

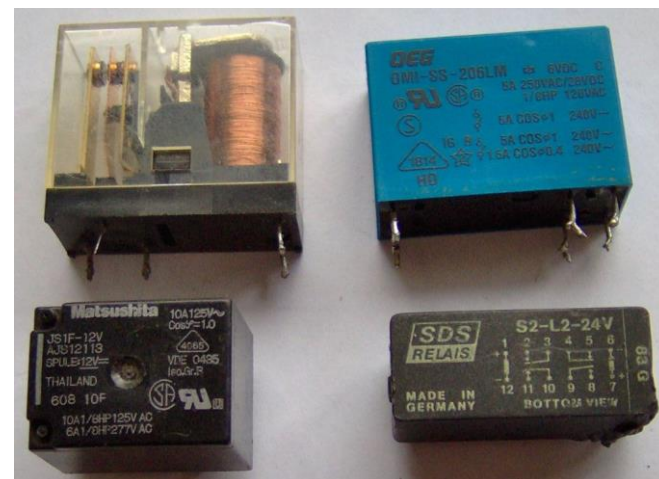
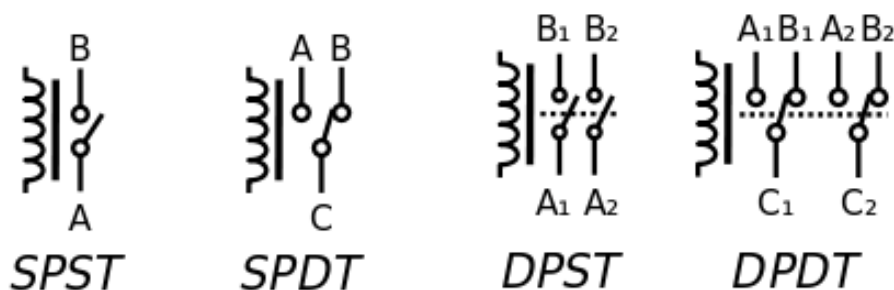
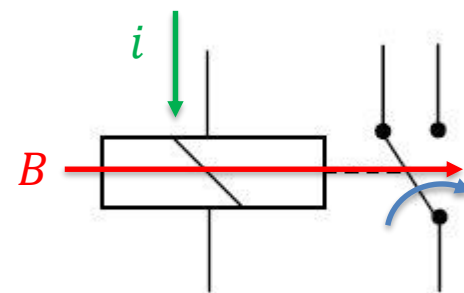


# Programação e IoT - Aula 04

## Atuadores

# ELEMENTO DE ACIONAMENTO: RELÉ

- É um componente eletromecânico capaz de comutar uma chave
  - Opera como um interruptor
- Ele possui um circuito com:
  - Uma bobina, responsável por comutar os polos (as chaves)
  - Pode ter um polo (SP), dois polos (DP) e até mais
    - Cada polo pode ter um (ST) ou dois (DT) atuadores
    - O nome dos atuadores referem-se ao relé desenergizado:
      - NF, normalmente fechado
      - NA, normalmente aberto



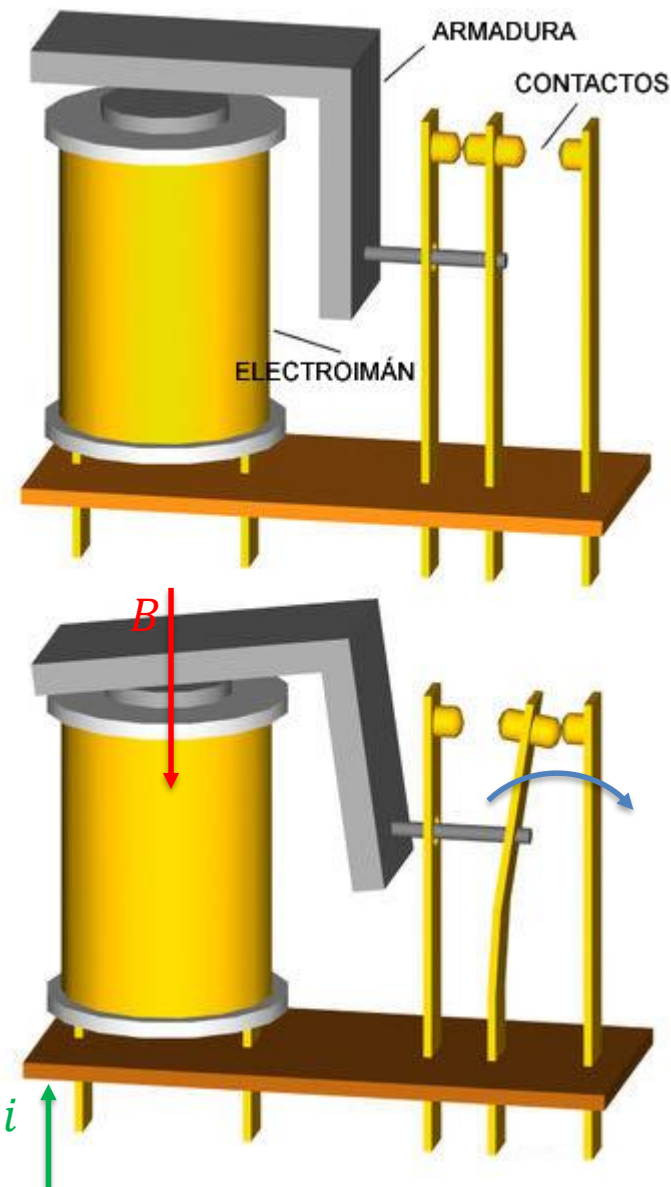
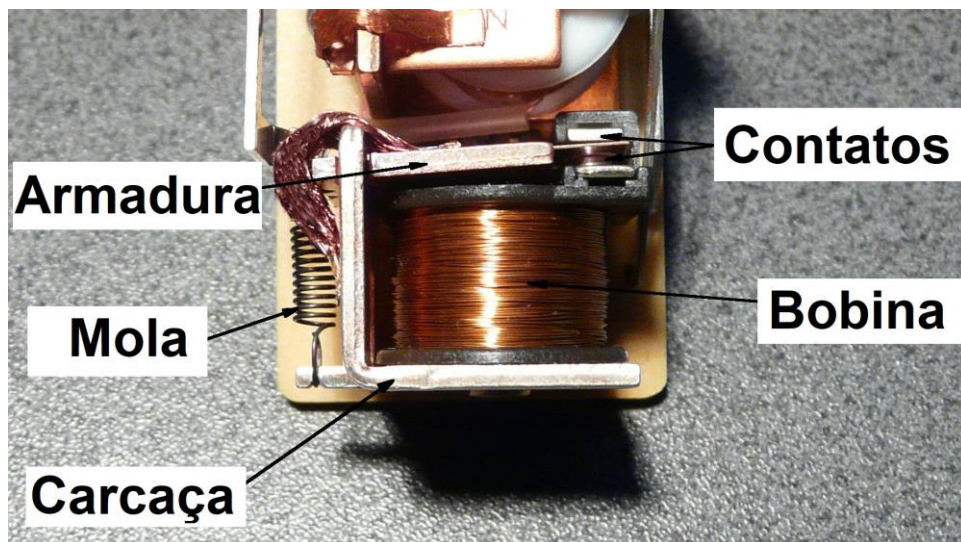
<https://pt.wikipedia.org/wiki/Rel%C3%A9>

<http://mundoprojetado.com.br/rele-o-que-e-e-como-funciona/>

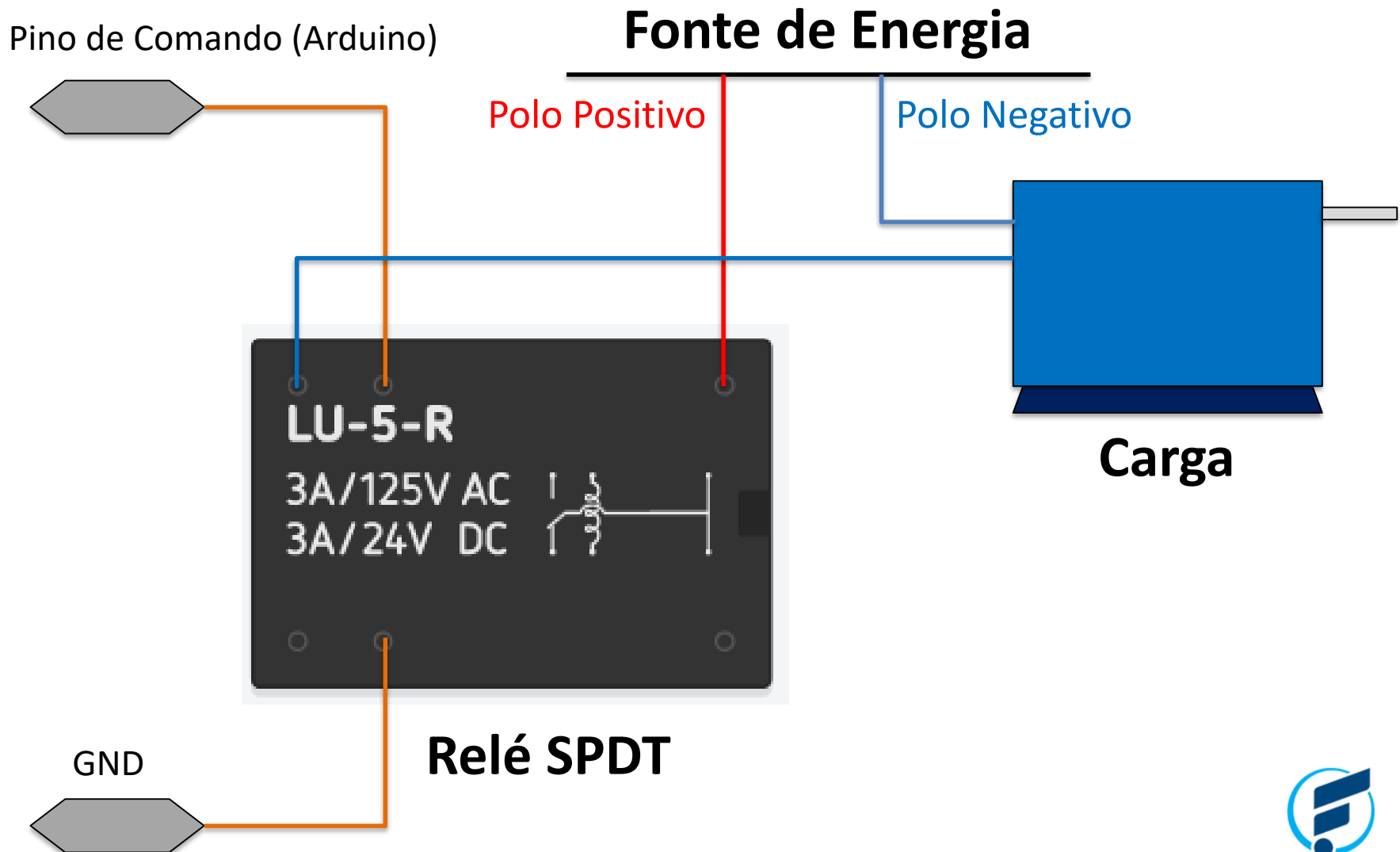
<https://pt.quora.com/O-que-significa-o-termo-SPST-SPDT-DPST-e-DPDT-relacionado-a-chaves-eletr%C3%B4nicas-nunca-entendi-isso-muito-bem>

# ELEMENTO DE ACIONAMENTO: RELÉ (Funcionamento)

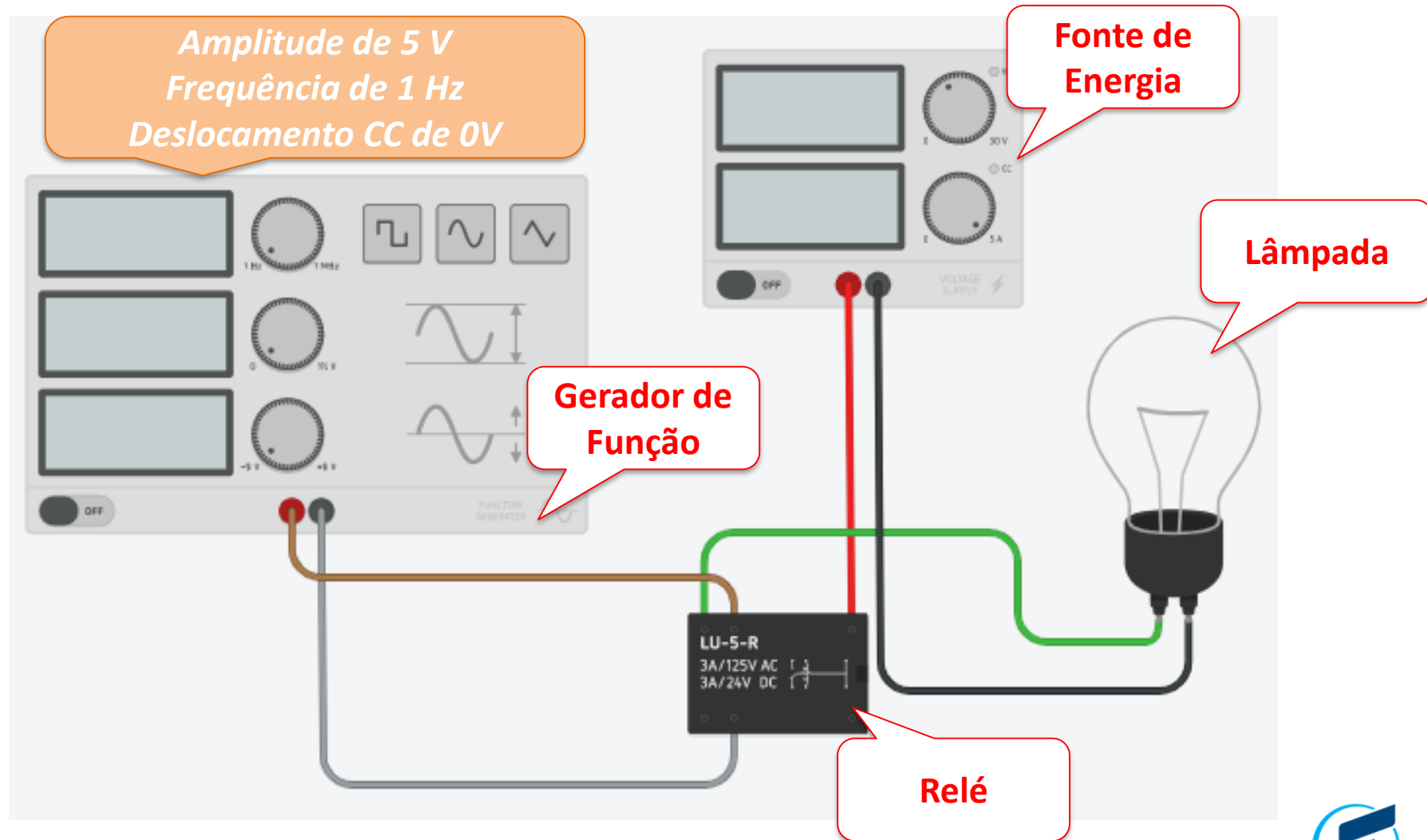
- Funcionamento:
  - Enquanto a bobina do relé não está energizada, a mola mantém o contato no polo NF;
  - Ao energizar a bobina, a armadura é atraída pela bobina e o contato migra para o polo NA.



# ELEMENTO DE ACIONAMENTO: RELÉ



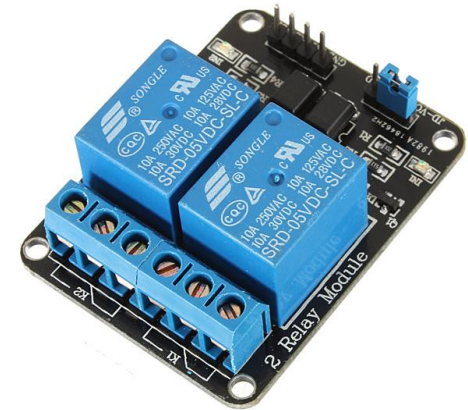
# ELEMENTO DE ACIONAMENTO: RELÉ



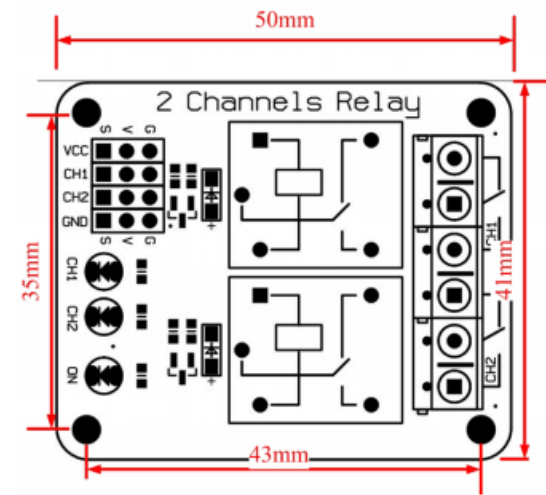
(Monte no Tinkercad)

# ELEMENTO DE ACIONAMENTO: RELÉ (Shield para Arduino)

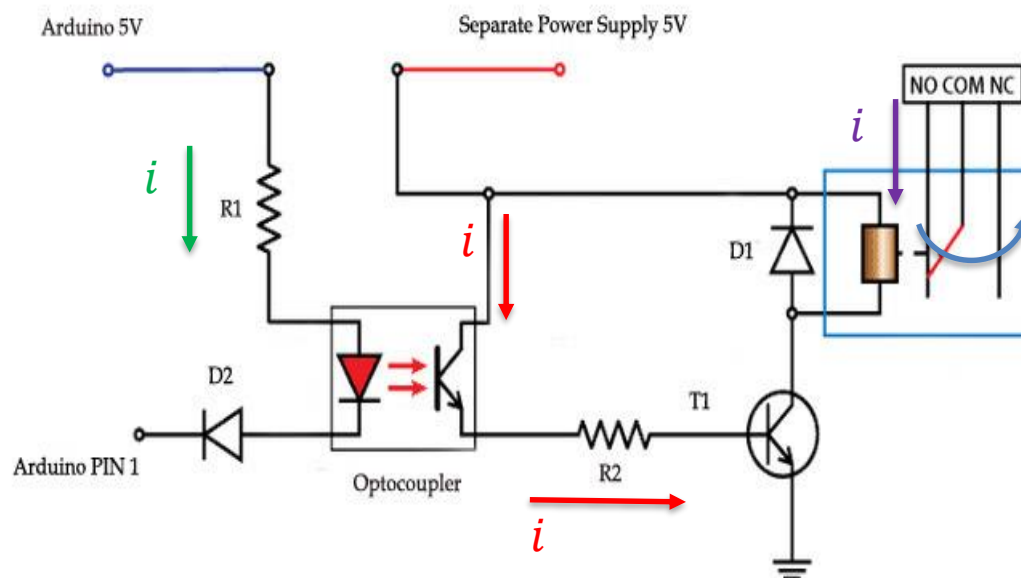
- Módulo Relé 5V, 2 Canais:
  - As linhas IN1 e IN2 são conectadas ao Arduino para controlar os relés K1 e K2, respectivamente;
  - Corrente típica de operação: 15~20mA;
  - Permite controlar cargas de (30 V<sub>CC</sub> @ 10A) ou (250V<sub>CA</sub> @ 10A).



# ELEMENTO DE ACIONAMENTO: RELÉ (Shield para Arduino)

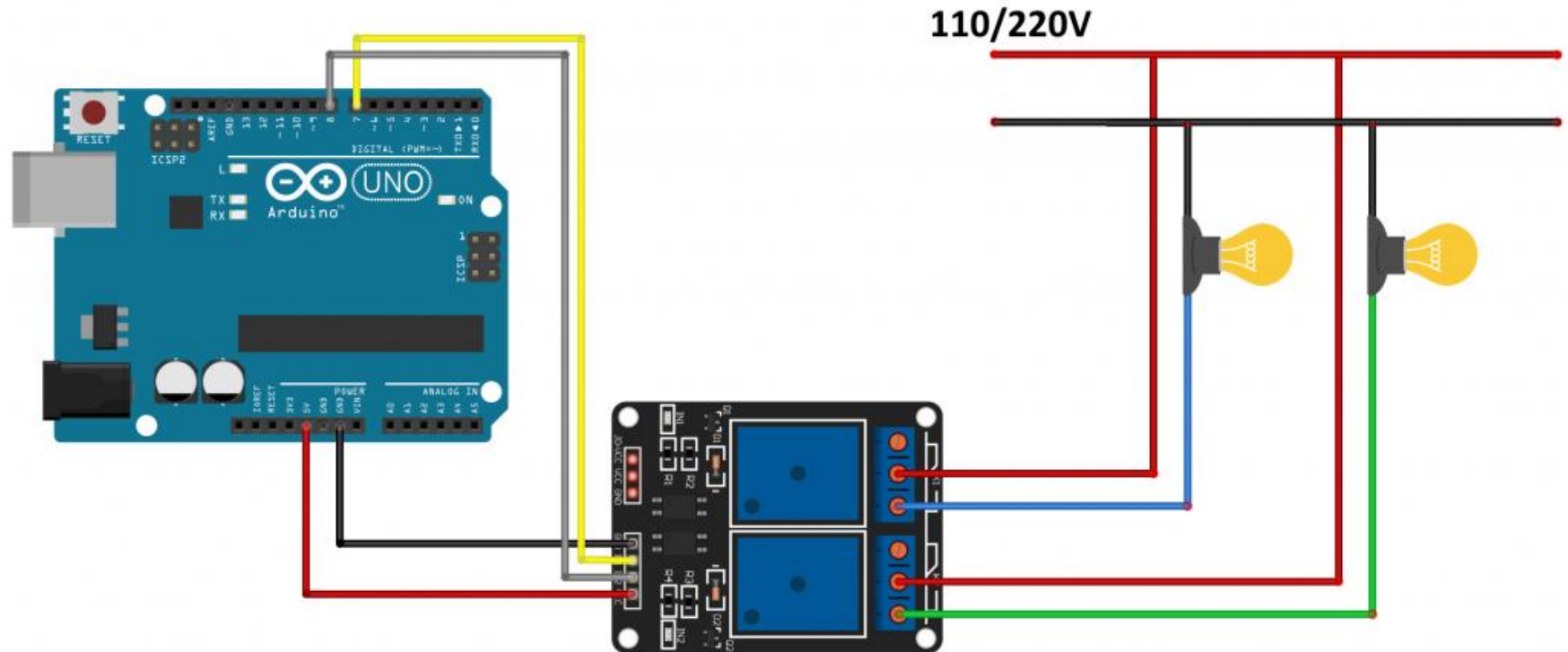


Para o funcionamento do relé deve-se mandar 0V no PIN1 (Shield)

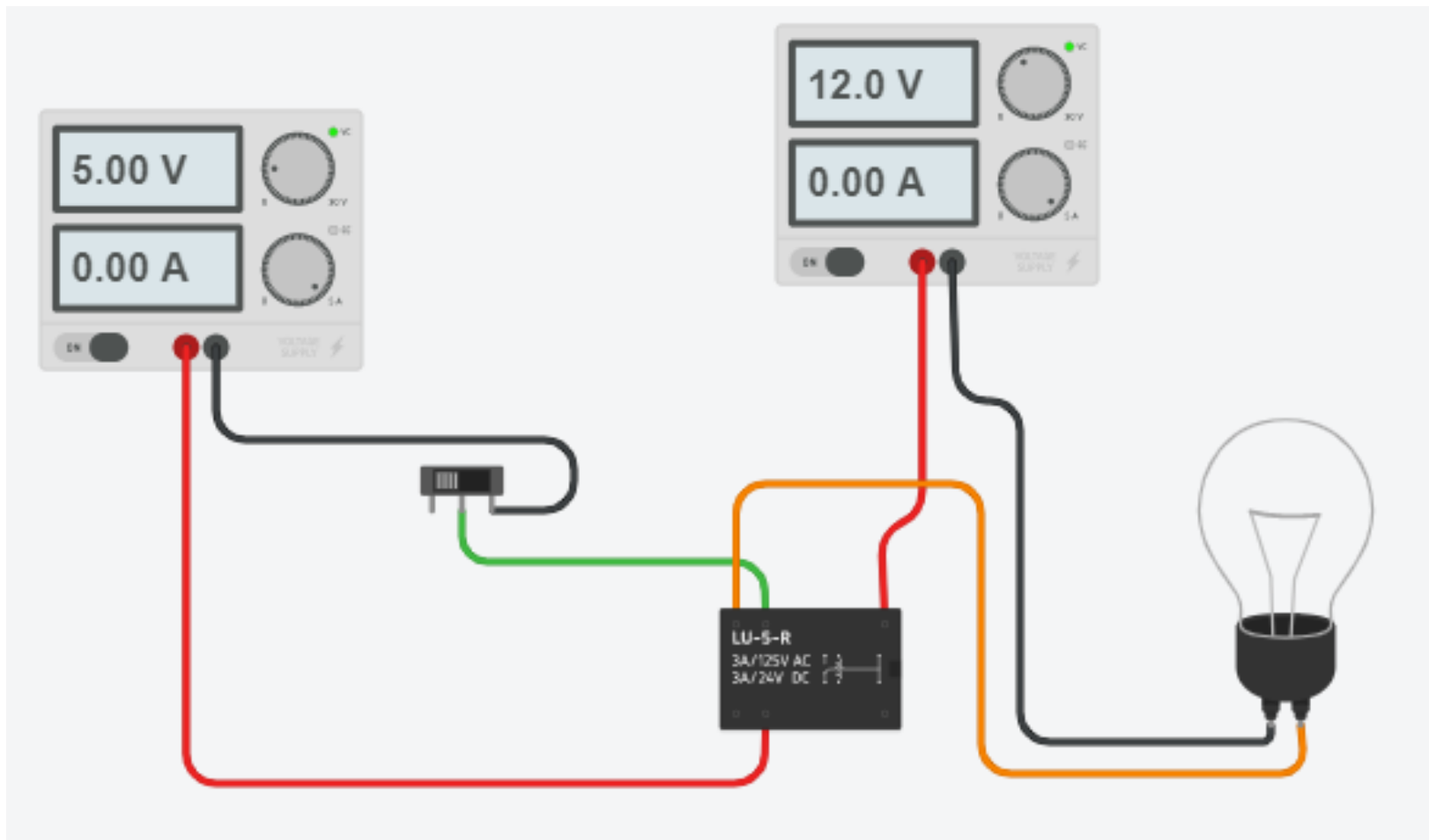




# ELEMENTO DE ACIONAMENTO: RELÉ (Exemplo)



# ELEMENTO DE ACIONAMENTO: RELÉ



**(Monte no Tinkercad)**

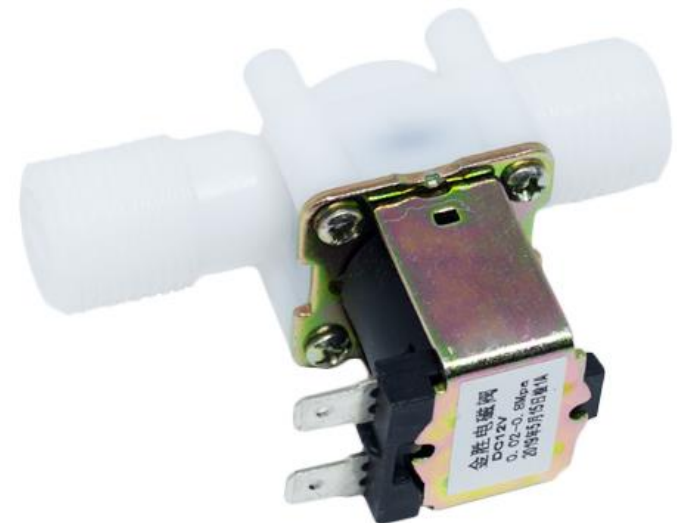
[Circuit design Swanky Jaagub | Tinkercad](#)





# ATUADORES: VÁLVULA

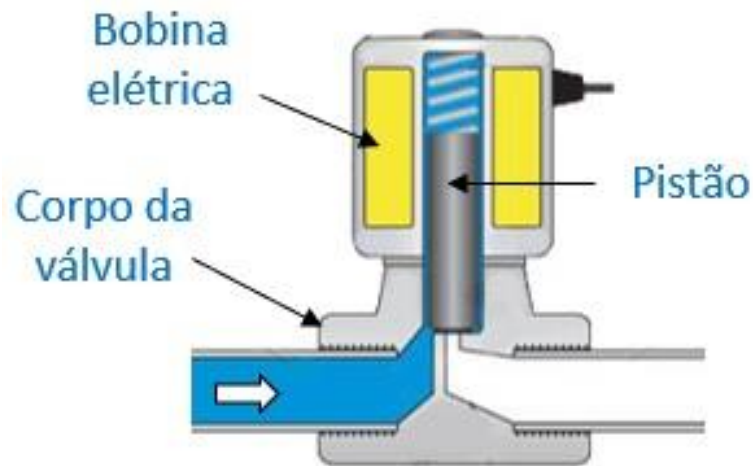
- Uma das aplicações dos relés é ligar/desligar válvulas solenoides de controle de vazão de líquidos
- A figura ilustra uma válvula solenoide 12VDC para controle de líquidos de baixa viscosidade, como a água
  - Opera com 12 V<sub>CC</sub> @ 200mA
  - Temperatura máxima do fluido: 100°C
  - Pressão: -0,02 ~ 0,8MPa
  - Normalmente fechada
  - Tipo de válvula: Diafragma (operada por servo)



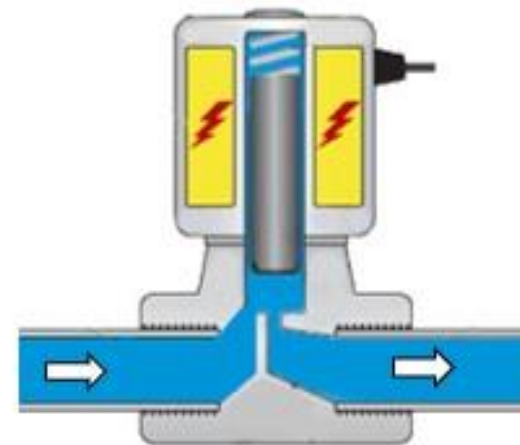
# ATUADORES: VÁLVULA

## Válvula solenoide

(2/2 de ação direta)



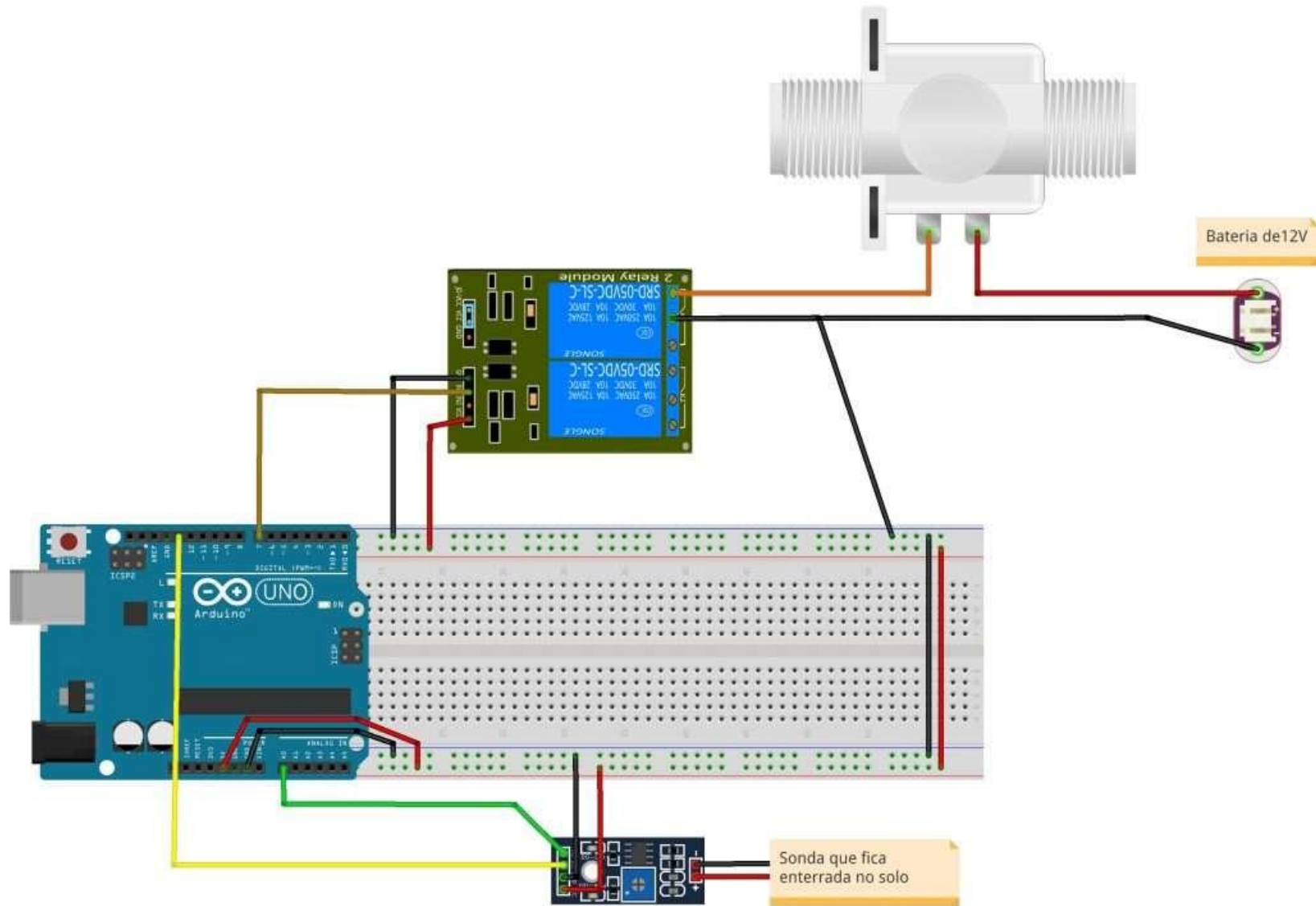
**Sem energia**



**Energizada**

[www.mtibrasil.com.br](http://www.mtibrasil.com.br)

# ATUADORES: RELÉ + VÁLVULA (Exemplo)



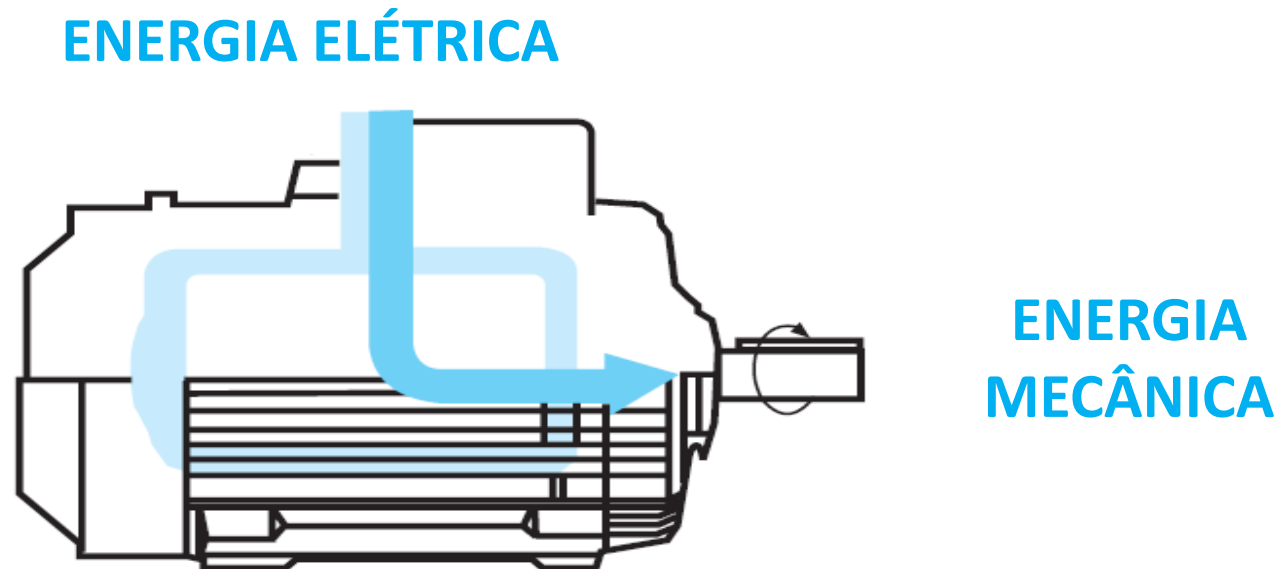
# Programação e IoT - Aula 04

ATUADOR: MOTORES

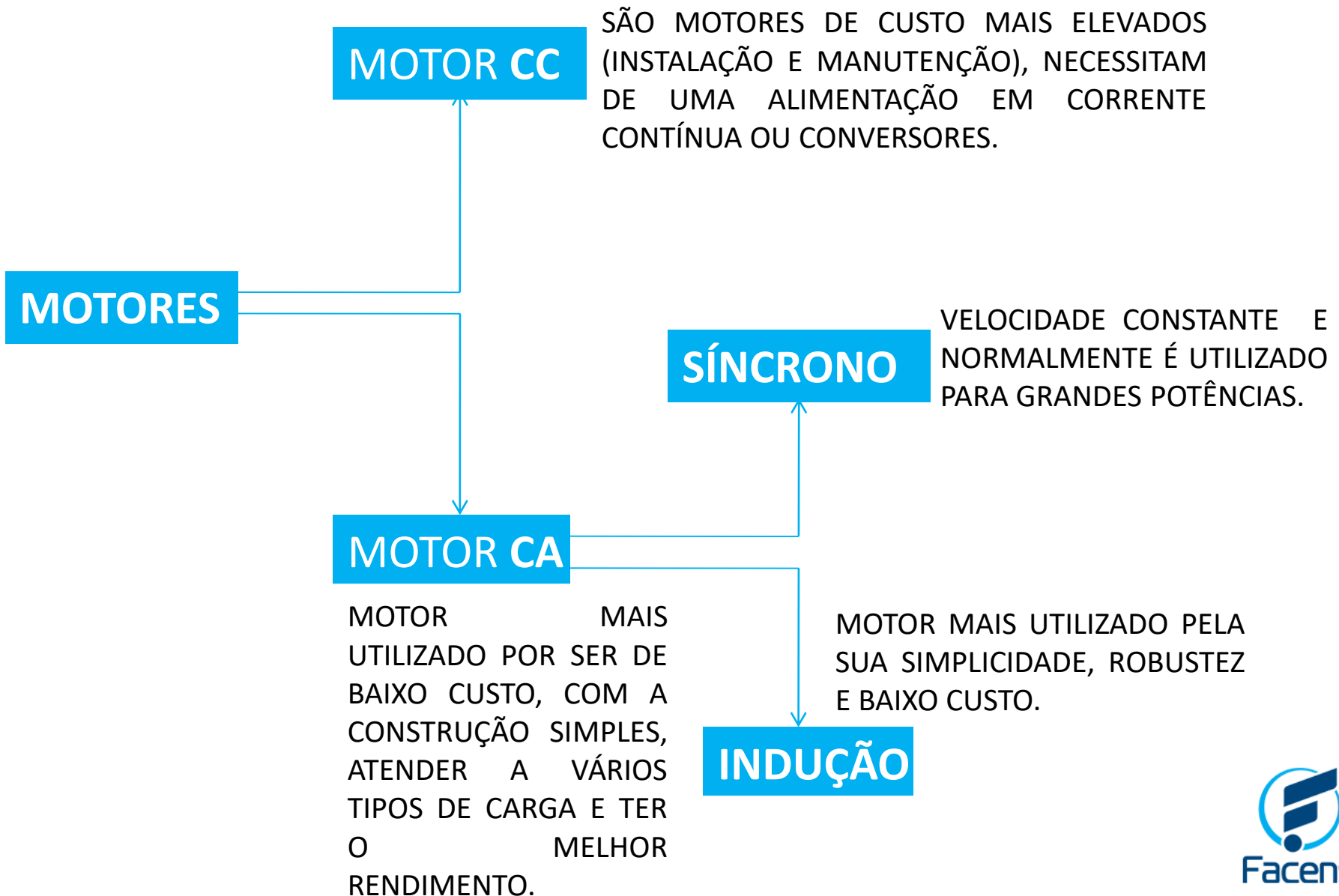


# ATUADOR: MOTORES

Motores são máquinas que transformam a energia elétrica em energia mecânica.



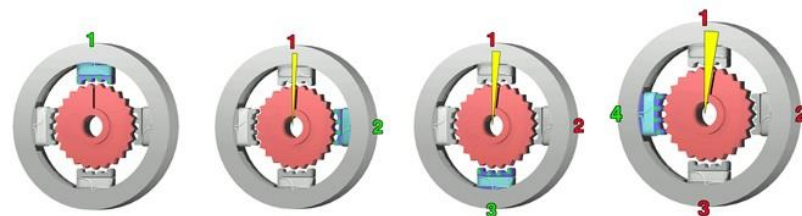
# ATUADOR: MOTORES



# ATUADOR: TIPO DE MOTORES

A escolha do motor depende do tipo de operação

- Motores de corrente contínua (DC motors)
  - **Com e sem escova;**
  - Ímã permanente ou relutância variável.
- Motores do tipo servo (servo):
  - **Movimentação precisa;**
  - Mantem a posição (forças externas);
  - **Opera em malha fechada;**
  - Motores CA ou CC;
  - Controle do torque no eixo de forma constante e em larga faixa de rotação.
- Motores de passo (stepper motors)
  - Unipolar ou bipolar;
  - **Dispensa realimentação.**



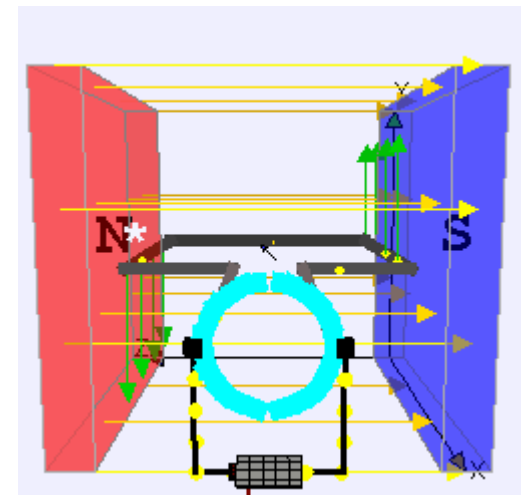
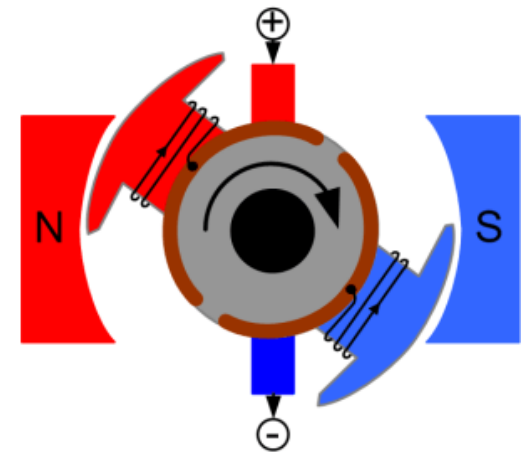
# ATUADOR: MOTOR DC (com escovas)

Em motor DC (com escovas), o rotor gira quando a corrente que passa em suas bobinas se contrapõem ao campo magnético formados pelos ímãs permanentes do estator.

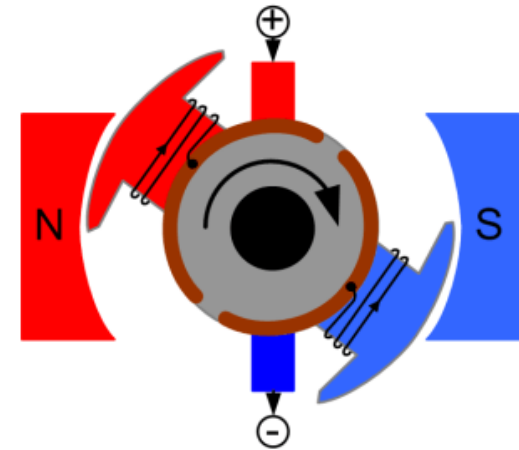
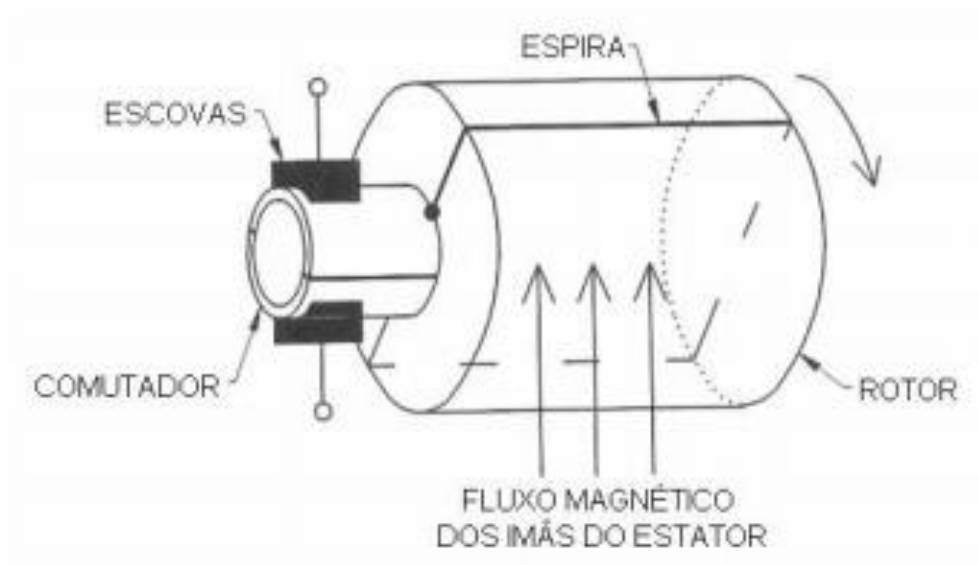
- Escova (grafite) são utilizadas para transferir energia às bobinas do rotor;
- O comutador garante o sentido da corrente nas bobinas, conforme o rotor gira.

Para variar a velocidade do motor podemos alterar a corrente nas bobina, que é diretamente proporcional a tensão sobre elas.

- Dessa forma, com a mudança da tensão, teremos uma alteração na velocidade de rotação do rotor;
  - Uma alternativa simples e eficiente para alterar a tensão sobre as bobinas é o uso de PWM.



# ATUADOR: MOTOR DC (com escovas)



No motor ideal a velocidade de rotação ( $\omega$ ) é proporcional a constante de proporcionalidade ( $J$ ) e a tensão ( $V$ ).

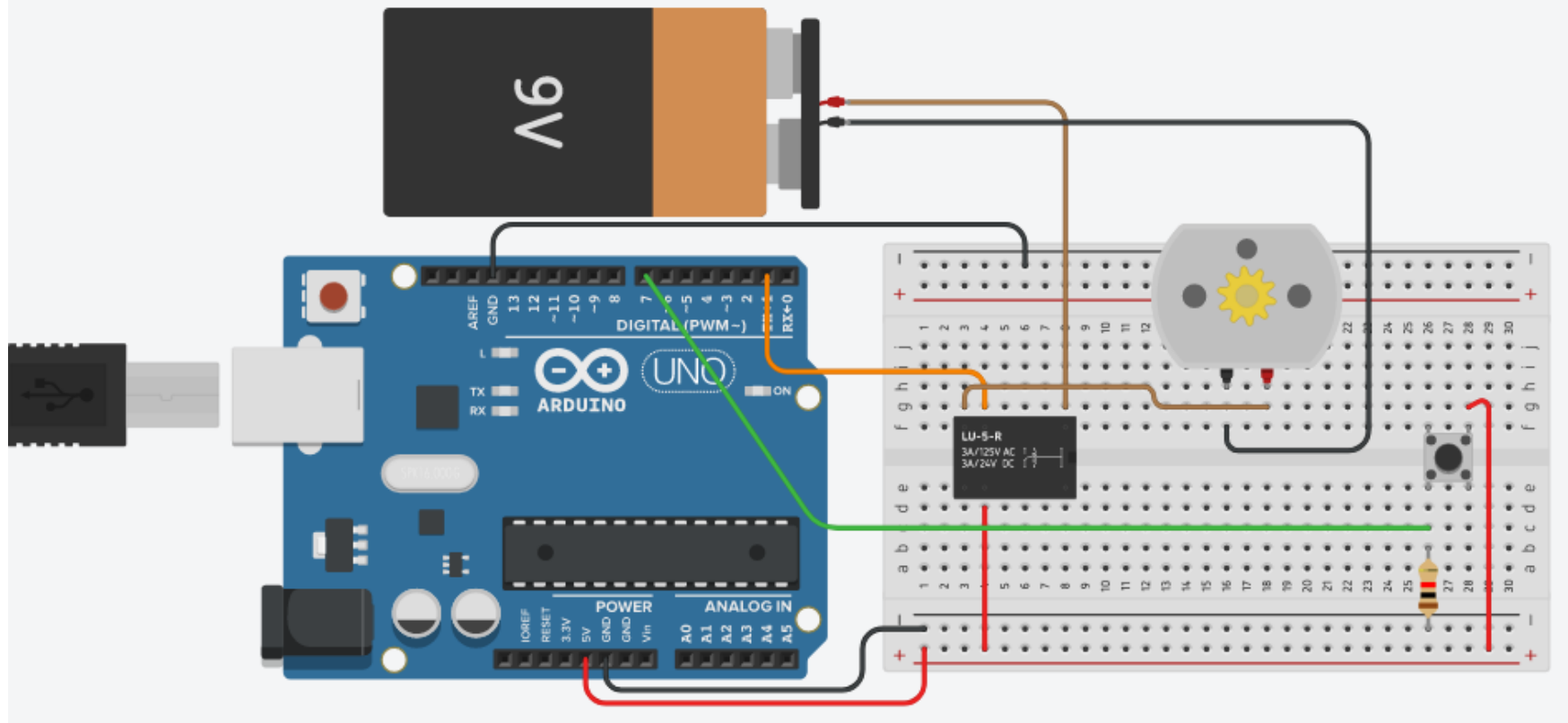
Sendo:

- $\omega$ : velocidade de rotação (rad/s);
- $J$ : Constante de proporcionalidade (rad/sV);
- $V$ : Tensão (V).

$$\omega = J \cdot V$$

# ELEMENTO DE ACIONAMENTO: RELÉ (Exemplo)

CONTROLE ON/OFF – SEM CONTROLE DE VELOCIDADE

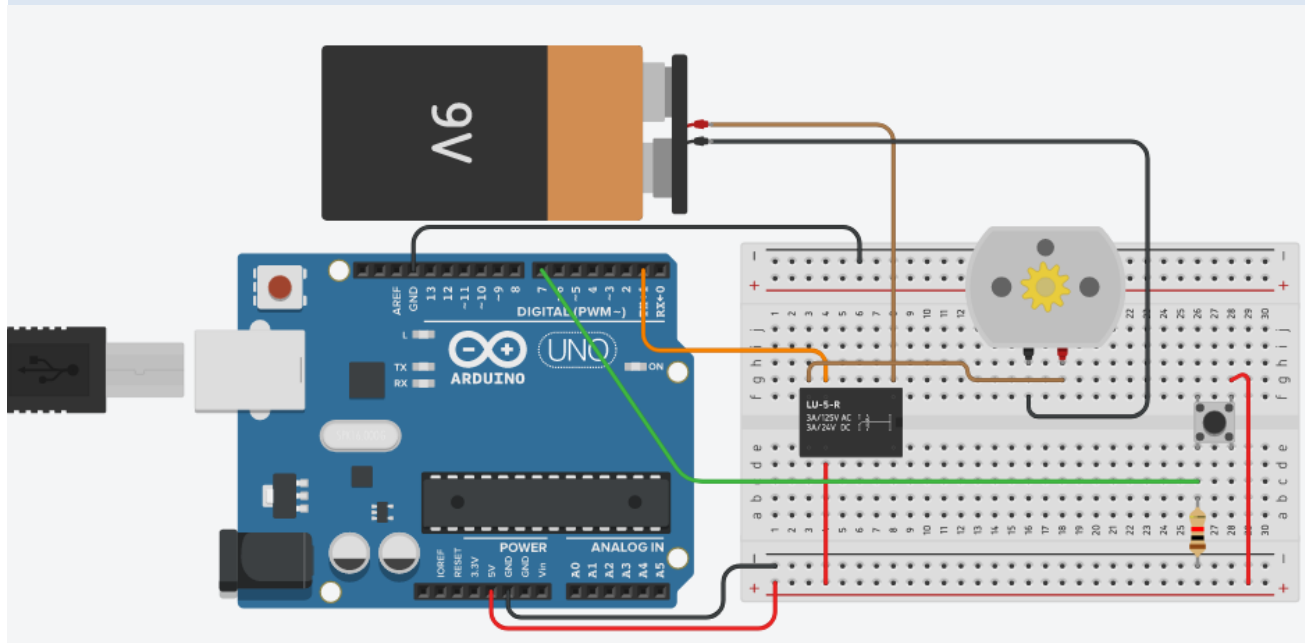


(Monte no Tinkercad e programe)

[Circuit design Smashing Krunk | Tinkercad](#)

# ELEMENTO DE ACIONAMENTO: RELÉ (Exemplo)

## CONTROLE ON/OFF – SEM CONTROLE DE VELOCIDADE



```
1  bool chave;  
2  void setup()  
3  {  
4      pinMode(7, INPUT);  
5      pinMode(1, OUTPUT);  
6  }  
7  
8  void loop()  
9  {  
10     chave = digitalRead(7);  
11  
12     if (chave == true)  
13     {  
14         digitalWrite(1, HIGH);  
15     }  
16     else  
17     {  
18         digitalWrite(1, LOW);  
19     }  
20 }
```

(Monte no Tinkercad e programe)

[Circuit design Smashing Krunk | Tinkercad](#)

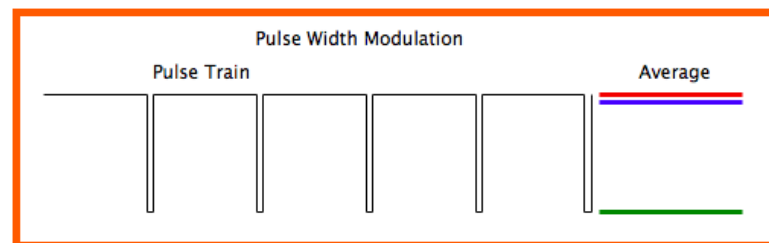
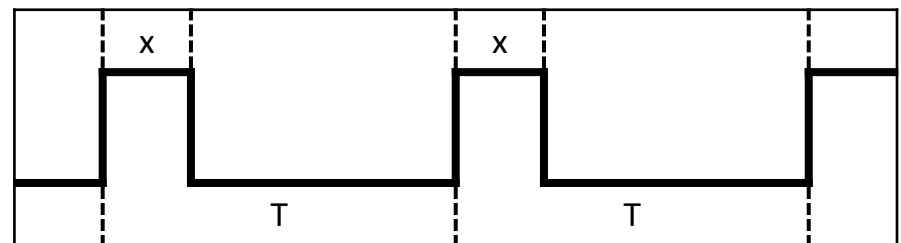
# ATUADOR: MOTOR DC (com escovas)

## PWM

PWM (Pulse Width Modulation – Modulação por Largura de Pulso) é uma técnica para obter resultados analógicos por meios digitais.

- Essa técnica consiste na geração de uma onda quadrada em que pode ser controlada a porcentagem do tempo em que a onda permanece em nível lógico alto;
- Esse tempo é chamado de “*Duty Cycle*” (Ciclo de trabalho) e sua alteração provoca mudança no valor médio da onda:
  - 0V (0% de Duty Cycle);
  - 5V (100% de Duty Cycle);
- O duty cycle é a razão do tempo em que o sinal permanece em HIGH sobre o tempo total de oscilação.

$$DutyCycle = \frac{x}{T} 100\%$$

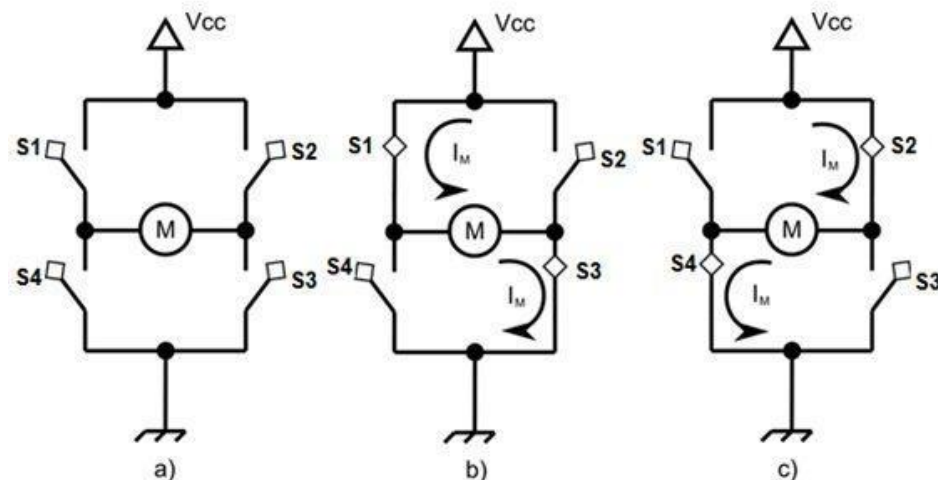




# ATUADOR: MOTOR DC (com escovas)

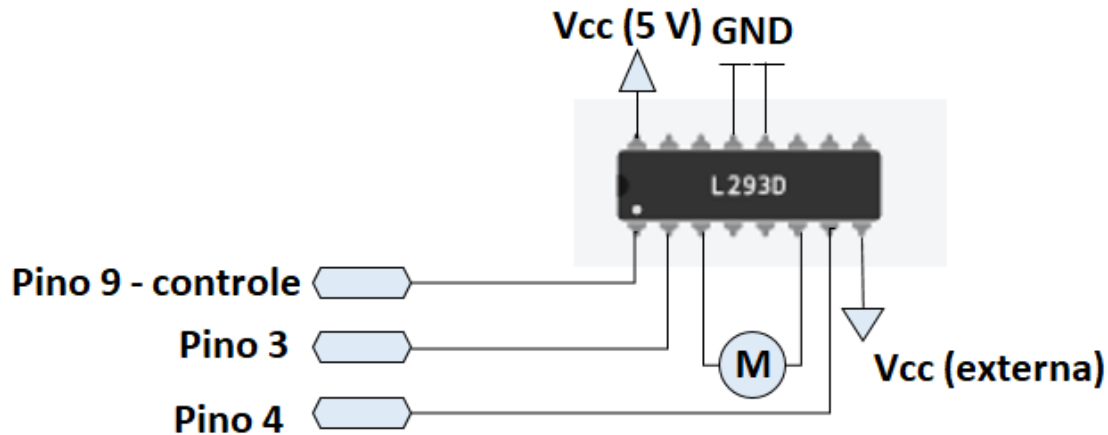
## Módulo Ponte H L298N

- Os motores DC (*“Direct Current”* ou corrente contínua) são cargas indutivas que, em geral, demandam uma quantidade de corrente superior à que as portas do Arduino conseguem fornecer.
- Uma solução simples seria utilizar um transistor para controlar o motor
  - Entretanto, não seria possível inverter o sentido de rotação do motor.
- Para ter o controle total do motor é utilizado uma ponte H:
  - a) Motor desligado;
  - b) O eixo do motor gira em um sentido;
  - c) O eixo do motor gira no sentido contrário.

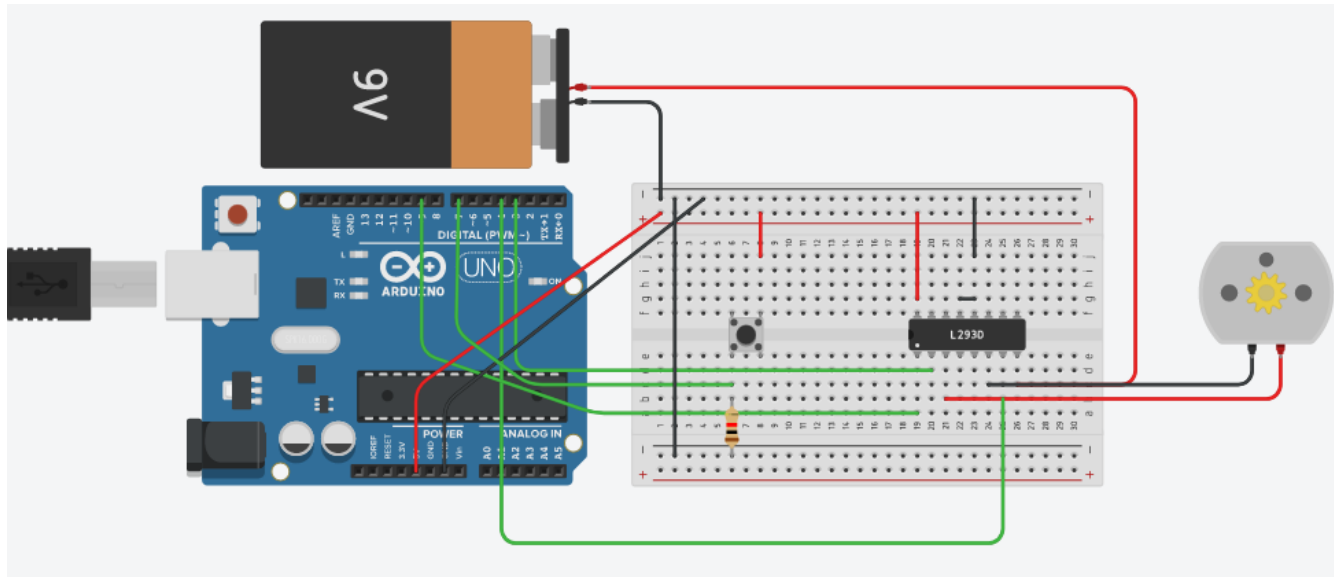


# ATUADOR: MOTOR DC (com escovas)

## Módulo Ponte H L293D



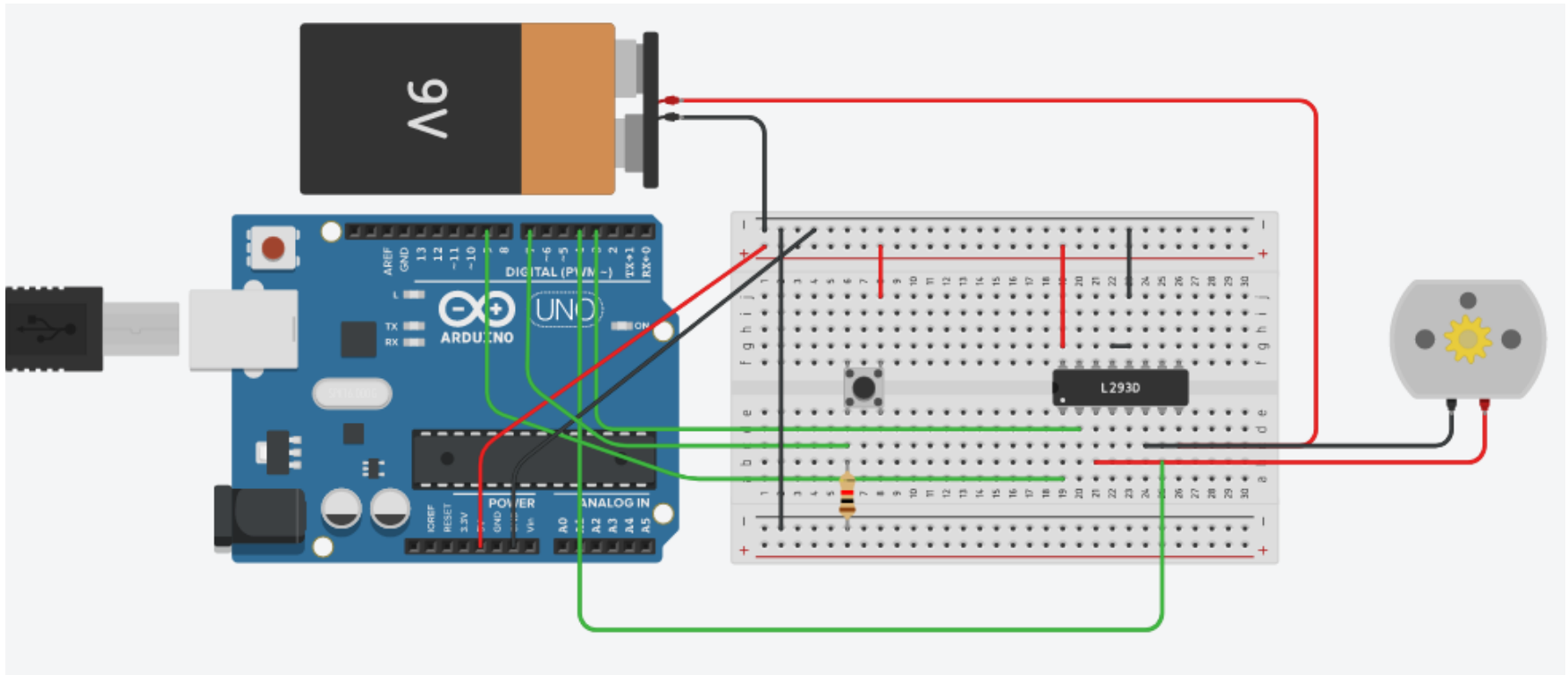
Pino 3 (0) e Pino 4 (1) - sentido de rotação 1  
Pino 3 (1) e Pino 4 (0) - sentido de rotação 2  
Pino 9 - Alimentação do motor



# ATUADOR: MOTOR DC (Exemplo)

## Módulo Ponte H L2930

### CONTROLE DE SENTIDO DE ROTAÇÃO – PONTE H



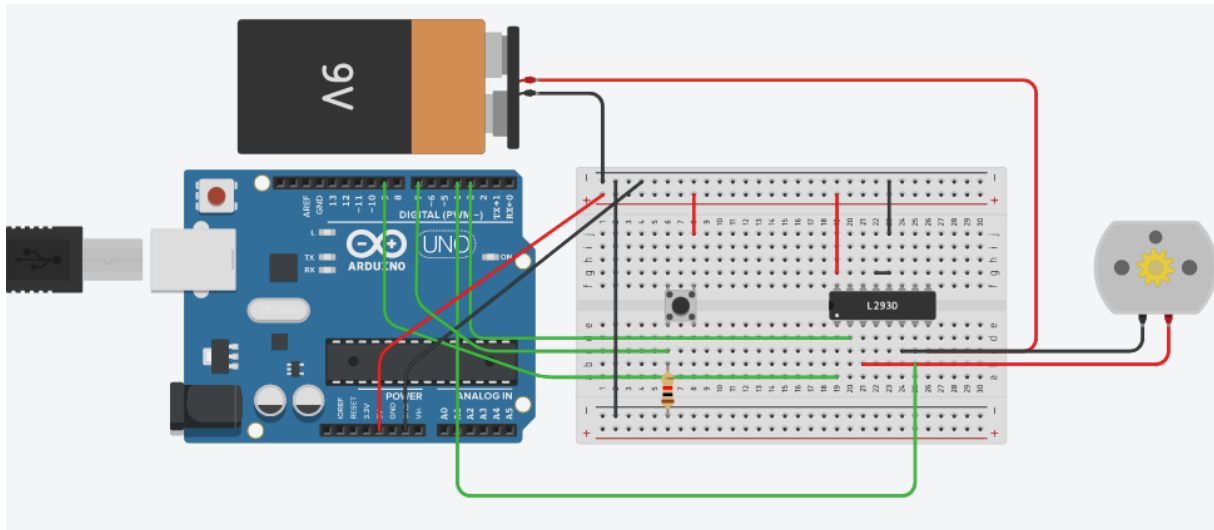
(Monte no Tinkercad e programe)

[Circuit design Acionamento motor ponteH | Tinkercad](#)

# ATUADOR: MOTOR DC (Exemplo)

## Módulo Ponte H L2930

### CONTROLE DE SENTIDO DE ROTAÇÃO – PONTE H



```
1  bool sentido;  
2  
3  void setup()  
4  {  
5      pinMode(3, OUTPUT);  
6      pinMode(4, OUTPUT);  
7      pinMode(7, INPUT);  
8      pinMode(10, OUTPUT);  
9  }  
10  
11 void loop()  
12 {  
13     sentido = digitalRead(7);  
14     if (sentido == true)  
15     {  
16         digitalWrite(4, HIGH);  
17         digitalWrite(3, LOW);  
18         digitalWrite(10, HIGH);  
19     }  
20     else  
21     {  
22         digitalWrite(3, HIGH);  
23         digitalWrite(4, LOW);  
24         digitalWrite(10, HIGH);  
25     }  
26 }
```

(Monte no Tinkercad e programe)

[Circuit design Acionamento motor ponteH | Tinkercad](#)

# CIRCUITOS DE POTÊNCIA: MOTOR DC (com escovas)

## Módulo Ponte H L298N

O Módulo Ponte H L298N oferece uma solução simples para o controle de pequenos motores DC

- Permite o controle do sentido do giro do rotor do motor;
- Permite o controle da velocidade do motor;
- Pode ser utilizado para controlar:
  - 2 motores DC;
  - 1 motor de passo bipolar;
- Possui circuito de proteção térmico;
- O controle da lógica por sinais TTL.

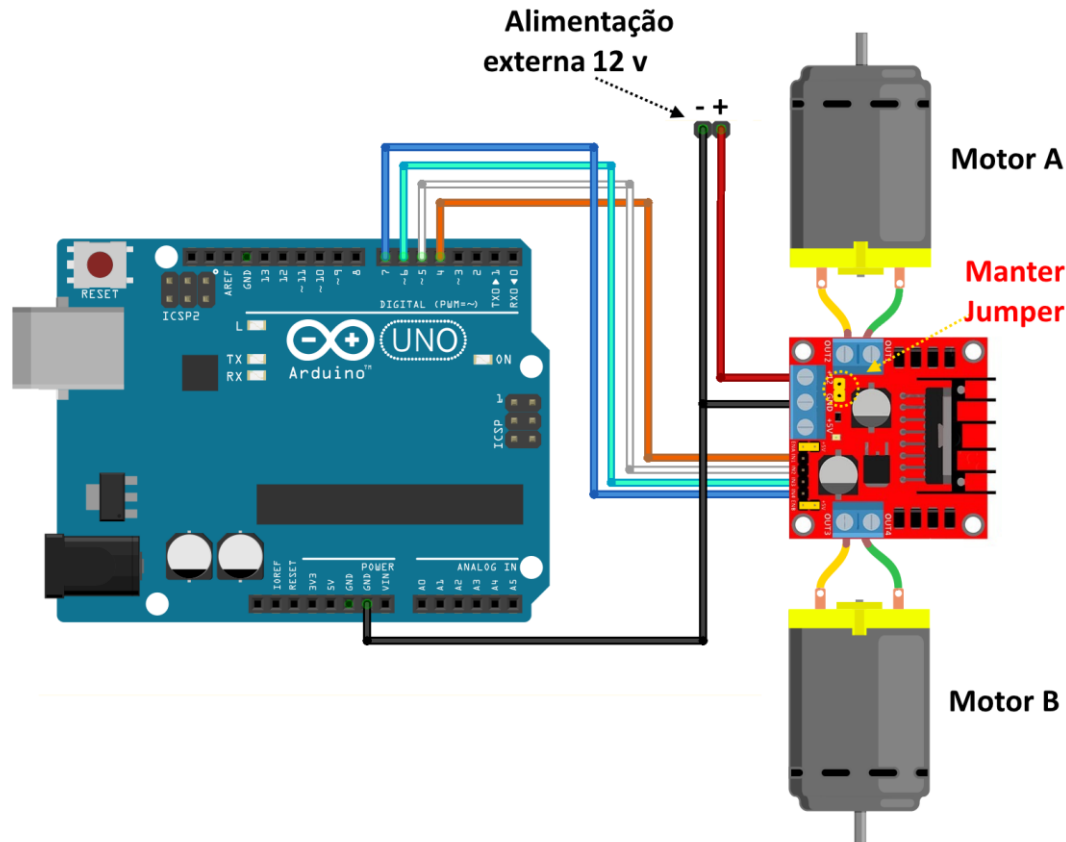


### NOTAS DE MANUSEIO:

- Manter a alimentação desligada enquanto altera as conexões elétricas;
- Deixar a placa sobre uma superfície isolante.

# CIRCUITOS DE POTÊNCIA: MOTOR DC (com escovas)

## Módulo Ponte H L298N



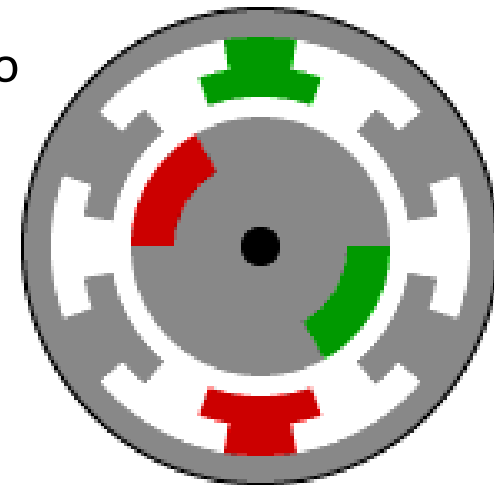
## ATUADOR: MOTOR DC (sem escovas)

Em motor DC (sem escovas), um rotor com ímãs permanentes gira quando a corrente que passa pelas bobinas do estator criam um campo magnético.

- Dispensa escovas, aumentando a vida útil do motor;
- O comutador é eletrônico, e precisa estimar a posição do rotor para controlar corretamente a alimentação das bobinas do estator.

Da mesma forma que o motor DC com escova, a corrente nas bobinas pode ser controlada por um PWM.

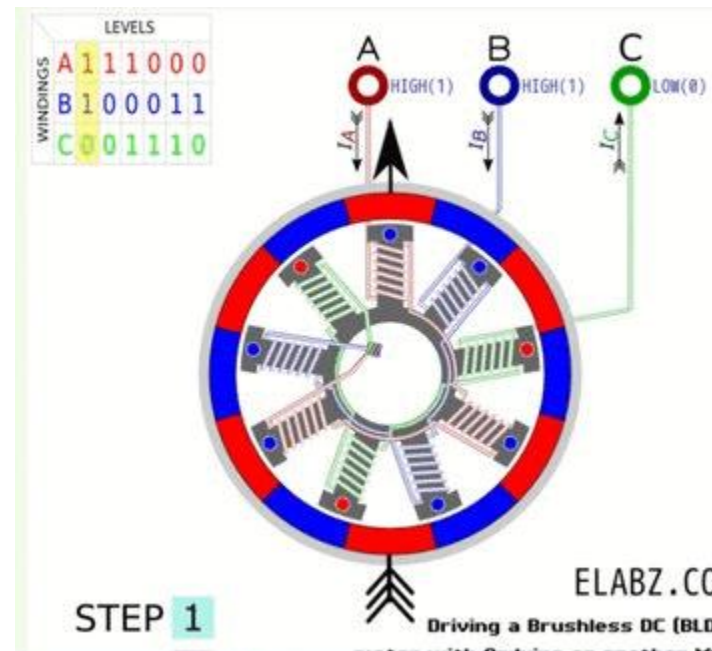
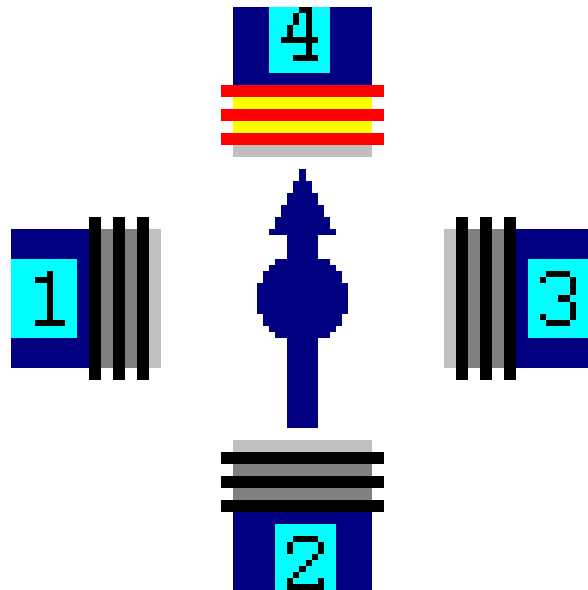
- A posição do rotor pode ser obtida através de um sensor do tipo “encoder óptico”.



# ATUADOR: MOTOR DE PASSO

Uma variação do motor DC (sem escovas) é o motor de passo.

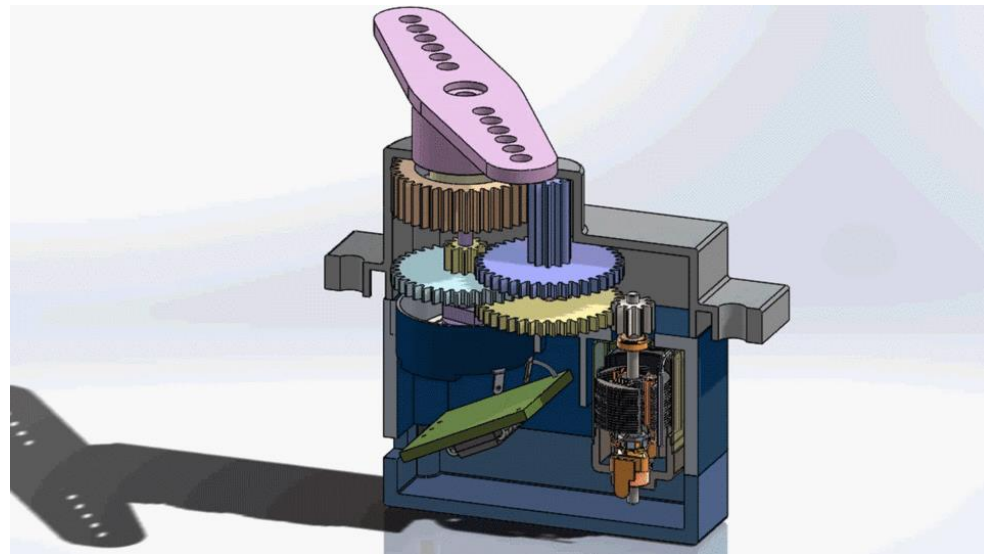
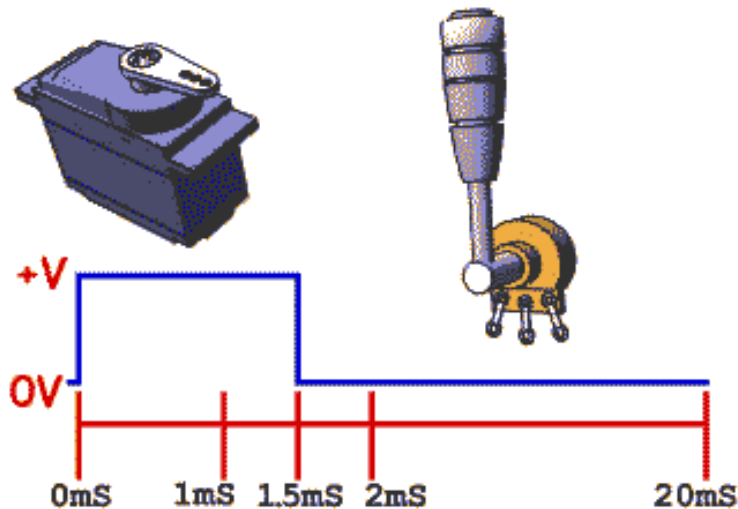
- Um arranjo diferenciado do enrolamento das bobinas geram um avanço angular do rotor constante;
- Utilizado quando o motor deve realizar movimentos controlados.





# ATUADOR: SERVO MOTOR

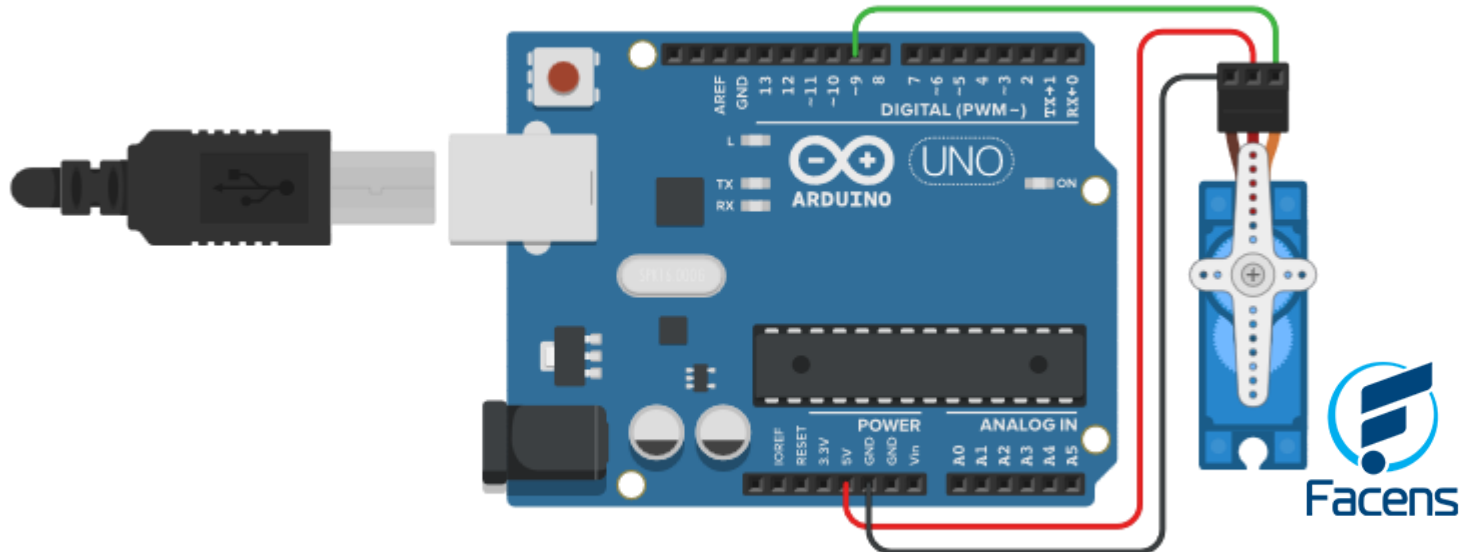
- Utiliza um motor com uma caixa de engrenagem para aumentar o torque e a precisão
  - Motores CA são utilizados quando é necessário um torque maior;
- Uma eletrônica com sensor e PID permite o controle do posicionamento do rotor;
- O controle (externo) é feito pela largura de um pulso (1ms ~2ms) repetido a uma cadência de 20ms.



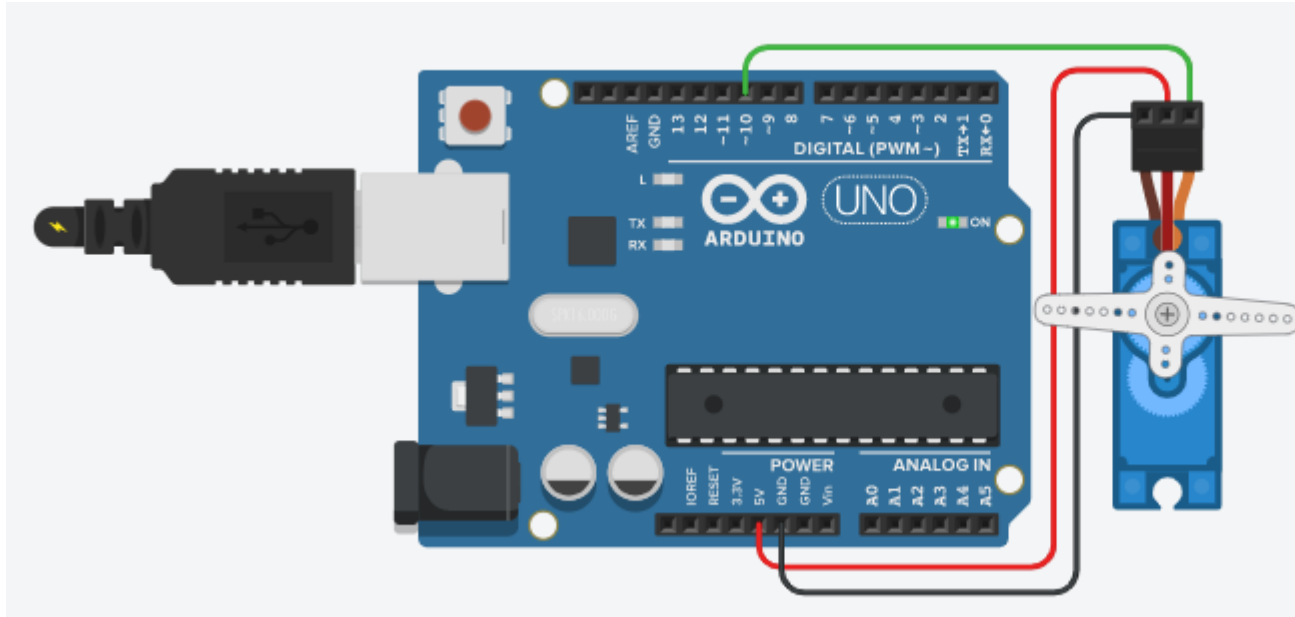
# ATUADOR: SERVO MOTOR (Exemplo)

## Micro Servo Motor 9g SG90

- Tensão de Operação: 3,0 – 7,2V
- Velocidade: 0,12 seg/60 Graus
  - 4,8V sem carga
- Torque: 1,2 kg.cm (4,8v) e 1,6 kg.cm (6,0v)
- Tipo de Engrenagem: Nylon
- Peso: 9g



# ATUADOR: SERVO MOTOR (Exemplo)



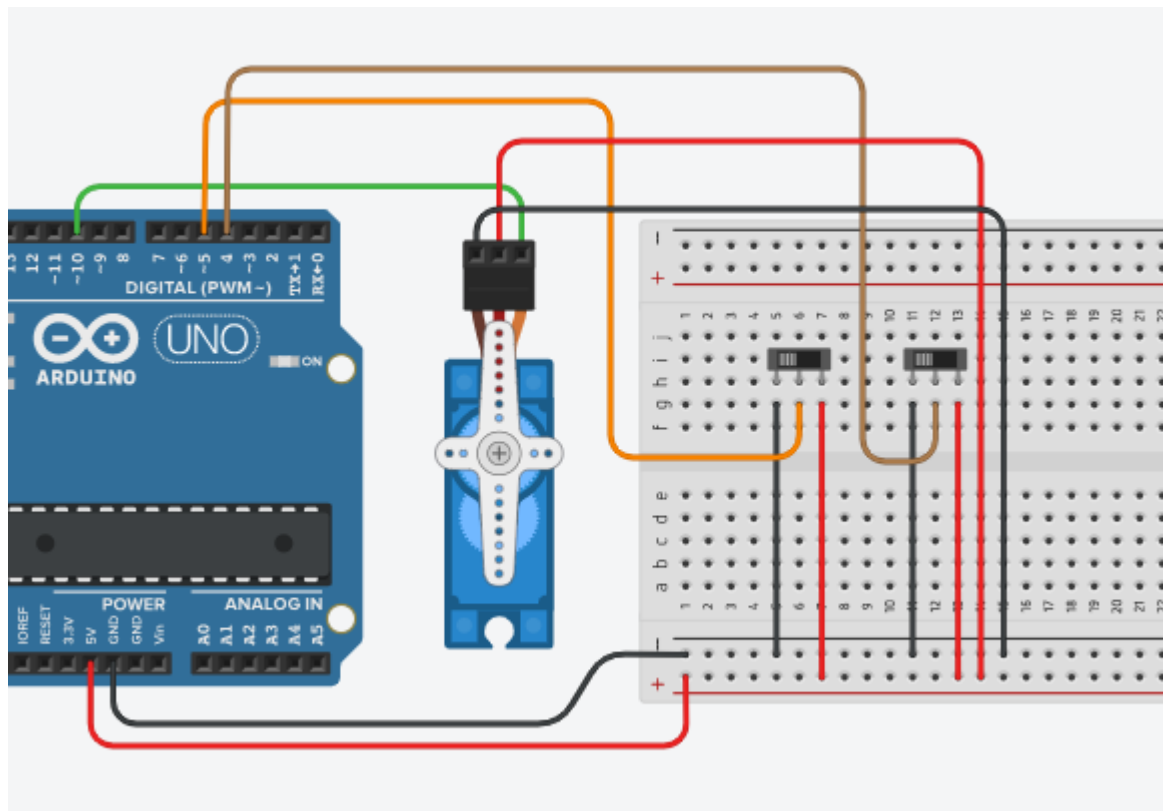
## Texto

```
1 #include<Servo.h>
2 Servo servo10;
3
4 void setup()
5 {
6     servo10.attach(10);
7 }
8
9 void loop()
10 {
11     servo10.write(90);
12 }
```

# ATUADOR: SERVO (Desafio)

## Servo

### ACIONAMENTO DO SERVO POR CHAVES



chave1	chave2	Ângulo
false	false	45°
false	true	90°
true	false	135°
true	true	180°

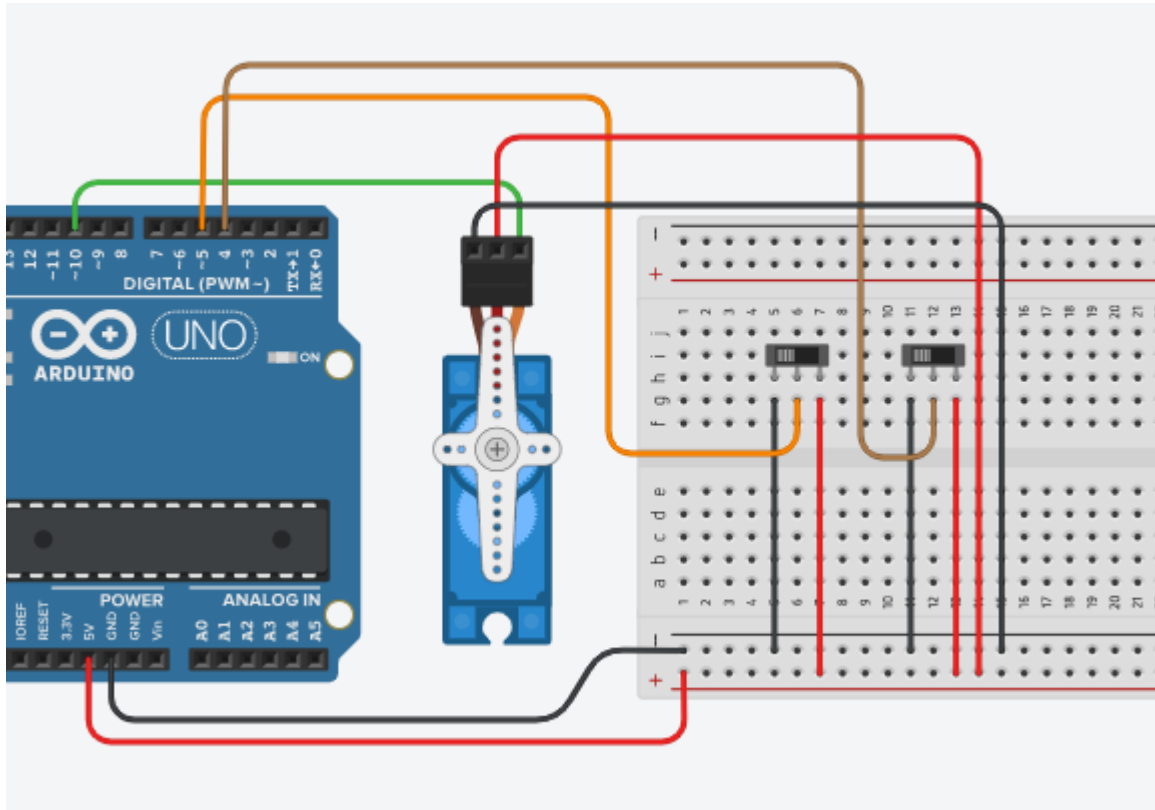
(Monte no Tinkercad e programe)

[Circuit design Acionamento servo chaves | Tinkercad](#)

# ATUADOR: SERVO (Desafio)

## Servo

### ACIONAMENTO DO SERVO POR CHAVES



```
1  #include<Servo.h>
2  Servo servo10;
3  bool chave1;
4  bool chave2;
5
6  void setup()
7  {
8      servo10.attach(10);
9  }
10
11 void loop()
12 {
13     chave1 = digitalRead(5);
14     chave2 = digitalRead(4);
15
16     if (chave1 == true)
17     {
18         if (chave2 == true)
19         {
20             servo10.write(180);
21         }
22         else
23         {
24             servo10.write(135);
25         }
26     }
27     else
28     {
29         if (chave2 == true)
30         {
31             servo10.write(90);
32         }
33         else
34         {
35             servo10.write(45);
36         }
37     }
38 }
```

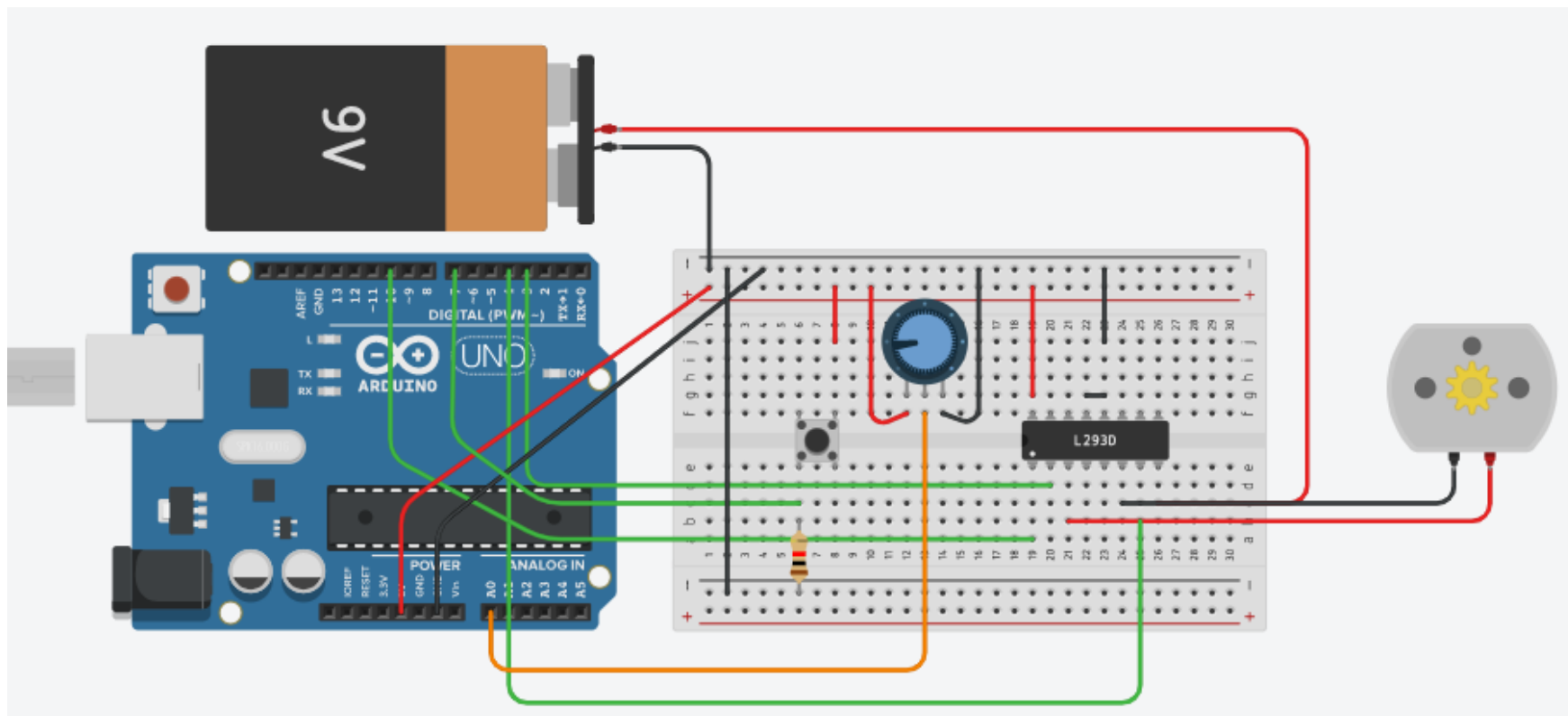
(Monte no Tinkercad e programe)

[Circuit design Acionamento servo chaves | Tinkercad](#)

# ATUADOR: MOTOR DC (Desafio)

## Módulo Ponte H L293D

### CONTROLE DE SENTIDO DE ROTAÇÃO – PONTE H



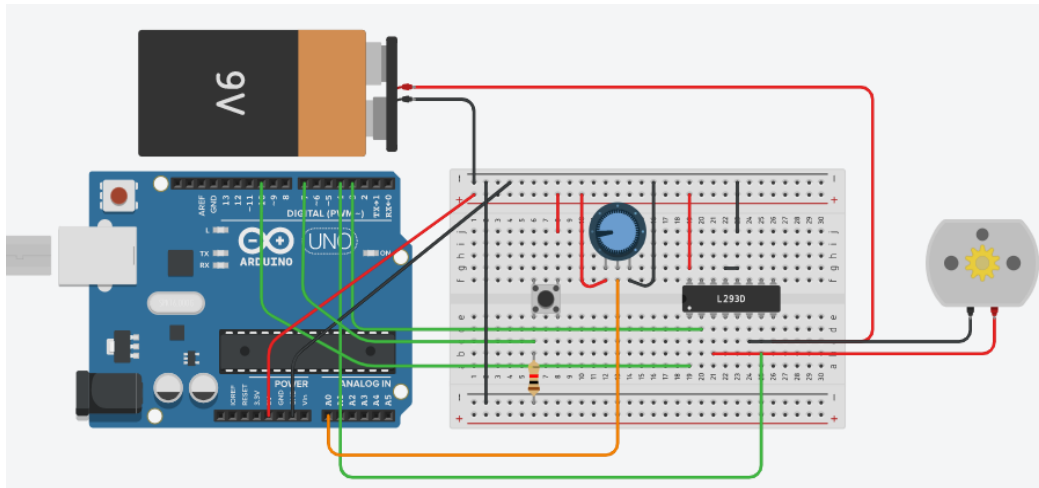
**(Monte no Tinkercad e programe)**

[Circuit design Acionamento motor ponteH potenciômetro | Tinkercad](#)

# ATUADOR: MOTOR DC (Desafio)

## Módulo Ponte H L293D

### CONTROLE DE SENTIDO DE ROTAÇÃO – PONTE H



```
1  int vel;  
2  
3  void setup()  
4  {  
5      pinMode(10, OUTPUT);  
6      pinMode(4, OUTPUT);  
7      pinMode(3, OUTPUT);  
8      pinMode(7, INPUT);  
9      pinMode(A0, INPUT);  
10 }  
11  
12 void loop()  
13 {  
14     vel = analogRead(A0);  
15     vel = map(vel, 0, 1023, 0, 255);  
16     if (digitalRead(7) == HIGH)  
17     {  
18         digitalWrite(3, HIGH);  
19         digitalWrite(4, LOW);  
20         analogWrite(10, vel);  
21     }  
22     if (digitalRead(7) == LOW)  
23     {  
24         digitalWrite(3, LOW);  
25         digitalWrite(4, HIGH);  
26         analogWrite(10, vel);  
27     }  
28 }
```

(Monte no Tinkercad e programe)

[Circuit design Acionamento motor ponteH potenciômetro | Tinkercad](#)