

**INSTITUTO  
FEDERAL**  
Santa Catarina

Câmpus  
Chapecó



**Arduino/Projeto AMA(2022)**

# O que é Arduino ?

- O Arduino foi criado em 2005 por um grupo de 5 pesquisadores : Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. O objetivo era elaborar um dispositivo que fosse ao mesmo tempo barato, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas amadores.
- <https://www.filipeflop.com/blog/o-que-e-arduino/>

Arduino é uma ferramenta de código aberto usado na construção de projetos eletrônicos(*protótipos*).

Arduino é composto por uma placa física programável, um circuito e um ambiente de desenvolvimento, ou IDE, que é executado em seu computador, é utilizado para escrever(linguagem c) e fazer upload de código do computador para a placa.

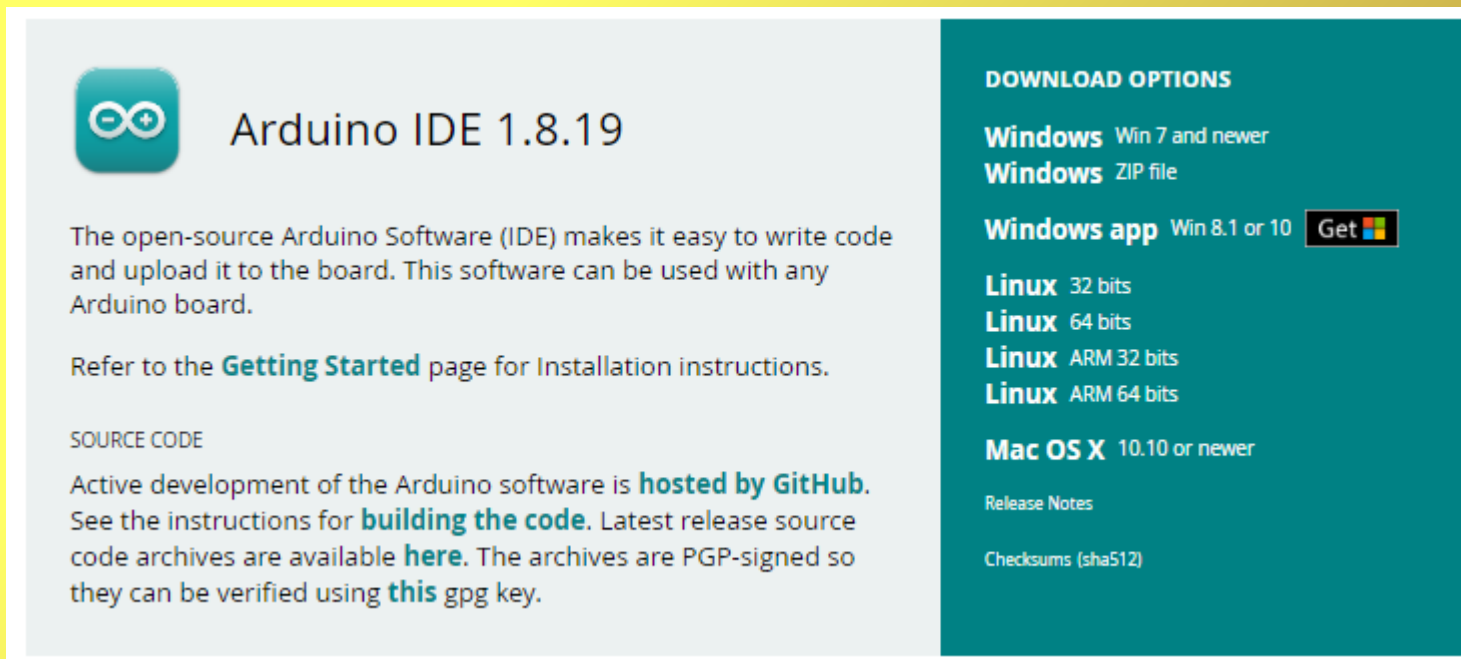
Depois de programado, o microcontrolador pode ser usado de forma independente, ou seja, você pode colocá-lo para controlar um robô, uma lixeira, um ventilador, as luzes da sua casa, a temperatura do ar condicionado, pode utilizá-lo como um aparelho de medição ou qualquer outro projeto que vier à cabeça.




# Como programar

- Para programar o Arduino é necessário realizar o Download do aplicativo no link abaixo:

<https://downloads.arduino.cc/arduino-1.8.19-windows.exe>



 **Arduino IDE 1.8.19**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.


Refer to the [Getting Started](#) page for Installation instructions.

**SOURCE CODE**

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

**DOWNLOAD OPTIONS**

**Windows** Win 7 and newer  
**Windows** ZIP file

**Windows app** Win 8.1 or 10 [Get](#) 

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

**Mac OS X** 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)

# Configuração do modelo e da porta de comunicação

The screenshot shows the Arduino IDE interface with the 'Ferramentas' menu open. The menu items are:

- Autoformatação (Ctrl+T)
- Arquivar Sketch
- Corrigir codificação e recarregar
- Gerenciar Bibliotecas... (Ctrl+Shift+I)
- Monitor serial (Ctrl+Shift+M)
- Plotter serial (Ctrl+Shift+L)
- WiFi101 / WiFININA Firmware Updater
- Placa: "Arduino Uno" (selected)
- Porta: "COM3"
- Obter informações da Placa
- Programador: "BusPirate as ISP"
- Gravar Bootloader

The 'Placa: "Arduino Uno"' item is expanded, showing a sub-menu with the following options:

- Gerenciador de Placas...
- Arduino AVR Boards (selected)
- ESP8266 Boards (3.0.2)

The 'Gerenciador de Placas...' sub-menu is also expanded, showing a list of board models:

- Arduino Yún
- Arduino Uno (selected)
- Arduino Duemilanove or Diecimila
- Arduino Nano
- Arduino Mega or Mega 2560
- Arduino Mega ADK
- Arduino Leonardo

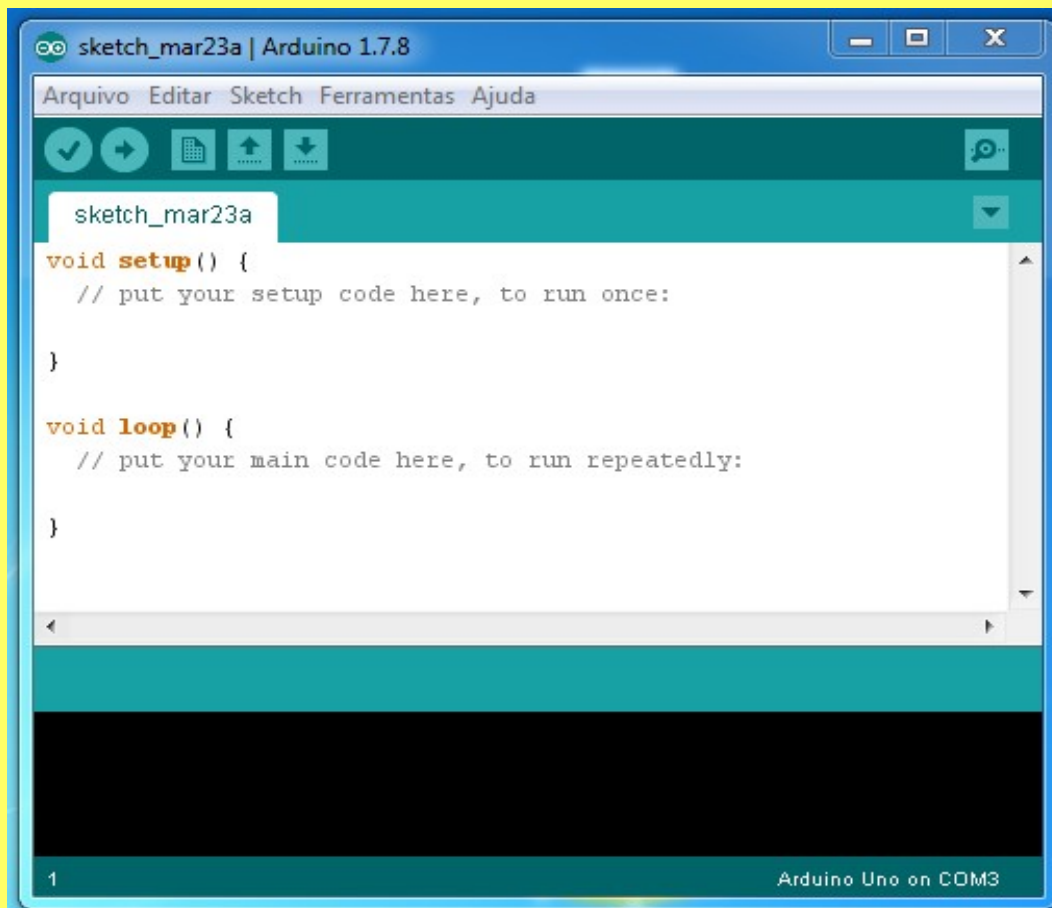
The background shows a code editor with the following code:

```
1 void se
2 // pu
3
4 }
5
6 void lo
7 // pu
8
```

Primeiro de tudo é importante explicar que a linguagem de programação (c) Arduino é sensível a maiúsculas: uma letra maiúscula não é o mesmo que uma letra minúscula.

O código a seguir representa o mínimo para que um programa possa ser compilado:

O "void setup ()" é normalmente usado para inicializar variáveis , modos de pinos , definição a velocidade de transmissão serial, etc. O software só vai executar uma vez.



```
sketch_mar23a | Arduino 1.7.8
Arquivo  Editar  Sketch  Ferramentas  Ajuda
sketch_mar23a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
1
Arduino Uno on COM3
```

O "void loop ()" é a parte do código em que se faz um loop, ou seja, cria-se um ciclo de repetição para que as instruções dentro do mesmo sejam repetidas.

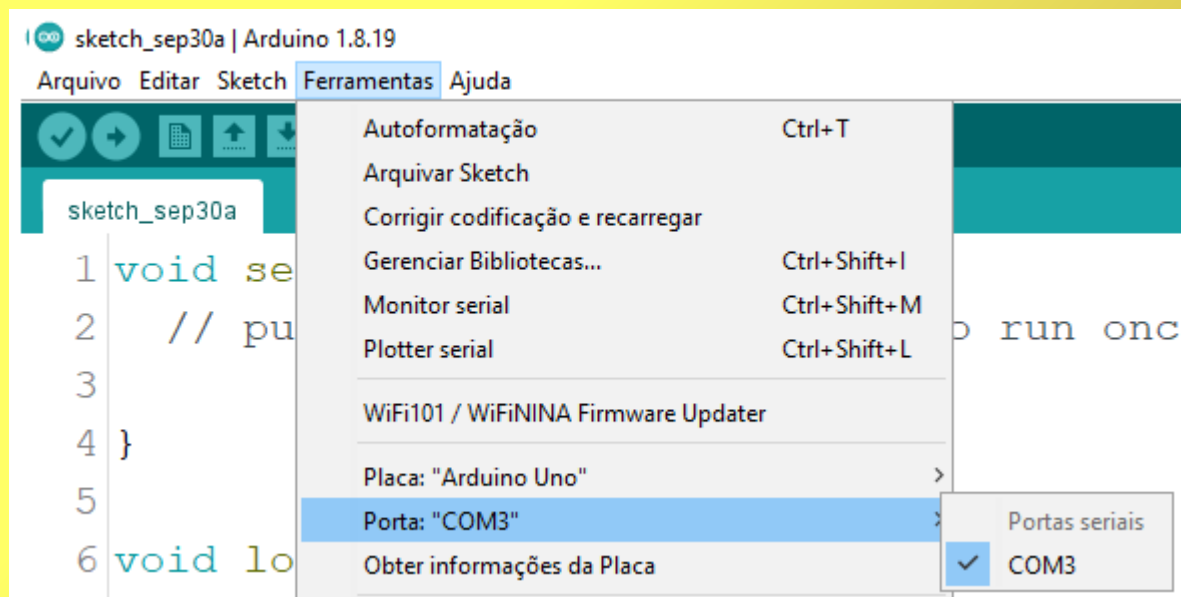
# O que você pode fazer com o Arduino ?

A lista de possibilidades é praticamente infinita. Você pode automatizar sua casa, seu carro, seu escritório, criar um novo brinquedo, um novo equipamento ou melhorar um já existente. Tudo vai depender da sua criatividade.

Para isso, o Arduino possui uma quantidade enorme de *sensores e componentes* que você pode utilizar nos seus projetos. Grande parte do material utilizado está disponível em módulos, que são pequenas placas que contêm os sensores e outros componentes auxiliares como resistores, capacitores e leds.

A porta de comunicação geralmente é maior que COM1 e COM2. Caso não apareça alguma porta maior é necessário executar um aplicativo simulador de portas através do seguinte link:

<https://projetoama.migueldebarba.com.br/CH341SER.EXE>

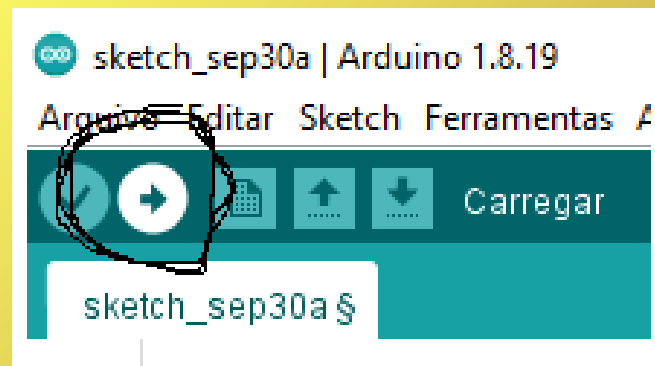


# Primeiro projeto

Vamos ligar o Led existente na placa do Arduino:

```
1 void setup() {  
2   pinMode(LED_BUILTIN, OUTPUT); // Configura o Led para saída  
3 }  
4  
5 void loop() {  
6   digitalWrite(LED_BUILTIN, HIGH); // Liga o Led  
7 }
```

Em seguida carregue o código para a placa realizar a atividade programada:



# Primeiro projeto

Vamos desligar o Led existente na placa do Arduino:

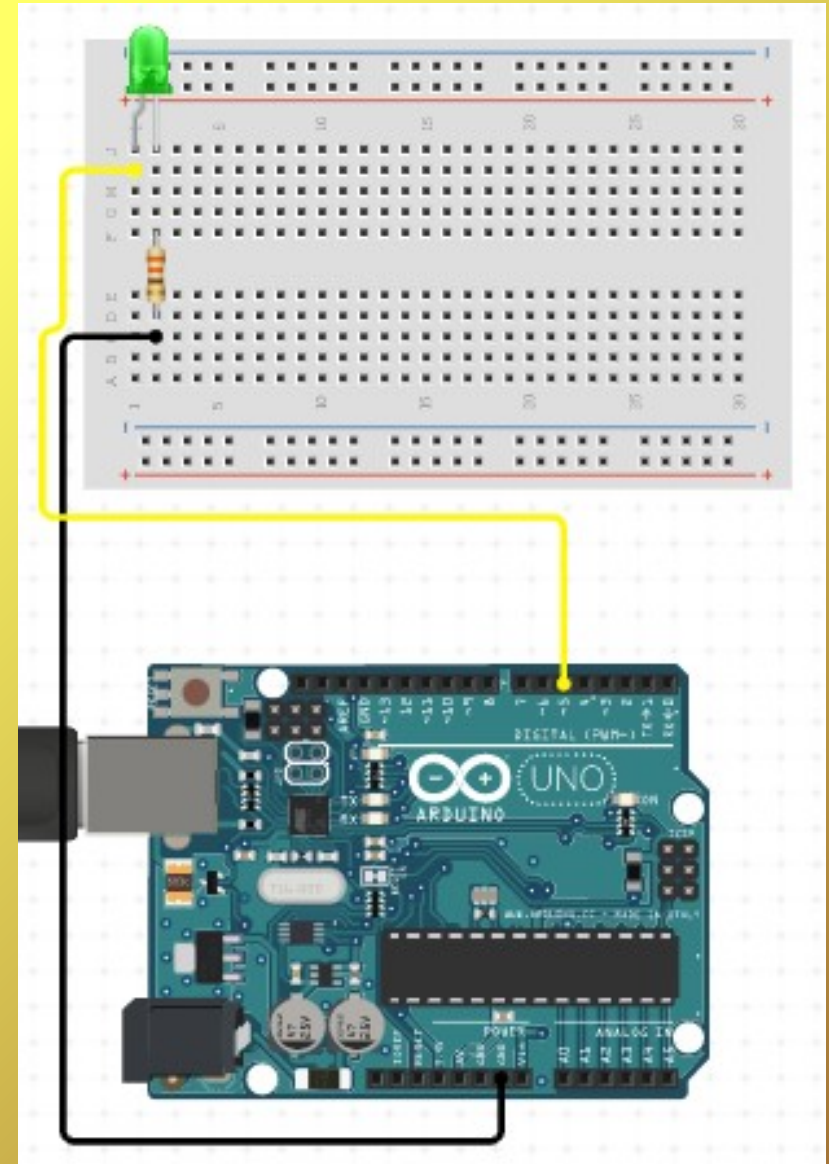
```
1 void setup() {
2   pinMode(LED_BUILTIN, OUTPUT); // Configura o Led para saída
3 }
4
5 void loop() {
6   digitalWrite(LED_BUILTIN, LOW); // Desliga o Led
7   delay(1000);
8 }
```

E para deixar o Led piscando?

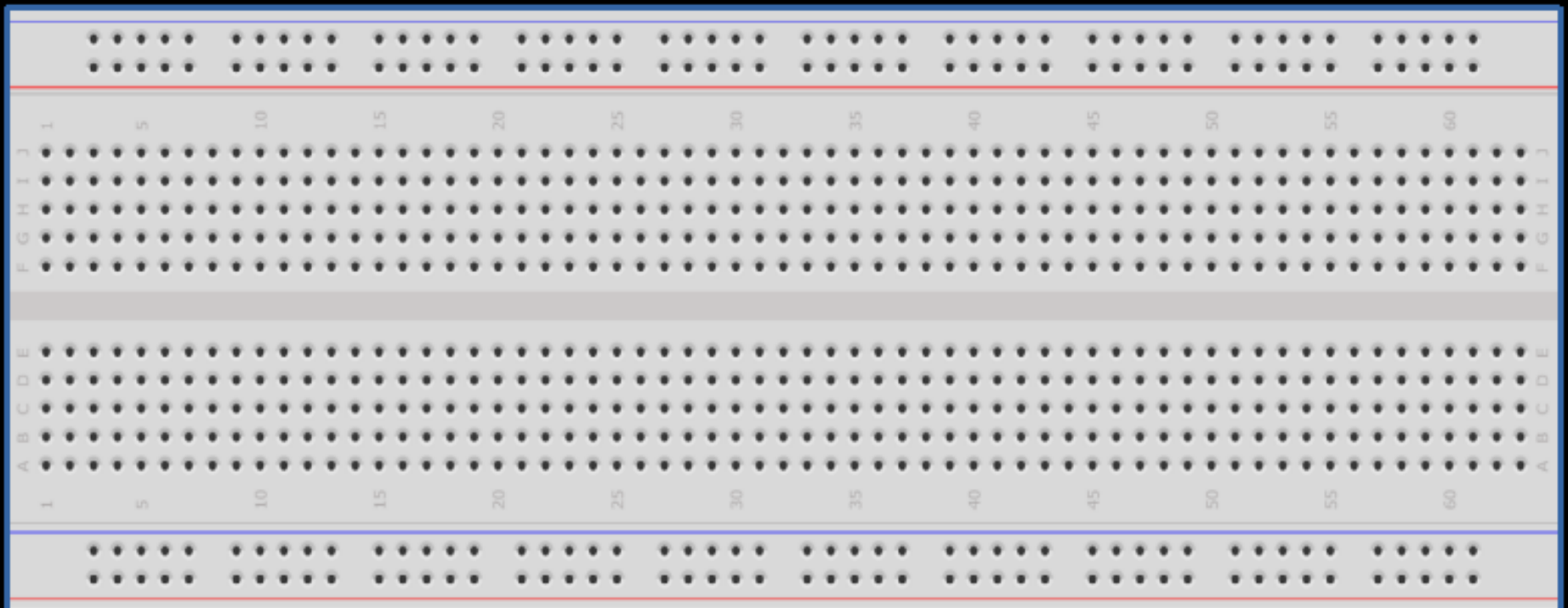
```
1 void setup() {
2   pinMode(LED_BUILTIN, OUTPUT); // Configura o Led para saída
3 }
4
5 void loop() {
6   digitalWrite(LED_BUILTIN, HIGH); // Liga o Led
7   delay(1000);
8   digitalWrite(LED_BUILTIN, LOW); // Desliga o Led
9   delay(1000);
10 }
```

# Próximo desafio

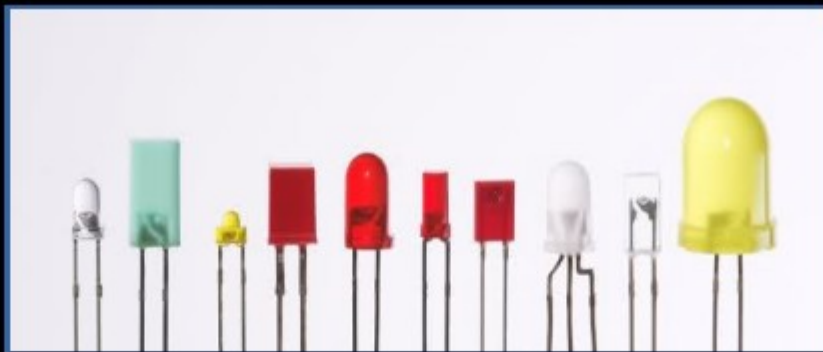
Acender um Led externo:  
Realizar a montagem do  
circuito conforme imagem  
ao lado. Utilizar  
protoboard e resistor.



A **“protoboard”** ou **“Matriz de contatos”** é utilizada para fazer montagens provisórias e/ou teste de projetos. É constituída por uma base plástica, contendo inúmeros orifícios destinados à inserção de terminais de componentes eletrônicos. Internamente existem ligações determinadas que interconectam os orifícios, permitindo a montagem de circuitos eletrônicos sem a utilização de solda.







O "**diodo emissor de luz**" também é conhecido pela sigla em inglês **LED** (Light Emitting Diode). Sua funcionalidade básica é a emissão de luz em locais e instrumentos onde se torna mais conveniente a sua utilização no lugar de uma lâmpada.



# Tensão dos Leds

Tensão: 3.1  
Amp: 0.02



1.8  
0.02



3.1  
0.02

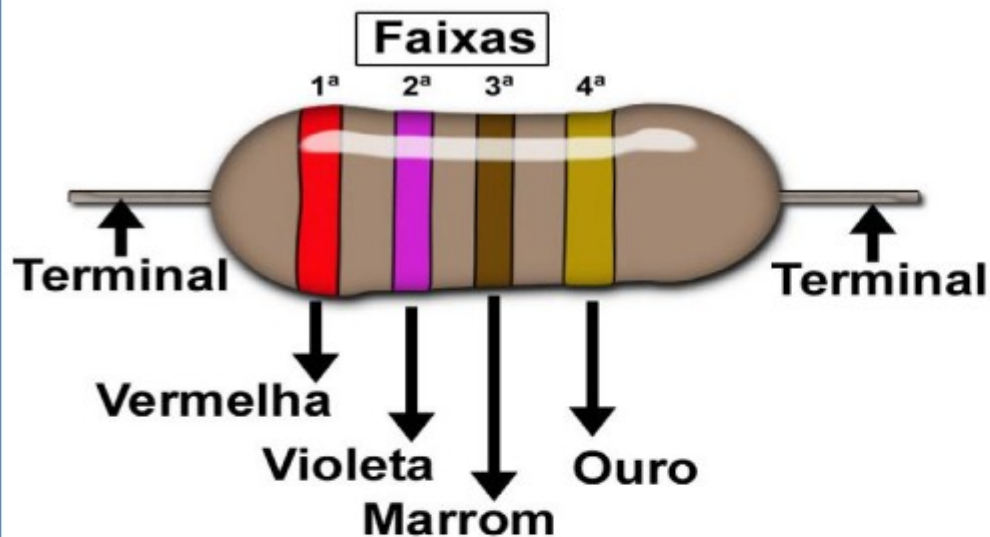


2.0  
0.015



2.1  
0.02





O valor da  
resistência é  
medido em OHM  
e seu símbolo é o  
Ômega Grego  $\Omega$

1 ohm ou  $1\Omega$   
1000 ohms =  $1K\Omega$   
1000  $K\Omega$  =  $1M\Omega$

Um **Resistor** (frequentemente chamado de resistência, que é na verdade a sua medida) é um dispositivo elétrico muito utilizado em eletrônica, ora com a finalidade de transformar energia elétrica em energia térmica por meio do efeito joule, ora com a finalidade de limitar a corrente elétrica em um circuito.

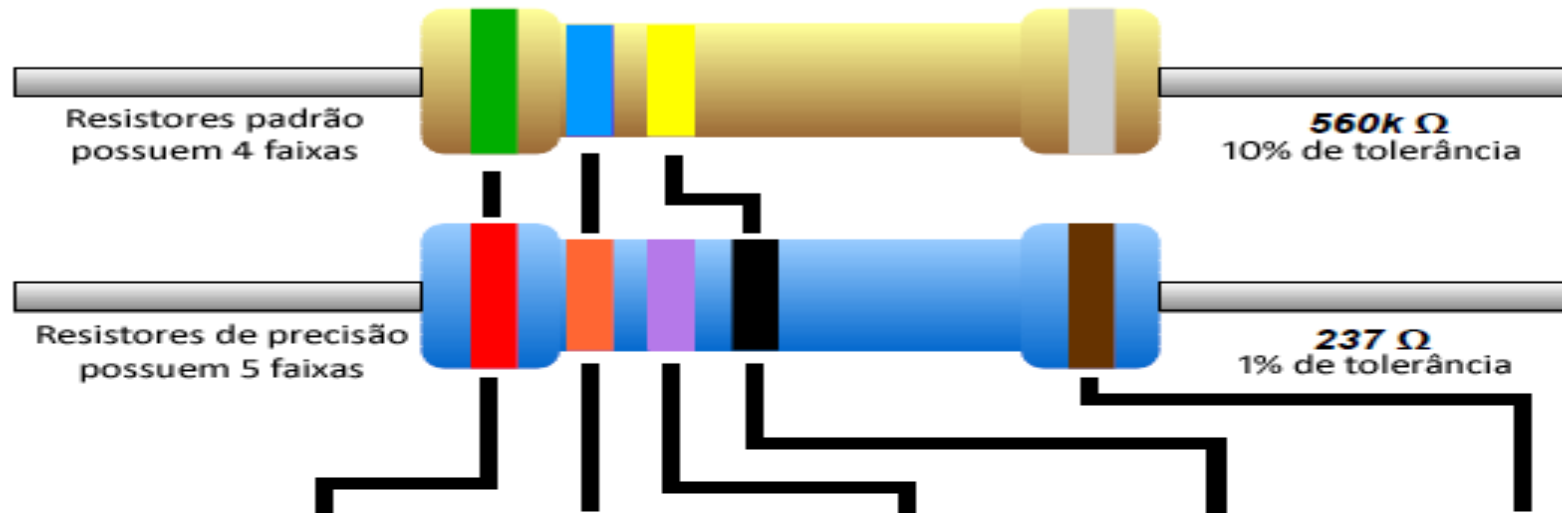
Tabela de tensão(v) para led's					
	Azul	Vermelho	Branco	Amarelo	Verde
Tensão(v)	3,1	1,8	3,1	2	2,1
Corrente(a)	0,020	0,020	0,020	0,015	0,020
Fonte	5	5	5	5	5
Tensão(v)	1,9	3,2	1,9	3	2,9
(R)Ohms	94	159	94	199	144

[http://www.audioacustica.com.br/exemplos/Valores\\_Resistores/Calculadora\\_Grafica\\_Resistores\\_4-bandas.html](http://www.audioacustica.com.br/exemplos/Valores_Resistores/Calculadora_Grafica_Resistores_4-bandas.html)

Tabela de resistores

# Código de Cores

A extremidade com mais faixas deve apontar para a esquerda



Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Preto	0	0	0	x 1 $\Omega$	
Marrom	1	1	1	x 10 $\Omega$	+/- 1%
Vermelho	2	2	2	x 100 $\Omega$	+/- 2%
Laranja	3	3	3	x 1K $\Omega$	
Amarelo	4	4	4	x 10K $\Omega$	
Verde	5	5	5	x 100K $\Omega$	+/- .5%
Azul	6	6	6	x 1M $\Omega$	+/- .25%
Violeta	7	7	7	x 10M $\Omega$	+/- .1%
Cinza	8	8	8		+/- .05%
Branco	9	9	9		
Dourado				x .1 $\Omega$	+/- 5%
Prateado				x .01 $\Omega$	+/- 10%

# Programação

```
1 const int LedVerde = 2;    // inicializa LED na porta digital 2
2 void setup() {
3     // inicializa LED como sendo output.
4     pinMode(LedVerde, OUTPUT);
5 }
6 void loop() {
7     digitalWrite(LedVerde, HIGH); // LED (HIGH tensão positiva +5v)
8     delay(1000);                 // aguarda 1000 milisegundos=1segundo
9     digitalWrite(LedVerde, LOW); // LED (LOW sem tensão de saída 0v)
10    delay(1000);                 // aguarda 1000 milisegundos=1segundo
11 }
```

Próximo desafio: ligar vários Led's na protoboard

