

# WTF is an Arduino?

George Brindeiro





**NÃO LEMBRO DE MAIS NADA DA SEMANA PASSADA...**

**CUIDADO**

**ENTRADA  
E SAÍDA  
DE VEÍCULOS**



**DIGITAL**



**ANALÓGICO**

**VS**











"The last thing I remember is something about a 'once in a lifetime' offer."

**SETUP**



**LOOP**



**IDE**

**SERIAL  
PORT**

**FUNÇÃO**

**RX/TX**

**BOARD**

**PROTOBOARD**





// COMENTÁRIO

COMANDO;

/\*COMENTÁRIO

MUITO

LONGO \*/

{BLOCO}

VERIFY UPLOAD



“ ARDUINO IS AN OPEN SOURCE ELECTRONICS  
PROTOTYPING PLATFORM BASED ON FLEXIBLE,  
EASY-TO-USE HARDWARE AND SOFTWARE.

IT'S INTENDED FOR STUDENTS, HOBBYISTS,  
AND PROFESSIONALS IN CREATING  
INTERACTIVE PROJECTS.





**B  
L  
I  
N  
K**



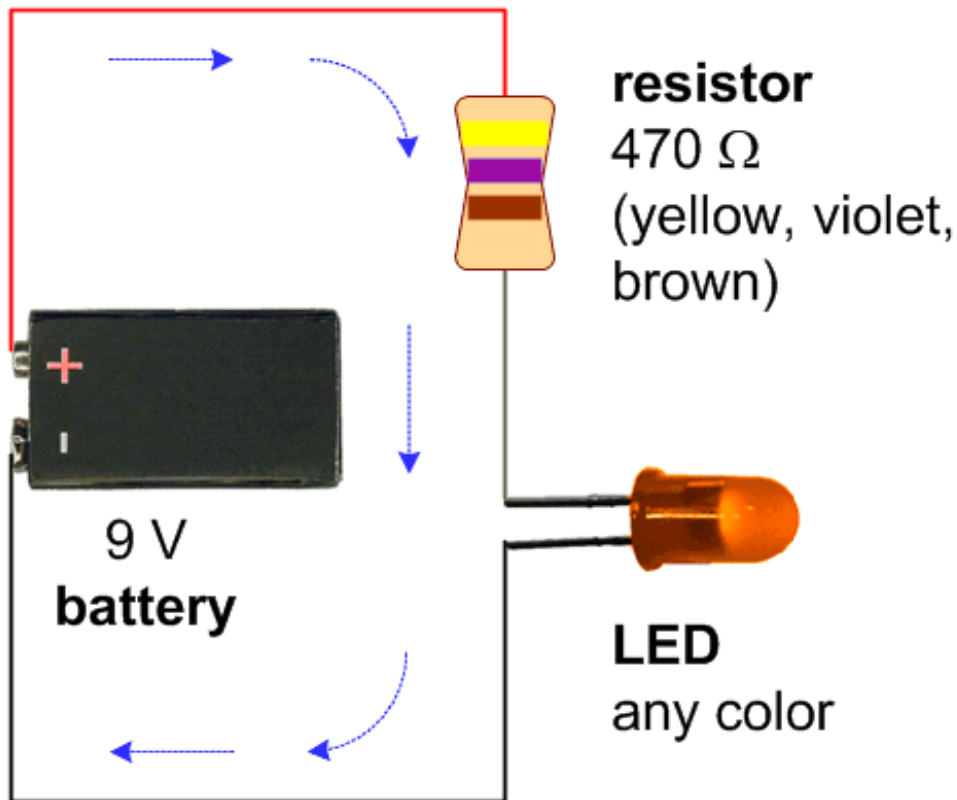
# Atualizando o sketchbook

- Arquivos do curso:  
<https://github.com/georgebrindeiro/wtf-is-an-arduino>
- Download ZIP e extrair em seguida
- File -> Preferences -> Sketchbook location
- Colocar a pasta extraída wtf-is-an-arduino/sketchbook

# 2

ENTRADA E SAÍDA DIGITAL:  
SOBRE BITS, BOTÕES E LEDs

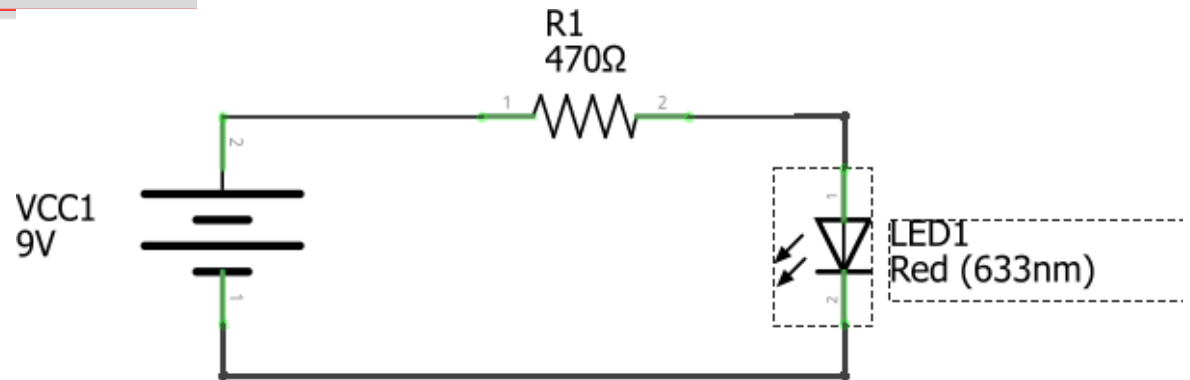
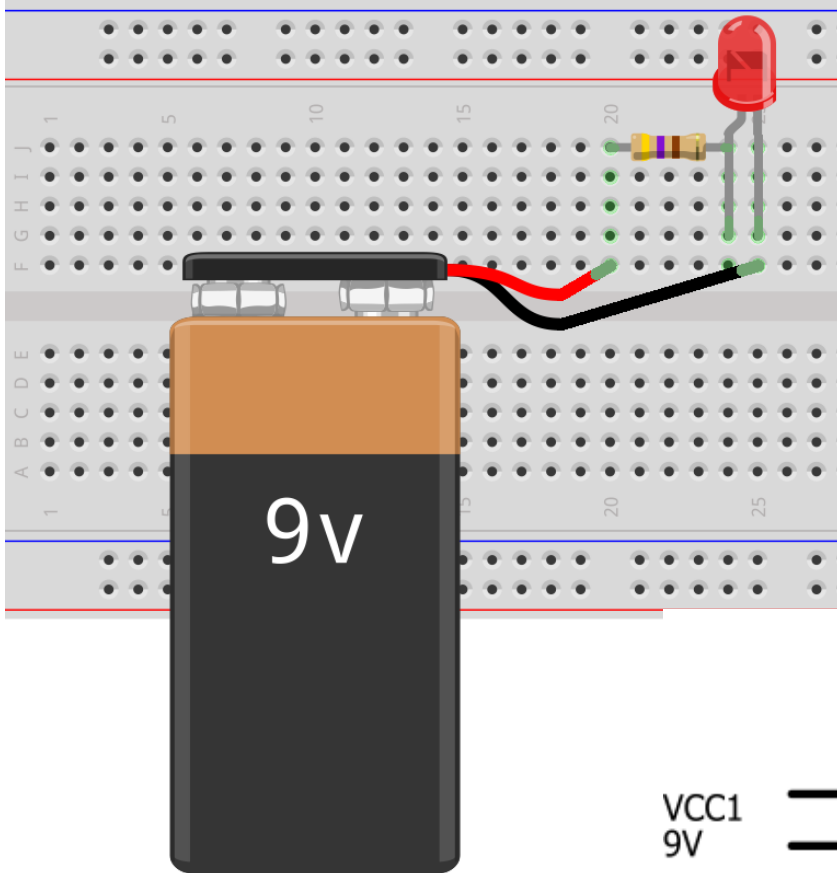
# Circuitos básicos: LED e Resistor



- Corrente flui do + para o -
- Lei de Ohm:  $V = RI$ 
  - V: tensão (9 V)
  - R: resistência (470  $\Omega$ )
  - I: corrente ( $9/470 \approx 19$  mA)
- Código de cores indica valor do resistor
- LED só deixa corrente fluir em um sentido



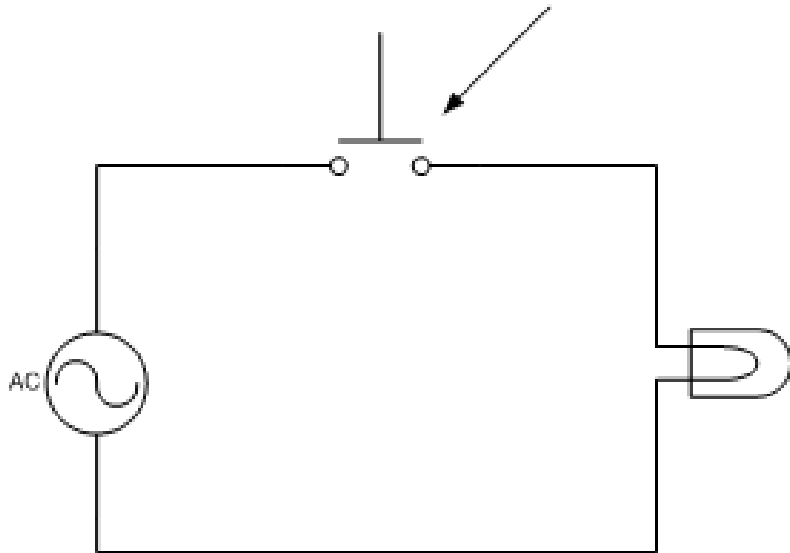
# Circuitos básicos: LED e Resistor



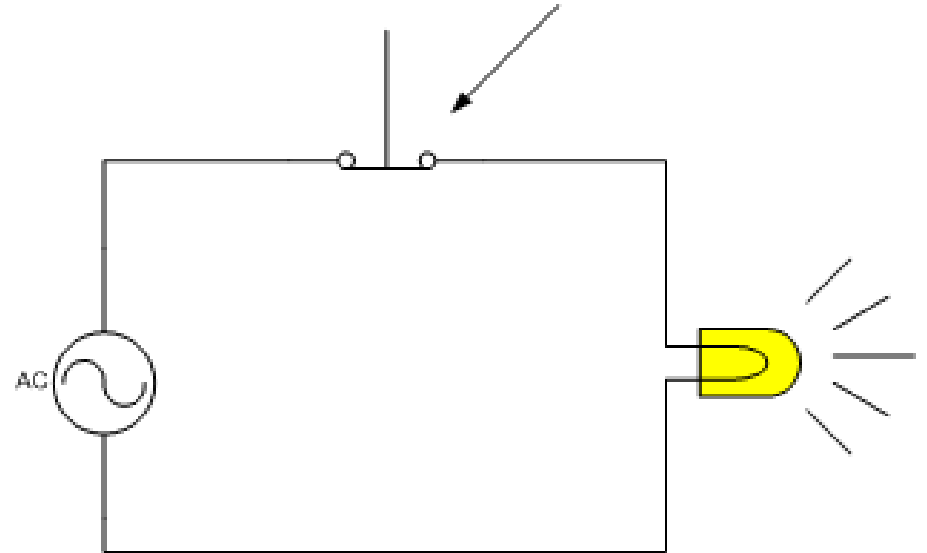
fritzing

# E pra termos... um interruptor?

Not connected (Open, Off)



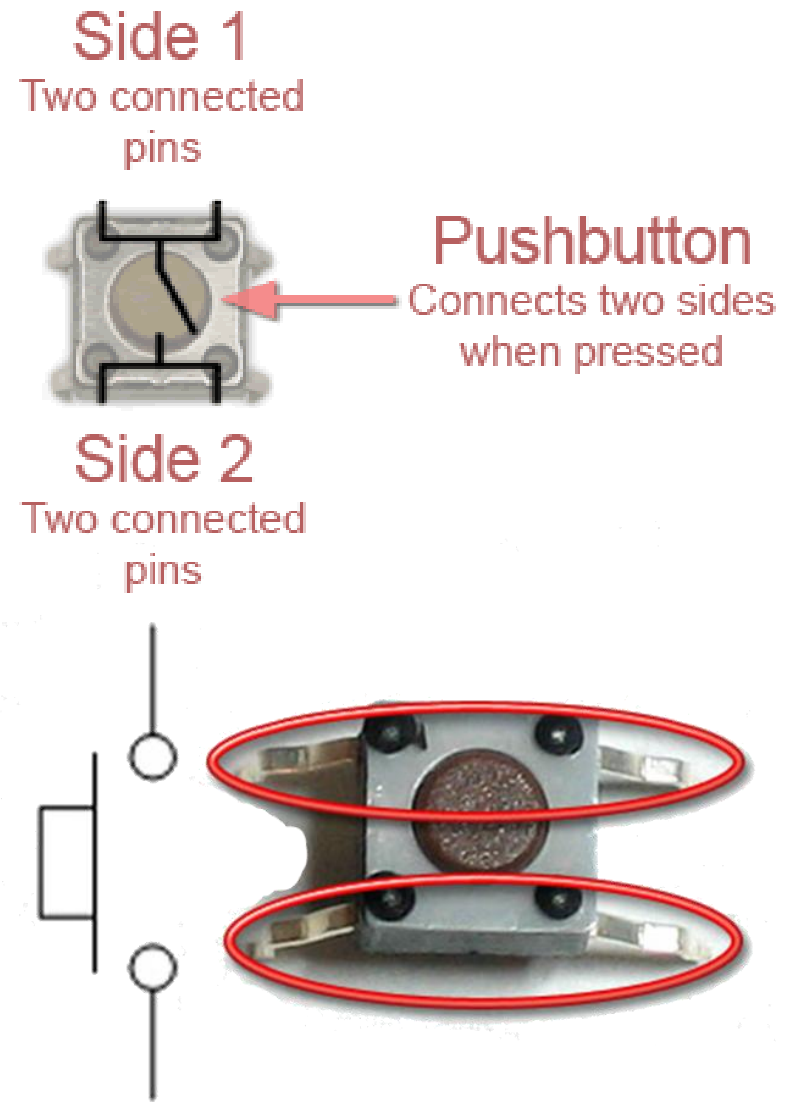
Connected (Closed, On)



# Botão “Pushbutton”



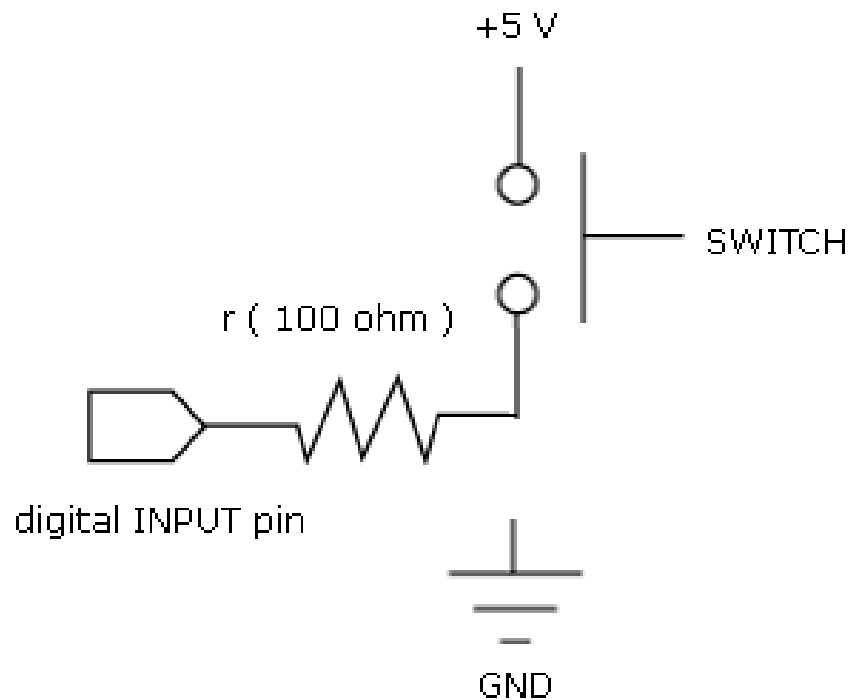
- Pares de pinos conectados (lados opostos)
- Normalmente aberto
- Apertar botão fecha circuito (conecta pinos eletricamente)



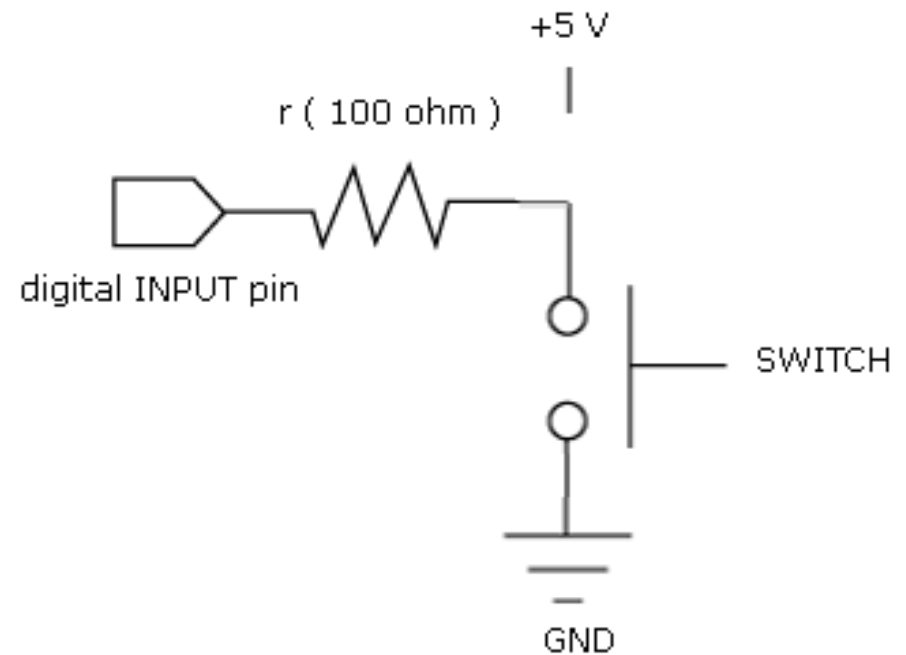


# Circuitos básicos: Botões

**CIRCUIT A**

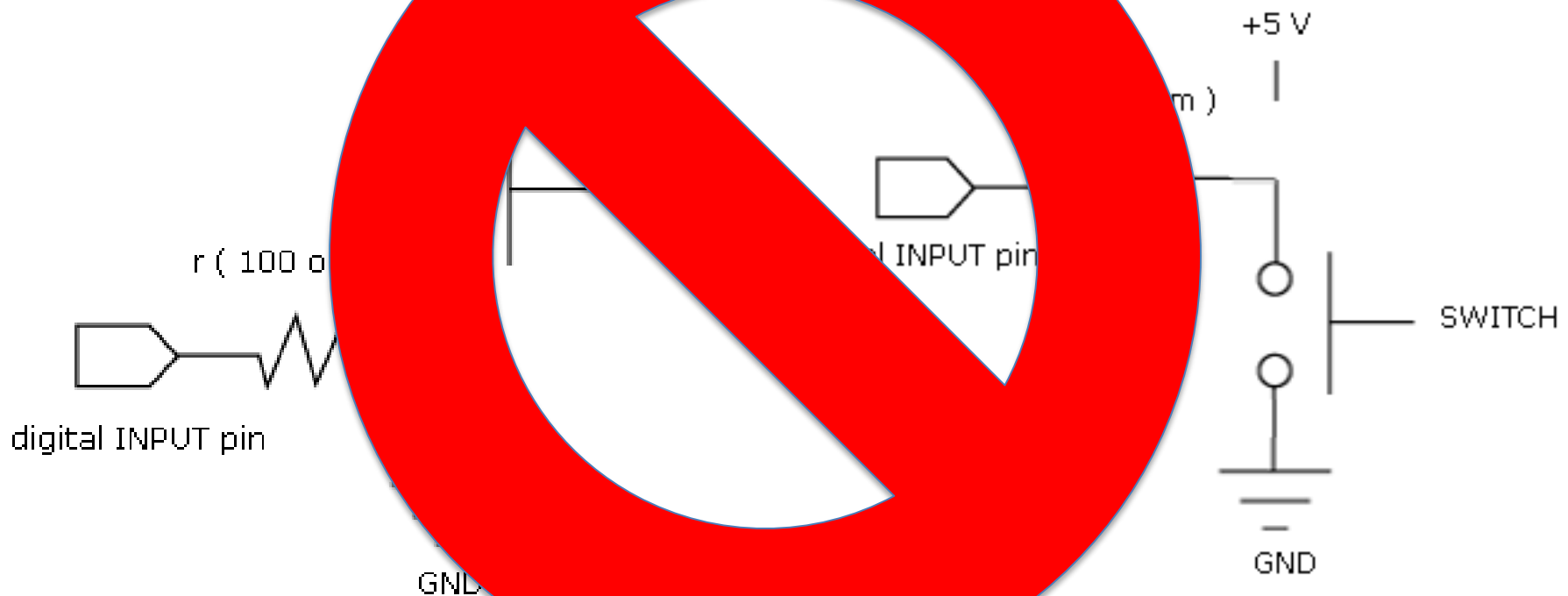


**CIRCUIT B**

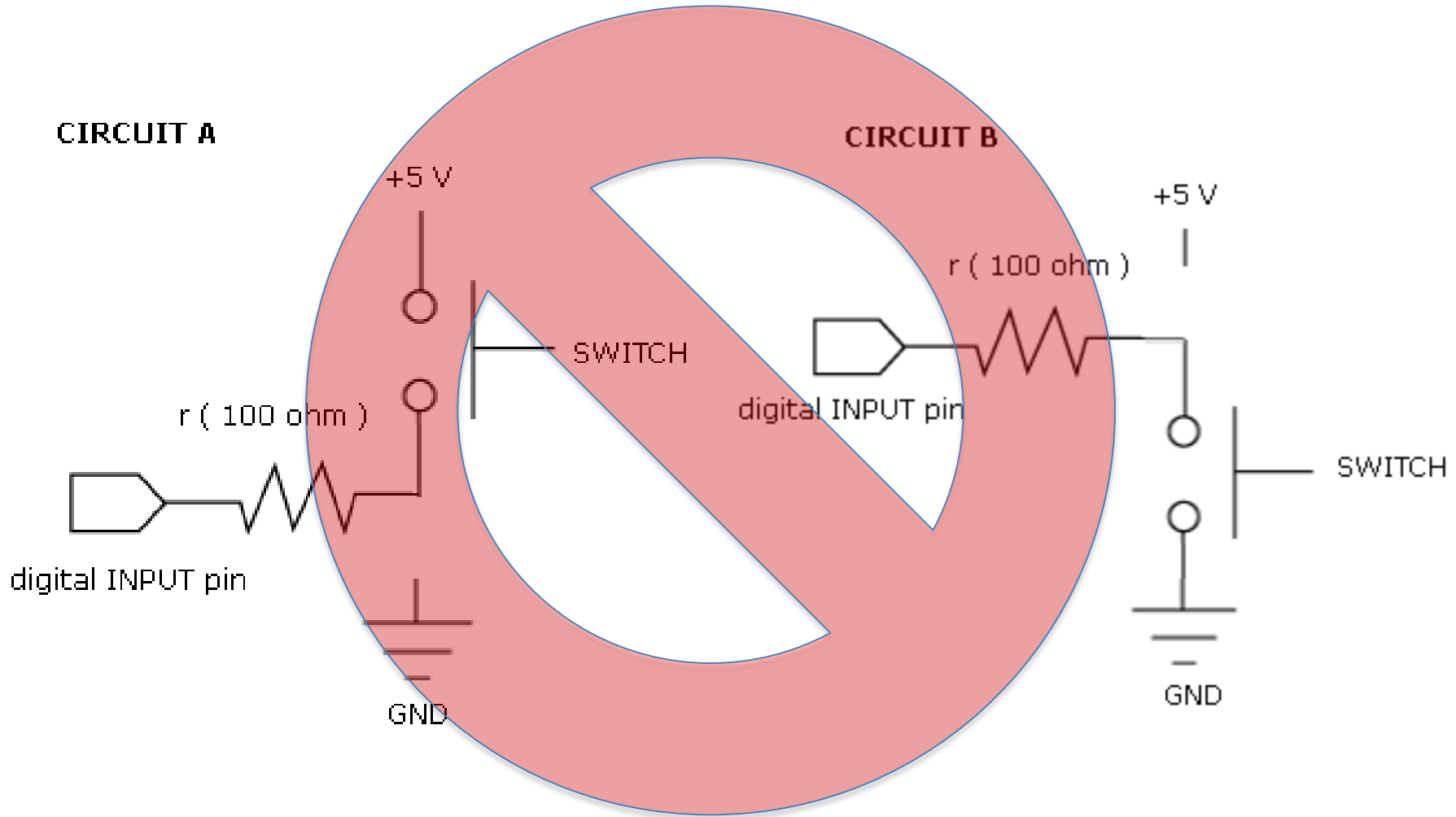


# Circuitos básicos: Botões

CIRCUIT A



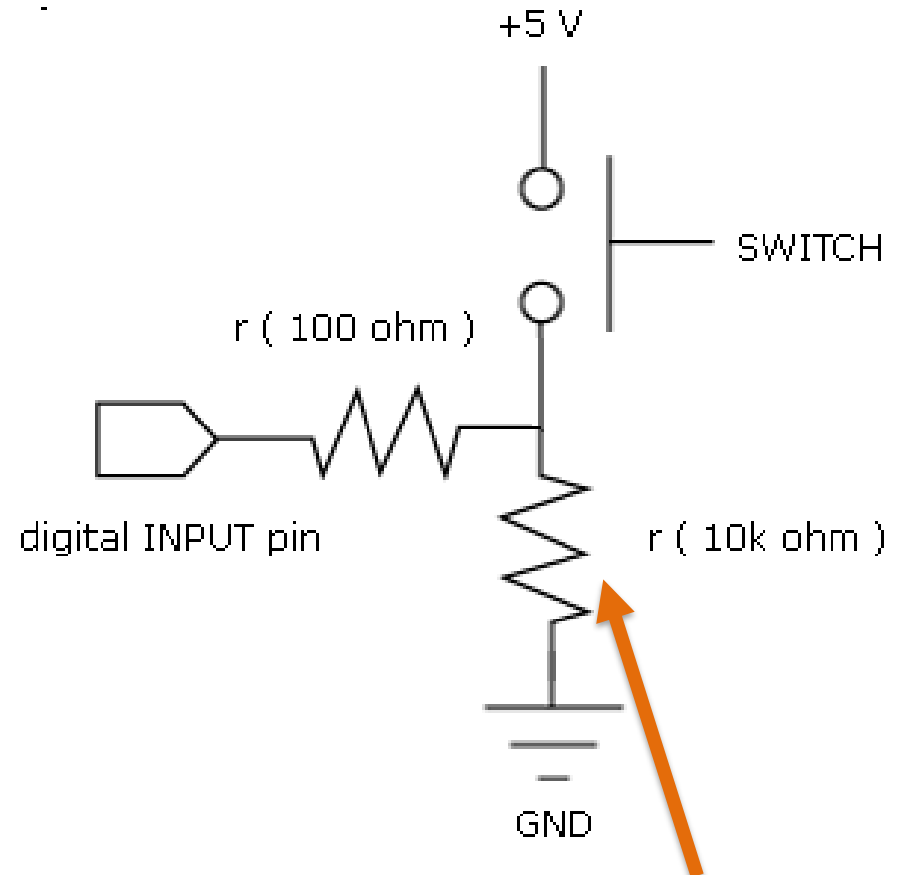
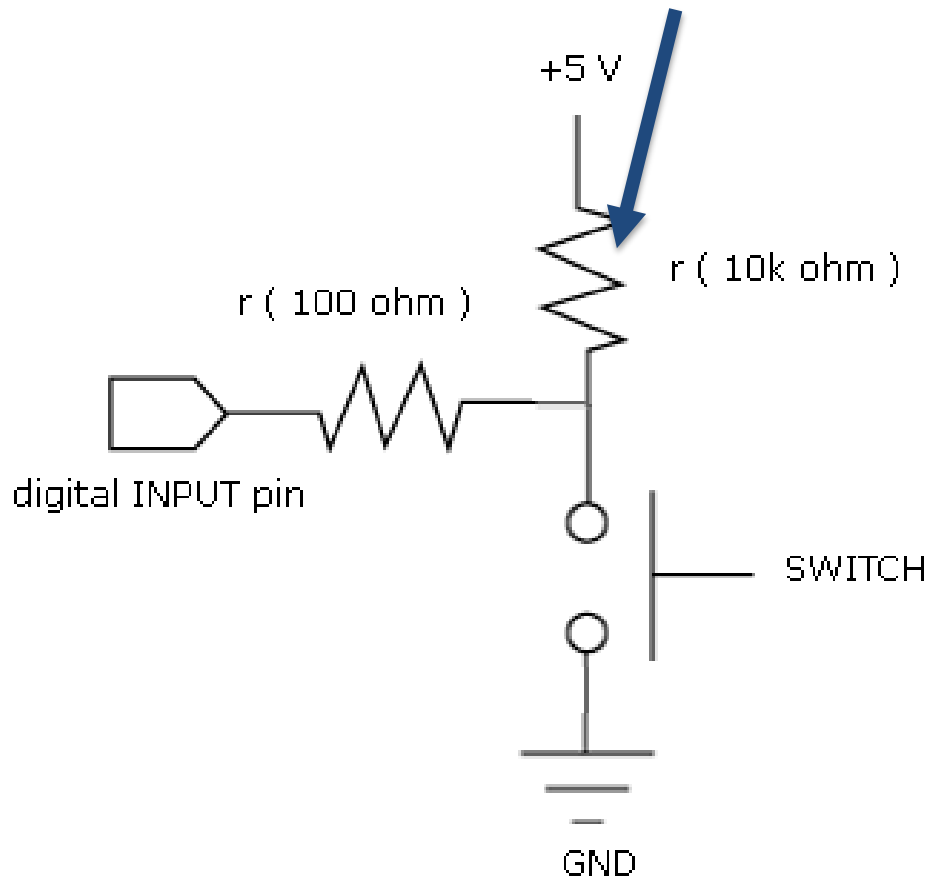
# Circuitos básicos: Botões



**PRECISAMOS ALTERNAR AS CONEXÕES**

# Truque: Pull-up ou Pull-down

## PULL-UP



## PULL-DOWN

# Programando: Entrada/Saída Digital

## Funções

`pinMode(pin, mode)`

`digitalWrite(pin, level)`

`digitalRead(pin)`

## Argumentos

`pin`: 0-13 (analog: A0-A7)

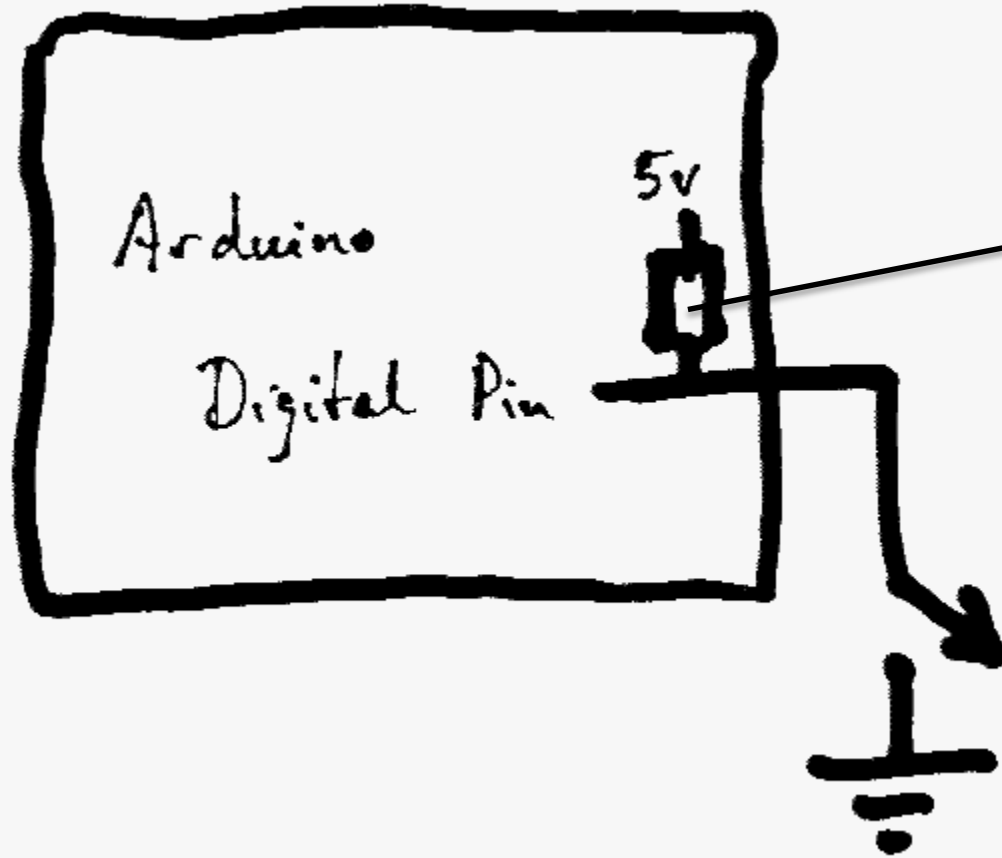
`mode`:

- INPUT para `digitalRead`
- INPUT\_PULLUP para `digitalRead` com pull-up interno (> Arduino 1.0.5)
- OUTPUT para `digitalWrite`

`level`: HIGH (5V) ou LOW (GND)



# Pull-up interno do Arduino



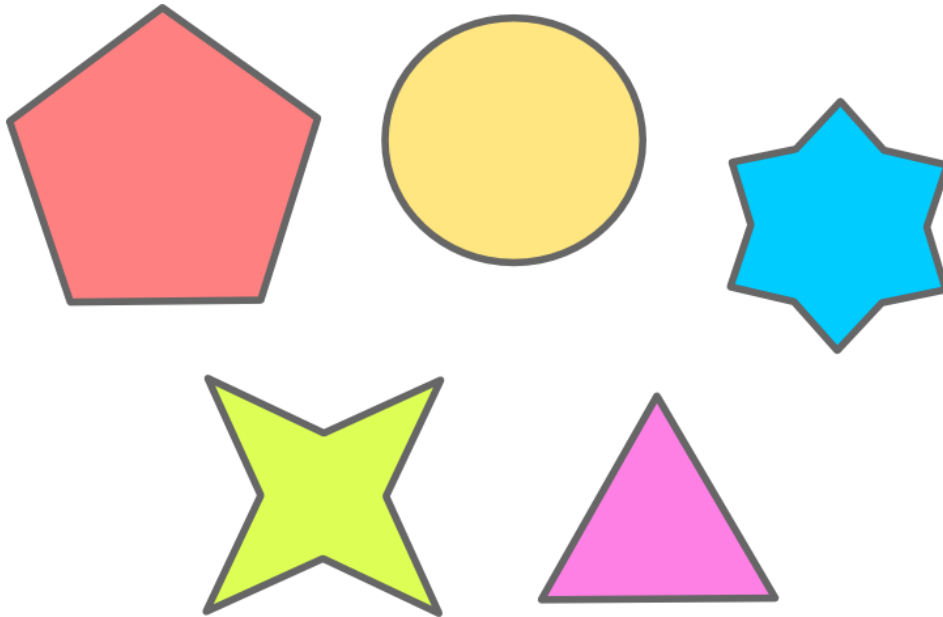
20 k $\Omega$  para a sua  
conveniência 😊

# ARDUINO REFERENCE

<http://arduino.cc/en/Reference/HomePage>

# Variáveis

- Às vezes você quer guardar um dado: número, texto, etc.
- Variáveis são “caixinhas” pra fazer isso



- [char](#)
- [byte](#)
- [int](#)
- [unsigned int](#)
- [long](#)
- [unsigned long](#)
- [float](#)
- [double](#)

# Variáveis: o que dá pra fazer?

- Atribuição: “guardar” um valor, pra usar depois

```
int outputPin;
```

```
outputPin = 13;
```

```
pinMode(outputPin, OUTPUT); // pino 13 usado como saída
```

- Operações lógicas/aritméticas

```
double v = 5, i = 0.02;
```

```
double r = v/i; // r guarda o valor do resistor pela lei de Ohm
```

# Expressões e operadores

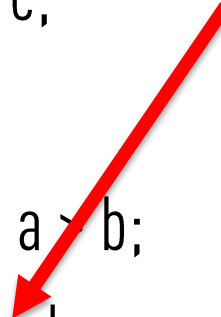
- Operadores aritméticos, por exemplo:
  - Soma/subtração: `int a = b+c;`
  - Multiplicação/divisão: `int delta = b*b-4*a*c;`
- Operadores comparação, por exemplo:
  - Maior/menor que: `bool greaterThanB = a > b;`
  - Igualdade: `bool equals_b = a == b;`
- Operadores booleanos
  - “E” lógico: `bool greaterOrEqual = (a > b) && (a == b);`

# Expressões e operadores

- Operadores aritméticos, por exemplo:
  - Soma/subtração: `int a = b+c;`
  - Multiplicação/divisão: `int delta = b*b-4*a*c;`
- Operadores comparação, por exemplo:
  - Maior/menor que: `bool greaterThanB = a > b;`
  - Igualdade: `bool equals_b = a == b;`
- Operadores booleanos
  - “E” lógico: `bool greaterOrEqual = (a > b) && (a == b);`



**== NÃO É =**





# Expressões e operadores

- Operadores aritméticos, por exemplo:
  - Soma/subtração: `int a = b+c;`
  - Multiplicação/divisão: `int delta = b*b-4*a*c;`
- Operadores comparação, por exemplo:
  - Maior/menor que: `bool greaterThanB = a > b;`
  - Igualdade: `bool equals_b = a == b;`
- Operadores booleanos
  - “E” lógico: `bool greaterOrEqual = (a > b) && (a == b);`



**PRECEDÊNCIA:  
NA DÚVIDA,  
USEM ()**



# Estruturas condicionais

- O caminho de um programa é guiada pela avaliação de variáveis e expressões lógicas/aritméticas

- Como na vida real:

`se(temperatura > 37)`

`ir_a_praia();`

- Estruturas condicionais permitem que um programa se comporte de formas diferentes em situações diferentes

# Condições

- Baseadas no resultado de expressões
  - `true`: qualquer resultado não-nulo
  - `false`: somente o resultado nulo
- Não precisam estar guardados numa variável  
`enquanto(4+3*2-2*5)`  
`corra(muito_rapido, pra_longe);`

# if

- Se condição for verdadeira, executa bloco de código

```
int a = 10;
```

```
if((a > 2) && (a < 15))
```

```
{
```

```
    // código que vai ser executado se  $2 < a < 15$ 
```

```
}
```

# if/else

- Se condição for verdadeira, executa primeiro bloco de código. Caso contrário, executa segundo bloco de código.

```
int luminosidade = analogRead(A1);
```

```
if(luminosidade < 50)
```

```
    digitalWrite(13, HIGH);    // tá escuro! :( vou ligar a luz
```

```
else
```

```
    digitalWrite(13, LOW);    // tá claro! :) vou apagar a luz
```

# switch/case

- Compara uma variável a várias possibilidades, executa a partir do rótulo de valor igual a variável
- O comando “break” possibilita sair do switch depois de executar o rótulo (sem isso, ele passa por todos rótulos)
- Bom para evitar uma cadeia longa de if/else

<http://arduino.cc/en/Reference/SwitchCase>

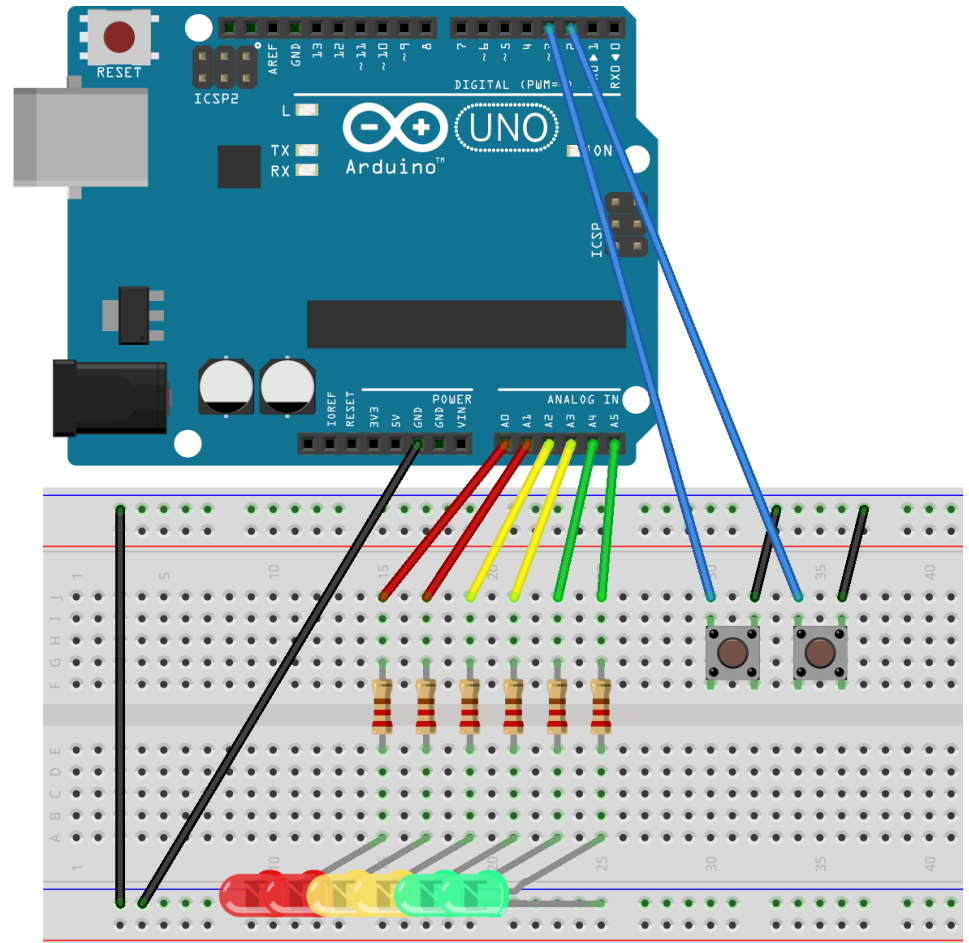


An ornate, symmetrical decorative frame in a light color, possibly white or cream, set against a dark background. The frame features intricate scrollwork, acanthus leaves, and flowing lines that curve upwards at the top and downwards at the bottom, creating a classic, elegant border.

Intermission

# Vamos montar!

- 6 resistores 330R
- 6 LEDs
- 2 botões
- Fios jumper
- Protoboard
- Arduino



# Lesson2\_DigitalInputPullup

Exemplo do Arduino com botão (10 min)

# Lesson2\_BlinkButton

Pequena modificação do código (10 min)

# Lesson2\_OnOffButton

Usar variável para armazenar estado, debounce (10 min)

# Desafio

- Lesson2\_ProgressBar:
  - “Animação” acendendo e apagando LEDs, como se estivesse enchendo uma barrinha de progresso.
- Lesson2\_UpDownBar:
  - Agora a barrinha enche apertando o botão 2 e volta com o botão 3

LEDs: pinos A0, A1, A2, A3, A4 e A5

Botões: pinos 2 e 3

Vocês têm até o final da aula pra fazer isso 😊