



# Programação e IoT

## Laboratório 04 - ATUADORES

# OBJETIVO

- Conhecer os periféricos de saída utilizados em IoT;
- Saber usar e aplicar atuadores discretos e não discretos;
- Conhecer a função PWM;
- Aprofundar sobre os recursos do Tinkercad.

# MATERIAL UTILIZADO

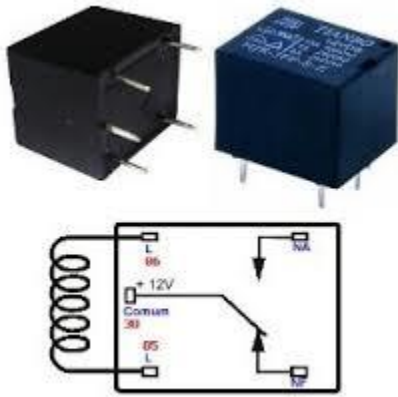
- Computador com:
  - Sistema operacional Windows;
  - Porta USB;
  - Acesso à internet;
- Tinkercad:
  - <https://www.tinkercad.com/>

# Programação e IoT

## Laboratório 04 - ATUADORES

TEORIA

# TEORIA – Dispositivos discretos de saída



Relés



Leds



Eletroválvula



Eletroválvula



Relé fotocélula

# TEORIA – Dispositivos discretos de saída



Motor de passo



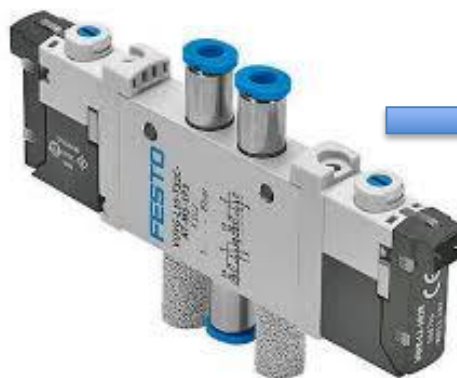
Motor de passo



Transistor



Eletroválvula pneumática



Eletroválvula pneumática



# TEORIA – Dispositivos não discretos de saída



Leds – intensidade do brilho



Servo motor



Transistor



Motor DC



Inversor de frequência



Motor AC



# LISTA DE FUNÇÕES DO ARDUINO JÁ TRABALHADAS

As seguintes funções já foram utilizadas nos sketches do Arduino:

- `pinMode(numeroPino, modo);`
  - `numeroPino`: 0 a 13
  - `modo`: pode ser INPUT, OUTPUT ou INPUT\_PULLUP
- `digitalRead(numeroPino);`
- `analogRead(nomePino);`
  - `nomePino`: A0 a A5
- `delay(ms);`
  - `ms`: valor do tempo de espera em milissegundos
- `Serial.print(val);`
- `Serial.println(val);`
  - `val`: texto (entre aspas dupla) ou uma variável de programa
- `Serial.begin(9600);`

# SINAIS DIGITAIS – digitalWrite()

Atribui um valor à um pino digital específico

- Sintaxe
  - digitalWrite(**numeroPino**, **valor**)
- Parâmetros
  - **numeroPino**: número do pino de saída
  - **valor**: HIGH ou LOW
- Função sem retorno

```
void setup() {  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

## SINAIS ANALÓGICOS – analogWrite()

Aciona uma onda PWM em um pino **digital**

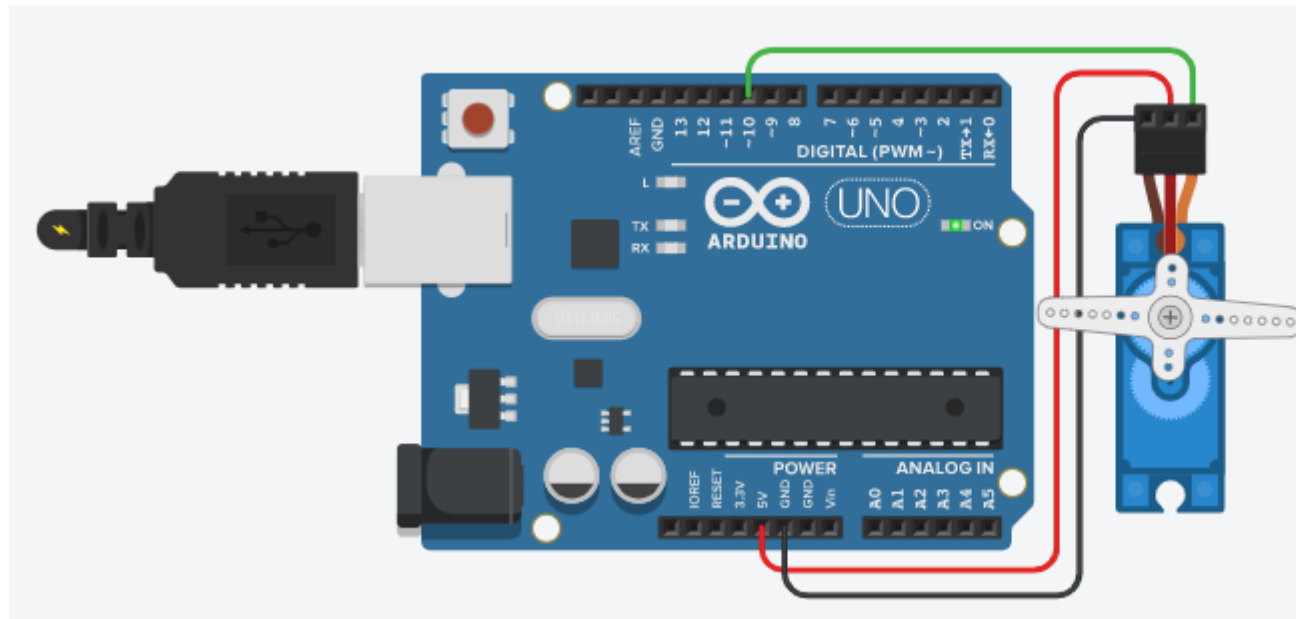
- Essa função funciona nos pinos 3, 5, 6, 9, 10 e 11
- A frequência é de 490Hz
- Nos pinos 5 e 6 a frequência é de 980Hz
  - Sofrem interferência das funções millis() e delay(), que compartilham o timer
- Sintaxe
  - analogWrite(**numeroPino**, **dutyCycle**);
- Parâmetros
  - **numeroPino**: número do pino de entrada
    - 3, 5, 6, 9, 10 ou 11
  - **dutyCycle**: duty cycle do PWM
    - 0 (0%) ~ 255 (100%)
- Sem Retorno

```
// LED conectado ao pino digital 9
int ledPin = 9;
// potenciômetro conectado ao pino analógico 3
int analogPin = 3;
// variável para guardar o valor lido
int val = 0;

void setup() {
    // configura o pino como saída
    pinMode(ledPin, OUTPUT);
}

void loop() {
    // lê o pino de entrada analógica
    val = analogRead(analogPin);
    // analogRead retorna valores de 0 a 1023
    // analogWrite recebe de 0 a 255
    analogWrite(ledPin, val / 4);
}
```

# PROGRAMAÇÃO DO SERVO



## 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 }
```

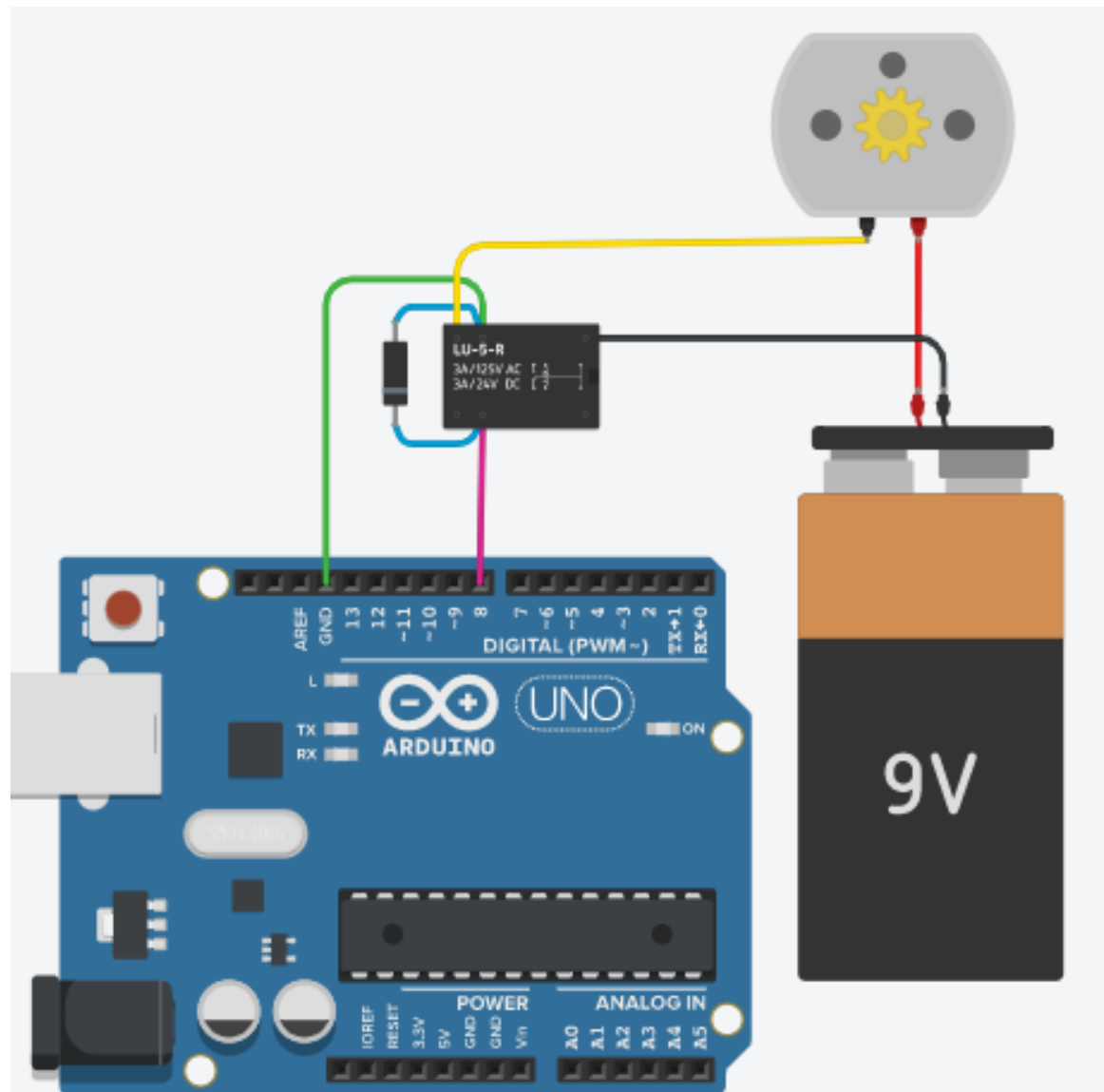
# Programação e IoT

## Laboratório 04 - ATUADORES

PROCEDIMENTOS

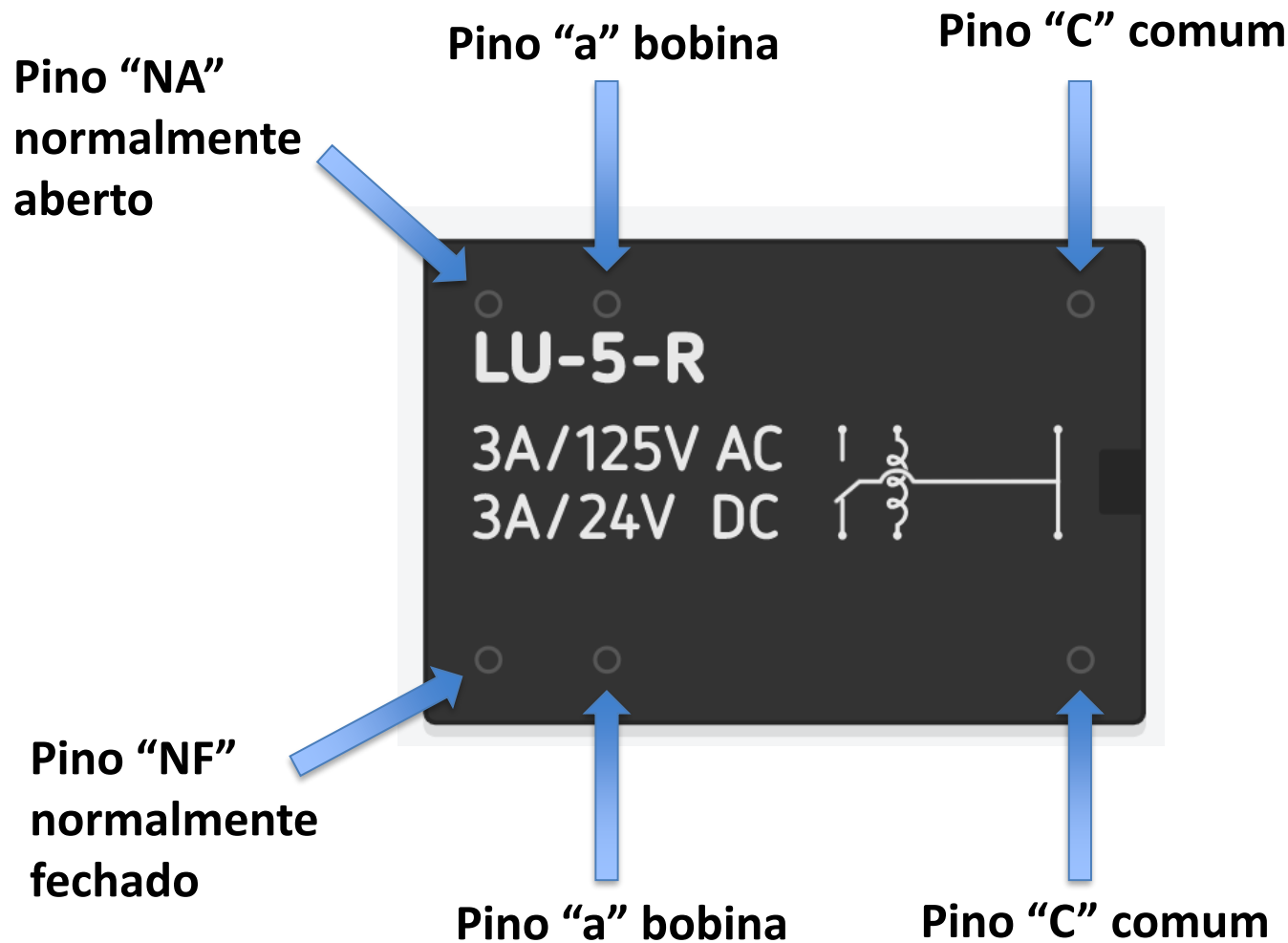
# PROCEDIMENTO 1 – USANDO RELÉ

1. Faça um programa em que o motor fica ligado por 3 segundos e desligado por 1 segundo.



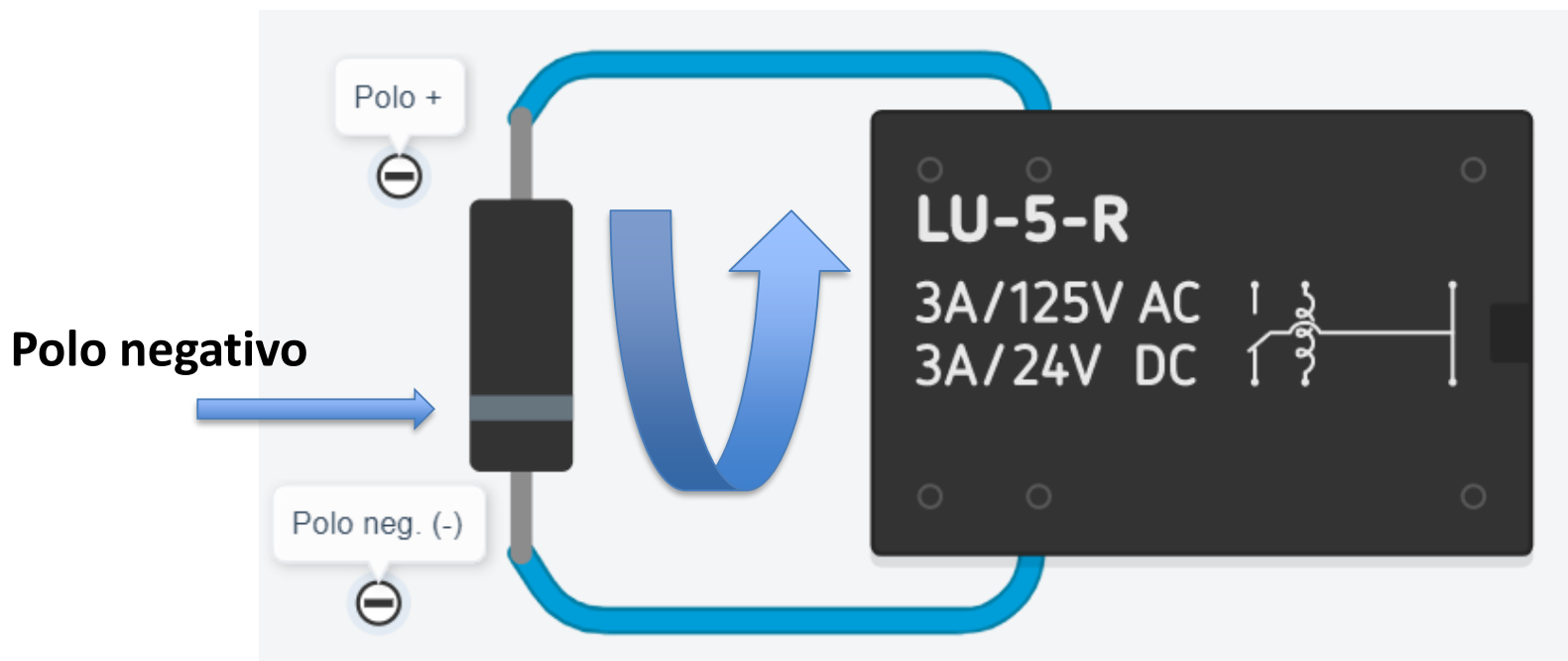
# PROCEDIMENTO 1 – USANDO RELÉ (Teoria)

a) O Tinkercad possui o LU-5-R, que tem a seguinte configuração de pinos:



# PROCEDIMENTO 1 – USANDO RELÉ (Cuidados)

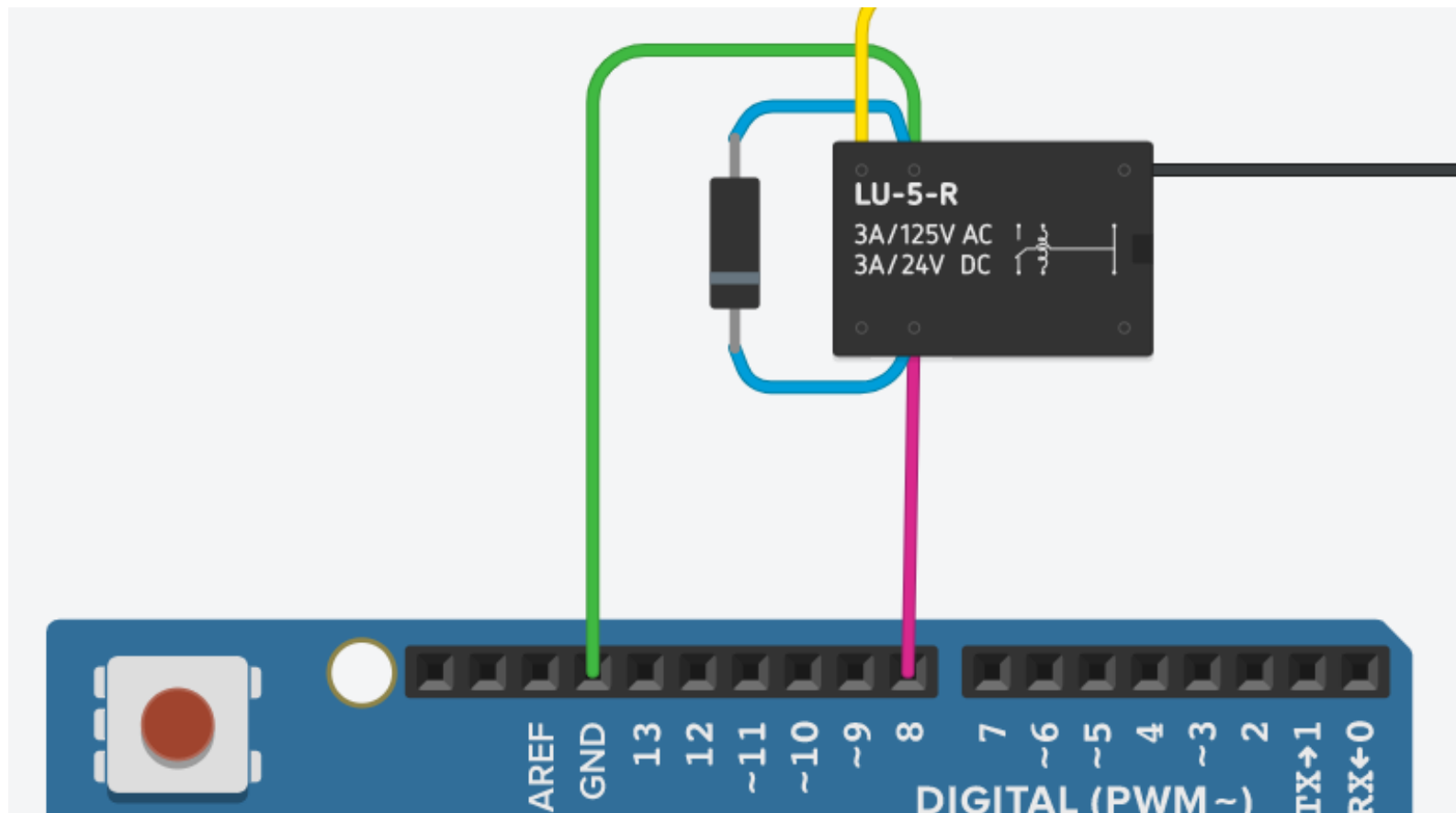
- b) Adicione um diodo invertido aos polos da bobina do relé
- Quando o relé é energizado, a bobina acumula energia magnética
  - Ao tentar desligar o relé, a bobina tenta manter a corrente
    - Gera uma tensão reversa
  - A função desse diodo é proteger o Arduino contra tensão reversa gerada pelo relé
    - Conhecido como “diodo de roda livre”!
    - O diodo absorve a tensão reversa gerada pela bobina do relé



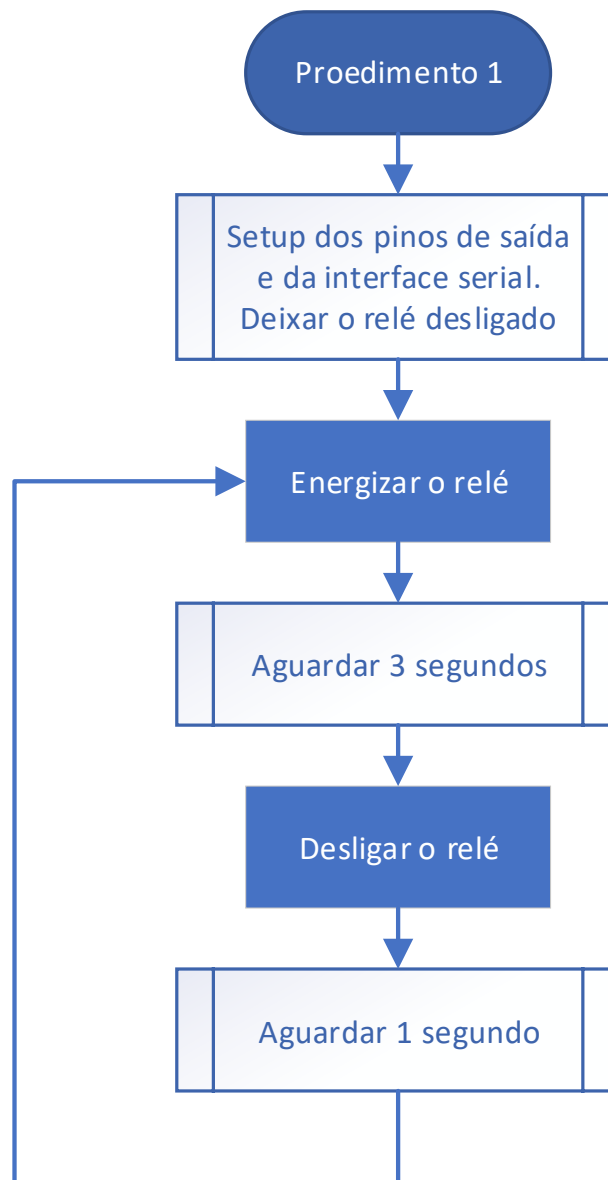


# PROCEDIMENTO 1 – USANDO RELÉ (Cuidados)

- c) Cuidado na conexão do relé ao Arduino
- Pino 8 na bobina “a”
  - GND na outra extremidade da bobina “a”



# PROCEDIMENTO 1 – USANDO RELÉ (Algoritmo)





# PROCEDIMENTO 2 – RELÉS CONTROLANDO MOTOR

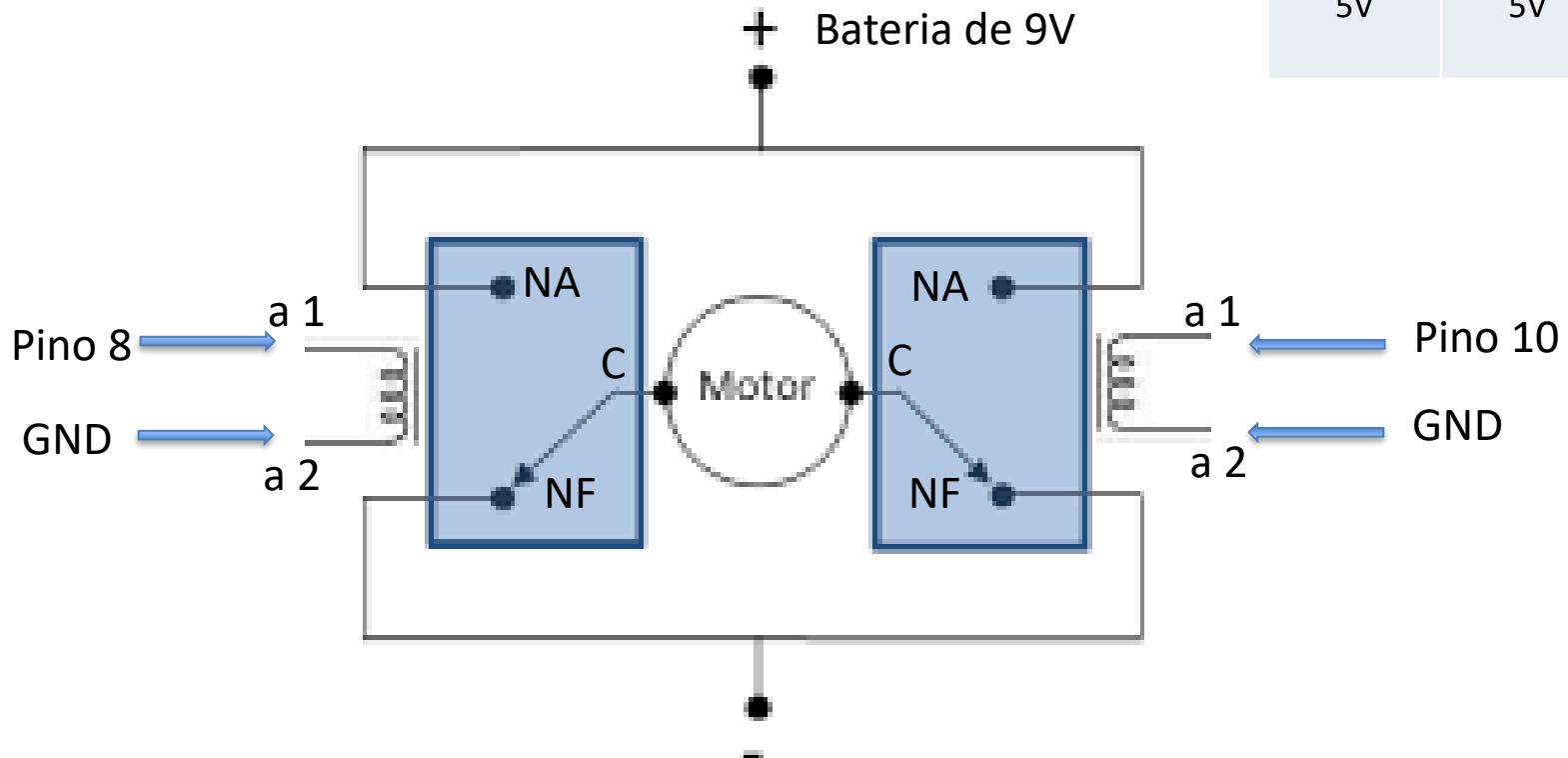
2. Modifique o projeto do Procedimento 1 para acrescentar um segundo relé

a) O motor DC será controlado pelo Arduino

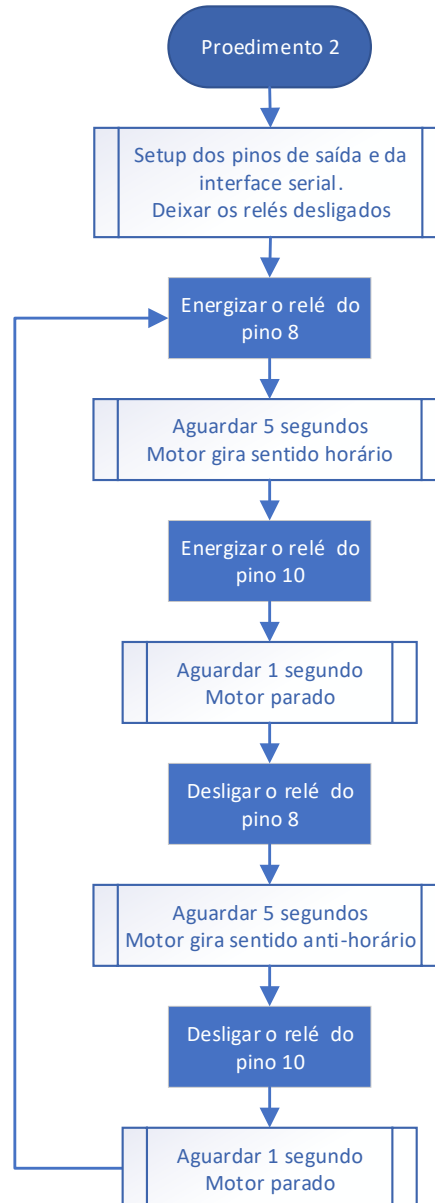
b) No loop, o motor deve:

- Girar no sentido horário por 5 segundos
- Parar por 1 segundo
- Girar no sentido anti-horário por 5 segundos
- Parar por 1 segundo

Pino 8	Pino 10	Motor
0V	0V	Off
0V	5V	Girando 
5V	0V	Girando 
5V	5V	Off



# PROCEDIMENTO 2 – RELÉS CONTROLANDO MOTOR



## PROCEDIMENTO 3 – CONTROLE DE UM SERVO MOTOR

3. Monte um circuito que faça o servo motor variar de  $0^\circ$  a  $90^\circ$  e depois  $90^\circ$  para  $0^\circ$ 
  - Variar o ângulo de  $1^\circ$  a cada 20ms

