

Unoverse: Estudante

Bem-vindo(a)!



Ligar e desligar um LED no Tinkercad

O que vamos aprender nessa aula?

- o que é um atuador e um sensor;
- o que é digital e analógico;
- o conceito de variável do tipo int;
- os comandos de leitura e escritura de sensores e atuadores, respectivamente;



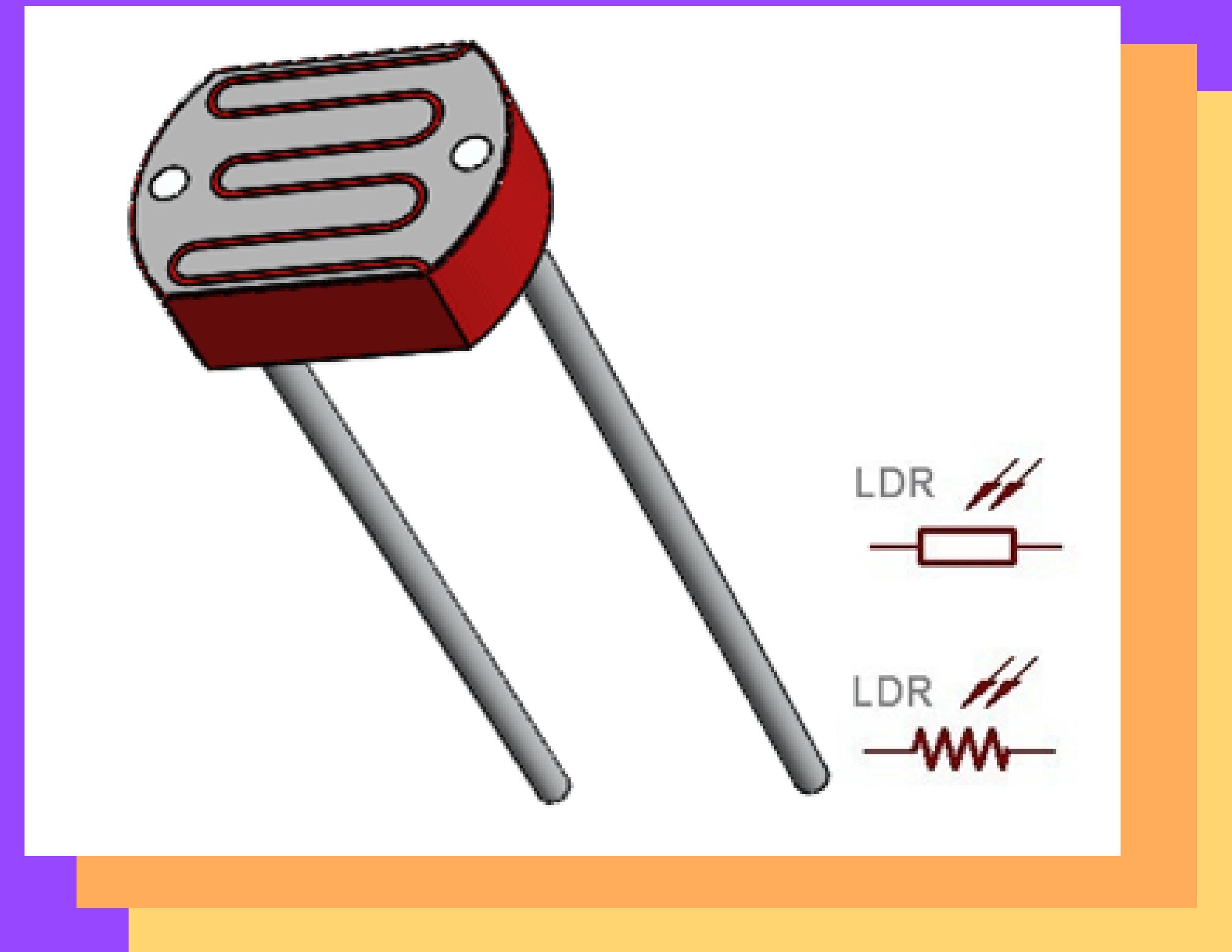
Passo 1

Entenda o LED e o resistor

LED -> um componente eletrônico que funciona como um atuante, recebendo informações do computador e realizando uma tarefa, nesse caso, acendendo.



Sensor -> um componente eletrônico que recebe algum estímulo do exterior e passa ao computador pelo Arduino, sendo semelhante a “informante”. Exemplo: sensor de luminosidade.



Tipo de sinais que os componentes eletrônicos podem emitir:

Digital -> possui dois estados - o ligado (1) e o desligado (0) - e está conectado na parte digital da placa.

Analógico -> pode ter valores variados, sinais eletrônicos com valores que variam de 0 a 1023, já que, no caso do sensor de luminosidade, por exemplo, possui vários tons.

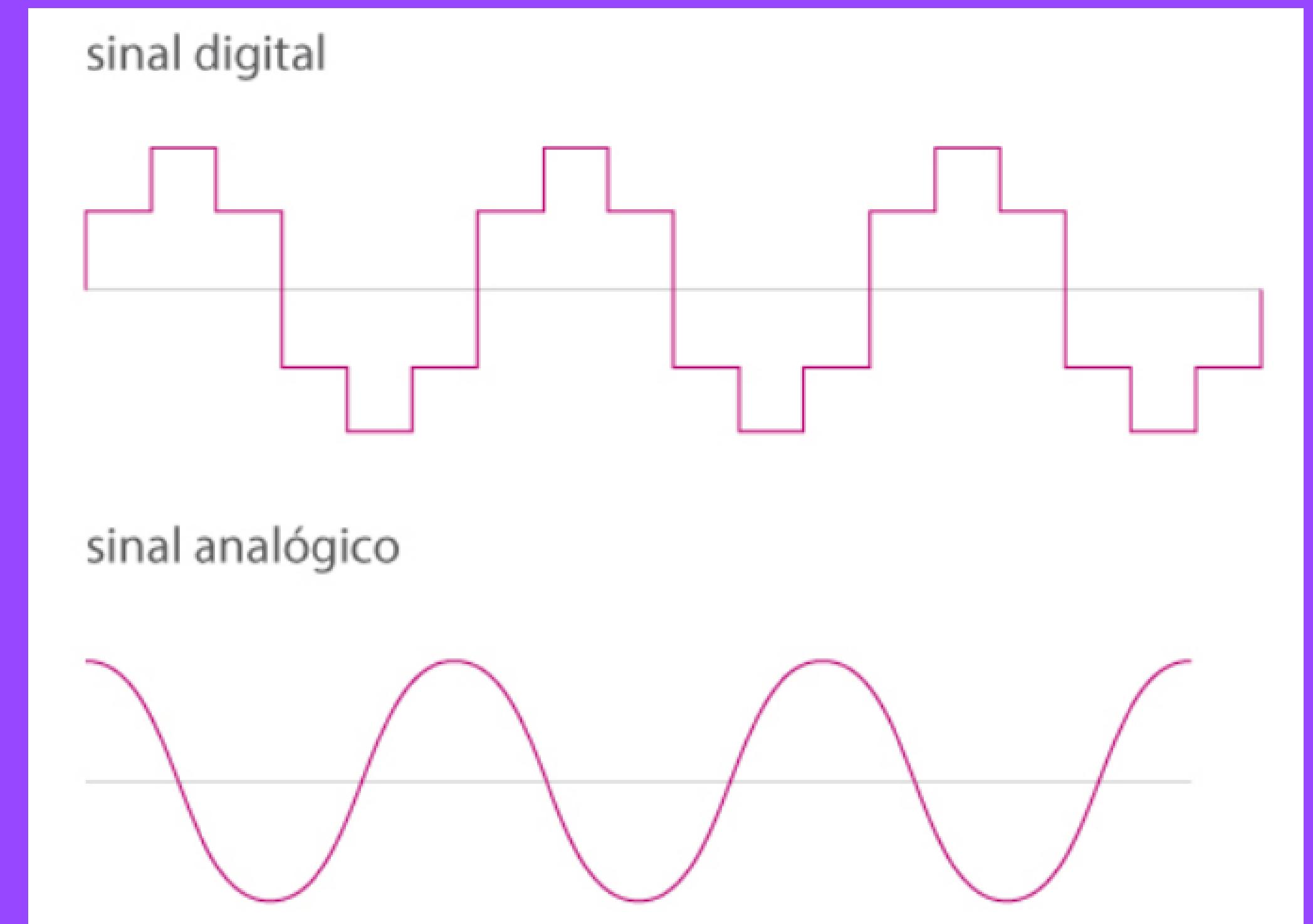
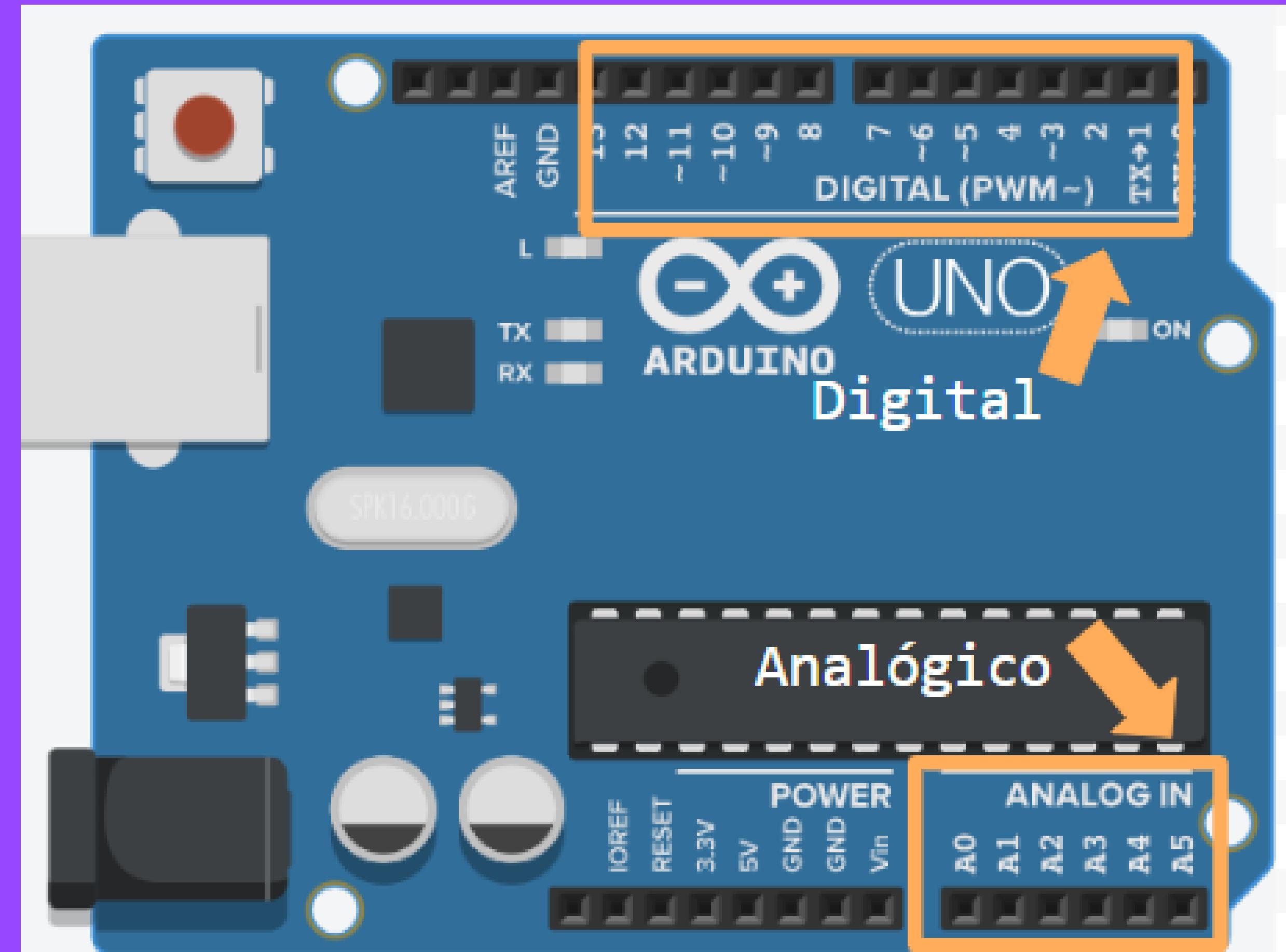


Gráfico de sinais

Portas digitais e analógicas do Arduino UNO



Entenda o resistor

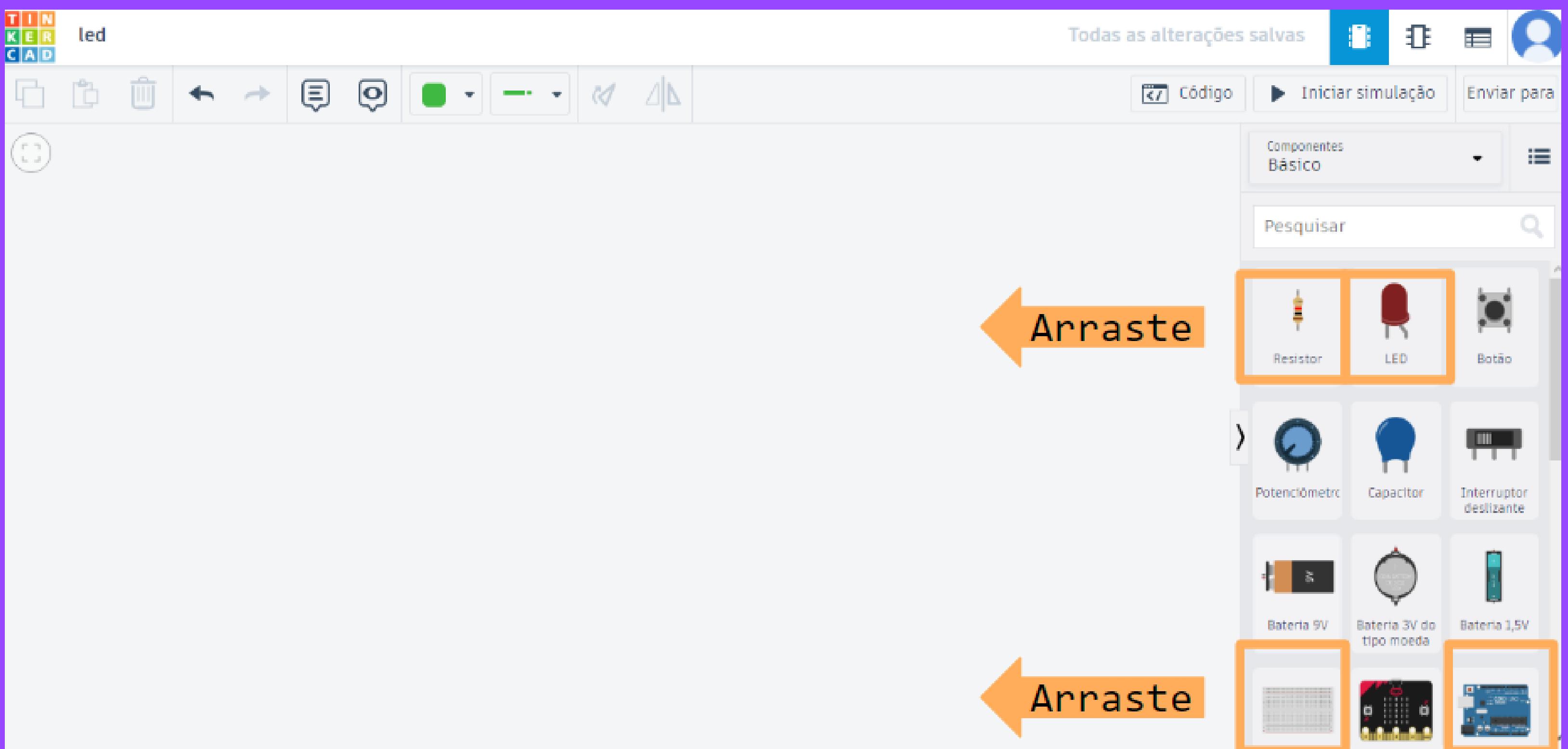
O resistor serve para criarmos uma resistência na corrente para que não ultrapasse o limite da resistência do componente que queremos utilizar, e dependendo da corrente que passa, se não houver resistência, pode queimá-lo.

- > Resistência: medida em ohms;
- > O resistor utilizado depende do componente;
- > Nos próximos passos, o resistor escolhido foi de 330Ω .

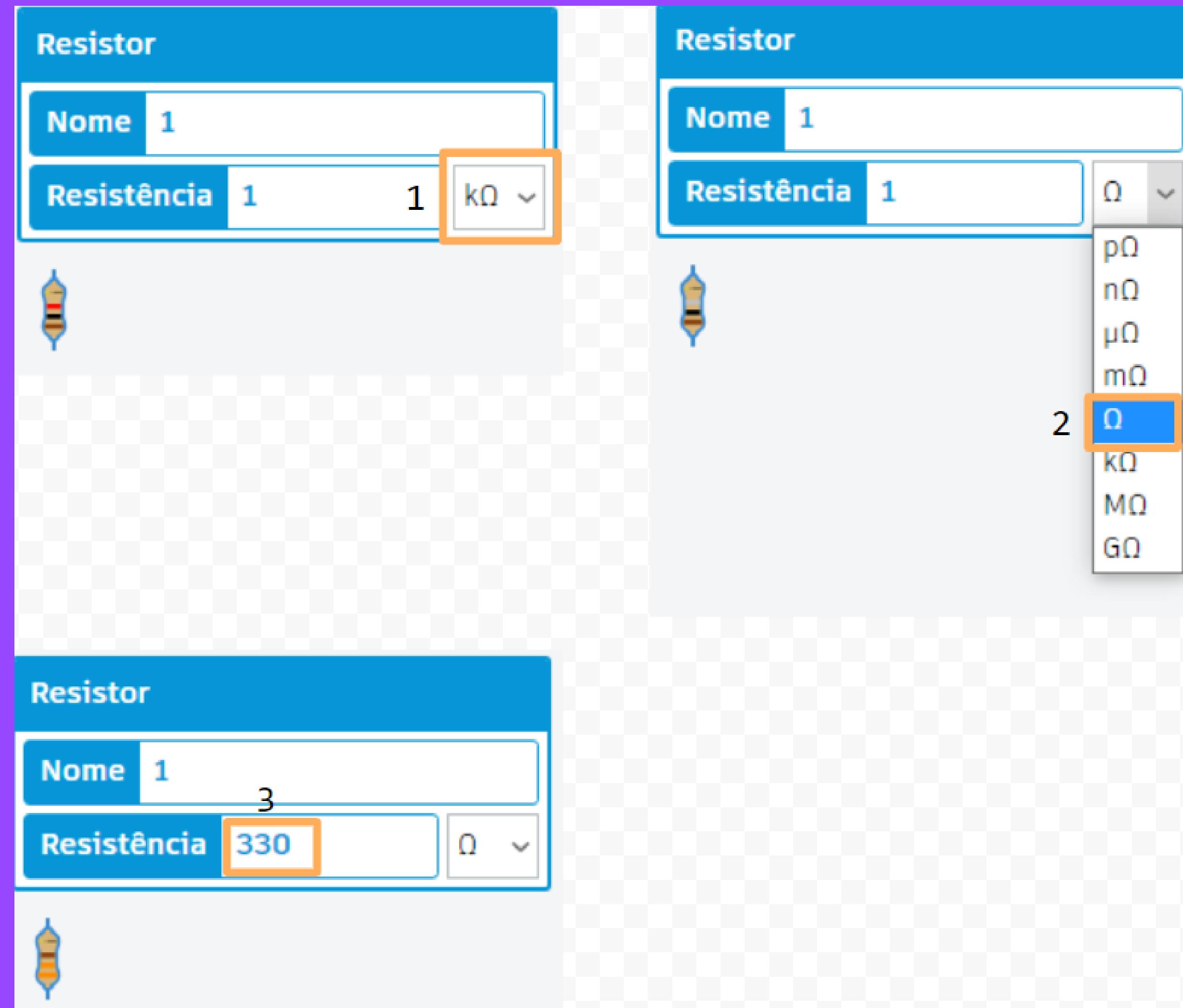
Passo 2

Vamos montar o circuito?

Primeiro, puxe uma tela em branco o Arduino Uno R3, uma protoboard, um LED e um resistor de 330Ω .

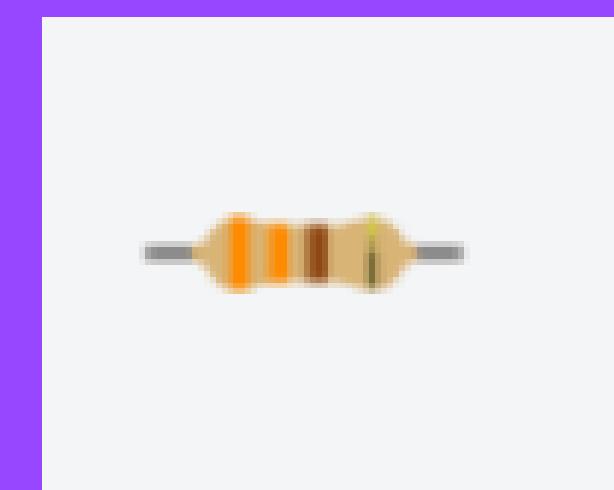
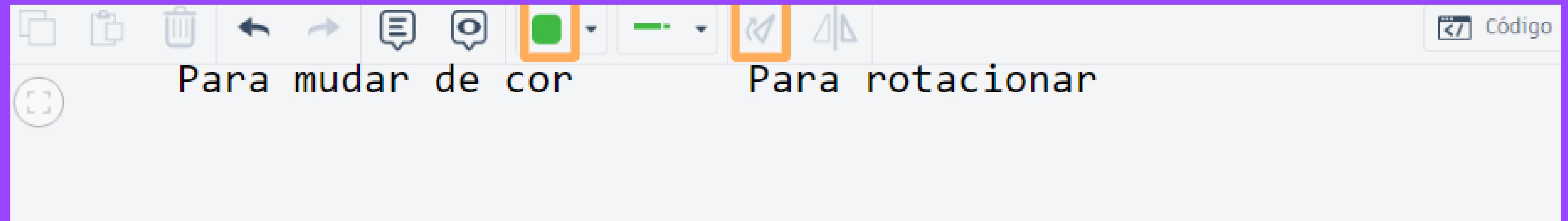


Para setar o resistor de 330Ω , após arrastá-lo ao quadro branco, vamos clicar em cima do resistor e seguir os passos abaixo.

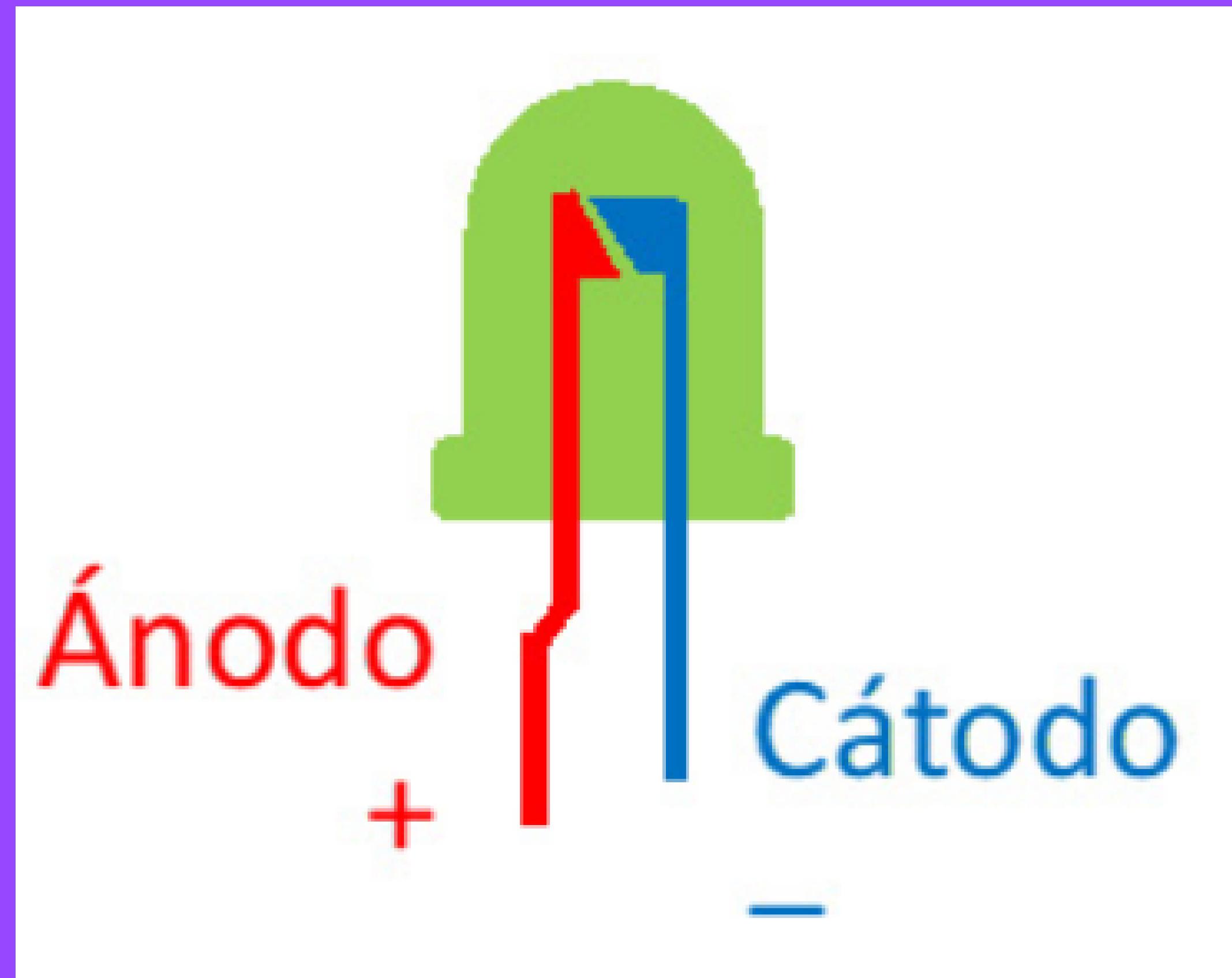


Para rotacionar os elementos, utilize um botão na tela, que estará na foto abaixo, ou a tecla “R” do teclado.

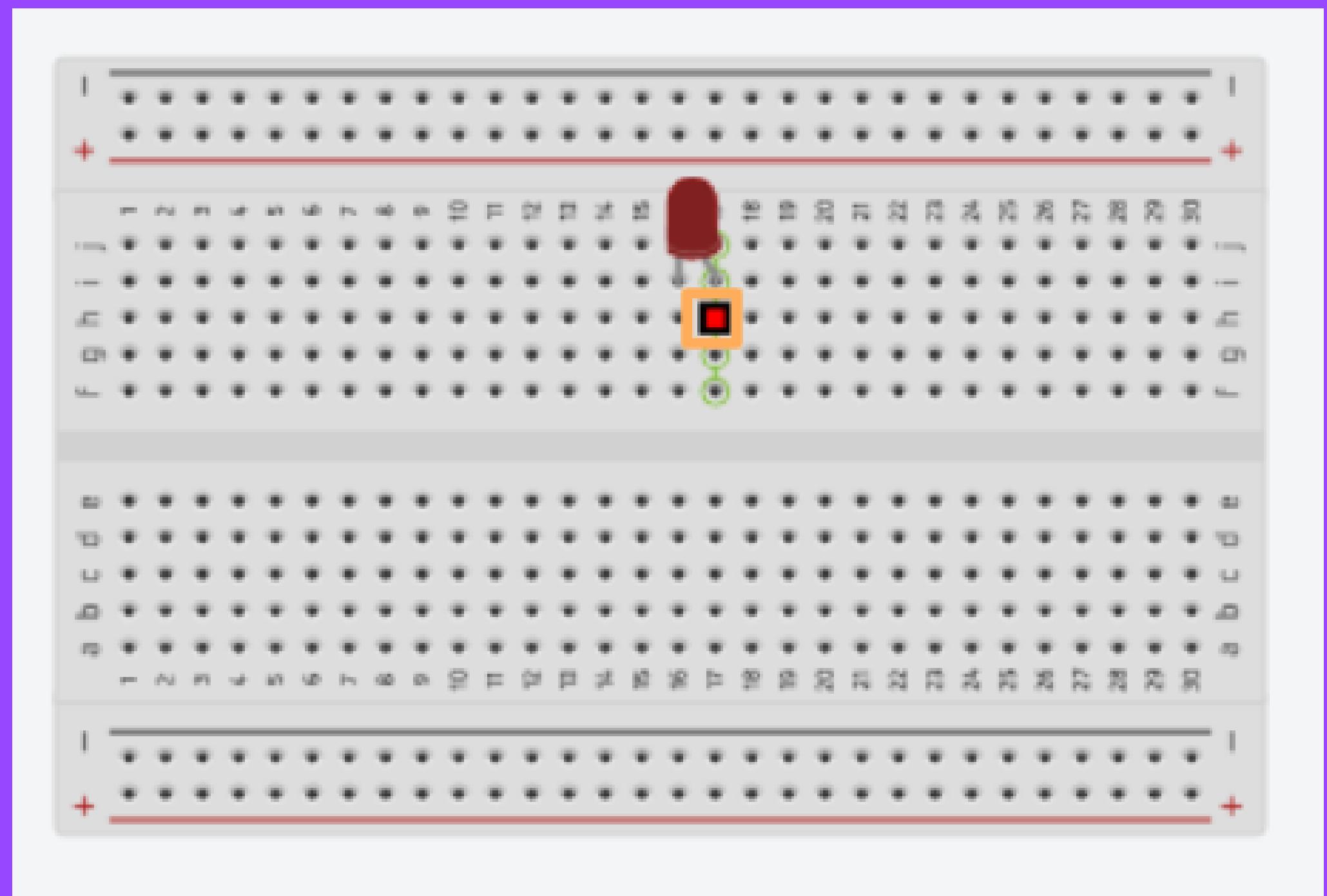
Escolhemos rotacionar o resistor até ficar em 180° e utilizar fios, que podem ter sua cor mudada em um botão próximo ao de rotacionar, observe:



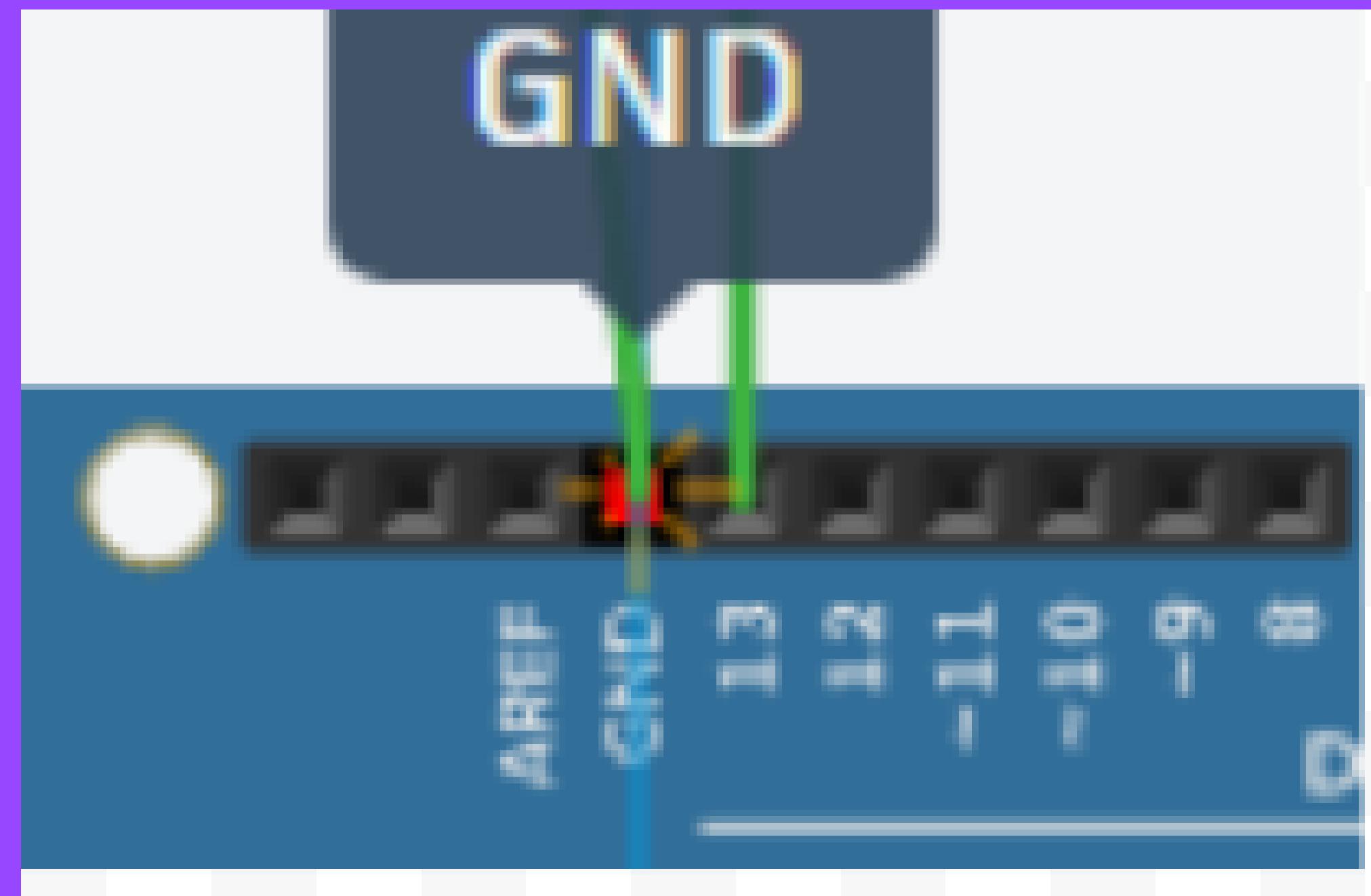
Para montarmos o circuito do LED, devemos saber que este componente possui duas terminações: a do ânodo e do cátodo, como demonstrado na figura:



Para puxar o fio, basta ir com o mouse em alguma extremidade da protoboard, apertar o botão esquerdo e puxar até a porta desejada. Perceba que há uma seleção que aparece na tela ao passar o mouse por cima dessas extremidades da protoboard:

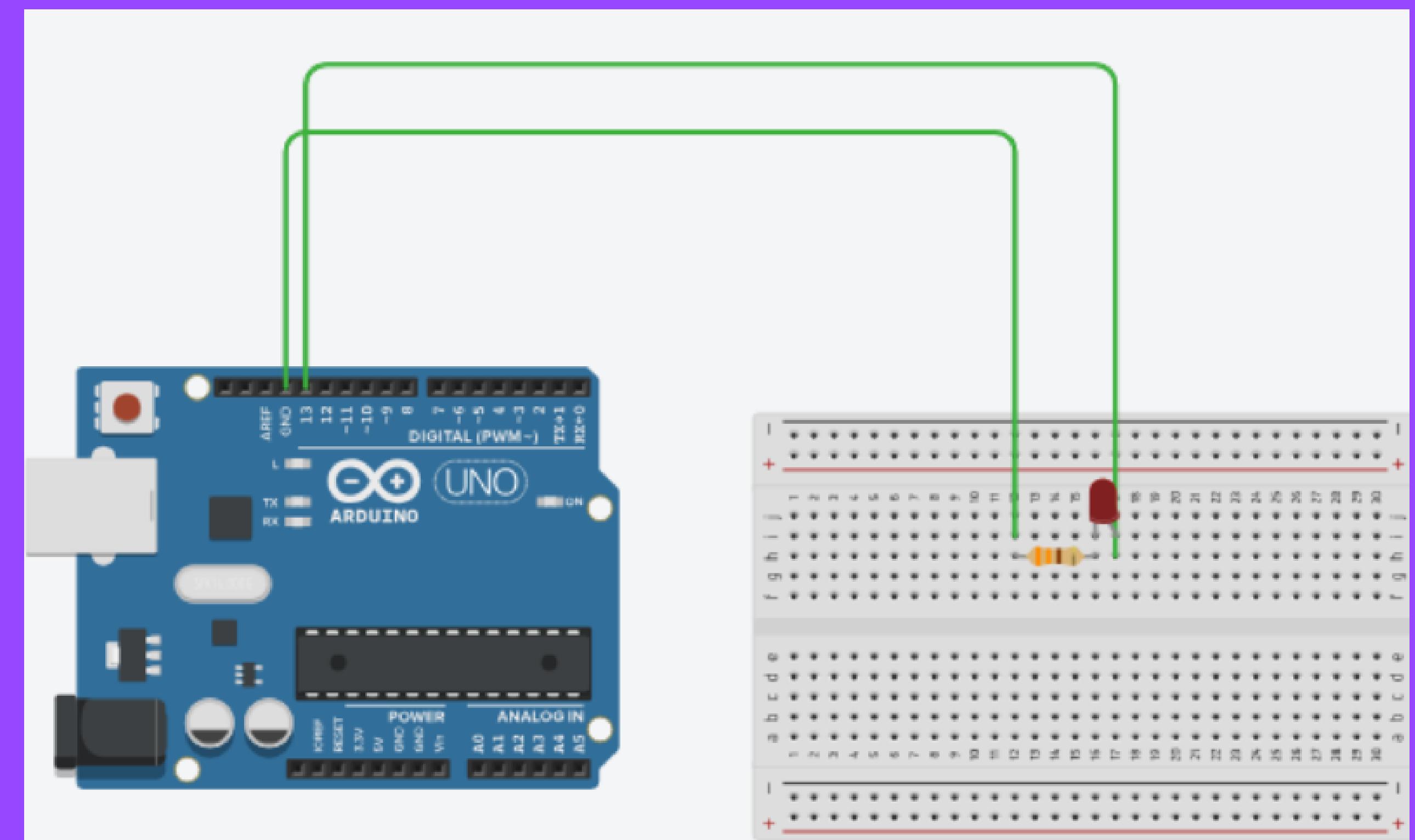


Perceba também que quando vamos colocar o fio na placa, é indicado em que porta está inserindo para melhor visualização:

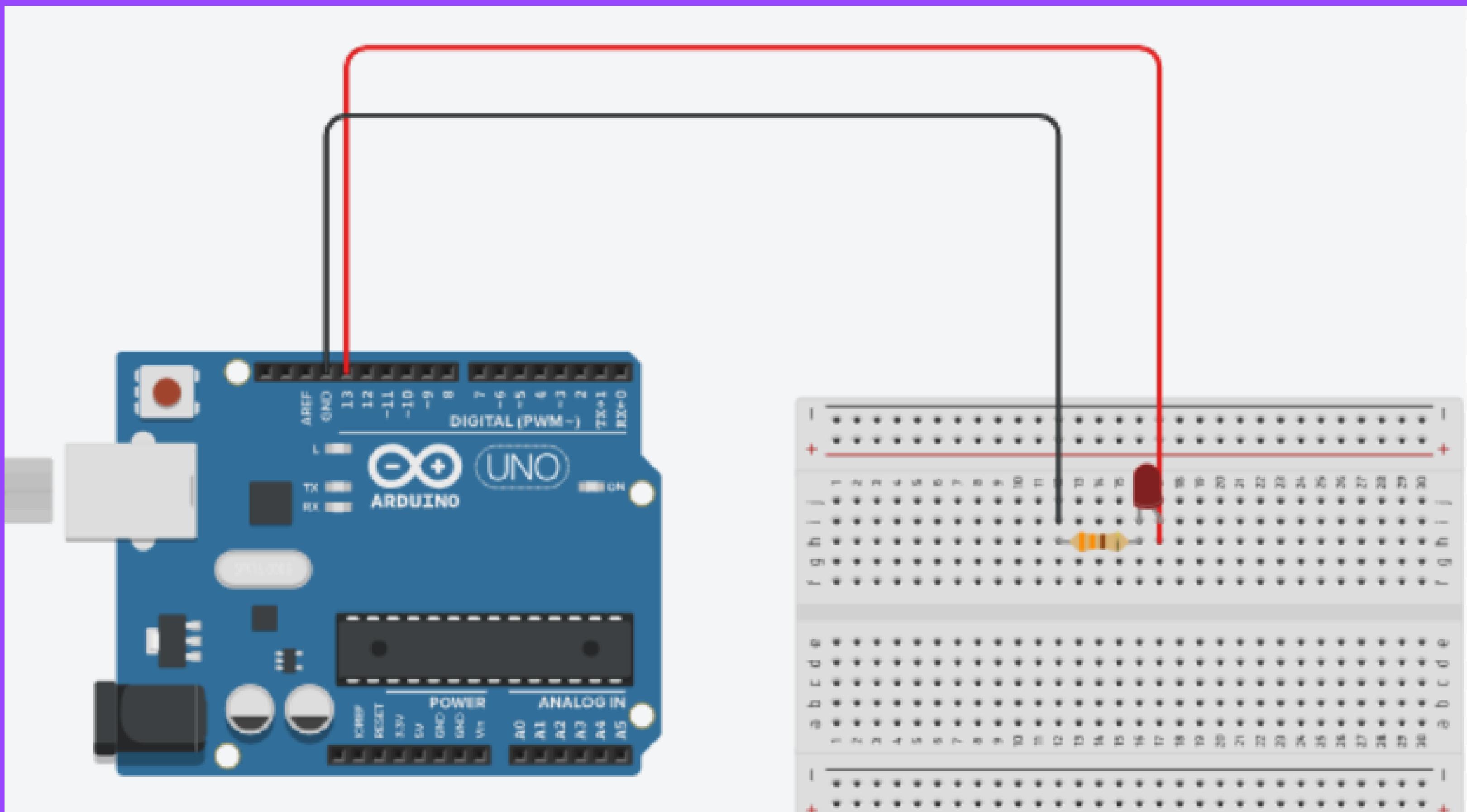


O ânodo (+) será ligado na placa onde o LED será energizado, ou seja, na porta digital. Enquanto o cátodo (-) será ligado a uma porta chamada ground (GND) que é a conexão terra, que serve como fase neutra.

-> Nesse circuito, ligamos o positivo na porta digital 13 do Arduino e o negativo em qualquer porta denominada “GND” na placa, como na figura:



Para facilitar a diferenciação dos dois fios, vamos deixar o cabo do GND sempre preto, por convenção. Já o de energia, será sempre vermelho. Para outras terminações além dessas duas, usaremos outras cores de nossa escolha.



Passo 3

Vamos entender a variável do tipo int?

Antes de começarmos o código, abra a seção de código, no formato de texto e começaremos já!

- > Variáveis: atribuir valores, como números, letras e palavras;
- > Atribuir: passar um valor para dentro da variável;

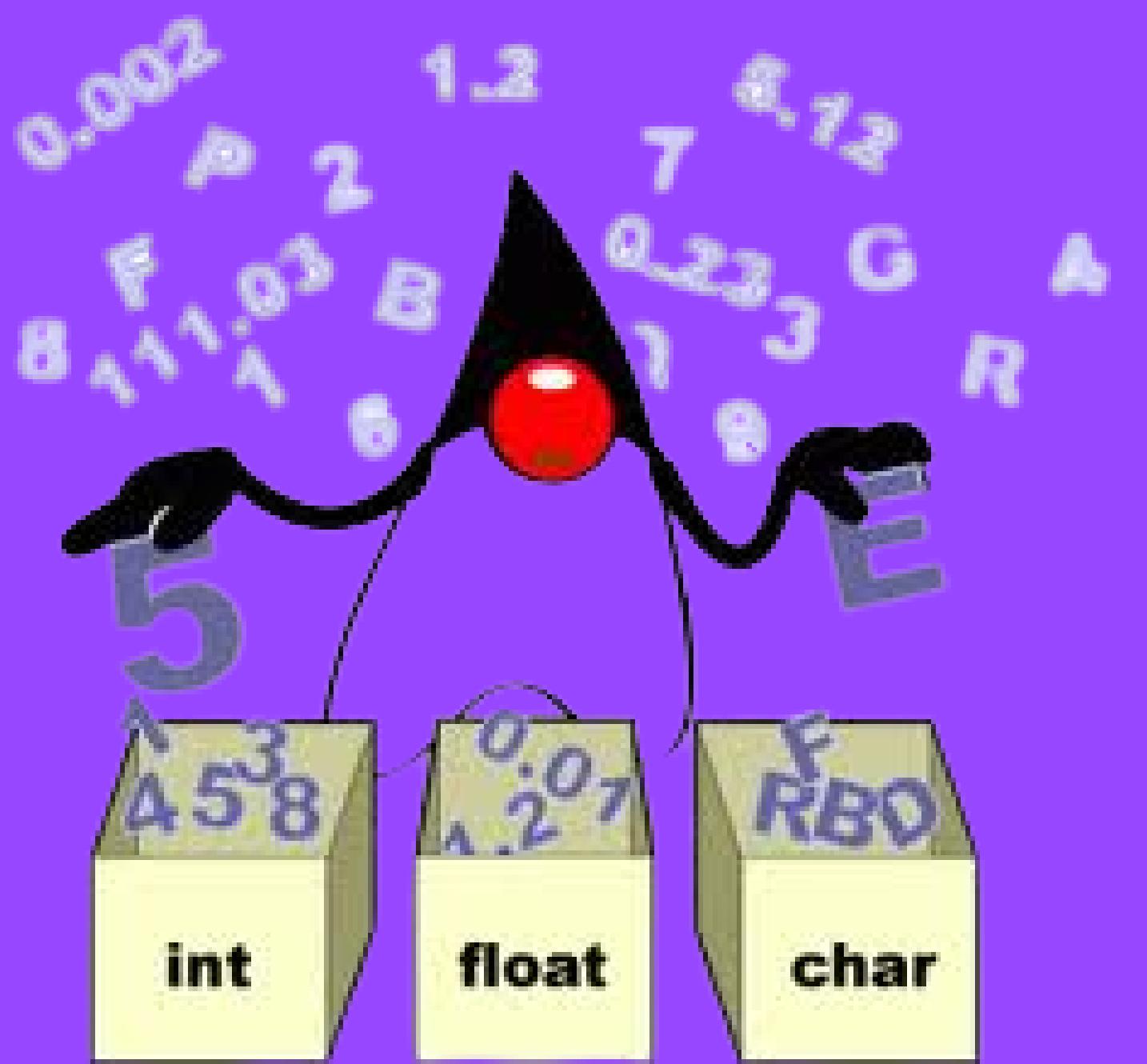
O nosso código utiliza variáveis, para criar uma variável de identificação do pino digital que vamos usar (13), com valores inteiros, chamadas de int.

Vamos entender a variável do tipo int?

- > Sintaxe - modo de escrever padronizado, sendo “tipo de variável + nome da variável (você escolhe) = valor”.

Regras para definir o nome da variável:

- > Não devem começar com números;
- > Pode usar símbolos como underline, letras maiúsculas e minúsculas;



Sintaxe utilizada

A sintaxe que usaremos será escrita dessa maneira: `int ledPin = 13;` Vamos colocar essa variável antes do `setup` e do `loop`, por ser global (todas as partes do programa podem utilizá-la).

- > Na função `setup`: colocar o inicializador do Monitor Serial e o comando de setar se nosso componente eletrônico é saída ou entrada, através do comando “`pinMode(ledPin, OUTPUT);`”.
- > `pinMode(ledPin, OUTPUT)` -> pegar a variável que criamos para relacionar a porta do Arduino. Separamos por uma vírgula e colocamos que é um componente de saída (`OUTPUT`). Se fosse um componente de entrada, usamos `INPUT`.

Sintaxe utilizada

- > Na função loop: colocar os comandos que irão fazer com que o LED acenda, escreva que está ligado no monitor serial e espere um segundo para desligar, escrever que está desligado e esperar um segundo novamente.
- > Comando do Monitor Serial: usaremos o que pula linha, para indicar se o LED está ligado ou desligado.
- > Comando que espera: usaremos 1000 milisegundos (1 segundo).

Comando que liga e desliga o LED

Chamado de escritura ou de leitura;

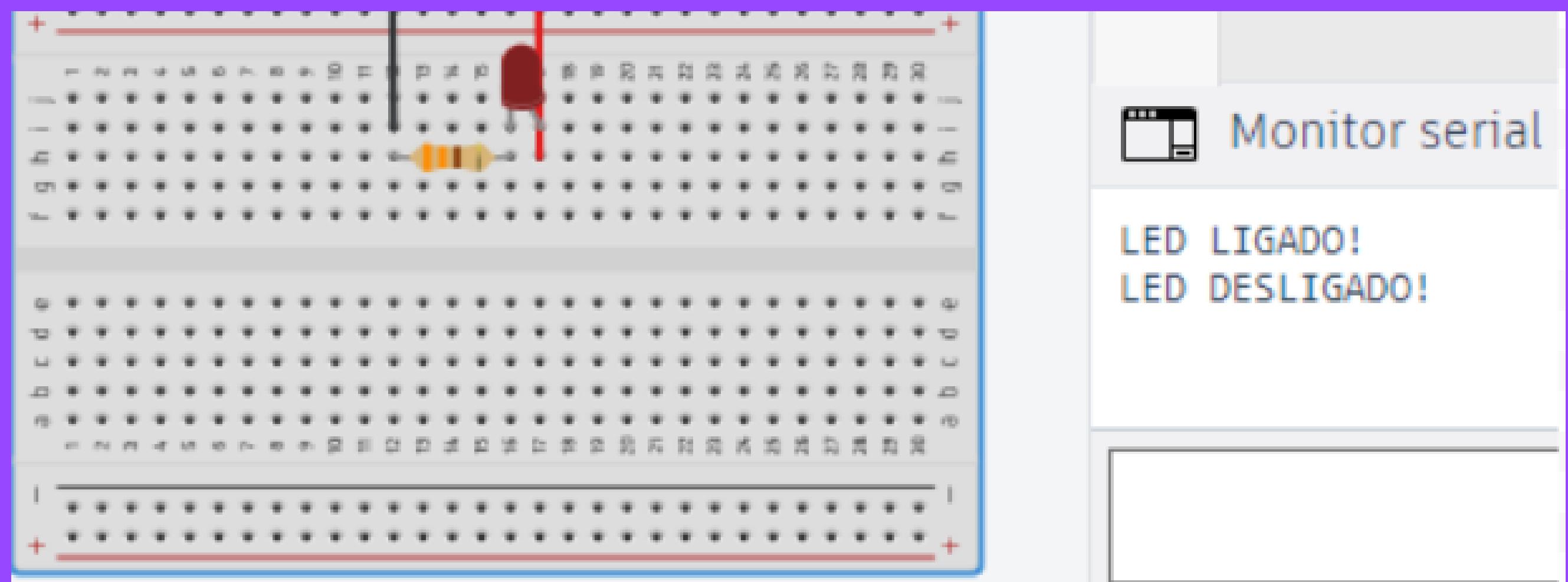
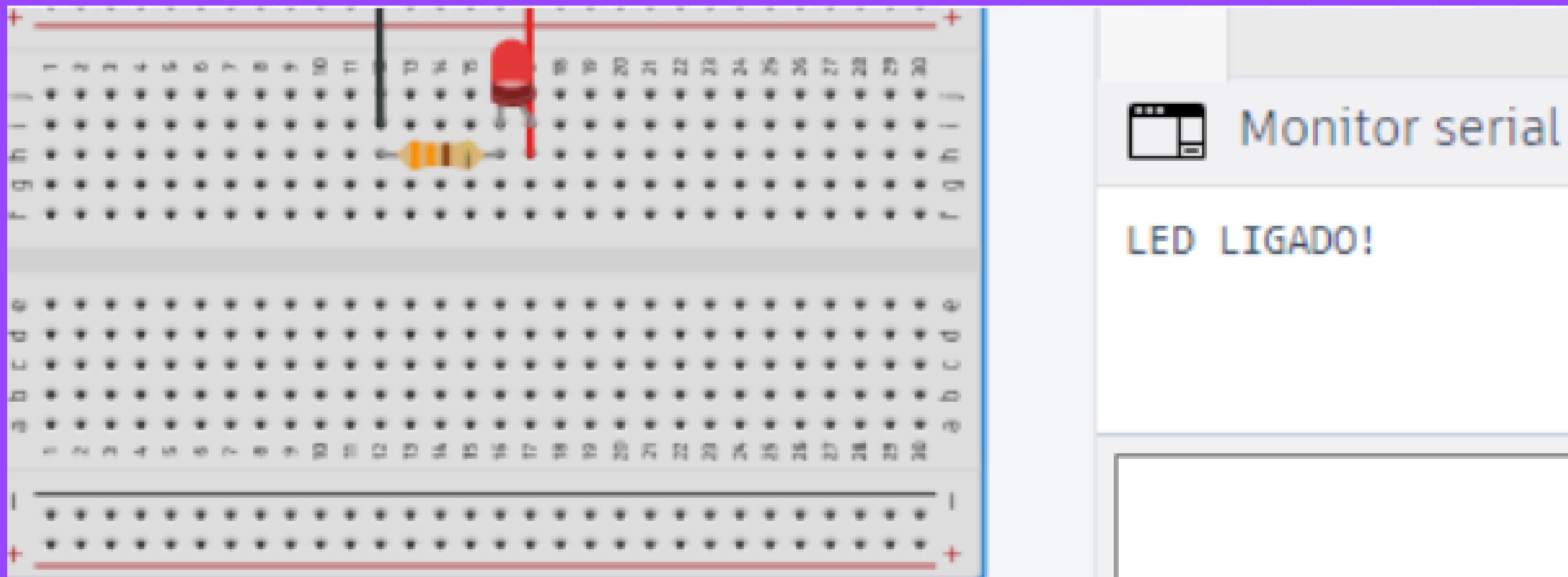
Em nosso código foi utilizado o de escritura:

- > `digitalWrite(ledPin,HIGH);` -> usa a variável que nos indica a porta 13 digital. Para indicar para ligar, usa-se HIGH ou 1.
- > `digitalWrite(ledPin, LOW);` -> esse comando usa a variável que nos indica a porta 13 digital, por isso é um comando digital, seguido de uma vírgula para indicar para desligar, usa-se LOW ou 0.

O programa está disponível no nosso website e aqui em uma foto anexa:

```
1 //programa de ligar e desligar o LED!
2 int ledPin = 13; //variavel do tipo int global
3
4 void setup()
5 {
6     Serial.begin(9600); //inicializa o Monitor Serial
7     pinMode(ledPin,OUTPUT); //declarando o LED como atuador
8
9 }
10
11 void loop()
12 {
13     digitalWrite(ledPin,HIGH); //comando para ligar o LED
14     Serial.println("LED LIGADO!"); //indicando o estado do LED
15     delay(1000); //aguarda 1 segundo
16     digitalWrite(ledPin,LOW); //comando para desligar o LED
17     Serial.println("LED DESLIGADO!"); //indicando o estado do LED
18     delay(1000); //aguarda 1 segundo
19 }
```

Resultado no monitor serial:



Agora você sabe o que ligar e desligar
um LED no Tinkercad. Continue para as
próximas aulas para aprender mais sobre
Robótica!!

