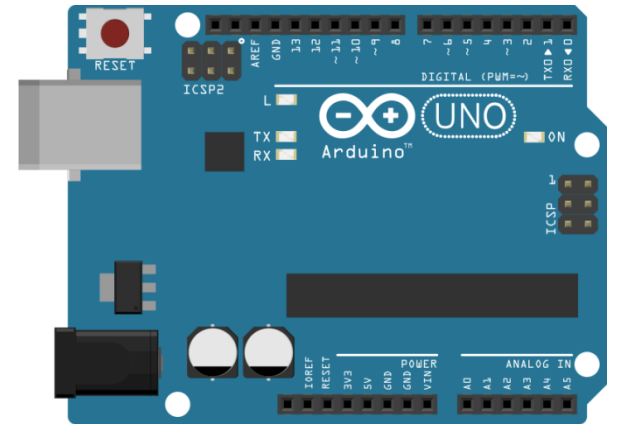


Workshop

Introdução ao Arduino



Arduino – História

O professor italiano, Massimo Banzi, desejava ensinar programação de computadores, mas aplicando este conhecimento a projetos de arte, automação e robótica.

Deparou-se, então, com um grande problema:



Não existiam placas que fossem didáticas e ao mesmo tempo poderosas e baratas.

Decidiu então desenvolver a sua própria placa juntamente com o engenheiro espanhol David Cuartielles, Massimo e com a ajuda de um de seus alunos chamado David Mellis, que ficou responsável pela linguagem de programação do **Arduino**.

Arduino - Modelos mais comuns

Arduino UNO



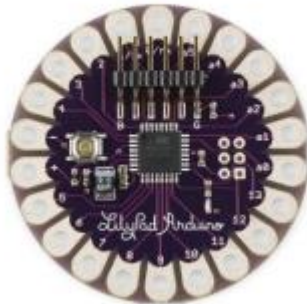
Arduino Mega



Arduino Mini



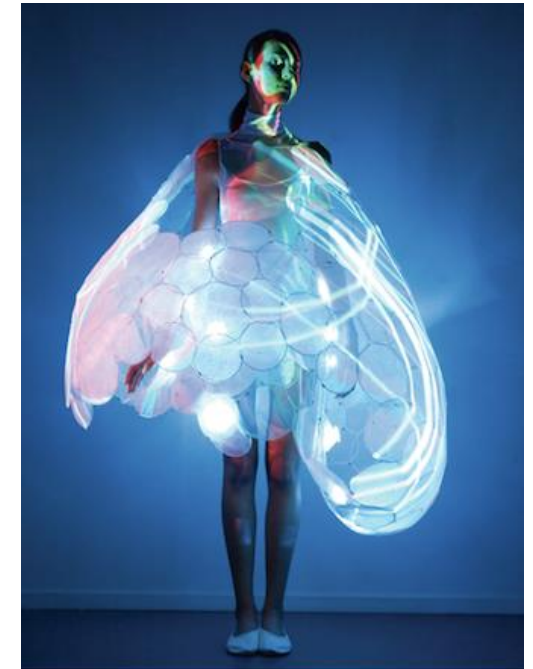
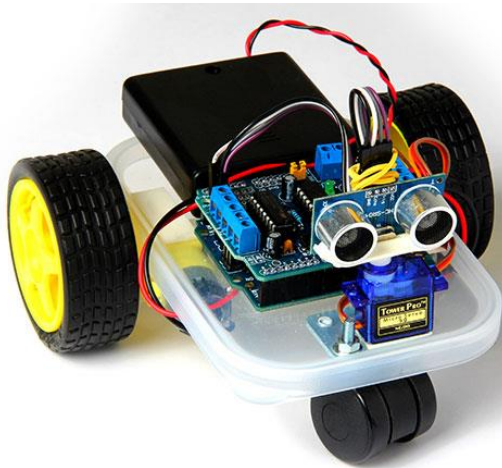
Lilypad



Arduino Nano



Arduino – Aplicações



Arduino

Input (Ler)



Potenciômetro



Botão



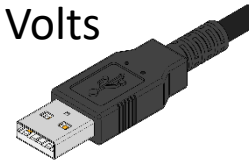
Sensor de luz



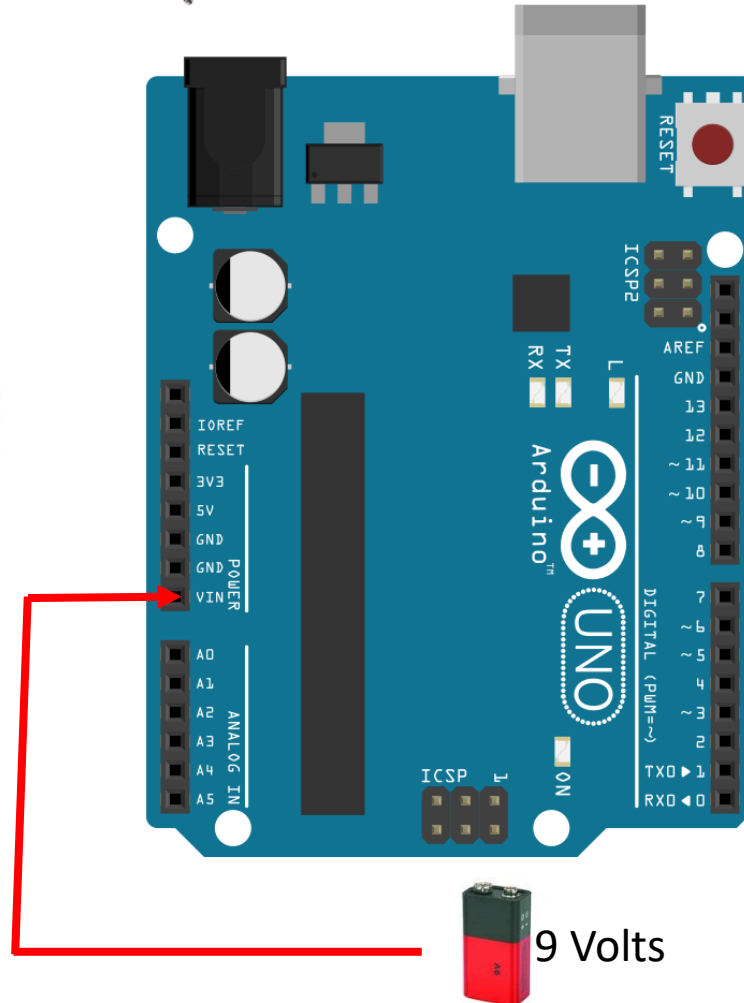
Sensor de temperatura



12 Volts



5 Volts



9 Volts

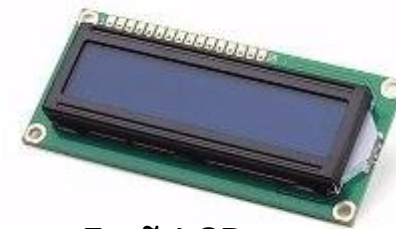
Output (Escrever)



Led



Buzzer

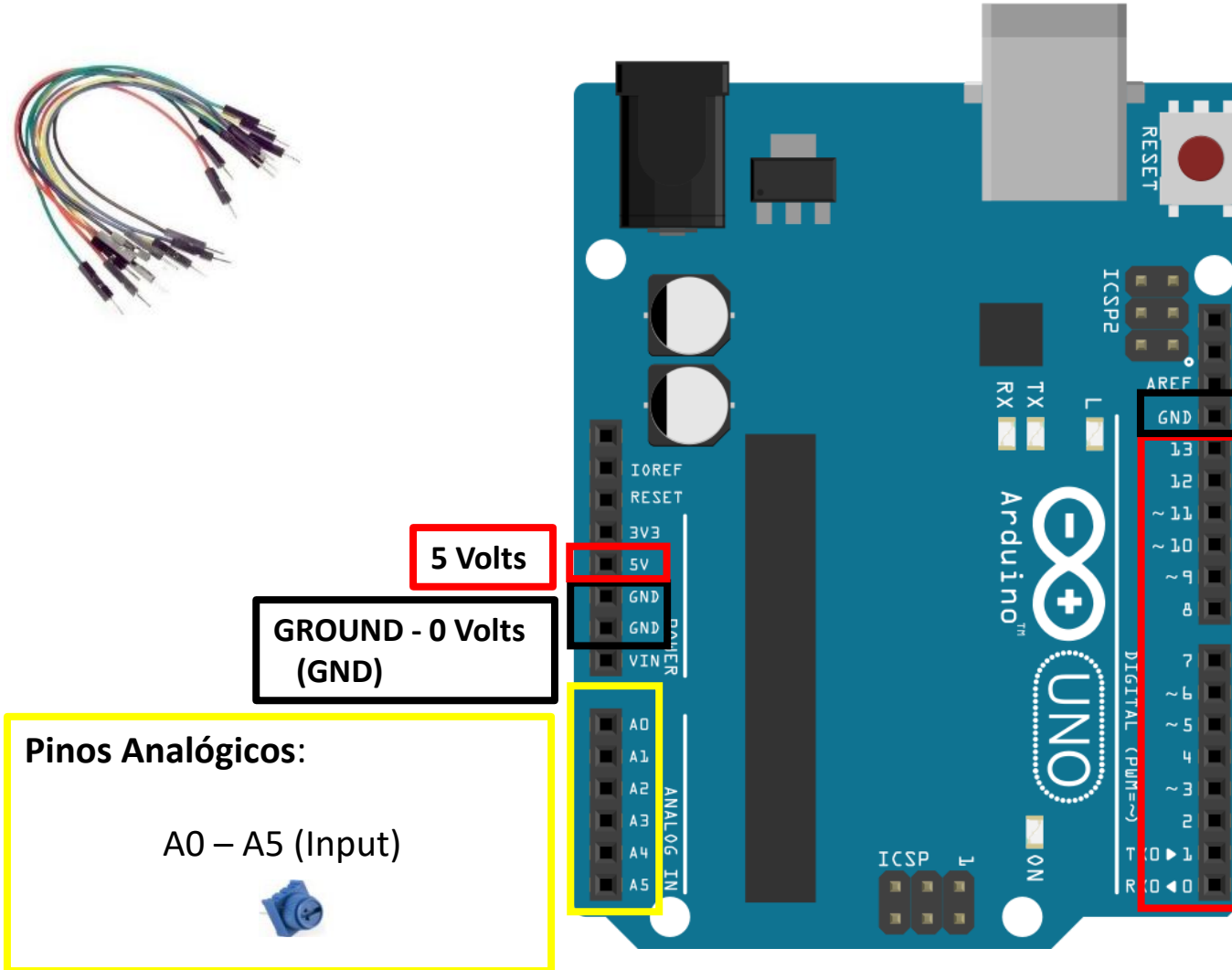


Ecrã LCD



Motor DC

Arduino



Pinos Analógicos:

A0 – A5 (Input)



Pinos Digitais:

0 – 13 (Input / Output)

~3

~5

~6

PWM

~9

~10

~11

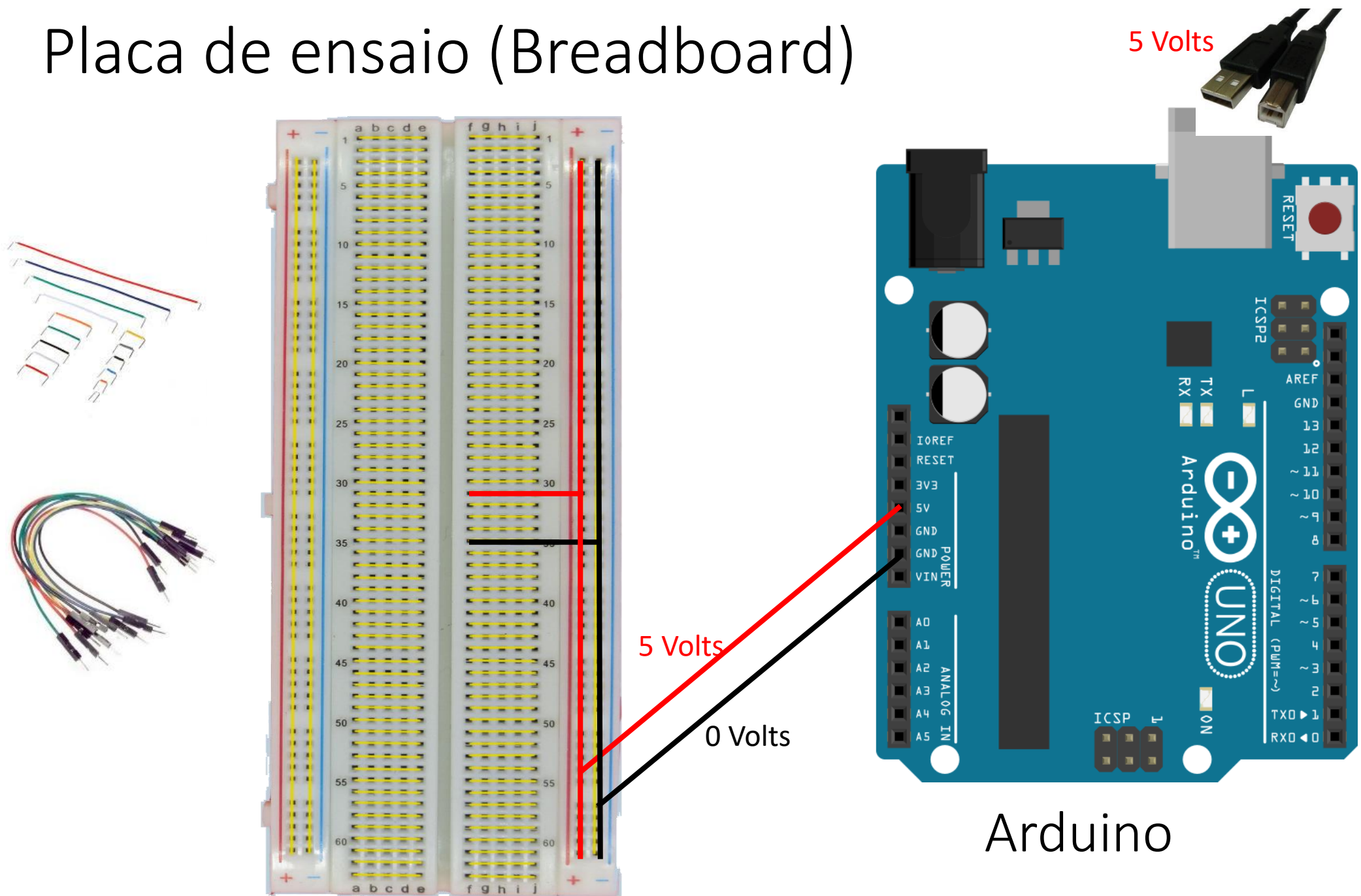


Exemplo:

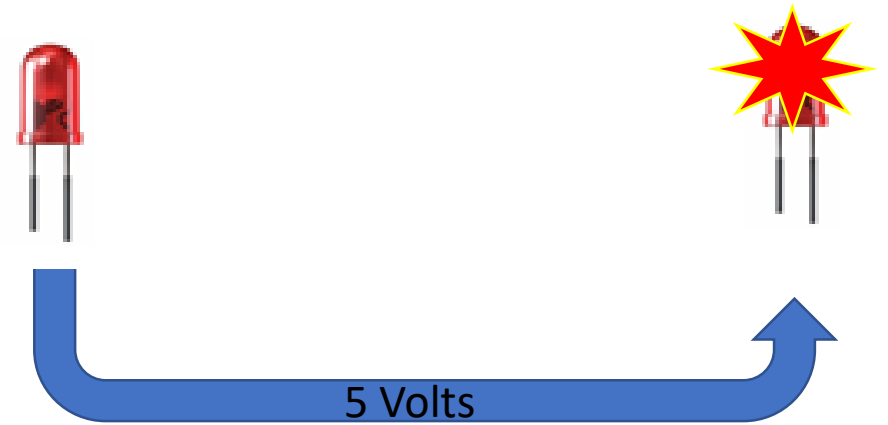
Para variar a velocidade
de um motor DC



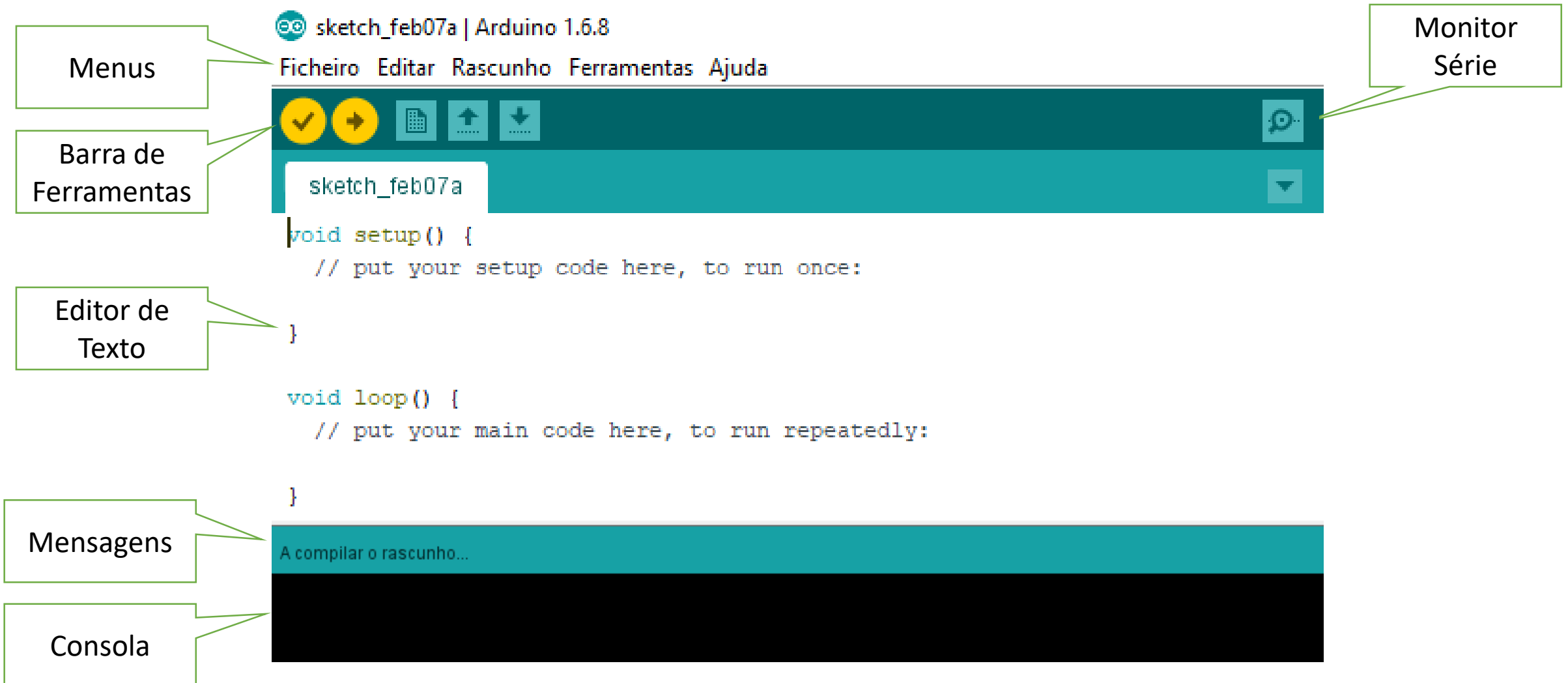
Placa de ensaio (Breadboard)



Led – Emissor de luz

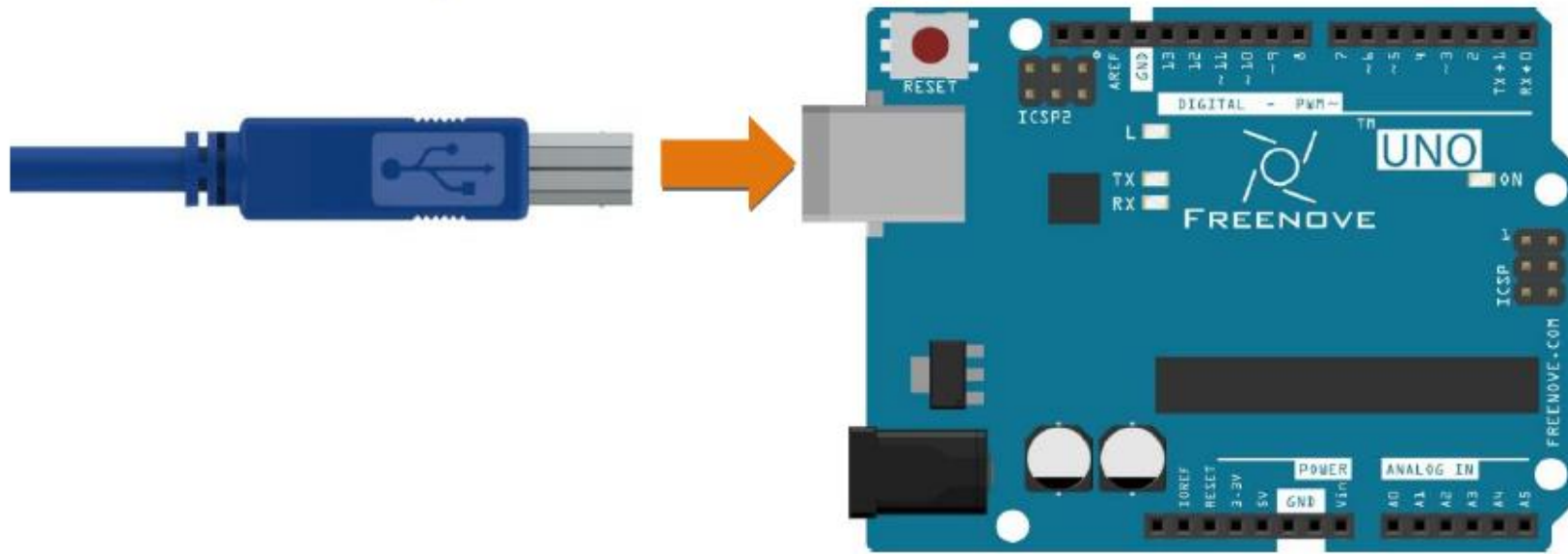


Ambiente de Desenvolvimento Integrado (IDE)



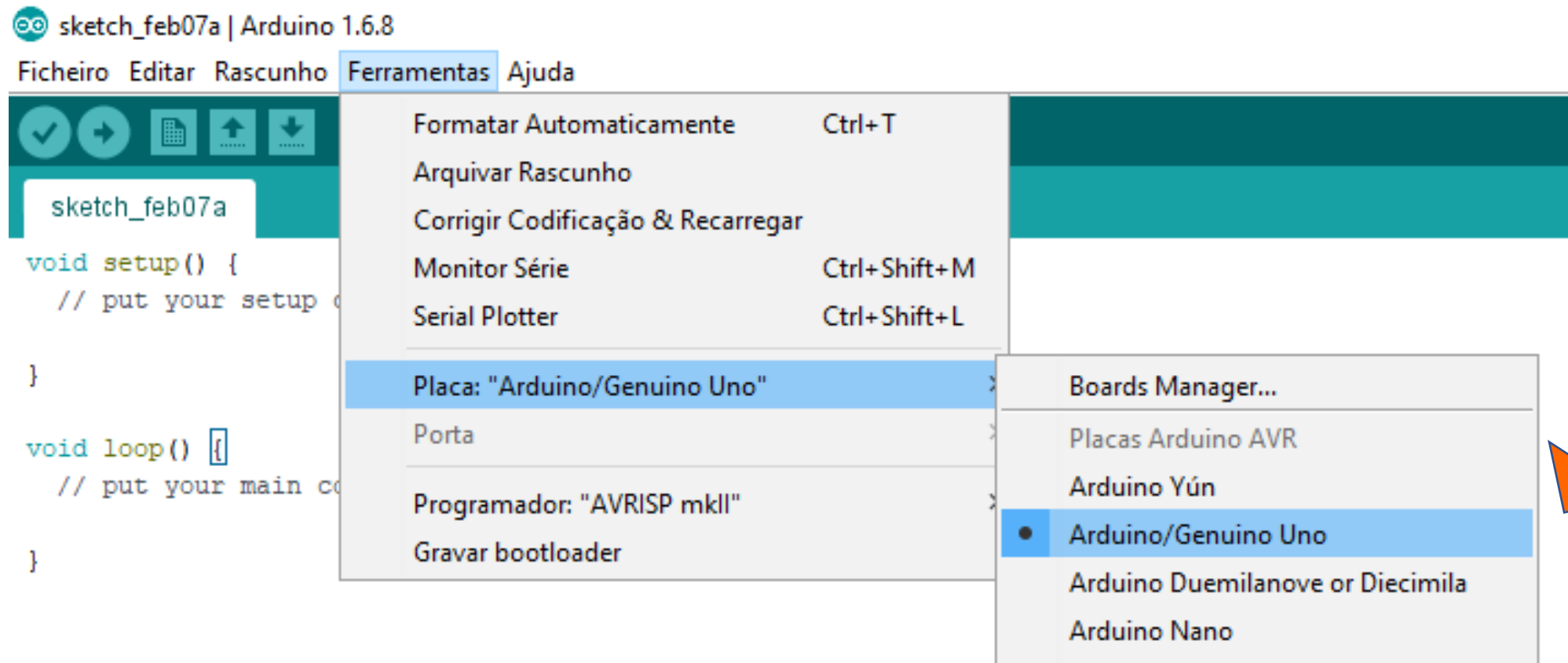
Ambiente de Desenvolvimento Integrado (IDE)

➡ 1º - Ligar o cabo USB ao Arduino após verificação das ligações



Ambiente de Desenvolvimento Integrado (IDE)

➡ 2º - Selecionar a placa



Ambiente de Desenvolvimento Integrado (IDE)

➡ 3º - Selecionar a porta

Verificar
código

Enviar
código

sketch_feb07a | Arduino 1.6.8

Ficheiro Editar Rascunho Ferramentas Ajuda

sketch_feb07a

```
void setup() {  
  // put your setup code here
```

```
}
```

```
void loop() {  
  // put your main code here
```

```
}
```

Formatar Automaticamente Ctrl+T

Arquivar Rascunho

Corrigir Codificação & Recarregar

Monitor Série Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

Placa: "Arduino/Genuino Uno" >

Port >

Programmer: "AVRISP mkII" >

Gravar bootloader

Serial ports

COM3 (Arduino/Genuino Uno)

Led aceso

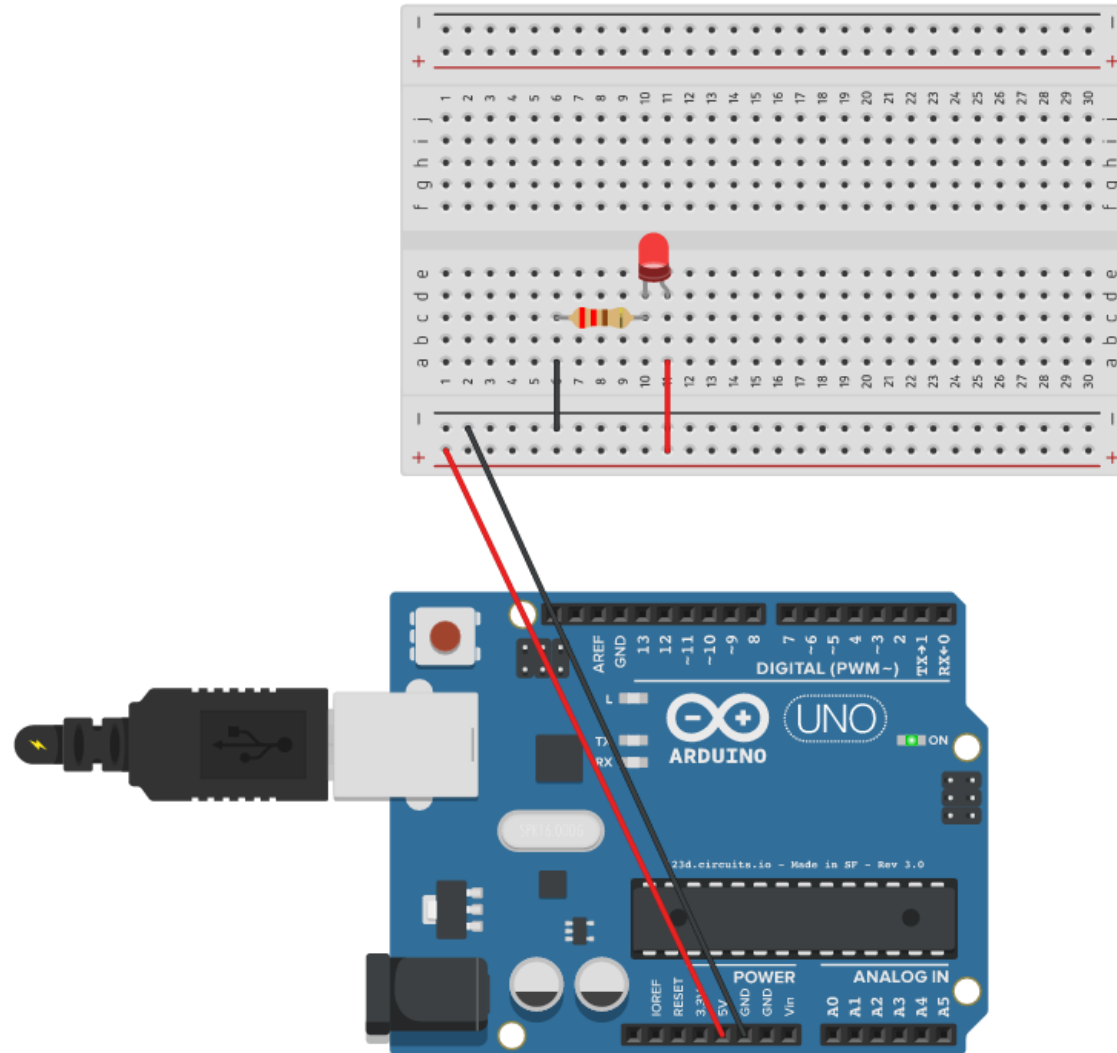
Material necessário:

Led  (Output Digital)

Resistência 

Fios 

Atenção:



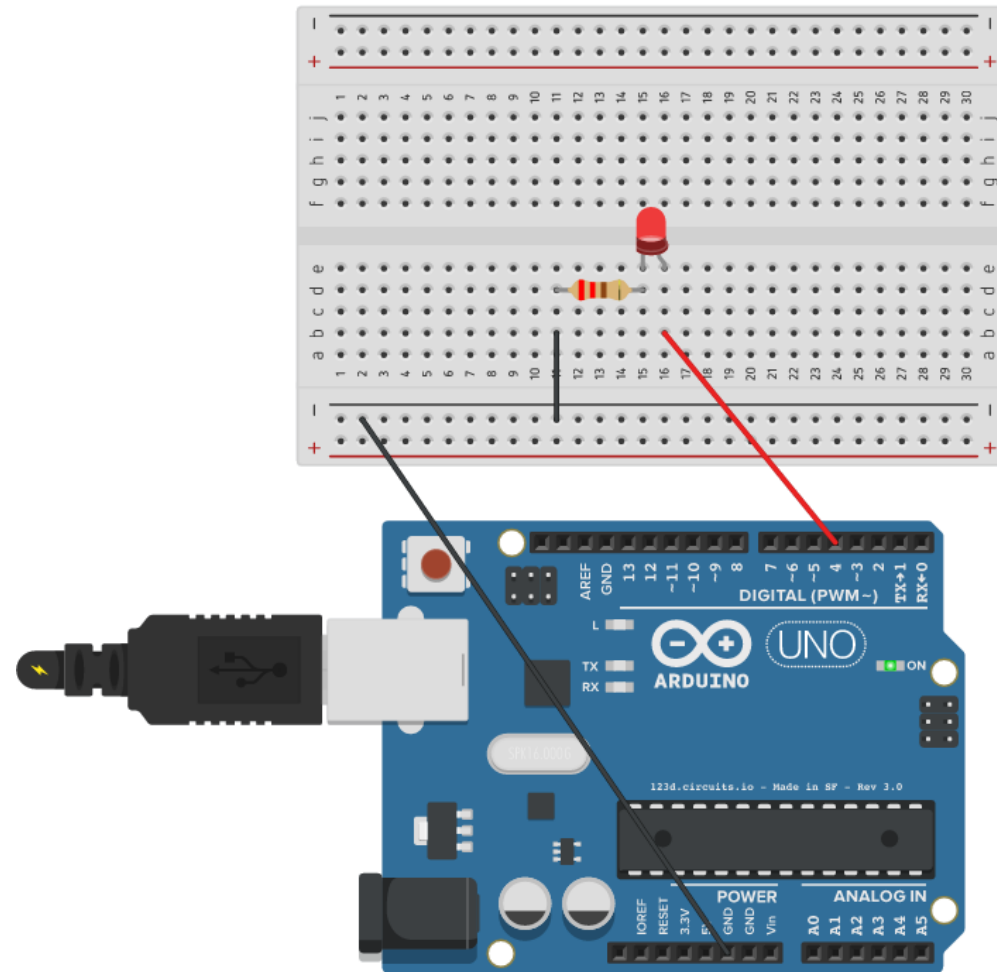
Led a piscar (Blink)

Material necessário:

Led  (Output Digital)

Resistância 

Fios 



Desafio 1 – Semáforo

Material necessário:

3 Leds  (Output Digital)

3 Resistências leds 

Fios



Regras de funcionamento do semáforo:

Só pode estar uma cor acesa.

Sequência de cores:

Acende **Verde** (5 segundos e passa a **Amarelo**)


Acende **Amarelo** (1 segundo e passa a **Vermelho**)


Acende **Vermelho** (5 segundos e passa a **Verde**)





Led acende com o botão

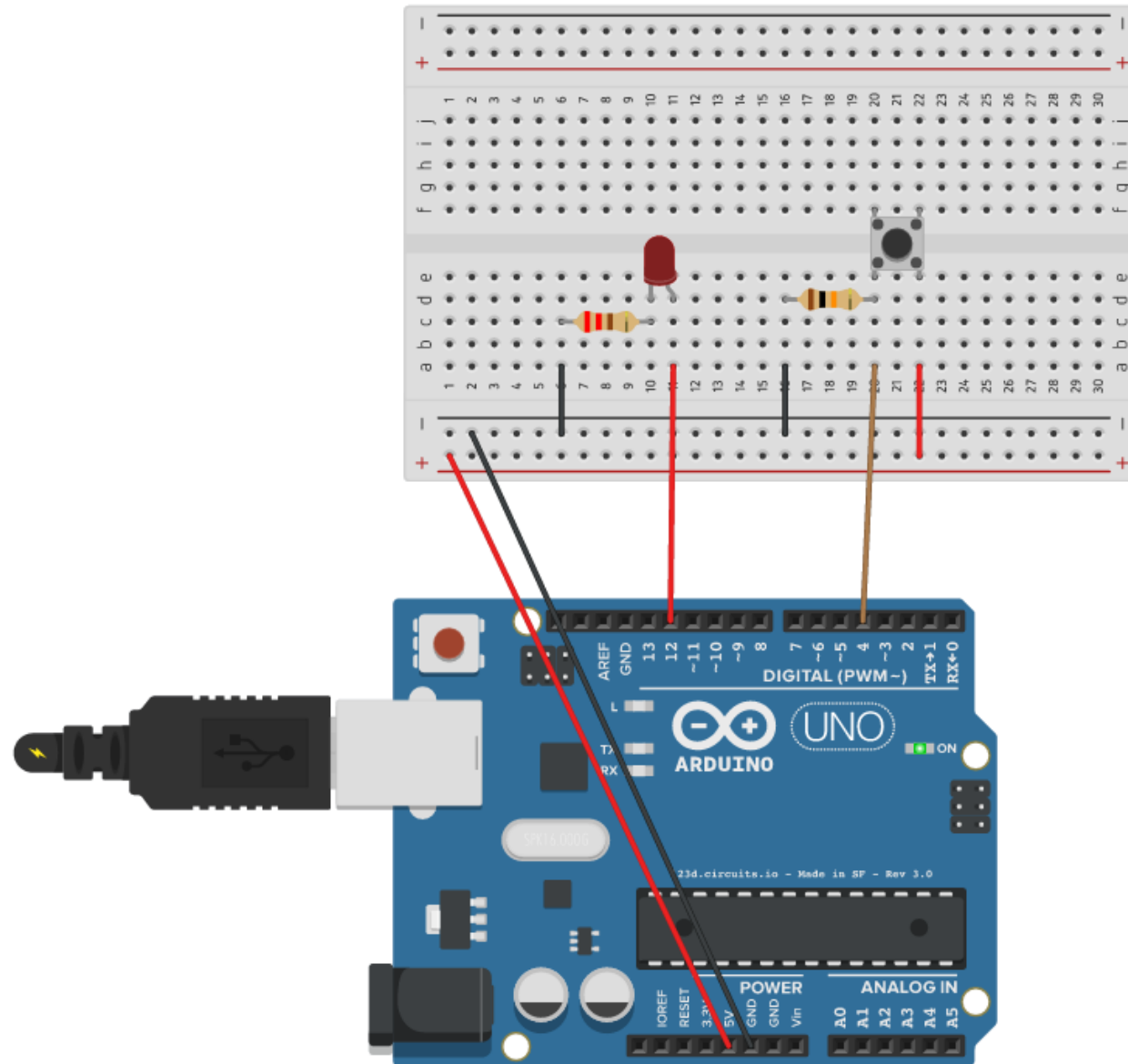
Material necessário:

Botão  (Input Digital)

Led  (Output Digital)

2 Resistências  Led
Botão 

Fios 




Sensor de Luz com LDR

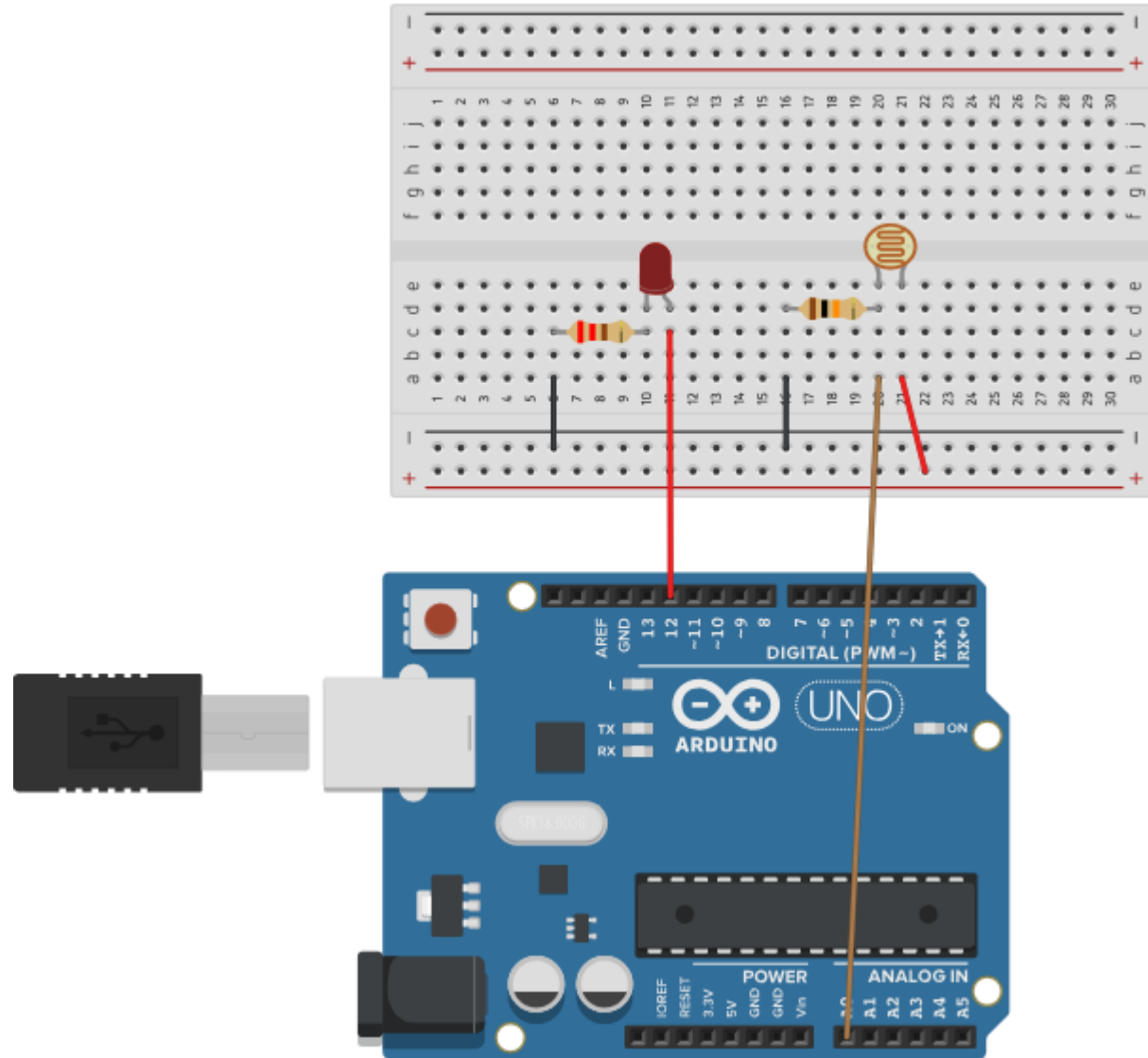
Material necessário:

LDR  (Input Analógico)

Led  (Output Digital)

2 Resistências 

Fios 



Led com Fade

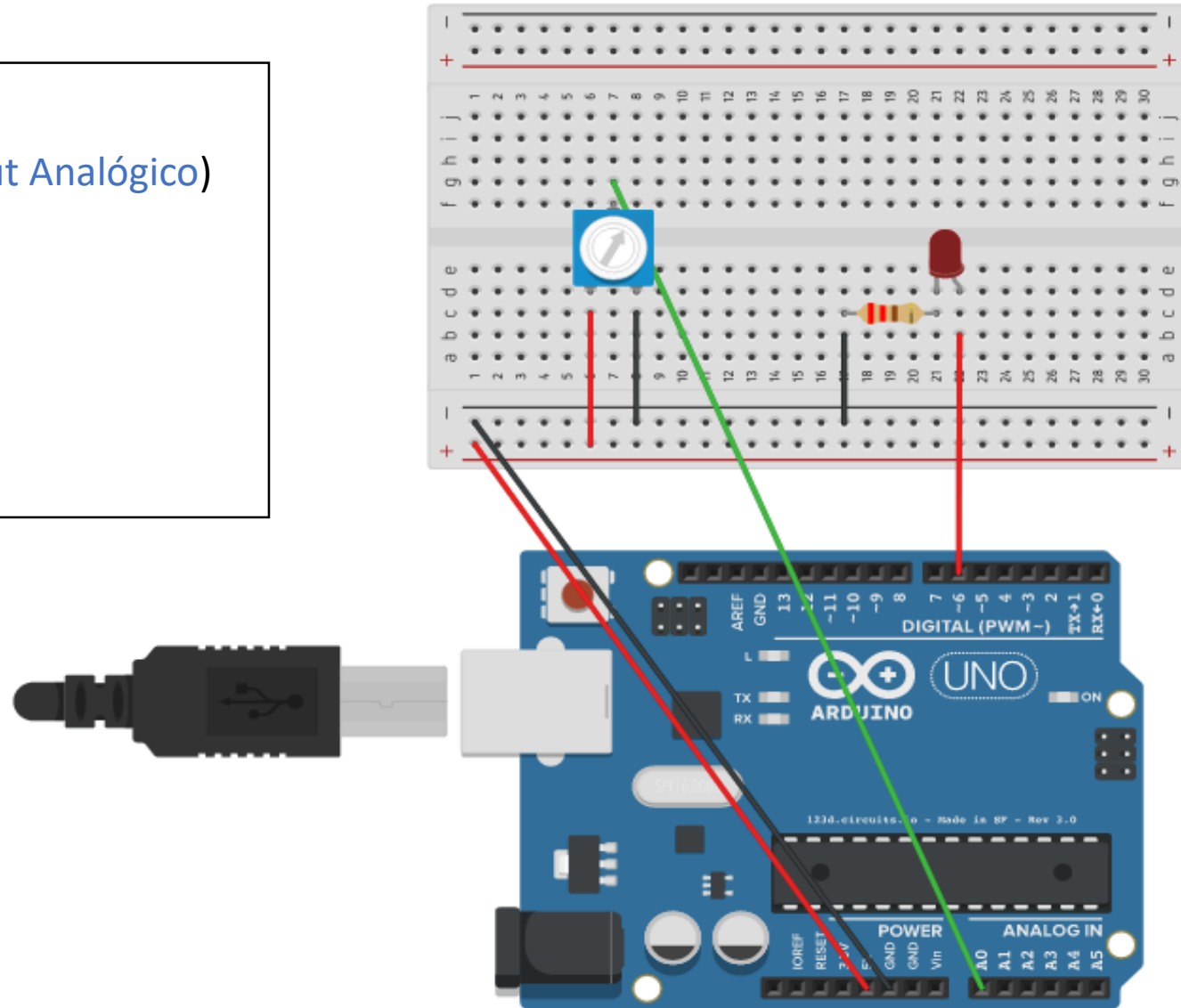
Material necessário:

Potenciômetro  (Input Analógico)

Led  (Output Digital)

Resistência Led 

Fios 



Leds que acendem quando o utilizador digita uma letra

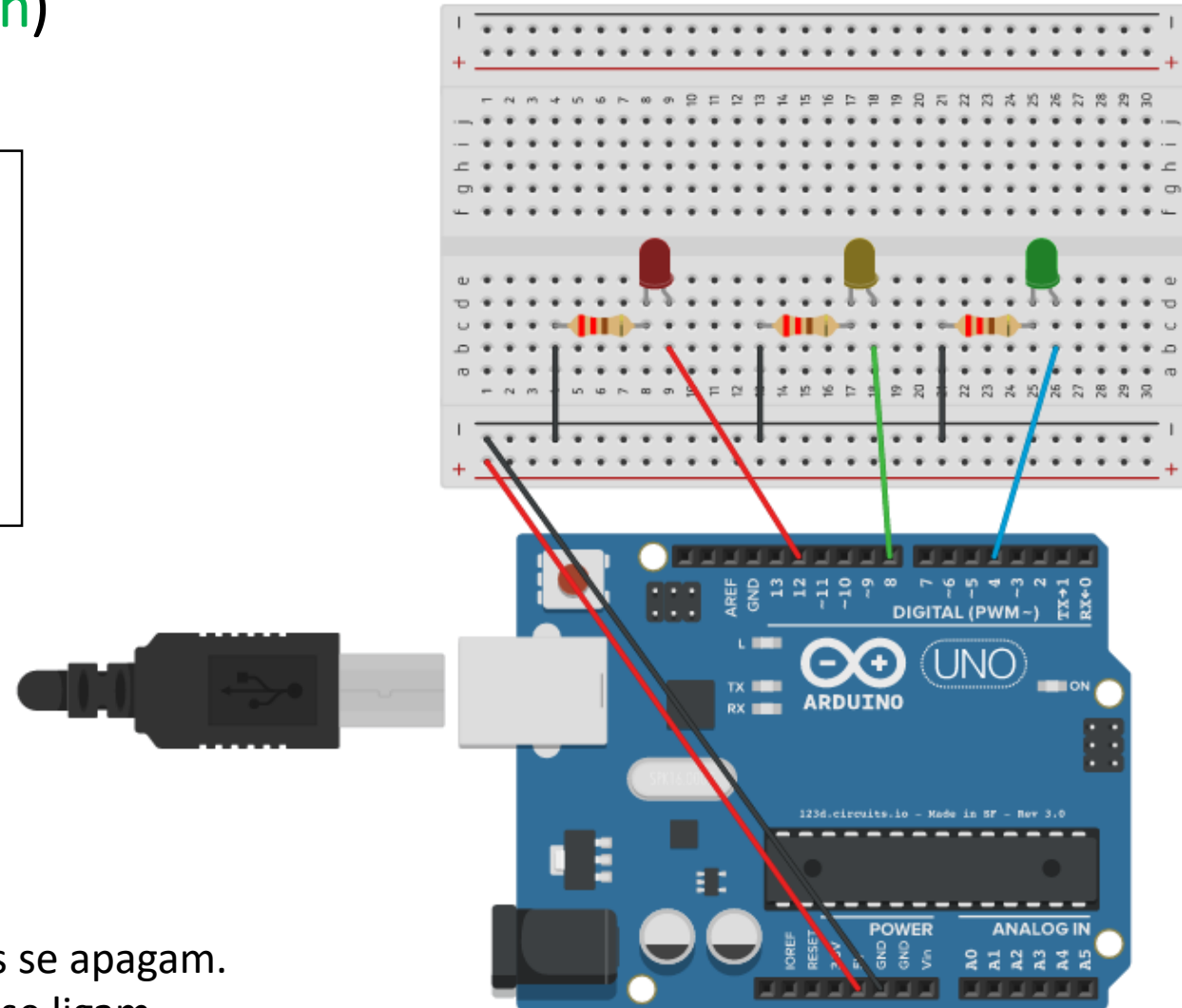
(R – Red, Y – Yellow, G – Green)

Material necessário:

3 Leds  (Output Digital)

3 Resistências leds 

Fios 



Nota:

Se o utilizador digitar A todos os leds se apagam.
Se o utilizador digitar L todos os leds se ligam.

Alarme

Material necessário:

1 Led RGB (Output Digital)



Buzzer (Output Analógico)



3 Resistências



1 Sensor Ultrassónico (Input Digital)



Fios

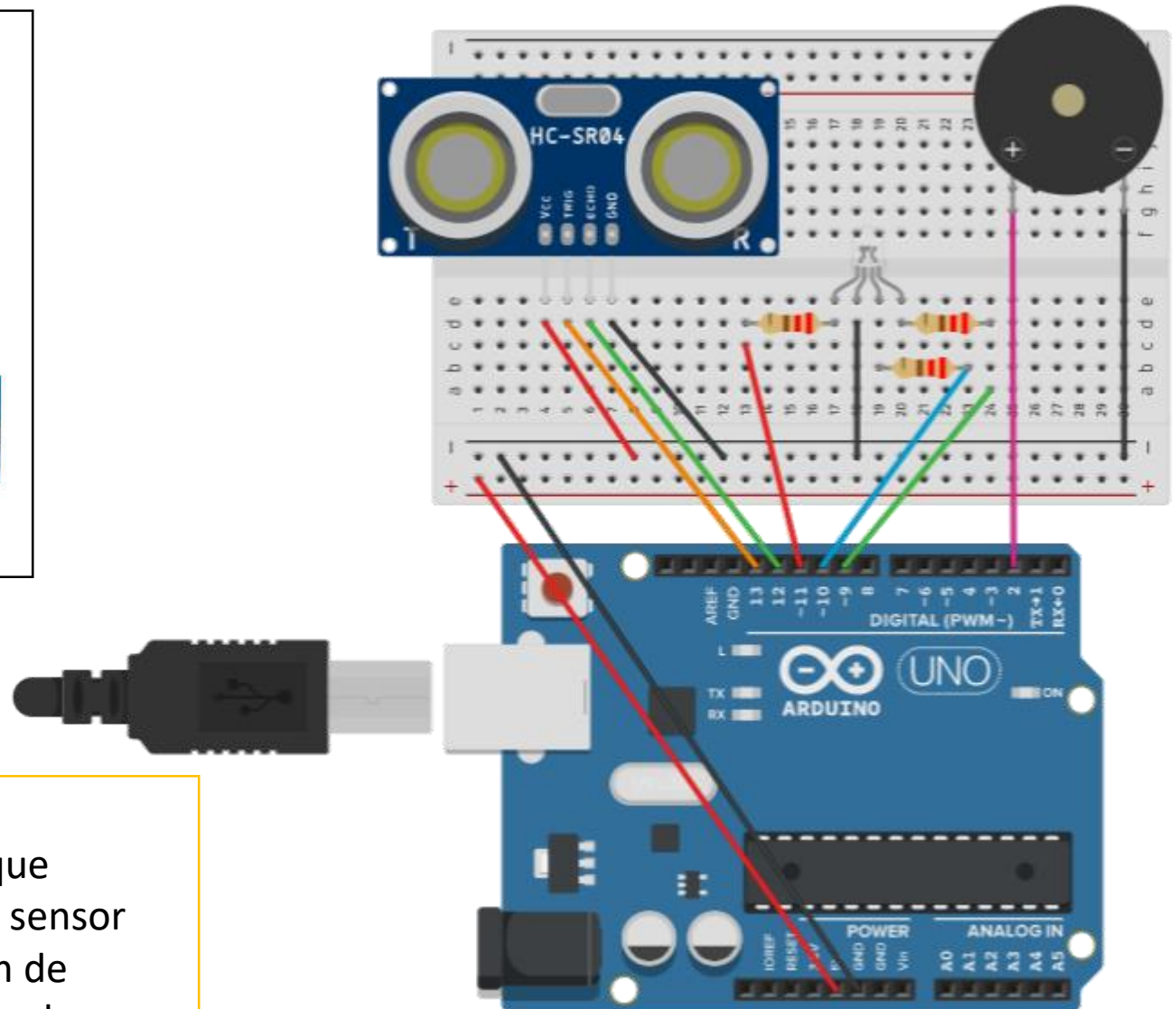


Funções:

tone() – Ativa um som no buzzer
noTone() – Desativa o som

Nota:

Nesta atividade pretende-se construir um circuito que permita construir um sistema de alarme. Quando o sensor ultrassónico detetar um indivíduo a menos de 20cm de distância, vai ativar o buzzer e o led RGB deverá acender com cores aleatórias em intervalos de 250ms



Piano

Material necessário:

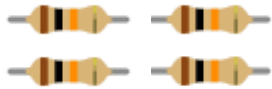
4 Botões (Input Digital)



Buzzer (Output Analógico)



4 Resistências



Fios



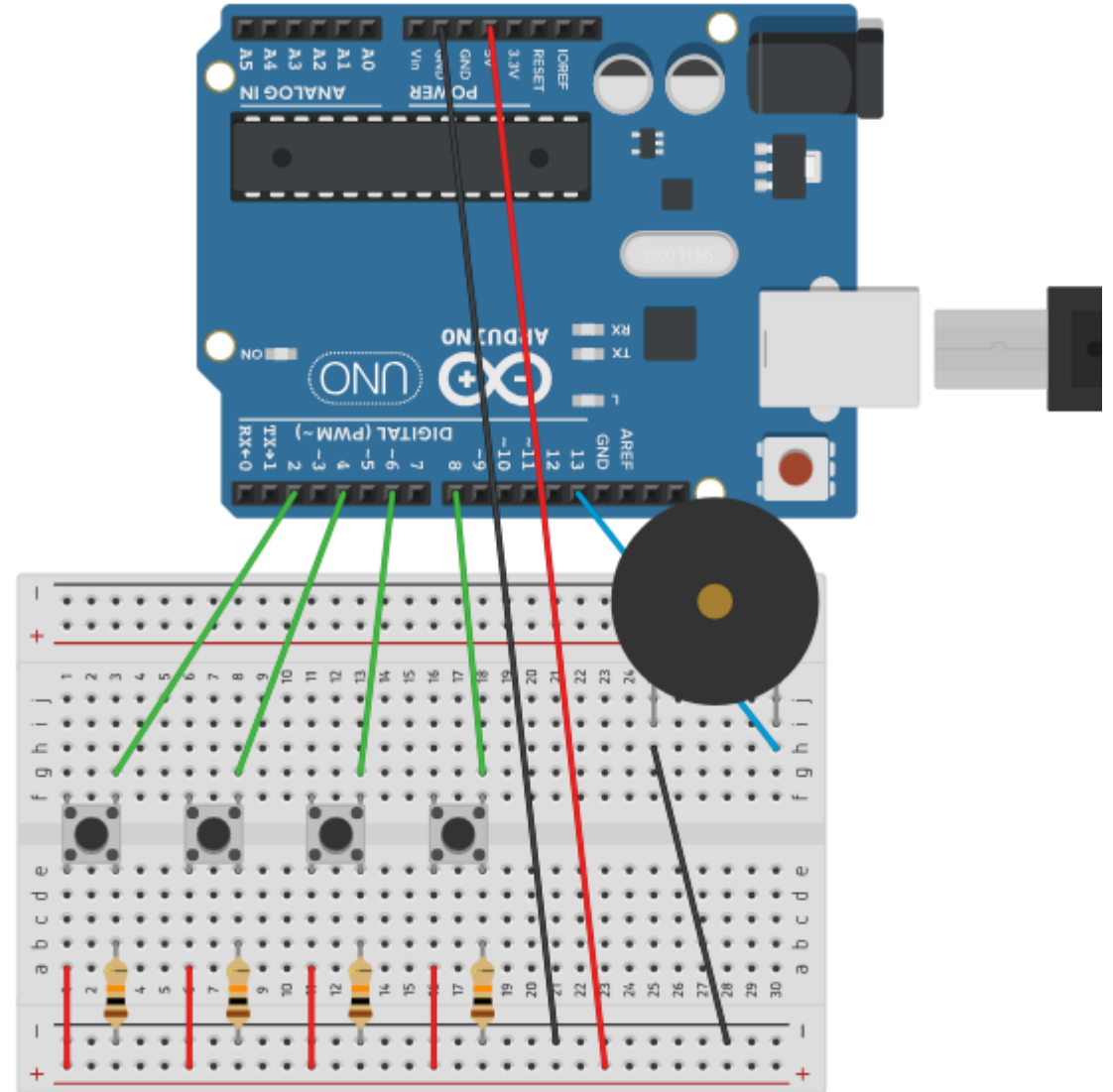
Funções:

tone() – Ativa um som no buzzer

noTone() – Desativa o som

Nota:

Definir um array de 4 posições com os valores de frequência (notas musicais) associados ao som de cada botão: {262,294,330,349}

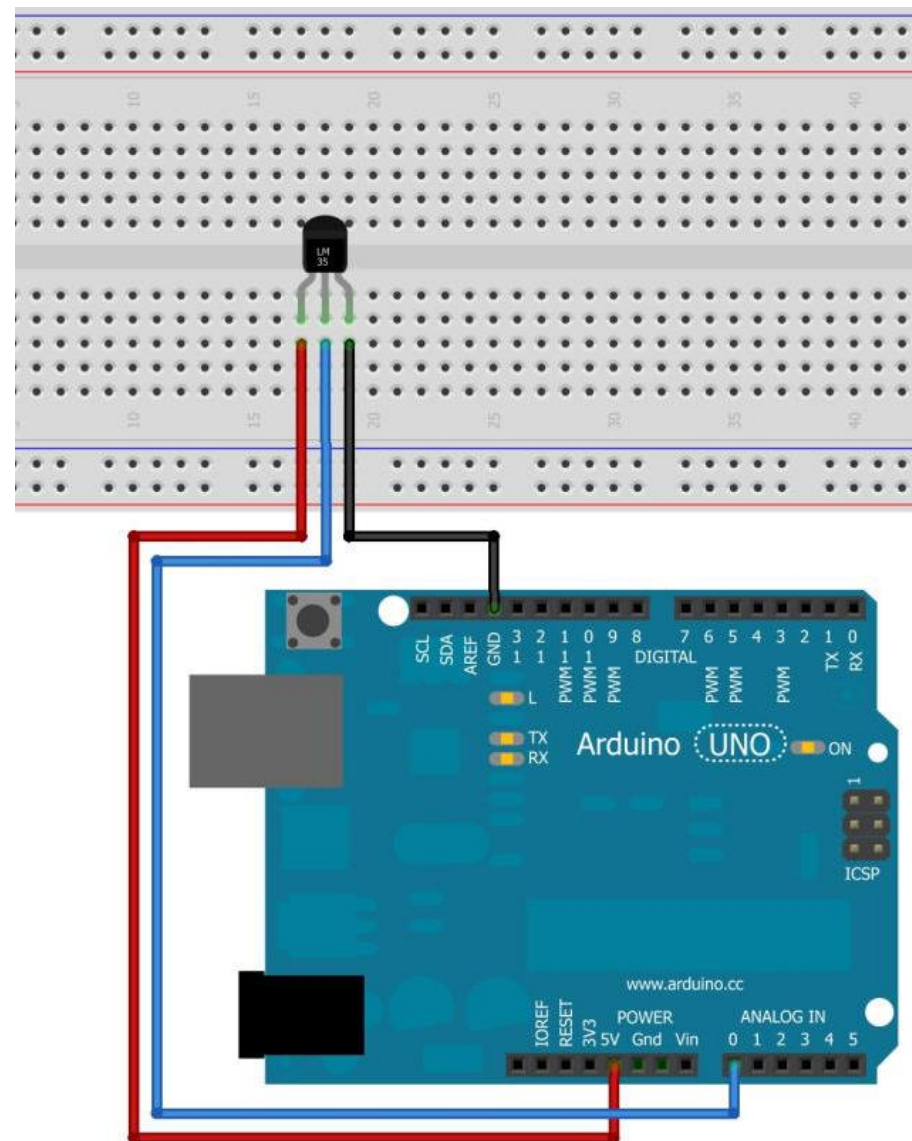


Sensor Temperatura

Material necessário:

1 Sensor Temperatura (Input Analógico)

Fios



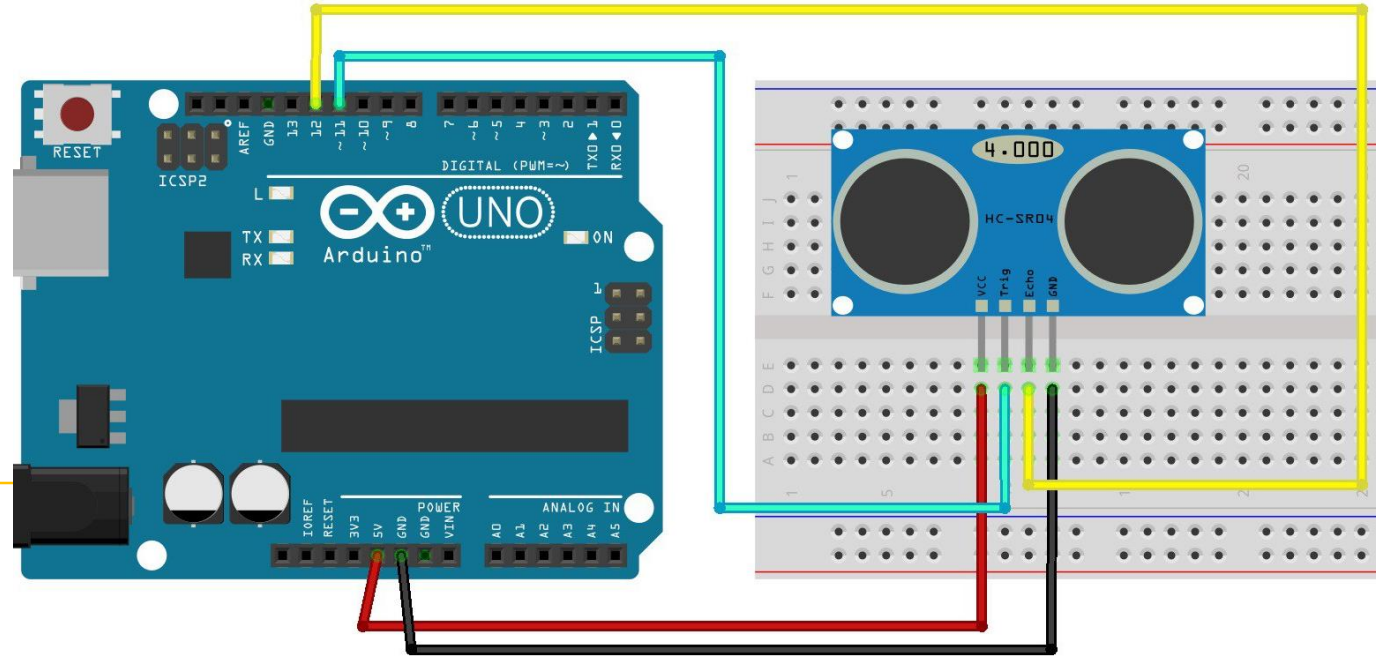
Sensor Ultrassónico

Material necessário:

1 Sensor Ultrassónico
(Input Digital)



Fios



Biblioteca Ultrasonic.h:

```
Ultrasonic nome_objeto(trigPin, echoPin);
```

Funções:

timing() – Tempo que o sinal demorou a colidir com o obstáculo e retornar

convert() – Conversão do tempo para distância (cm)

Motor Servo

Material necessário:

1 Potenciômetro  (Input Analógico)

2 Condensadores 100µF

1 Motor Servo

Fios

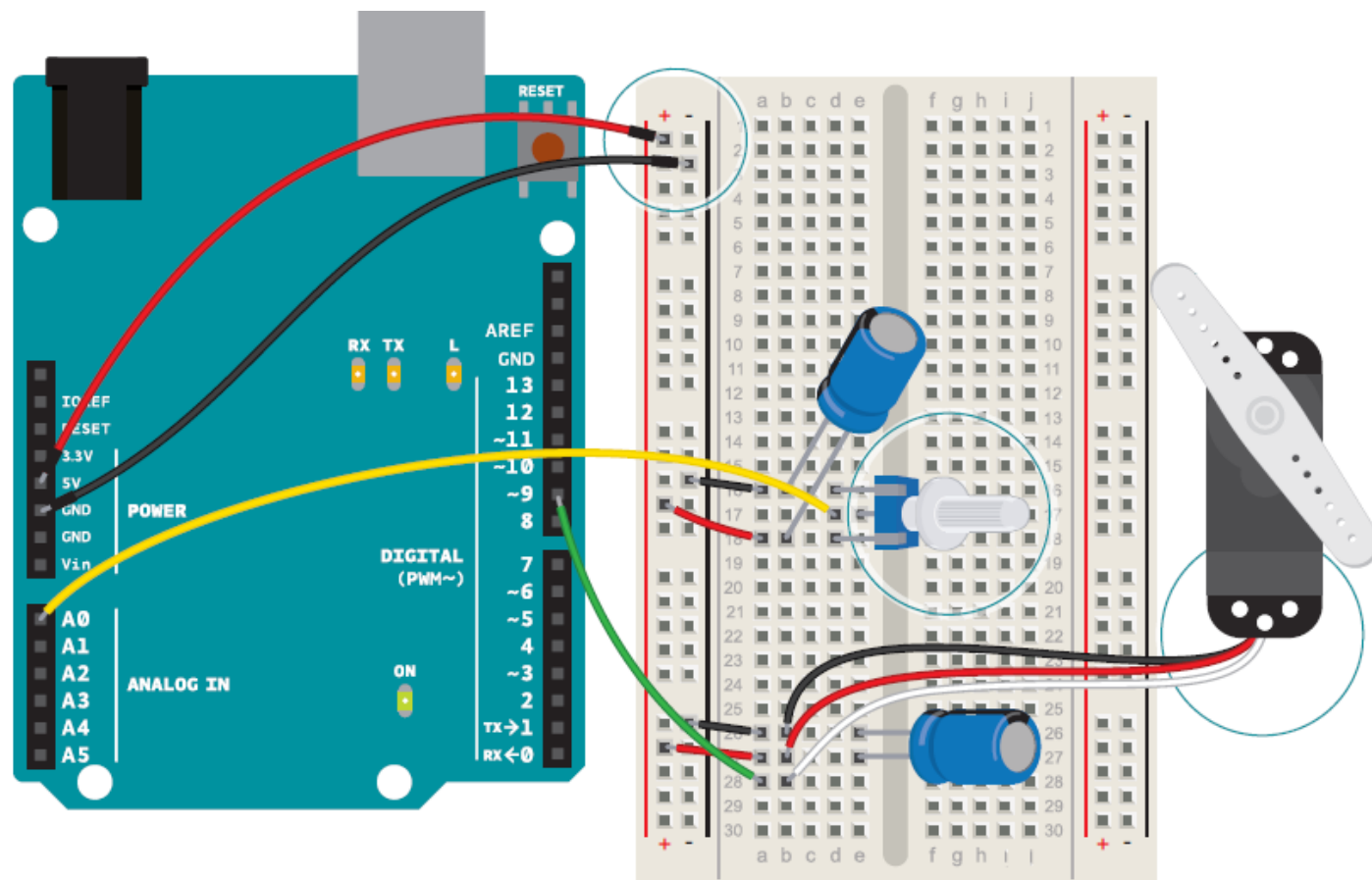


Biblioteca Servo.h:

```
Servo myServo;
```

```
void setup() {  
    myServo.attach(pin);  
}
```

```
myServo.write(angle);
```



Motor DC – Ventoinha

Material necessário:

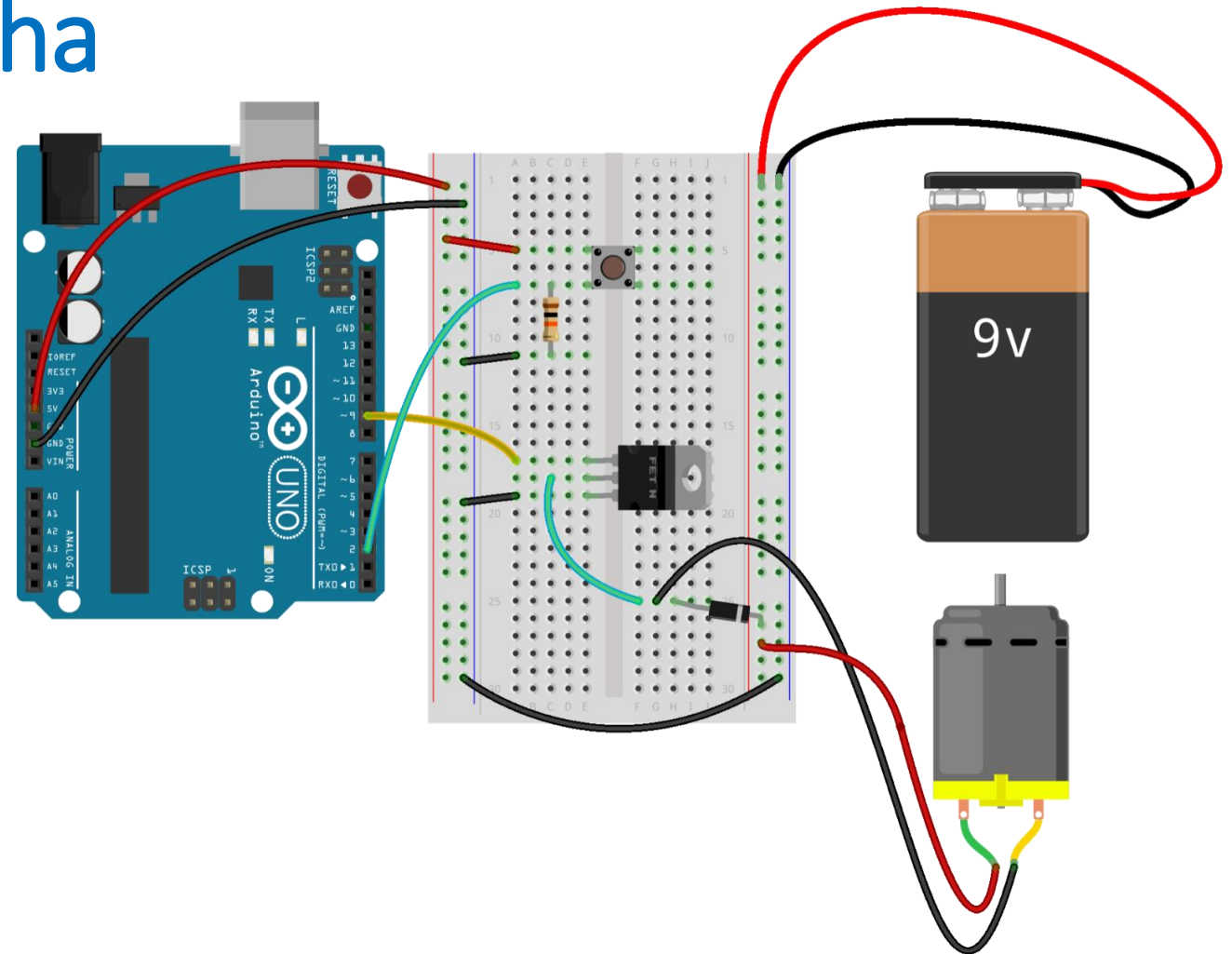
1 Botão  (Input Digital)

1 Mosfet 

1 Resistência 

1 Díodo 

Fios 



Bluetooth e Anprino

Material necessário:

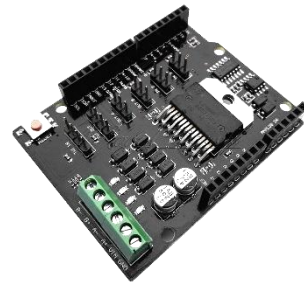
1 x Sensor Ultrassónico HC – SR04



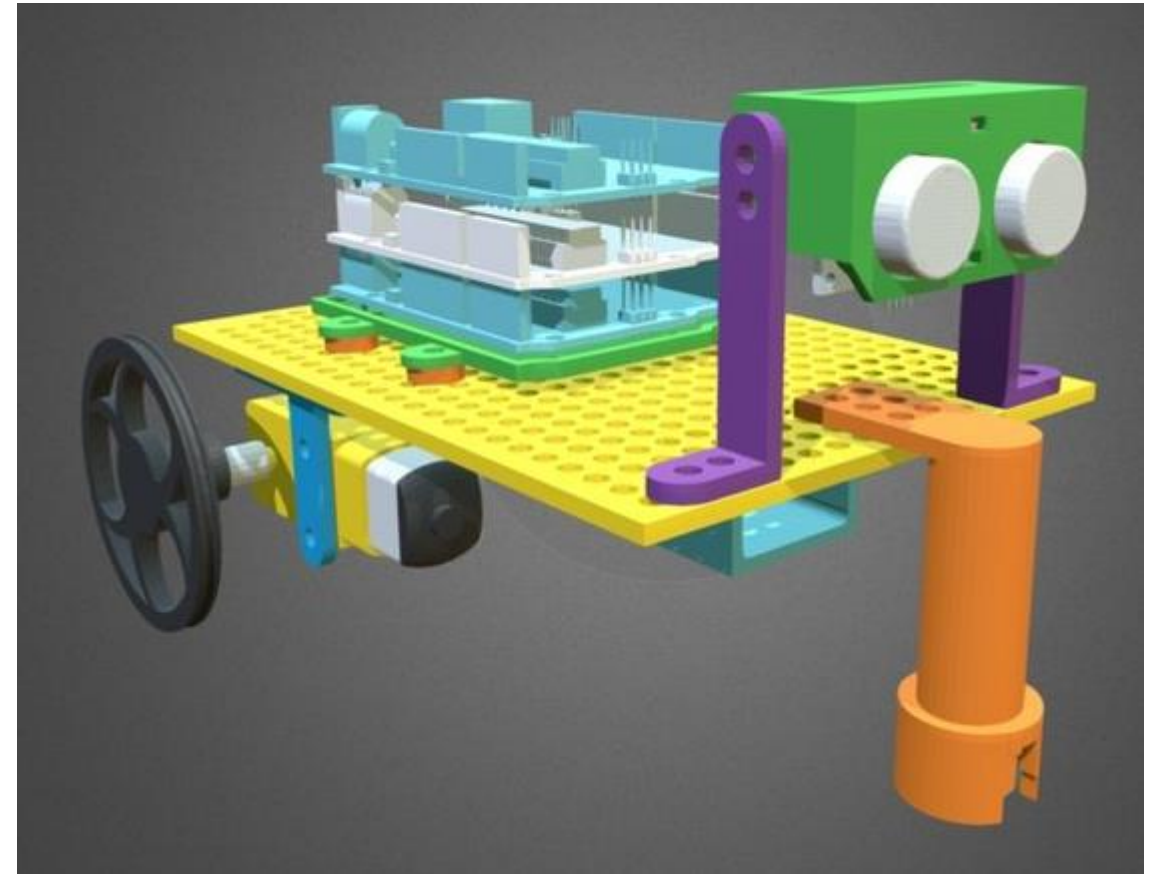
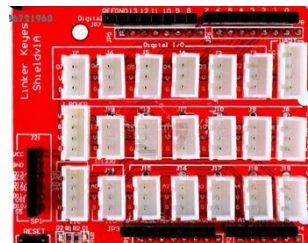
1 x Módulo Bluetooth (HC-06)



1 x L298NH Motor Driver



1 x Base Shield Sensor I/O
Shield Expansion Board



2 x Motor DC



Fim da sessão