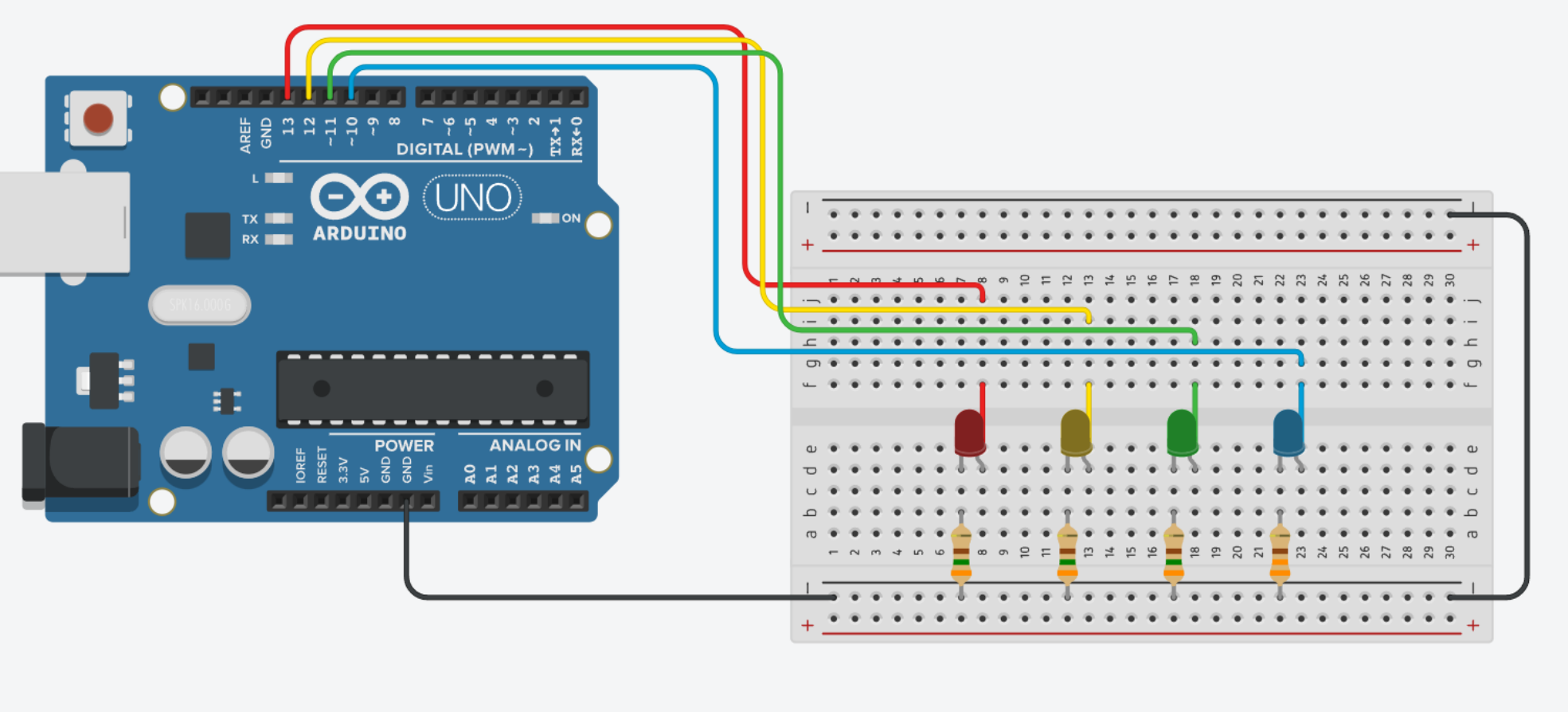


EP03

Exercício 1 -

Circuito



Código

```
// 853355

// Definir LEDs
int RLed = 13;
int YLed = 12;
int GLed = 11;
int BLed = 10;

void setup()
{
  pinMode (RLed, OUTPUT);
  pinMode (YLed, OUTPUT);
  pinMode (GLed, OUTPUT);
  pinMode (BLed, OUTPUT);
}

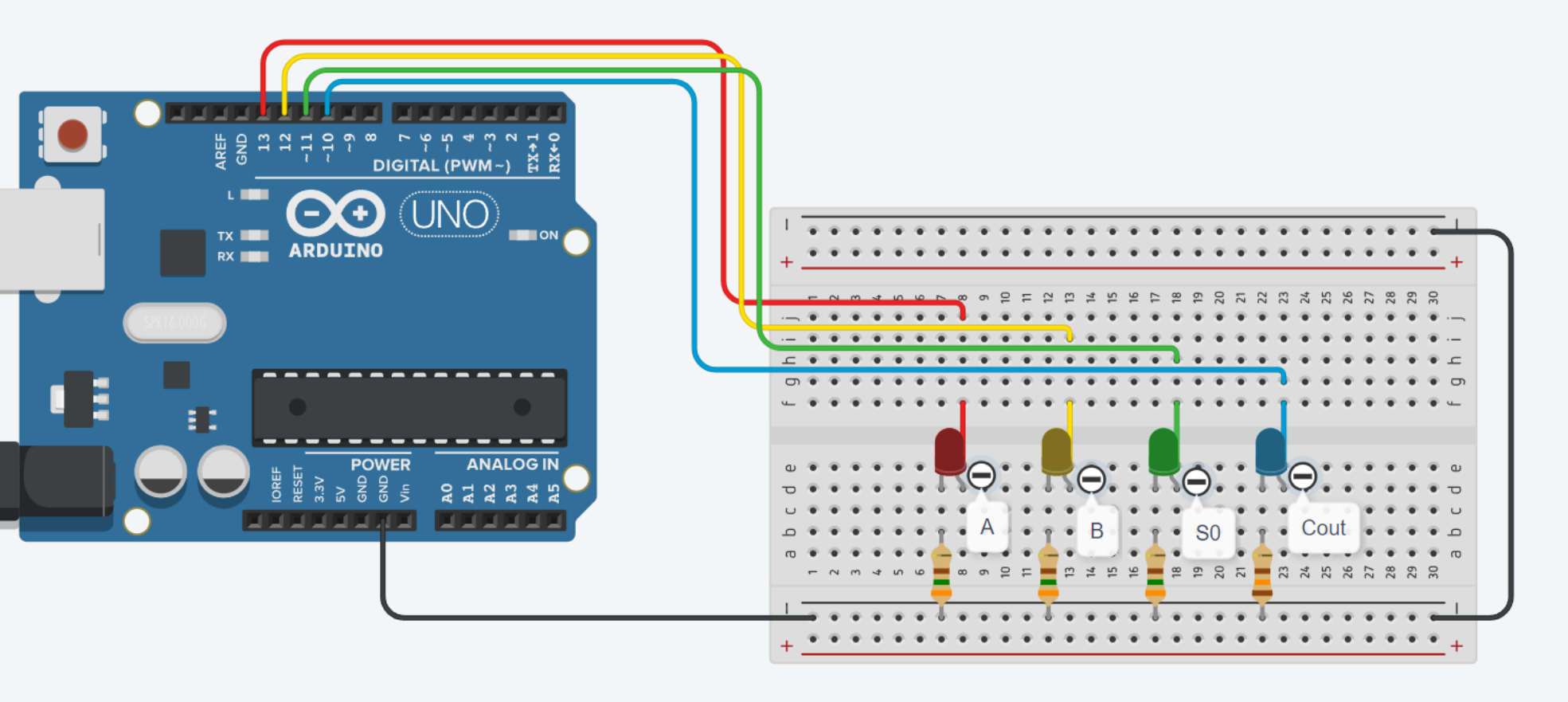
void loop()
{
  cycle (RLed, 3);
  cycle (GLed, 4);
  cycle (YLed, 2);
}

void cycle (int Led, int T)
{
  for (int i = 0; i < T; i++)
  {
    LedC(Led);
  }
  digitalWrite(Led, LOW);
}
```

```
void LedC (int Led)
{
    digitalWrite(BLed, HIGH);
    digitalWrite(Led, HIGH);
    delay(1000); // 1s
    digitalWrite(BLed, LOW);
    delay(1000); // 1s
}
```

Exercício 2 -

Circuito



Código

```
// 853355

// Definir dados

// Leds
int RLed = 13;
int YLed = 12;
int GLed = 11;
int BLed = 10;

// Entradas
char A = '\0';
char B = '\0';
char OP = '\0';

// Status
bool RL = false;
bool YL = false;
bool GL = false;
bool BL = false;

void setup()
{
    Serial.begin(9600);

    pinMode(RLed, OUTPUT);
    pinMode(YLed, OUTPUT);
    pinMode(GLed, OUTPUT);
```

```

pinMode(BLed, OUTPUT);
}

void loop()
{
    if (Serial.available() > 0)
    {
        // Ler input
        String input = Serial.readString();

        // Checar input
        if (input.length() != 3)
        {
            Serial.println ("ERRO: Input invalido!");
        }
        else
        {
            // Limpar dados
            RL = false;
            YL = false;
            GL = false;
            BL = false;

            // Atribuir valores
            A = input[0];
            B = input[1];
            OP = input[2];

            // Checar input
            if (A > '1' || B > '1' || OP > '3' || A < '0' || B < '0' || OP < '0')
            {
                Serial.println("ERRO: Input invalido!");
            }
            else
            {
                // Operar
                ULA (A, B, OP);

                // Mostrar LEDs
                print();
            }
        }
    }
}

void ULA (char A, char B, char OP)
{
    // "Ler" A e B
    if (A == '1')
    {
        RL = true;
    }
    if (B == '1')
    {
        YL = true;
    }

    switch (OP) // "MUX"
    {
        case '0':    // AND
            GL = (RL && YL);
            break;

        case '1':    // OR

```

```
        GL = (RL || YL);
    break;

    case '2':    // NOT A
        GL = (!RL);
    break;

    case '3':    // SUM
        if (RL && YL)
        {
            BL = true;
        }
        else if (RL || YL)
        {
            GL = true;
        }
    break;
}

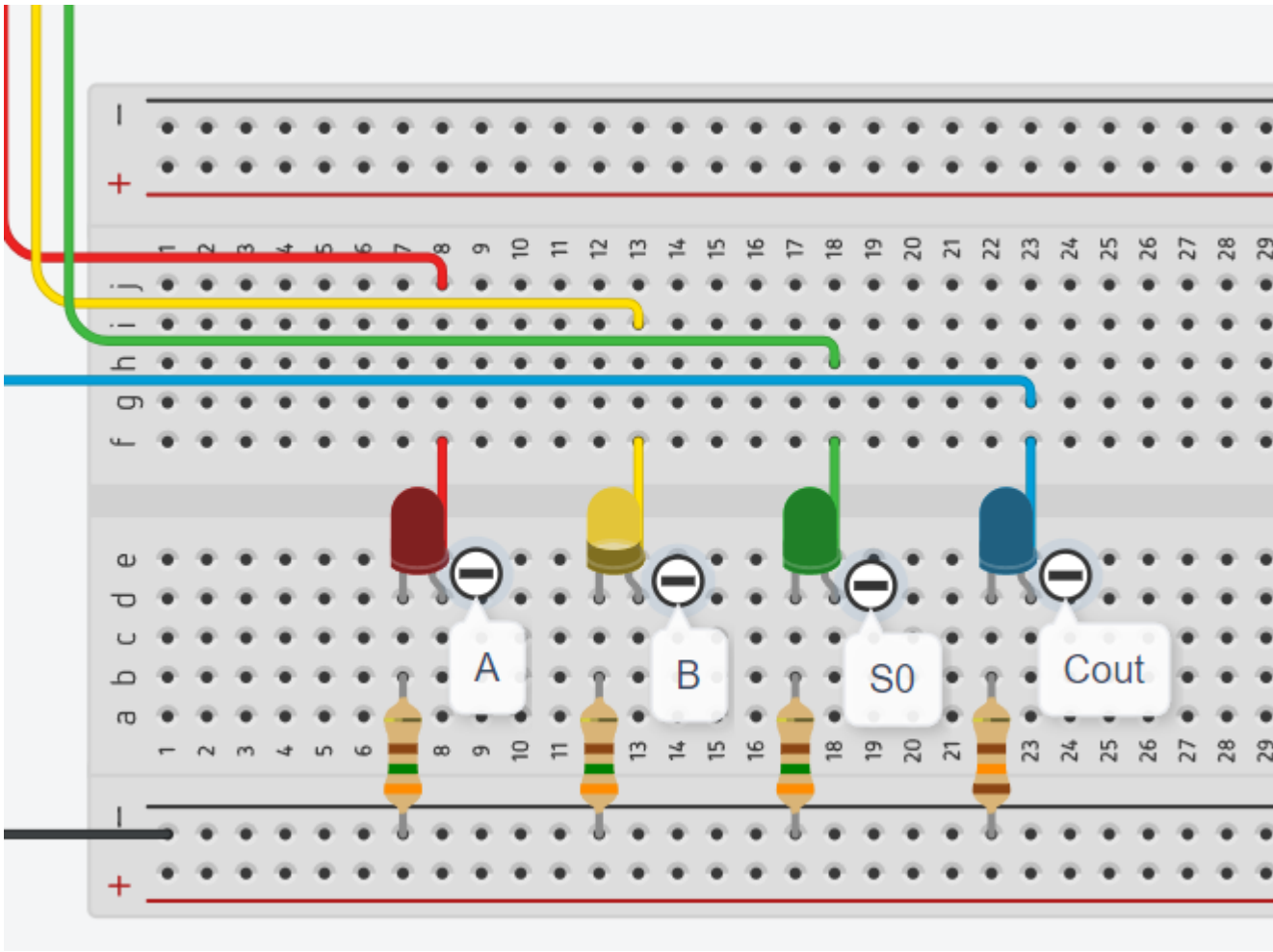
// Controlar LEDs
void print ()
{
    digitalWrite (RLed, RL);
    digitalWrite (YLed, YL);
    digitalWrite (GLed, GL);
    digitalWrite (BLed, BL);
}
```

Tabela

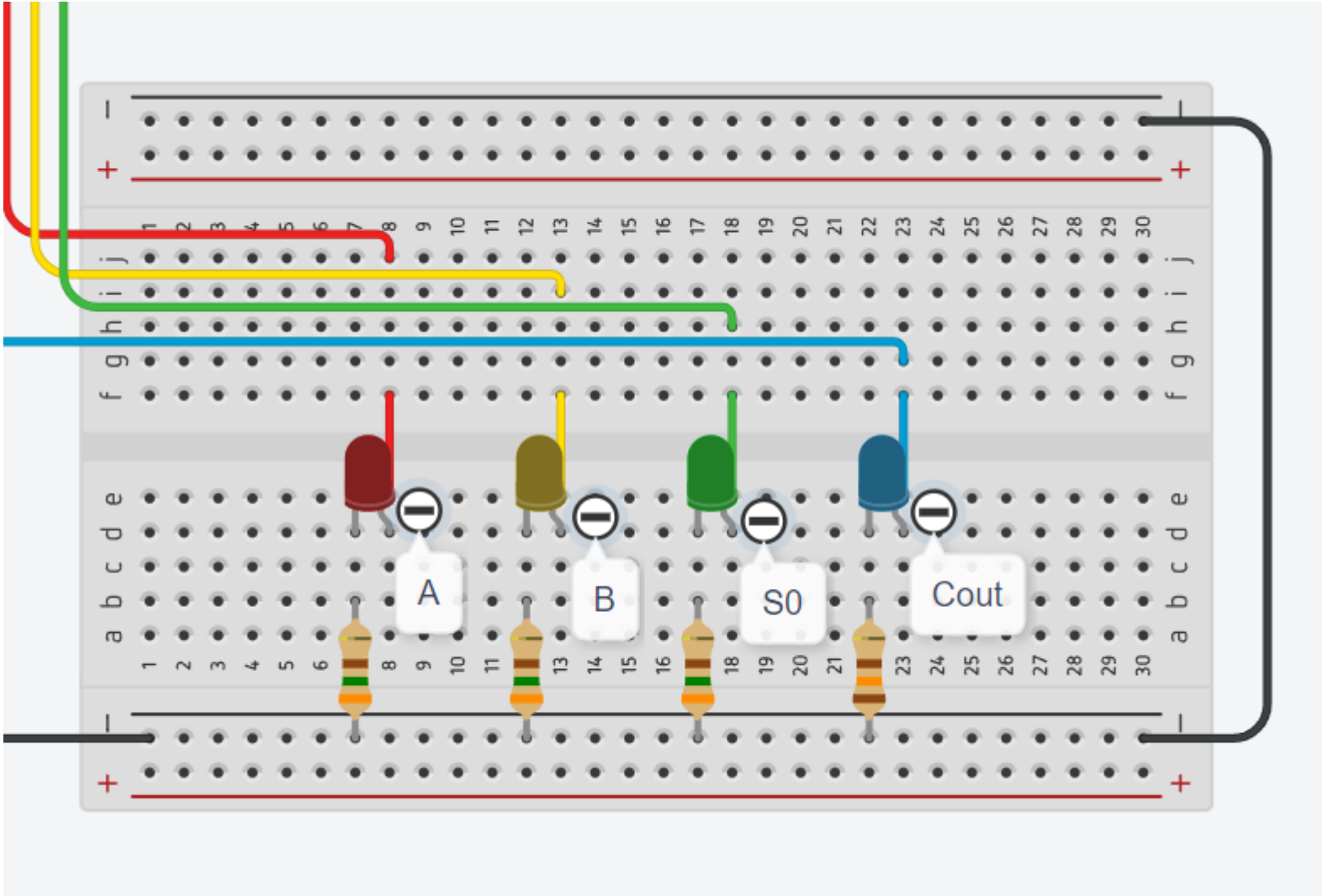
Instrução	Binário	Hexa	Resultado
AND (A, B)	0 1 00	0x4	0
OR (A, B)	0 0 01	0x1	0
SOMA (A, B)	1 1 11	0xF	0
NOT (A, B)	0 1 10	0x6	1
AND (B, A)	1 1 00	0xC	1

Testes

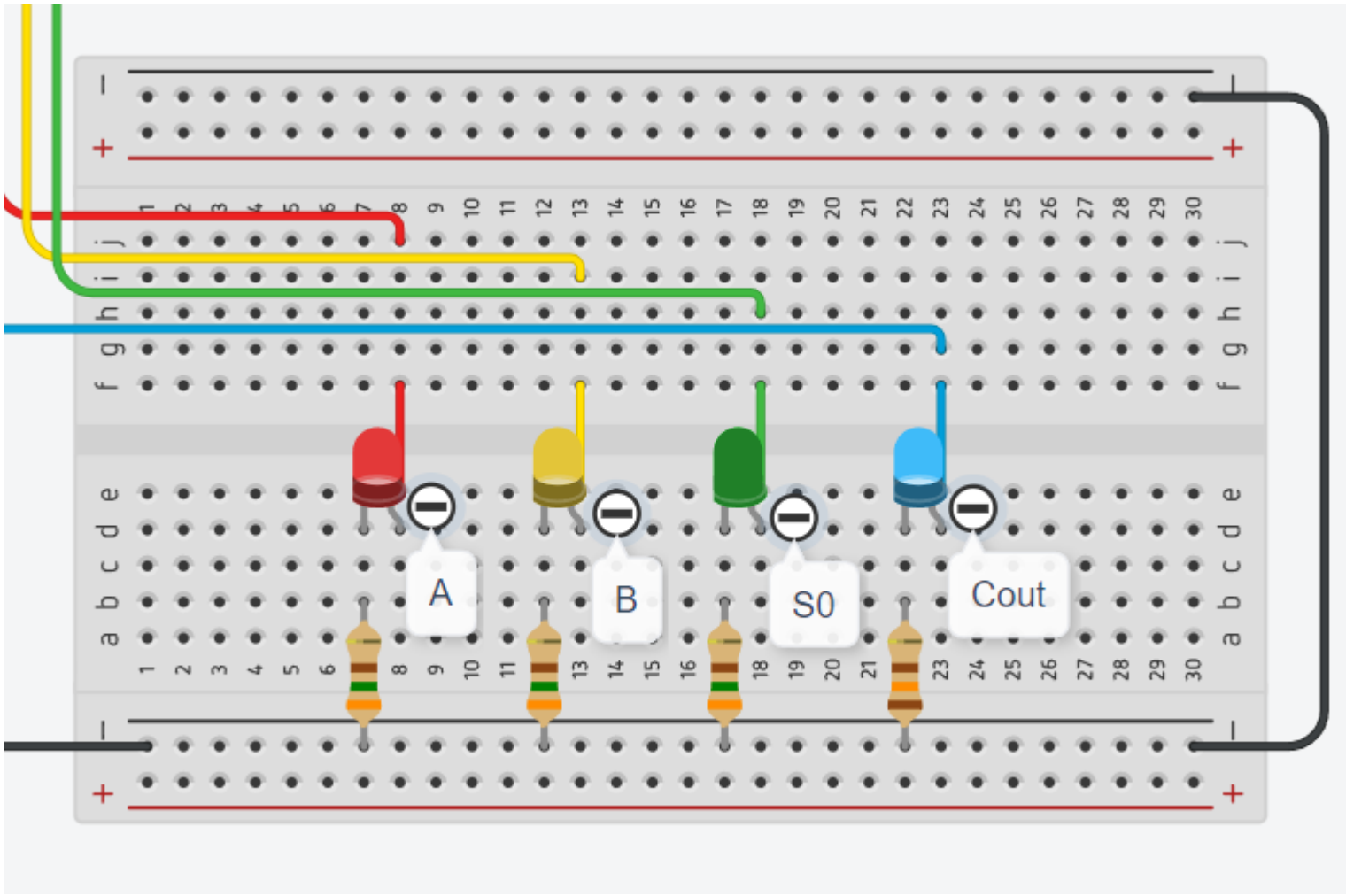
Teste 1



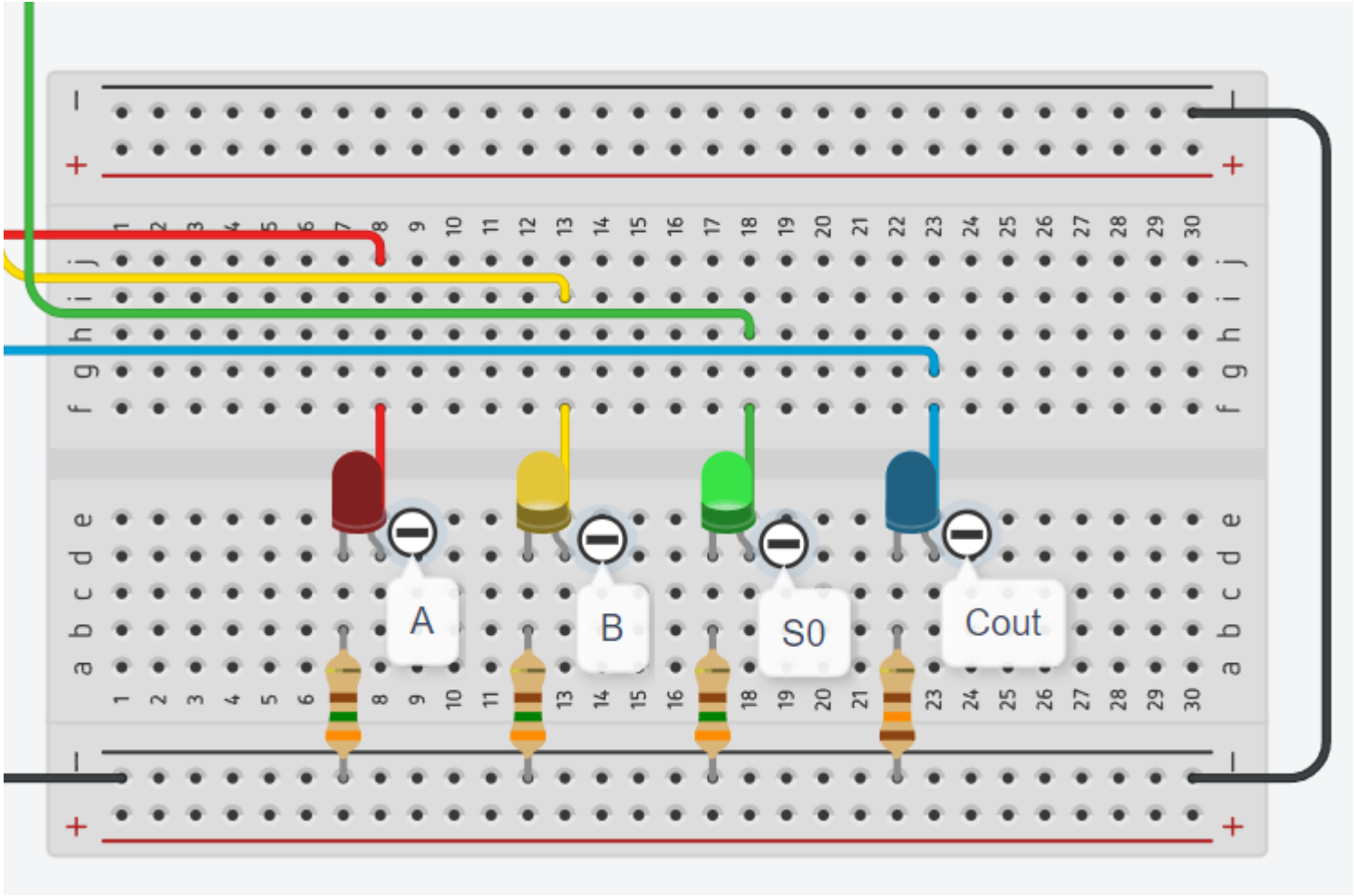
Teste 2



Teste 3



Teste 4



Teste 5

