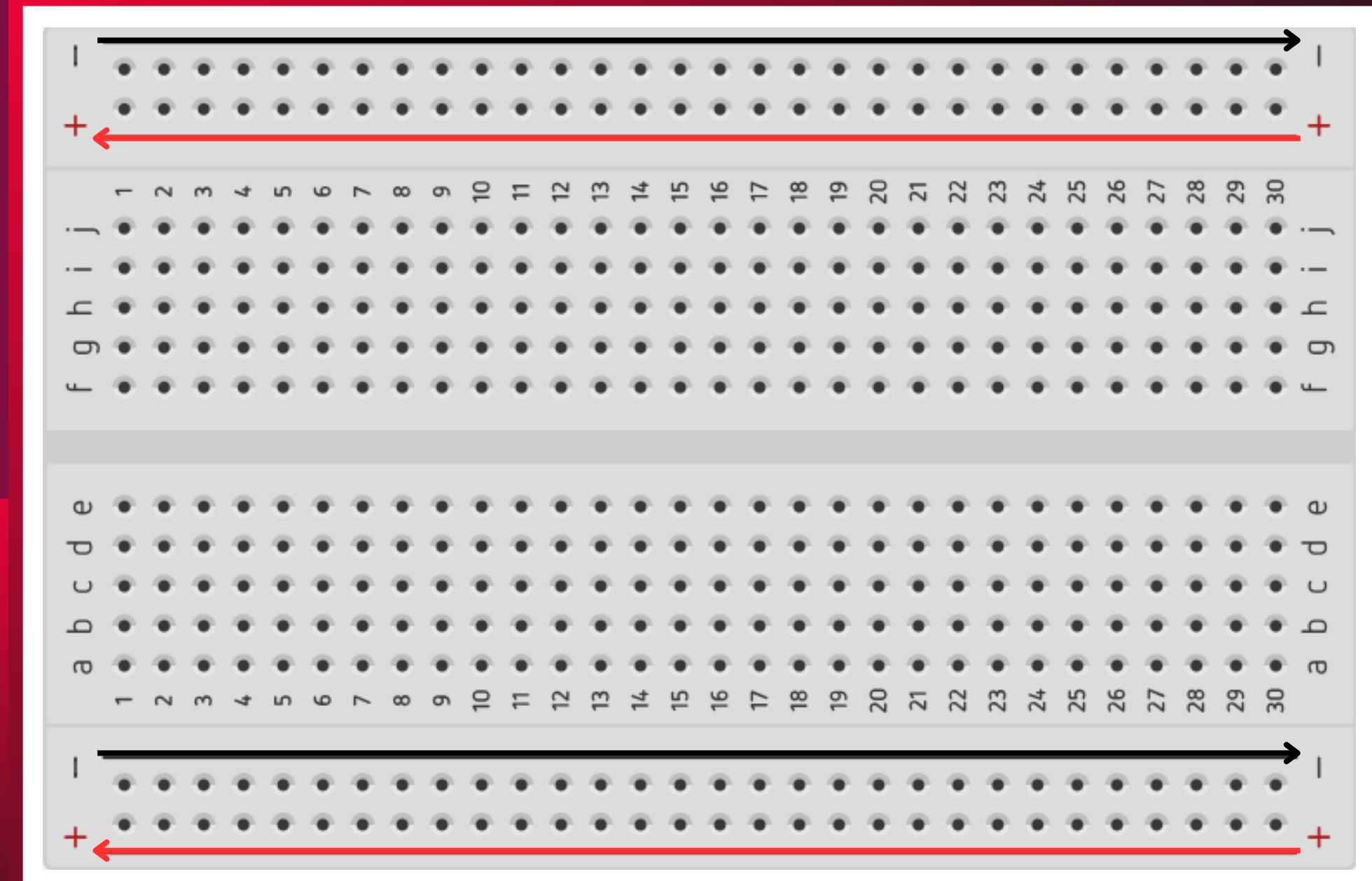
- PLACA DE ENSAIO
  - FACILITA A MONTAGEM
  - PERMITE MÚLTIPLAS CONEXÕES ENTRE OS COMPONENTES

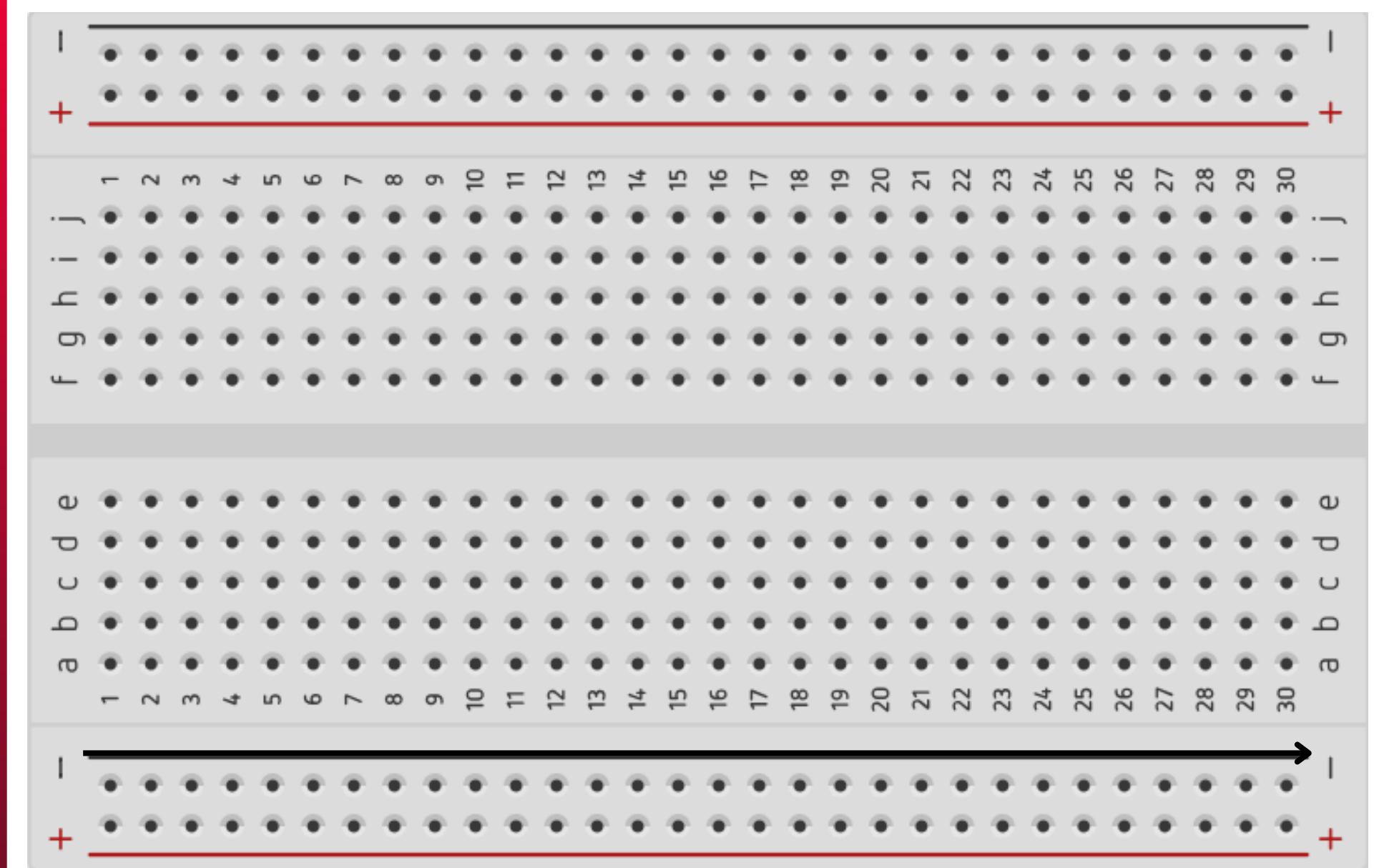


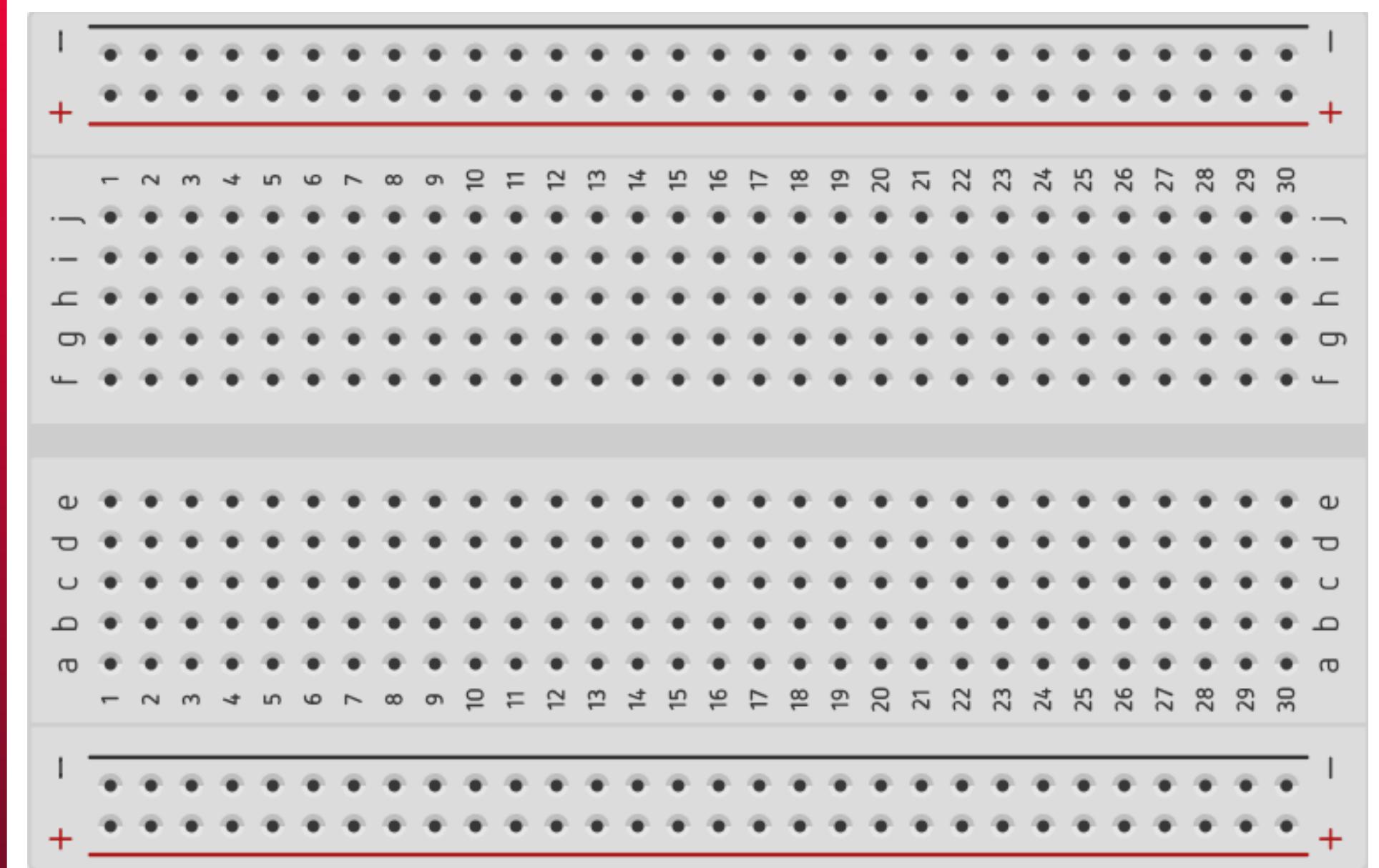
# COMO USAR A PROTOBOARD?

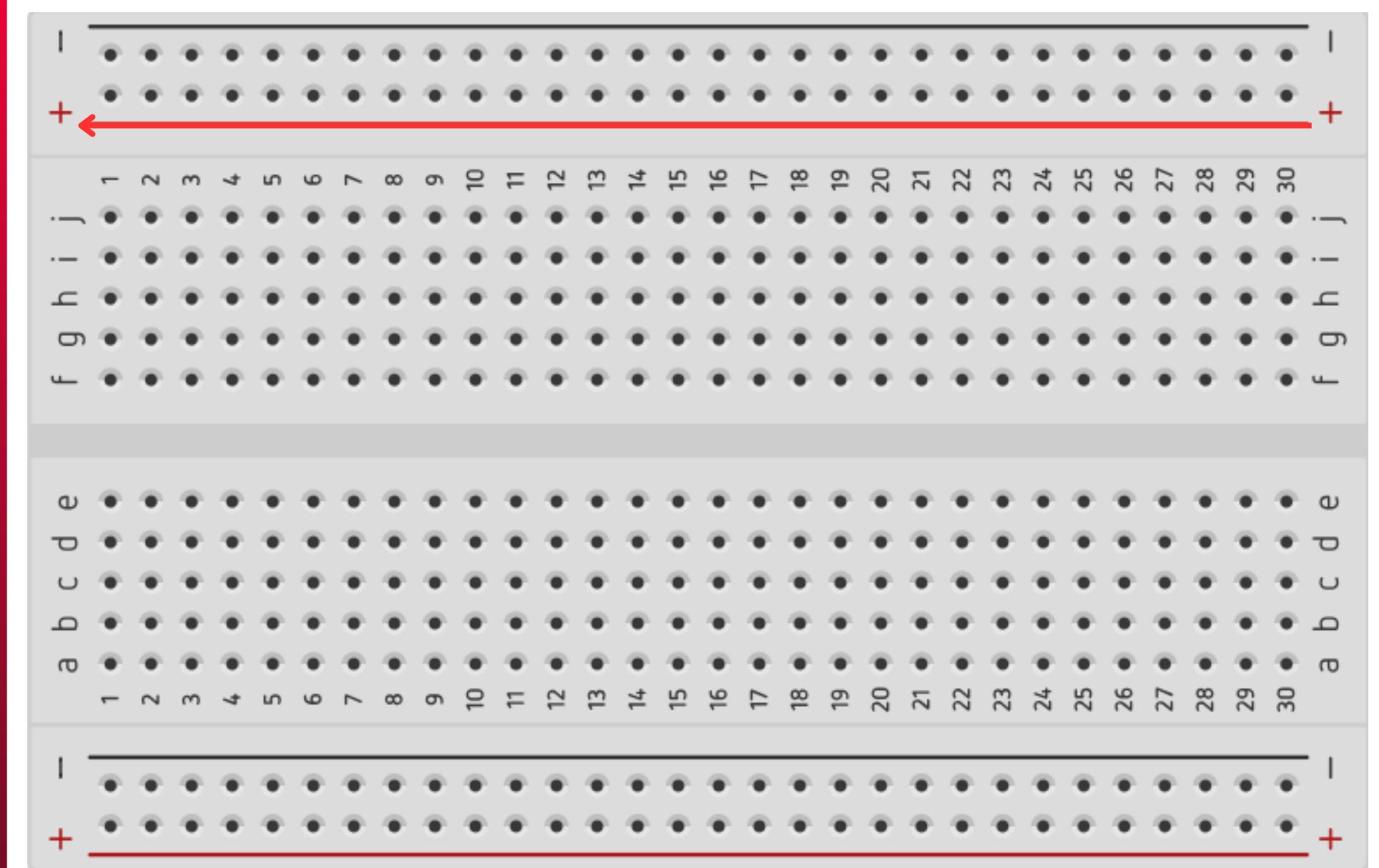


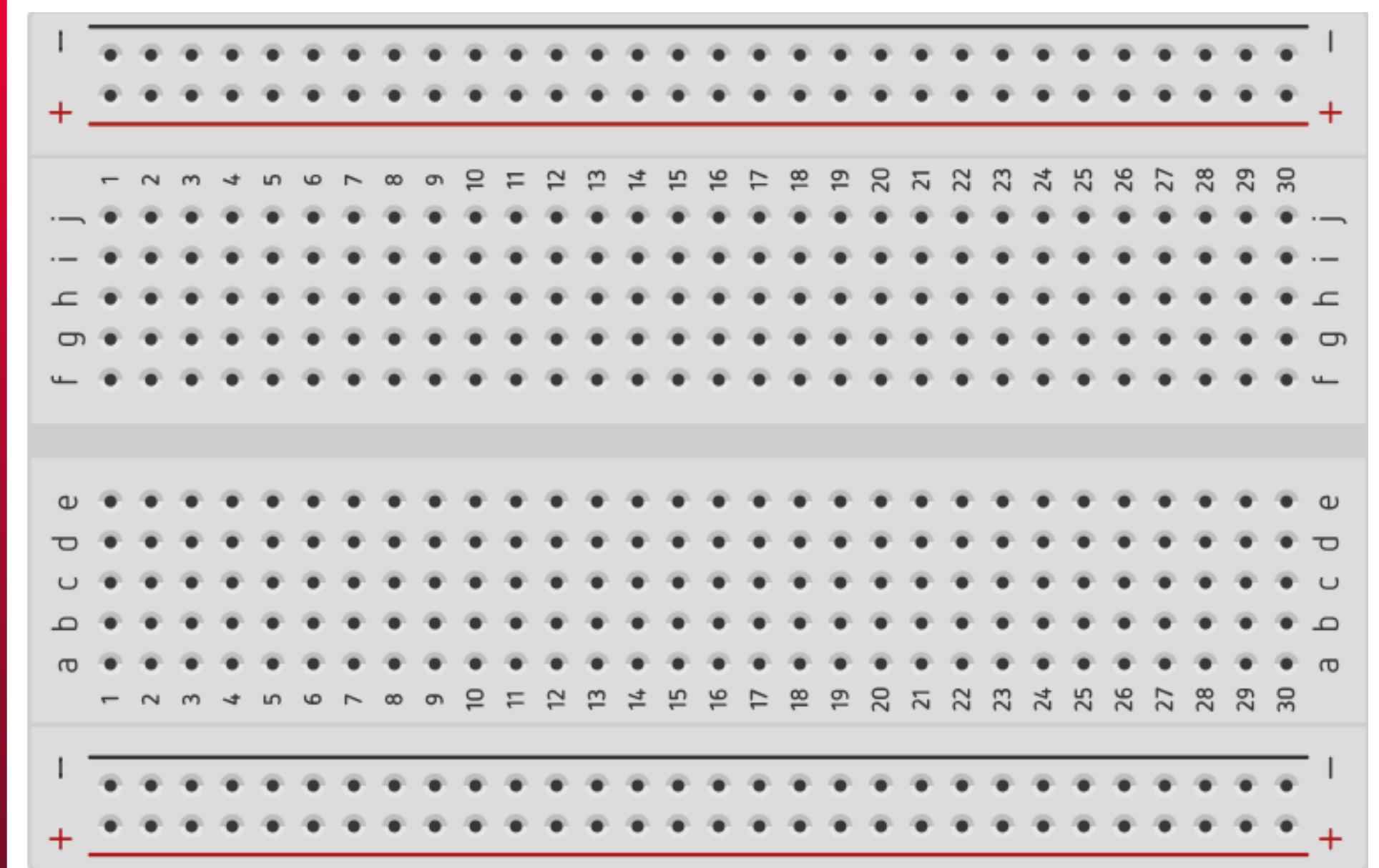
# PERMITE INTERCONEXÃO ENTRE OS FUROS HORIZONTAIS

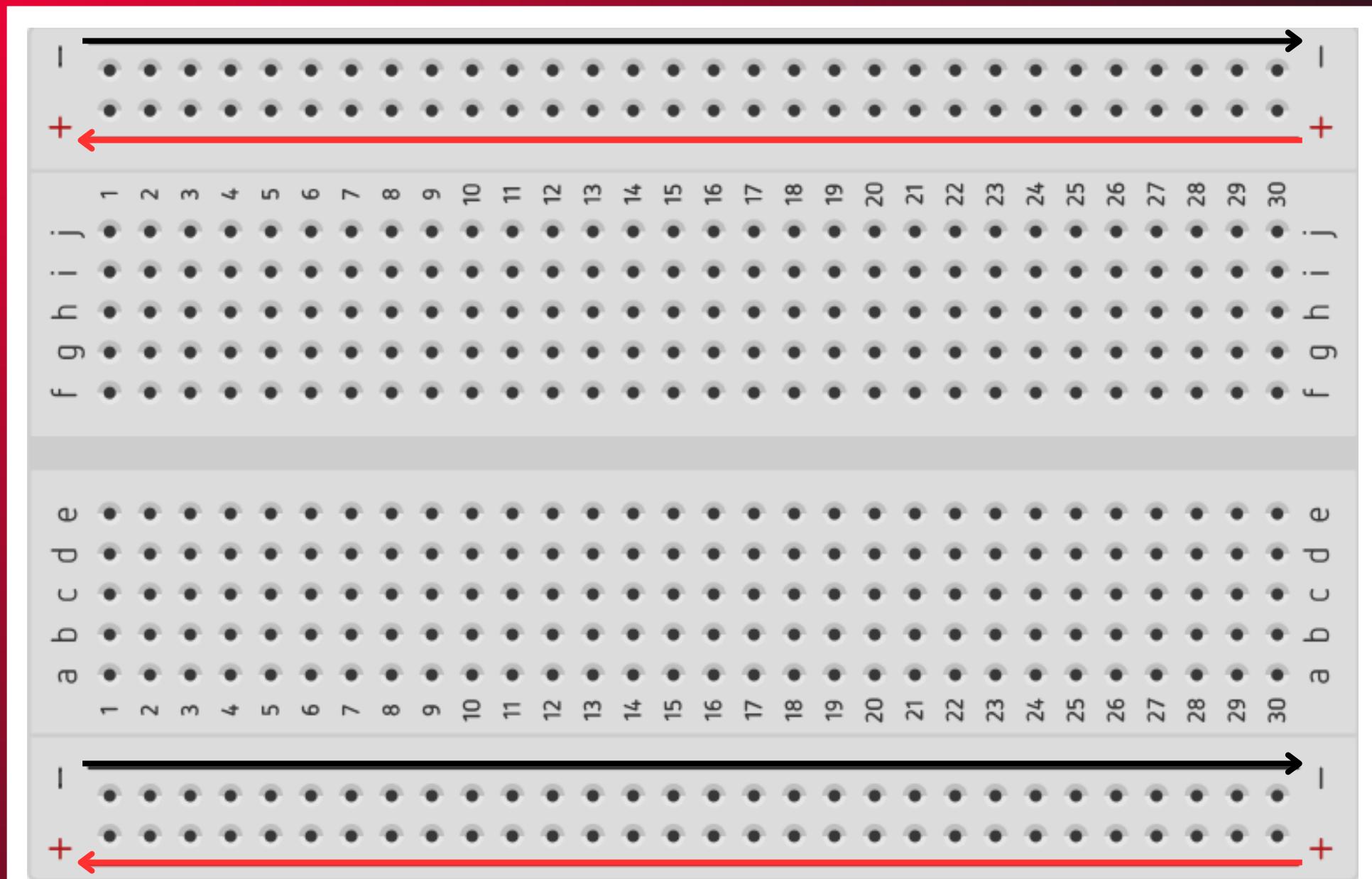




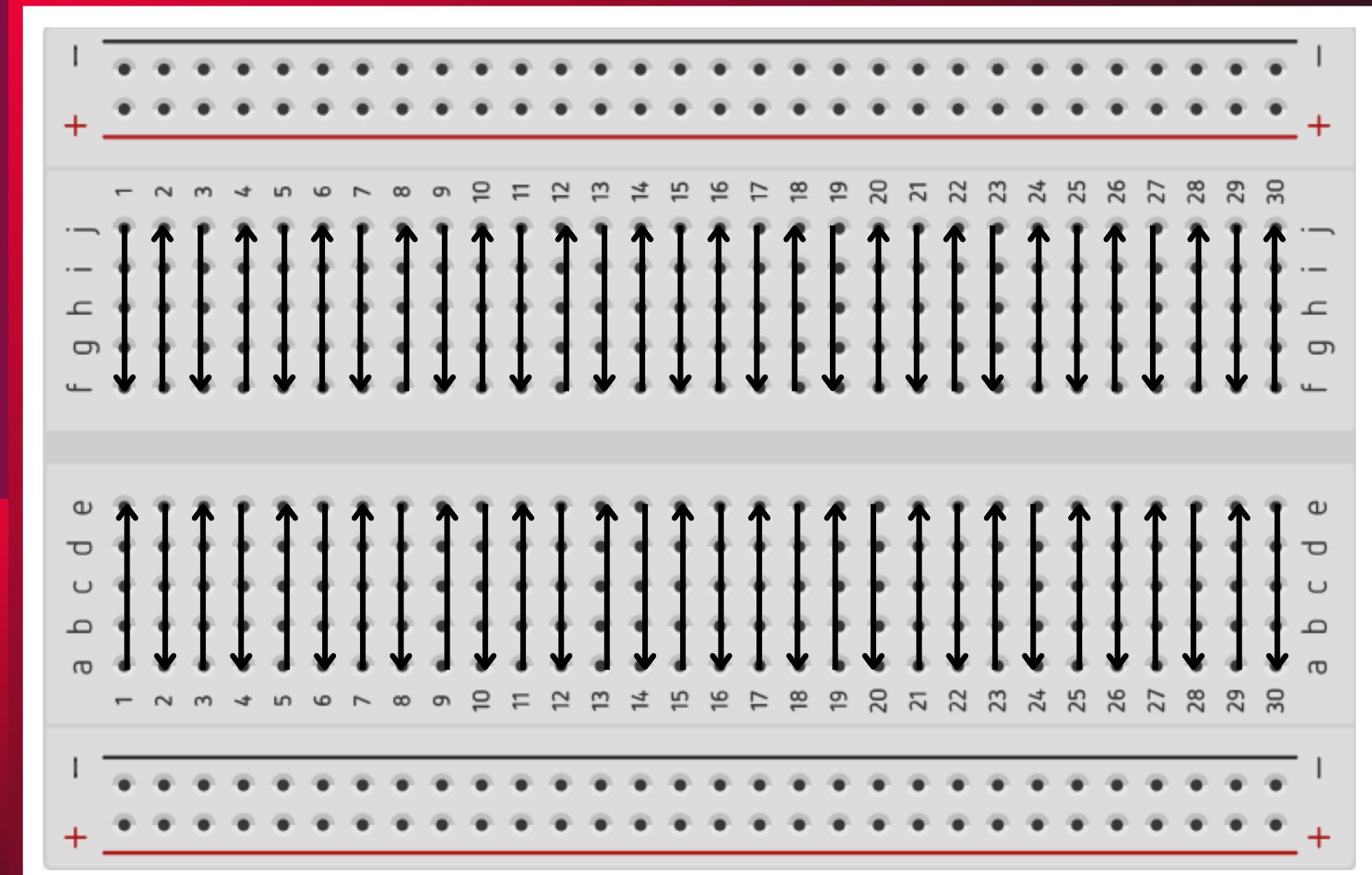


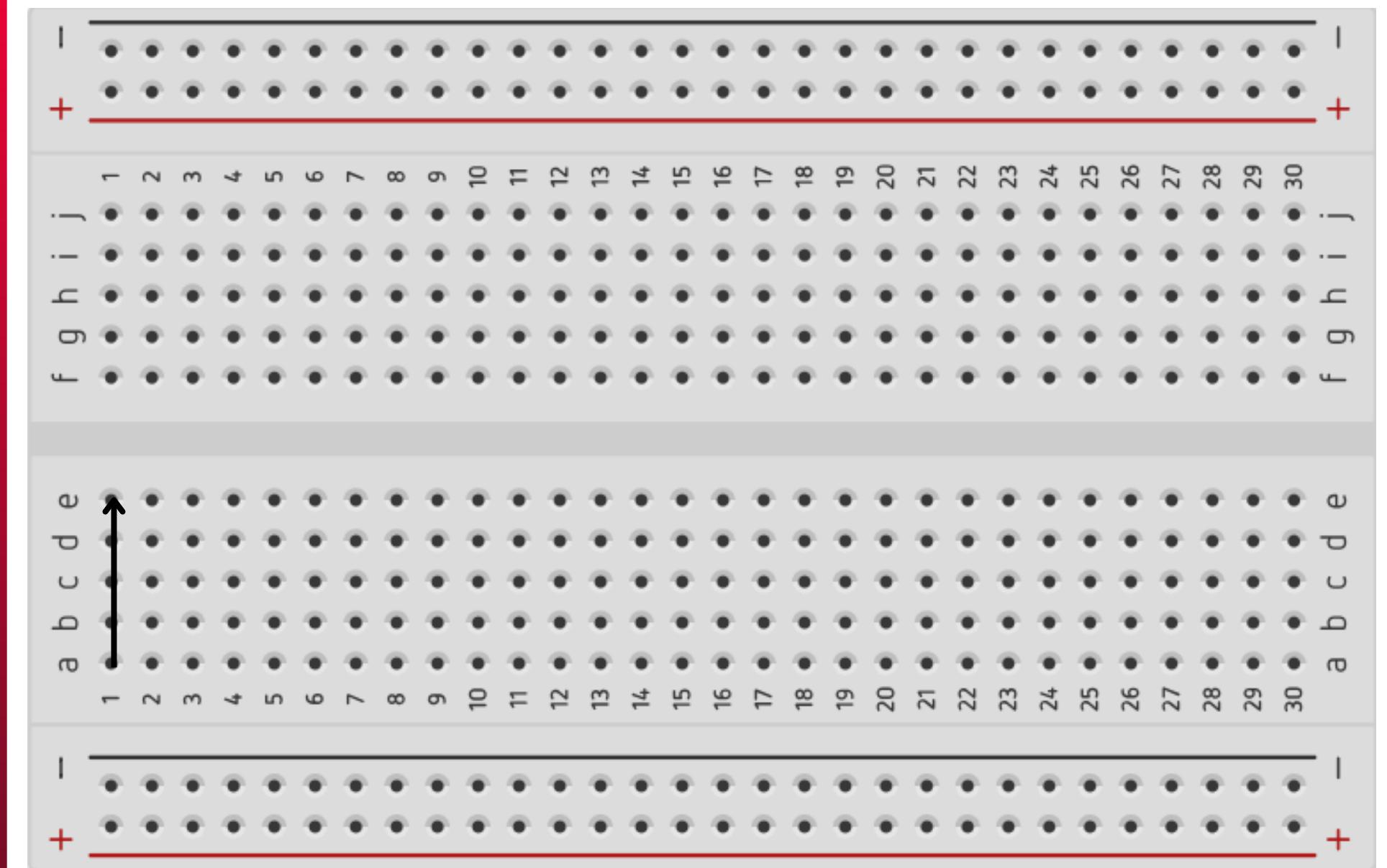


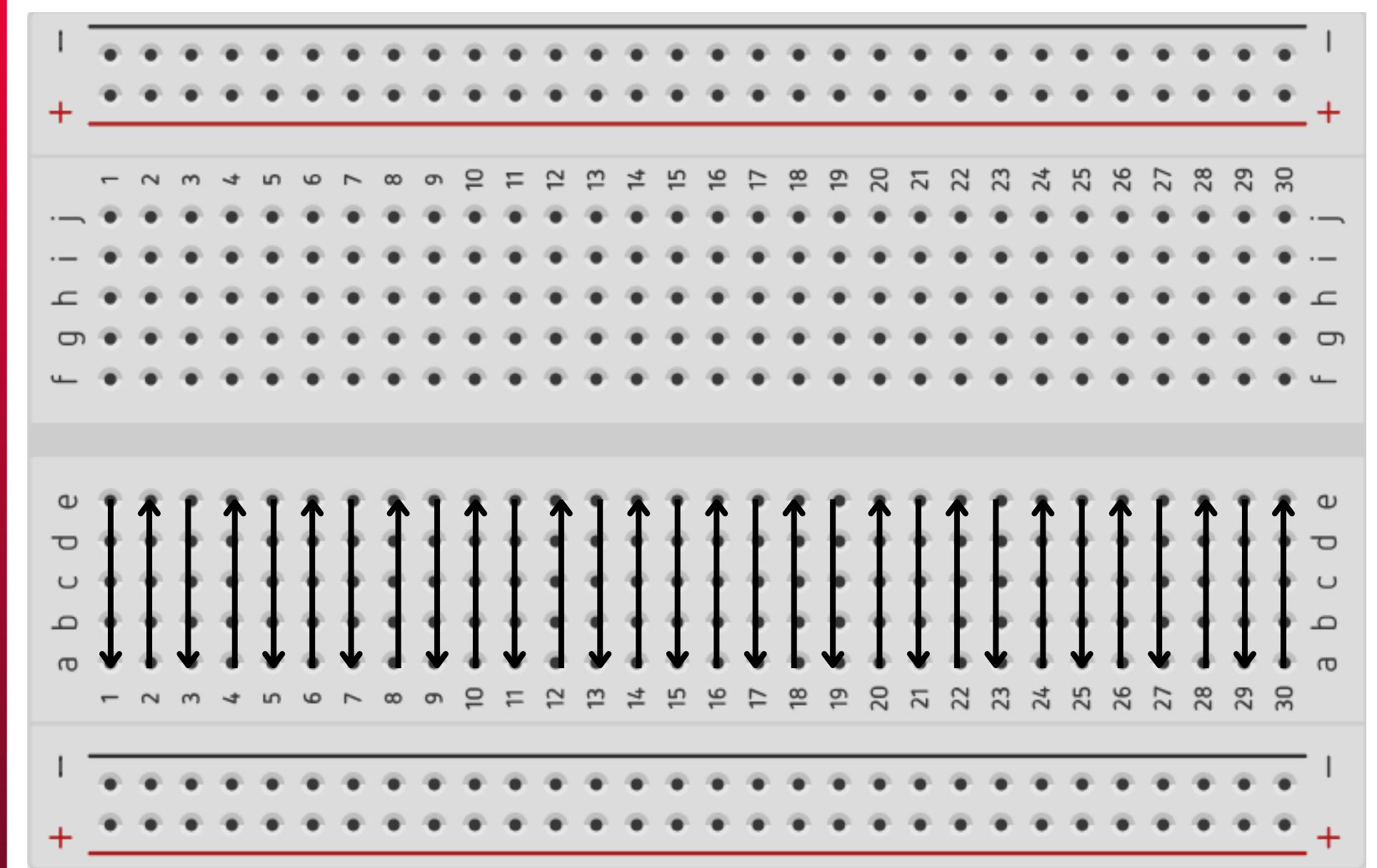


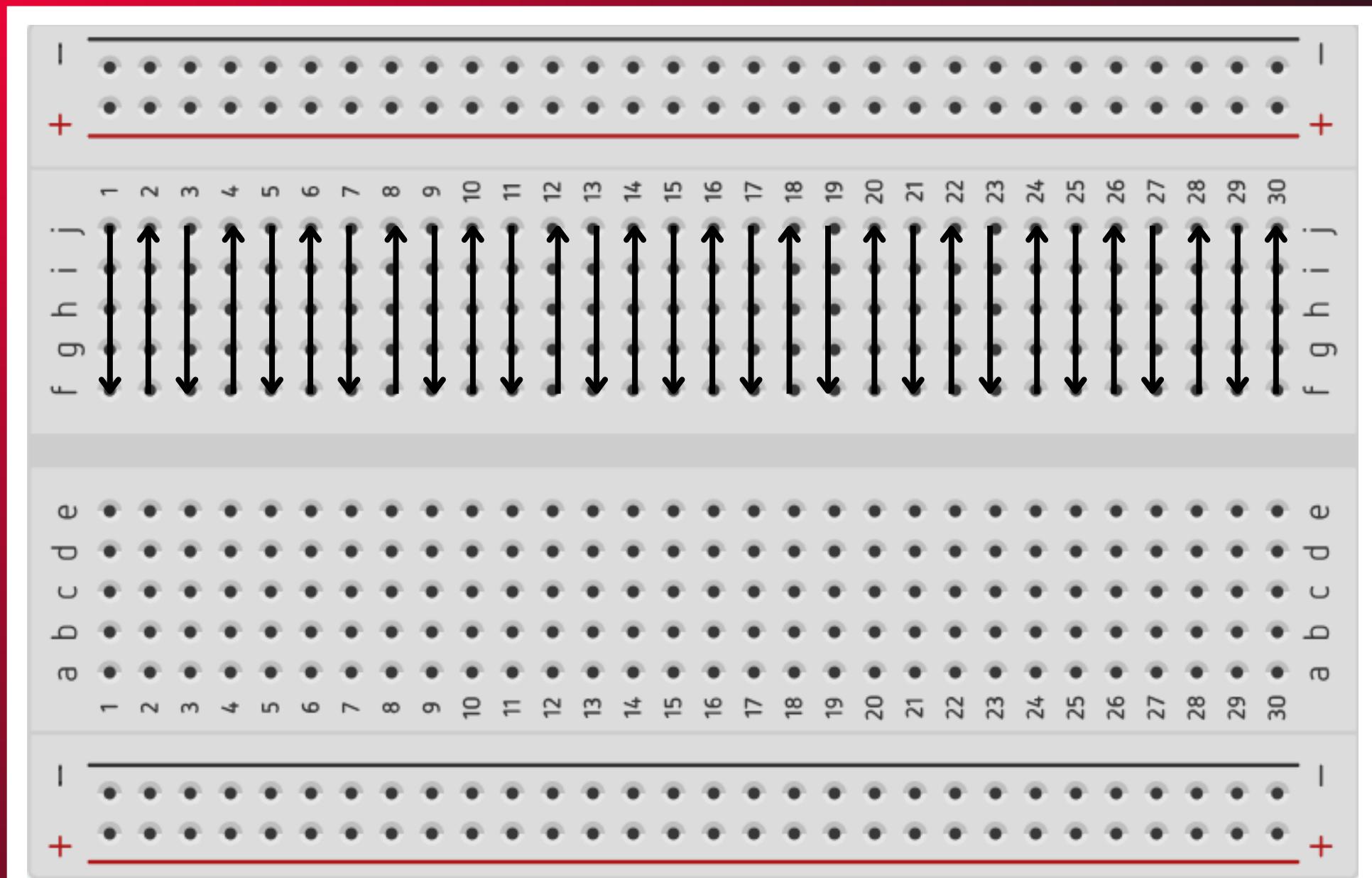


# PERMITE INTERCONEXÃO ENTRE OS FUROS VERTICais

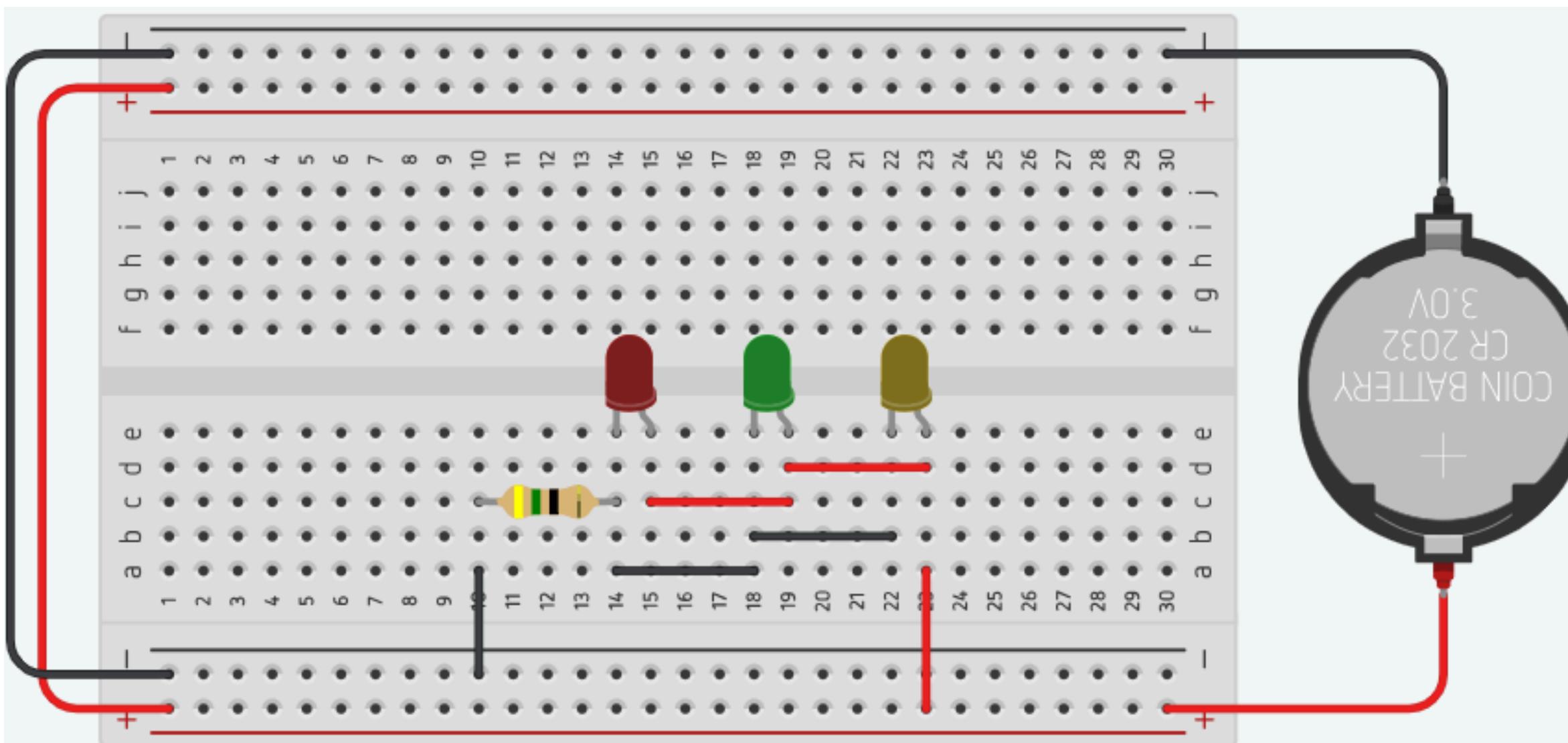


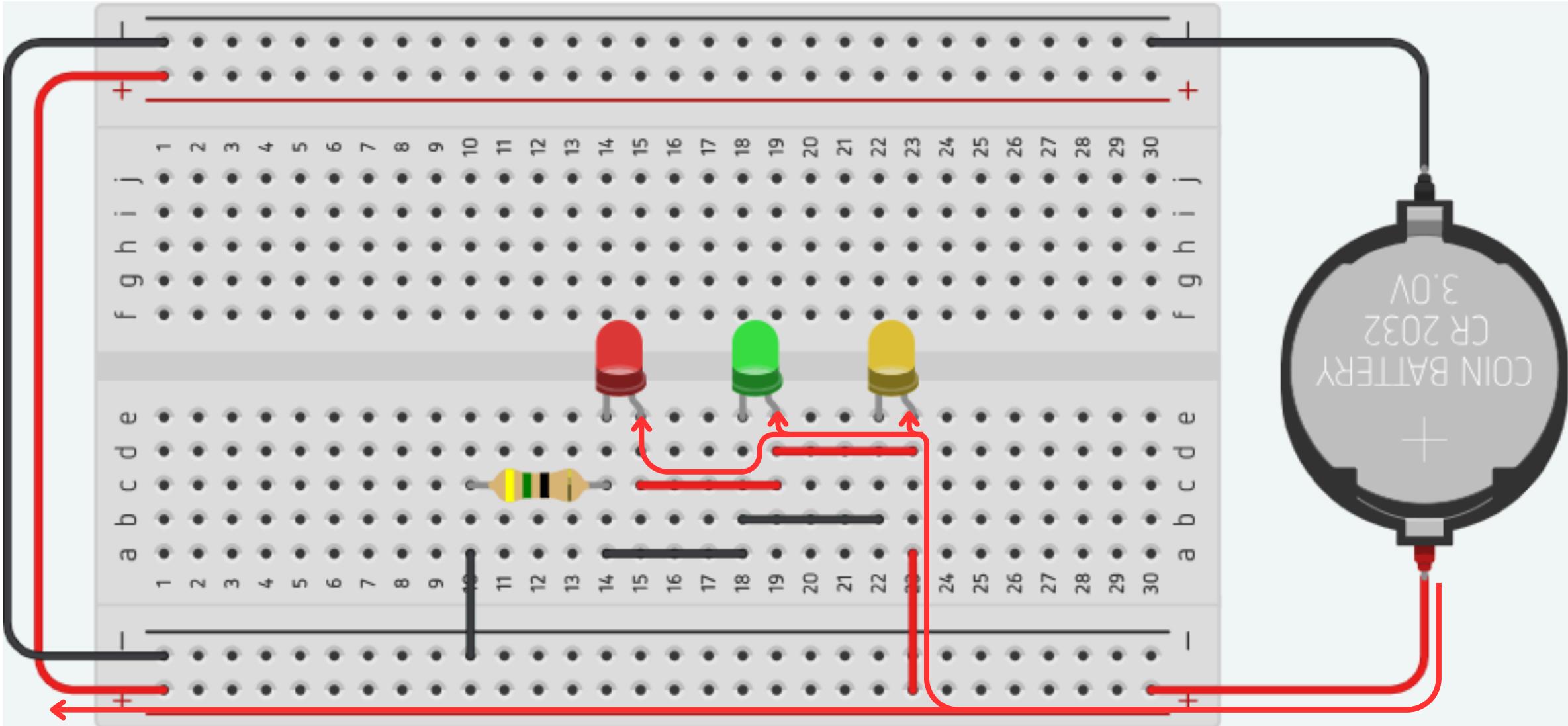


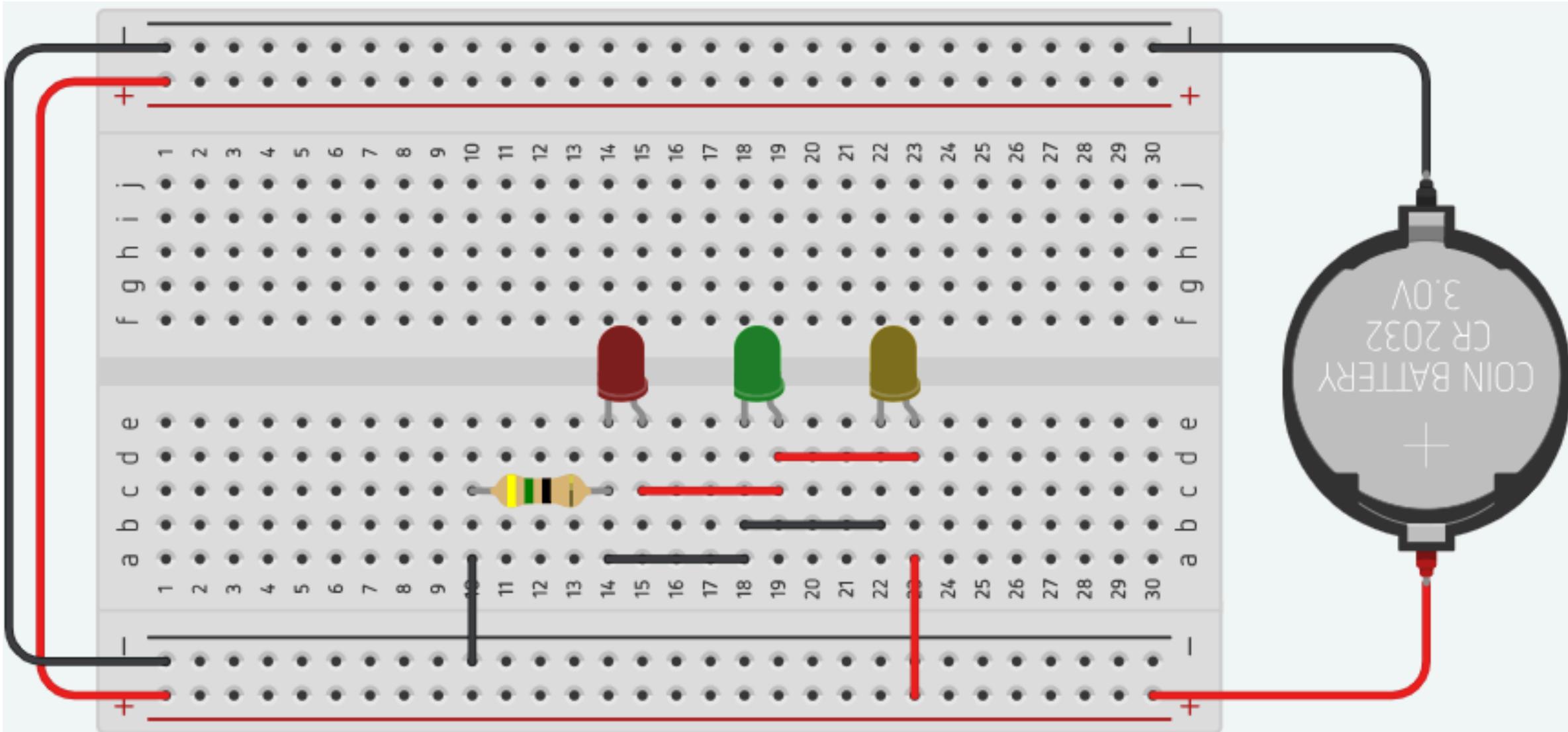


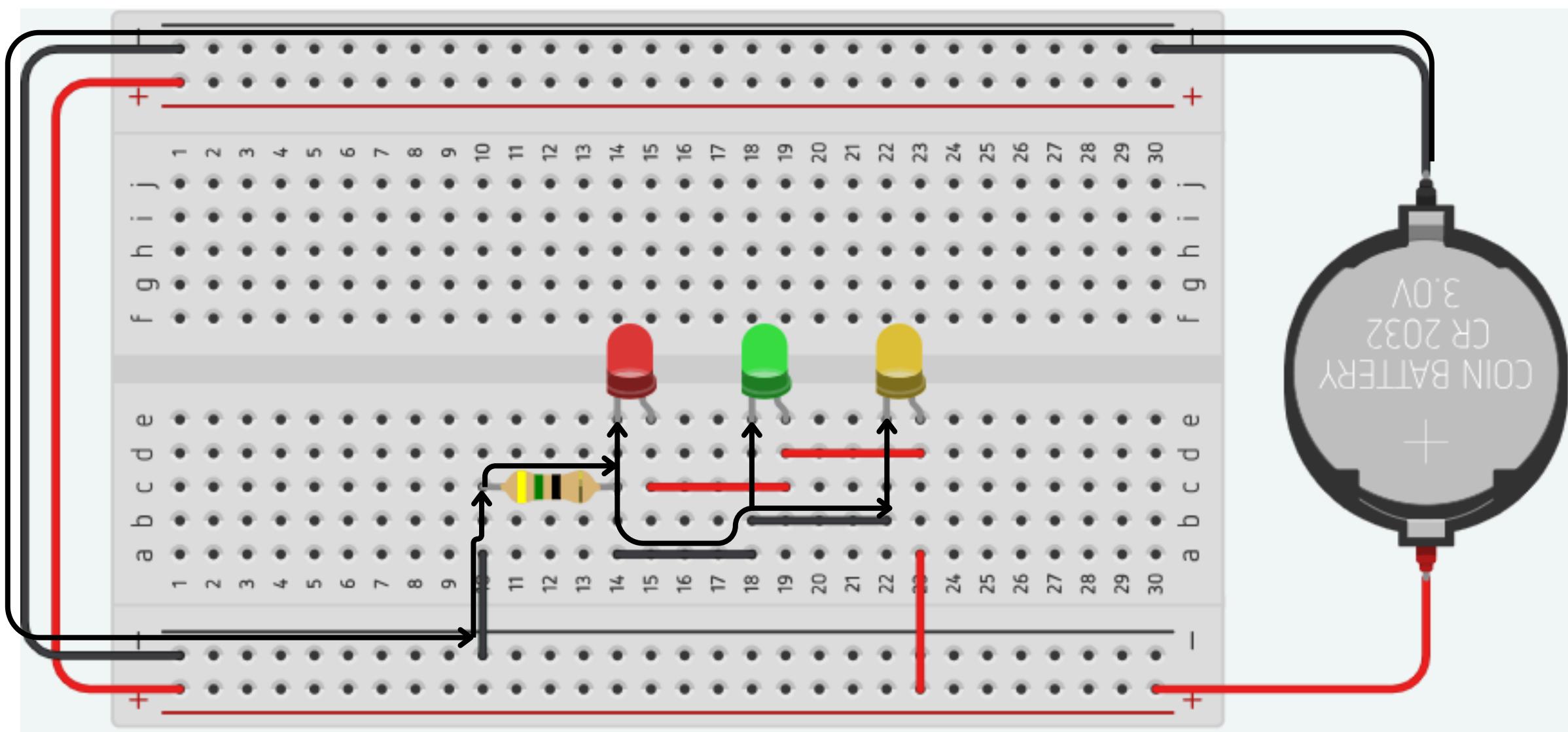


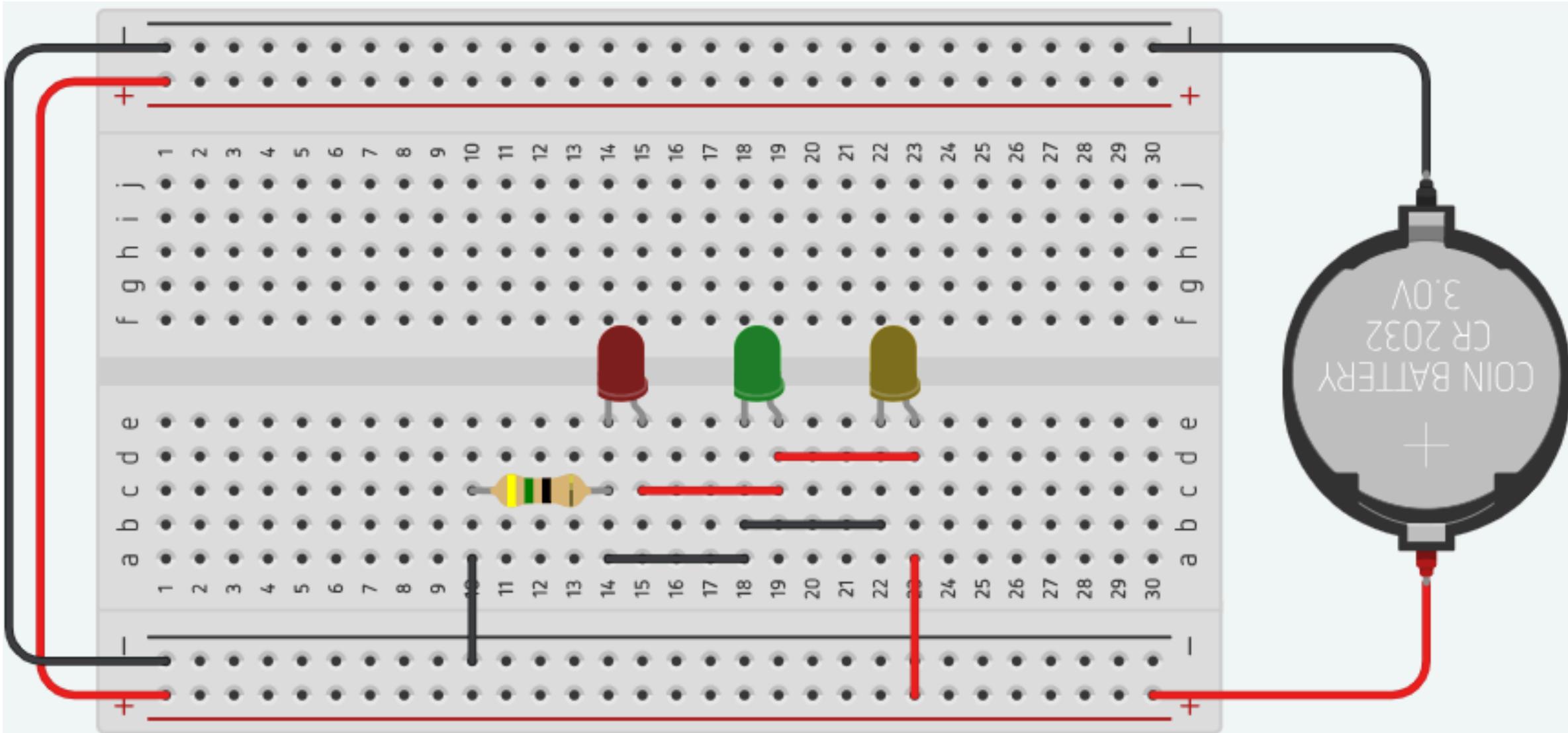
# FACILITA MONTAGEM E PROTOTIPAGEM DE CIRCUITOS

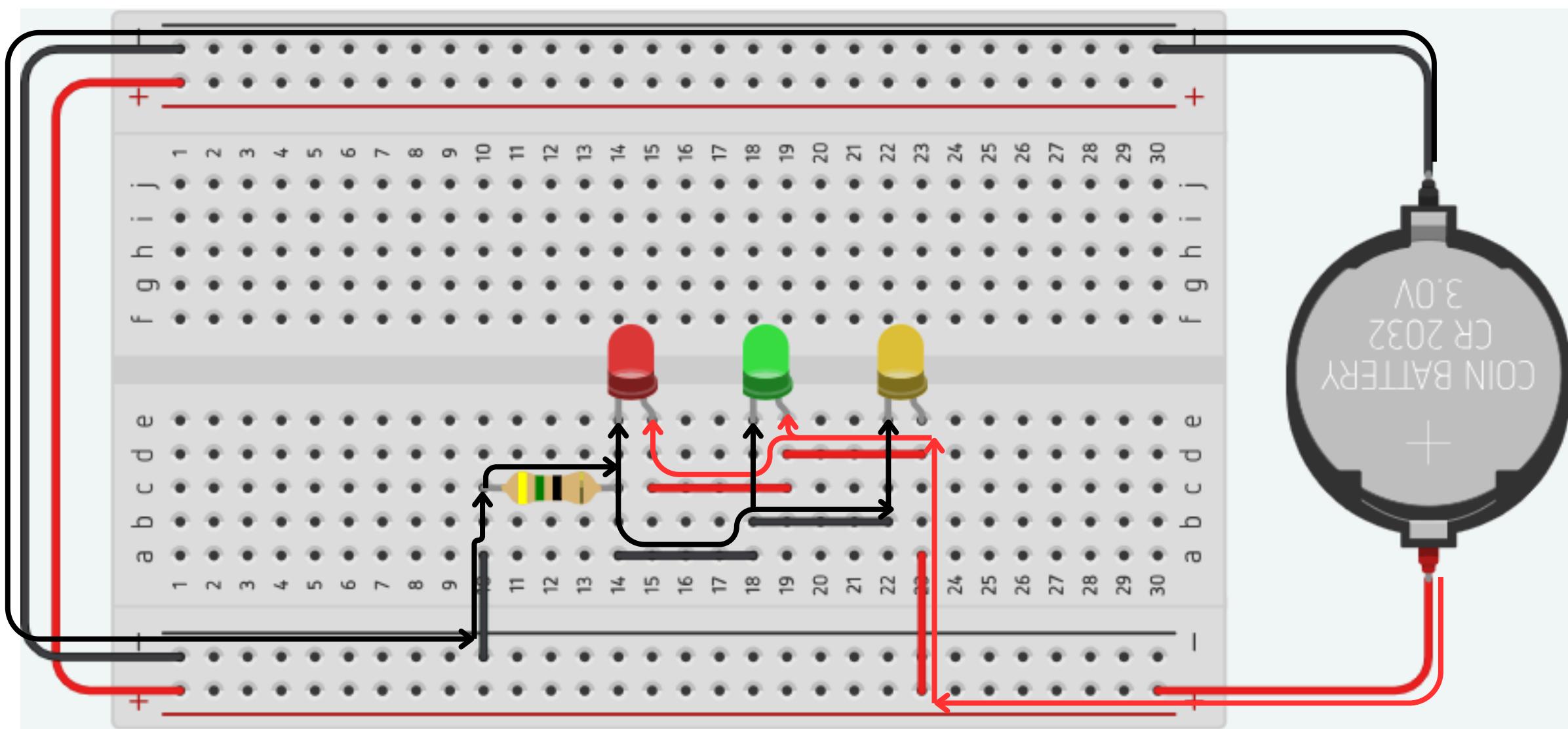


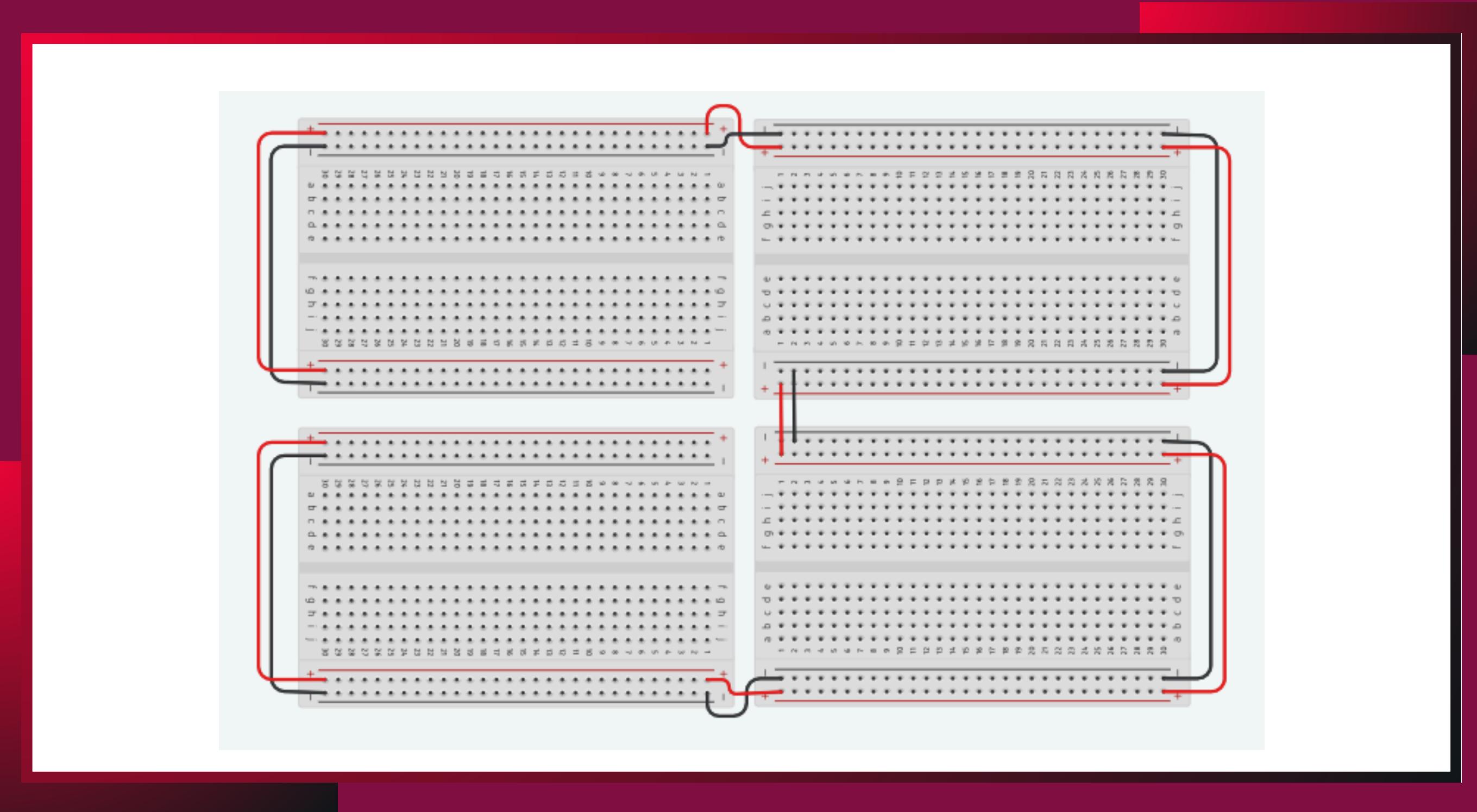












# CONCEITOS FUNDAMENTAIS

## ⚡ CORRENTE ELÉTRICA (I)

- É A QUANTIDADE DE CARGA ELÉTRICA QUE PASSA POR UM PONTO DO CIRCUITO POR SEGUNDO.
- UNIDADE DE MEDIDA DADA EM AMPÈRE (A)

## ⚡ TENSÃO ELÉTRICA (U OU V)

- É A DIFERENÇA DE POTENCIAL ELÉTRICO ENTRE DOIS PONTOS.
- NA ELETRÔNICA, A TENSÃO É A FORÇA QUE EMPURRA OS ELÉTRONS PELO CIRCUITO.
- UNIDADE DE MEDIDA DADA EM VOLT (V)

## brick RESISTÊNCIA ELÉTRICA (R)

- É A OPOSIÇÃO À PASSAGEM DA CORRENTE ELÉTRICA EM UM COMPONENTE OU MATERIAL.
- NA ELETRÔNICA, USAMOS RESISTORES PARA CONTROLAR A CORRENTE E PROTEGER OS COMPONENTES.
- UNIDADE DE MEDIDA DADA EM OHM ( $\Omega$ )



# LEI DE OHM

## 🧠 DEFINIÇÃO

- A LEI DE OHM DESCREVE A RELAÇÃO ENTRE TENSÃO (V), CORRENTE (I) E RESISTÊNCIA (R) EM UM CIRCUITO ELÉTRICO.
- SE A RESISTÊNCIA DE UM CIRCUITO É CONSTANTE, A TENSÃO É DIRETAMENTE PROPORCIONAL À CORRENTE.

## ⟳ FÓRMULAS DERIVADAS:

$$I = \frac{V}{R} \quad V = R \times I$$



## FÓRMULA PRINCIPAL:

$$R = \frac{V}{I}$$

• ONDE:

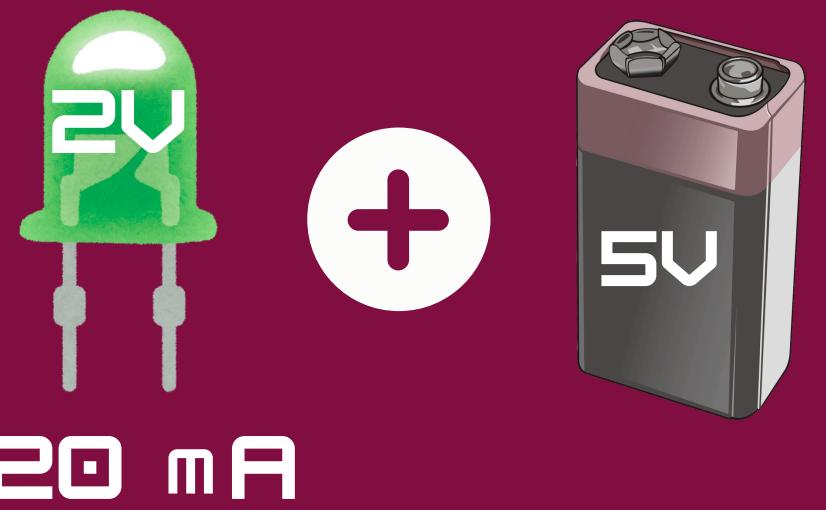
- V = TENSÃO (VOLTAGEM)  
[VOLTS - V]

- I = CORRENTE ELÉTRICA  
[AMPÈRES - A]

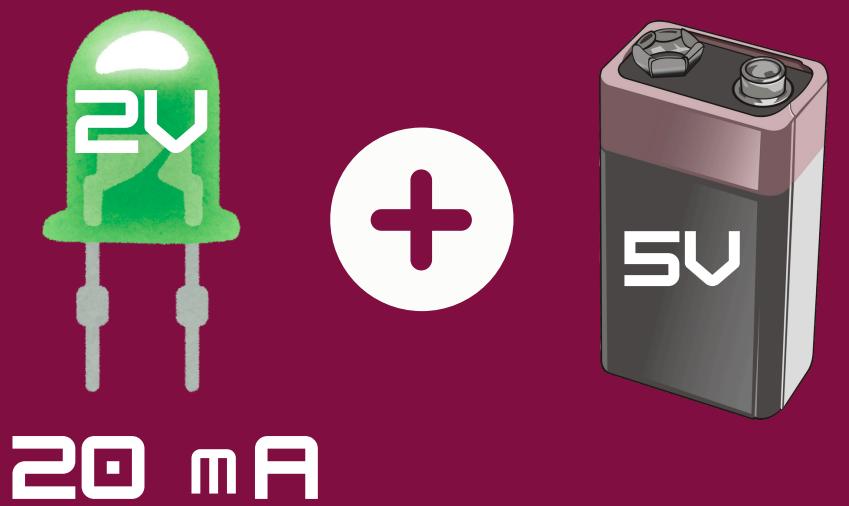
- R = RESISTÊNCIA ELÉTRICA  
[OHMS - Ω]



# DIMENSIONAMENTO DE RESISTORES

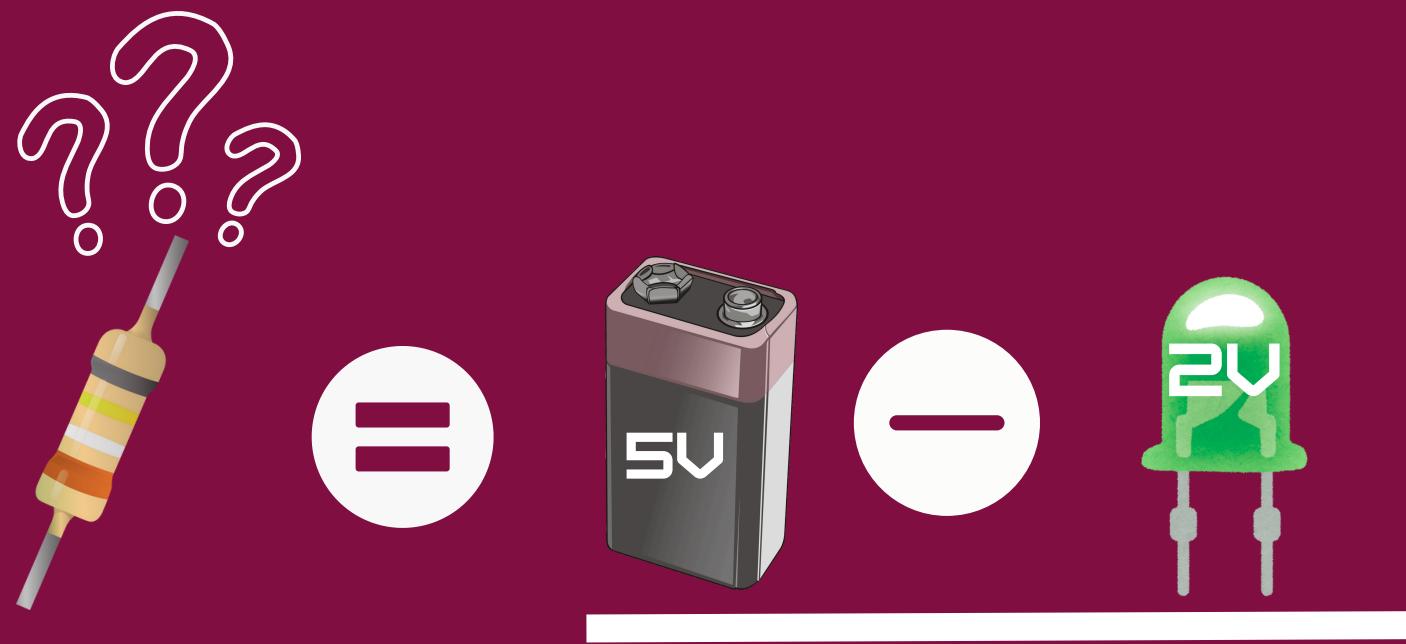


# DIMENSIONAMENTO DE RESISTORES



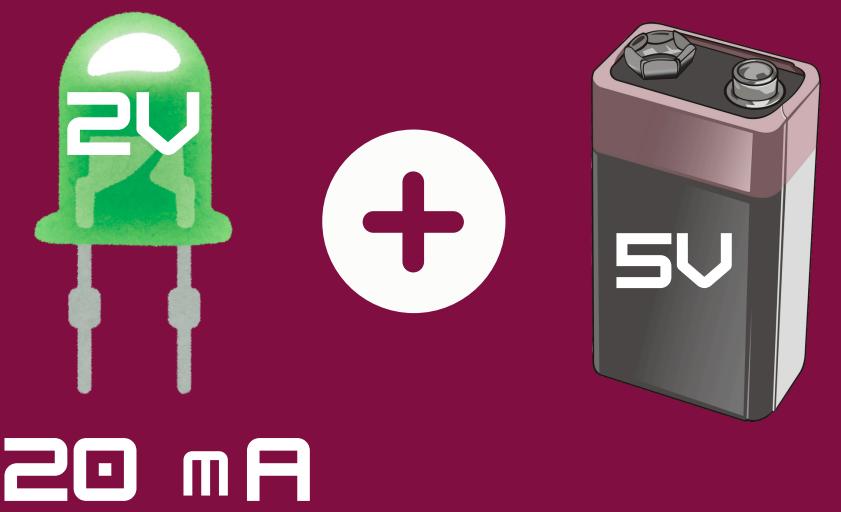
20 mA

$$R = \frac{3}{0.02}$$



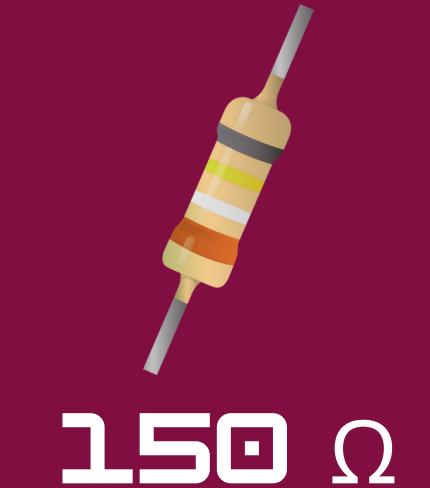
20 mA

# DIMENSIONAMENTO DE RESISTORES

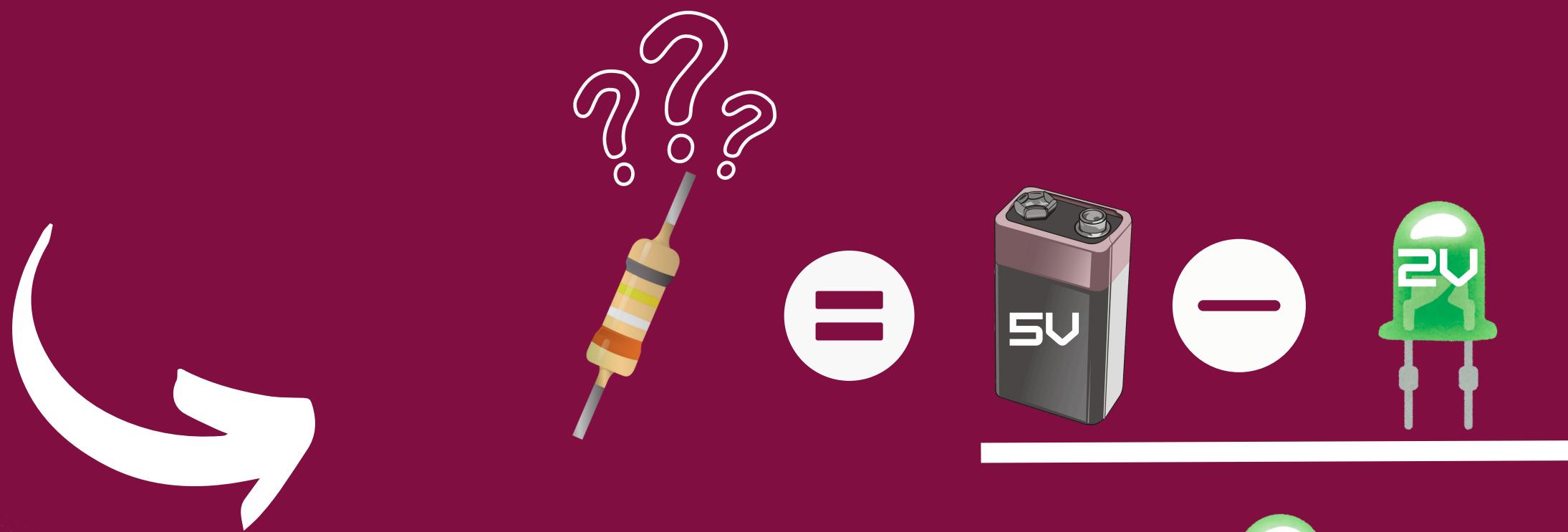


20 mA

$$R = \frac{3}{0.02} \quad \therefore R = 150 \Omega$$

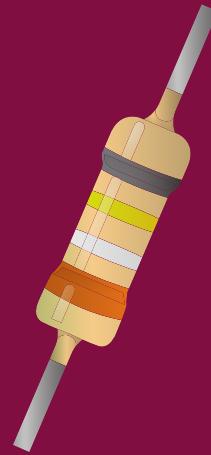
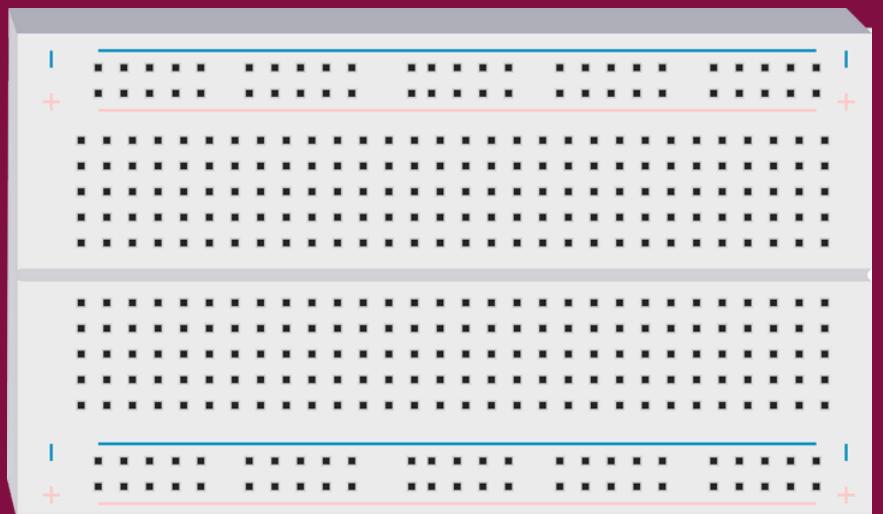
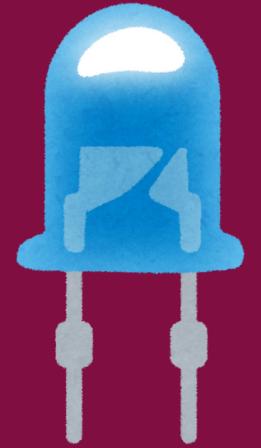


150 Ω



20 mA

# MONTAGEM NA PROTOBOARD



# USO DE CONSTANTES E VARIÁVEIS

## • CONSTANTES

- UM DADO É CONSTANTE QUANDO NÃO SOFRE NENHUMA VARIAÇÃO NO DECORRER DO TEMPO.
- DO INÍCIO AO FIM DO PROGRAMA O VALOR PERMANECE INALTERADO.

## COMO DECLARAR VARIÁVEIS

// Uso da const

```
const int x = 100;
```

// Uso do #define

```
#define X 100
```



# USO DE CONSTANTES E VARIÁVEIS

## • CONSTANTES

- UM DADO É CONSTANTE QUANDO NÃO SOFRE NENHUMA VARIAÇÃO NO DECORRER DO TEMPO.
- DO INÍCIO AO FIM DO PROGRAMA O VALOR PERMANECE INALTERADO.

## COMO DECLARAR CONSTANTES

```
// Uso da const  
// const tipo nome = valor;  
  
const int x = 100;  
  
// Uso do #define  
// #define Substituído Substituto  
#define X 100
```



# CONSTANTES NO ARDUINO

## • CONSTANTES PREDEFINIDAS

- SÃO PALAVRAS RESERVADAS DA LINGUAGEM.
- NÃO PODEM SER USADAS COMO NOMES DE VARIÁVEIS.

## • EXEMPLOS DE CONSTANTES

- TRUE E FALSE
  - REPRESENTAM VERDADEIRO E FALSO.
- HIGH E LOW
  - CONTROLAM AS PORTAS DIGITAIS.
  - HIGH → PORTA ATIVADA (5V).
  - LOW → PORTA DESATIVADA (0V).

## • INPUT E OUTPUT

- USADAS COM A FUNÇÃO PINMODE ( ) PARA CONFIGURAR OS PINOS.
  - INPUT
    - O PINO SERÁ USADO PARA RECEBER SINAIS.
    - EXEMPLO: SENsoRES.
  - OUTPUT
    - O PINO SERÁ USADO PARA ENVIAR SINAIS.
    - EXEMPLO: ACENDER UM LED, ACIONAR UM MOTOR.

# USO DE CONSTANTES E VARIÁVEIS

- **VARIÁVEIS**

- **ESPAÇO NA MEMÓRIA QUE ARMAZENA UM DADO**
- **ESSE DADO PODE SER MODIFICADO EM TEMPO DE EXECUÇÃO**

```
// Tipo nomeVariável;  
int valor_x = 10;  
char letra = 'a'  
int valor_x = 30;  
char letra = 'd'
```



Tipo	Definição
<b>void</b>	Indica tipo indefinido. Usado geralmente para informar que uma função não retorna nenhum valor.
<b>boolean</b>	Os valores possíveis são true (1) e false (0). Ocupa um byte de memória.
<b>char</b>	Ocupa um byte de memória. Pode ser uma letra ou um número. A faixa de valores válidos é de -128 a 127.
<b>unsigned char</b>	O mesmo que o <b>char</b> , porém a faixa de valores válidos é de 0 a 255.
<b>byte</b>	Ocupa 8 bits de memória. A faixa de valores é de 0 a 255.
<b>int</b>	Armazena números inteiros e ocupa 16 bits de memória (2bytes). A faixa de valores é de -32.768 a 32.767.
<b>unsigned int</b>	O mesmo que o <b>int</b> , porém a faixa de valores válidos é de 0 a 65.535.
<b>word</b>	O mesmo que um <b>unsigned int</b> .
<b>long</b>	Armazena números de até 32 bits (4 bytes). A faixa de valores é de -2.147.483.648 até 2.147.483.647.
<b>unsigned long</b>	O mesmo que o <b>long</b> , porém a faixa de valores é de 0 até 4.294.967.295.
<b>short</b>	Armazena número de até 16 bits (2 bytes). A faixa de valores é de -32.768 até 32.767.
<b>float</b>	Armazena valores de ponto flutuante (com vírgula) e ocupa 32 bits (4 bytes) de memória. A faixa de valores é de -3.4028235E+38 até 3.4028235E+38
<b>double</b>	O mesmo que o <b>float</b> .



# LAÇOS CONDICIONAIS

- LAÇO CONDICIONAL IF ()
  - EXECUTA UM BLOCO DE CÓDIGO CASO VALOR SEJA VERDADEIRO
- LAÇO CONDICIONAL ELSE ()
  - EXECUTA UM BLOCO DE CÓDIGO CASO O IF () SEJA FALSO

```
if (condição) {  
    // bloco de código  
}  
  
else {  
    // bloco de código  
}
```



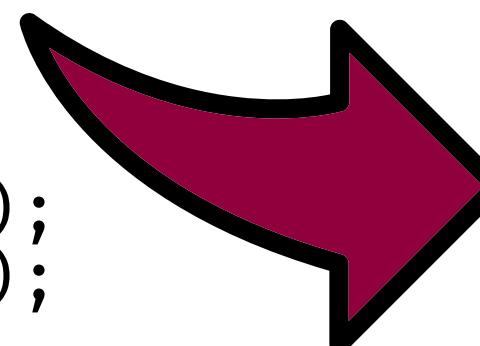
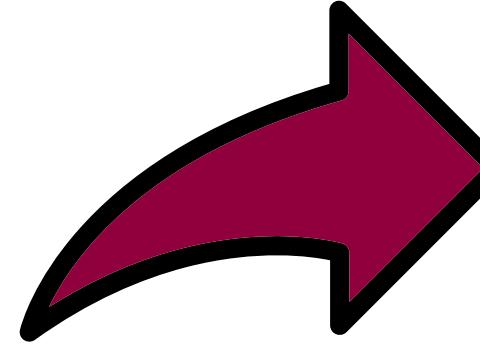
```
void setup() {
  pinMode(10, OUTPUT)
  pinMode(13, OUTPUT)
  pinMode(A2, INPUT)
}

void loop()  {

  if ((analogRead(A2) * 5.0 / 1023.0)
      * 100.0) < 50)
  {
    digitalWrite(13, HIGH);
    digitalWrite(10, HIGH);
  }
  else
  {
    digitalWrite(13, LOW);
    digitalWrite(10, LOW);
  }
  delay(1000);
}
```

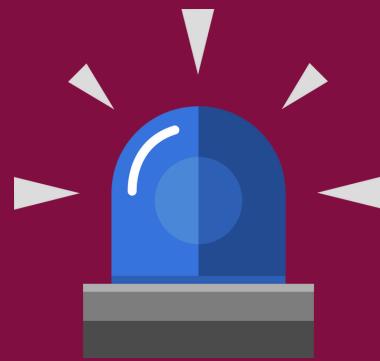
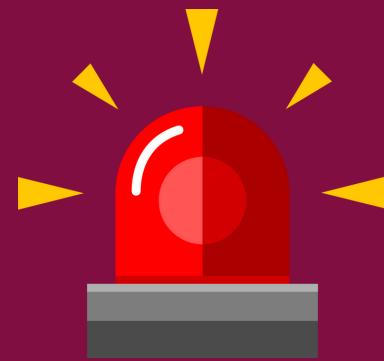


```
void setup() {  
    pinMode(10, OUTPUT)  
    pinMode(13, OUTPUT)  
    pinMode(A2, INPUT)  
}  
  
void loop() {  
  
    if ((analogRead(A2) * 5.0 / 1023.0)  
        * 100.0) < 50)  
    {  
        digitalWrite(13, HIGH);  
        digitalWrite(10, HIGH);  
    }  
    else  
    {  
        digitalWrite(13, LOW);  
        digitalWrite(10, LOW);  
    }  
    delay(1000);  
}
```



```
#define TEMPO_CICLO 1000  
const int led = 13;  
const int motor = 10;  
const int sensorTemp = A2;  
const int tempSeguranca = 50;  
  
void setup() {  
    pinMode(led, OUTPUT)  
    pinMode(motor, OUTPUT)  
    pinMode(sensorTemp, INPUT)  
}  
  
void loop() {  
    int L = analogRead(sensorTemp);  
    float T = (L * 5.0 / 1023.0) * 100.0;  
  
    if (T < tempSeguranca) {  
        digitalWrite(led, HIGH);  
        digitalWrite(motor, HIGH);  
    }  
    else {  
        digitalWrite(led, LOW);  
        digitalWrite(motor, LOW);  
    }  
    delay(TEMPO_CICLO);  
}
```

# SIRENE LED



# OBRIGADO ! !



EQUIPE DE  
TRAINING :

DIRETOR DE TREINAMENTO :  
PEDRO S. CONCEIÇÃO