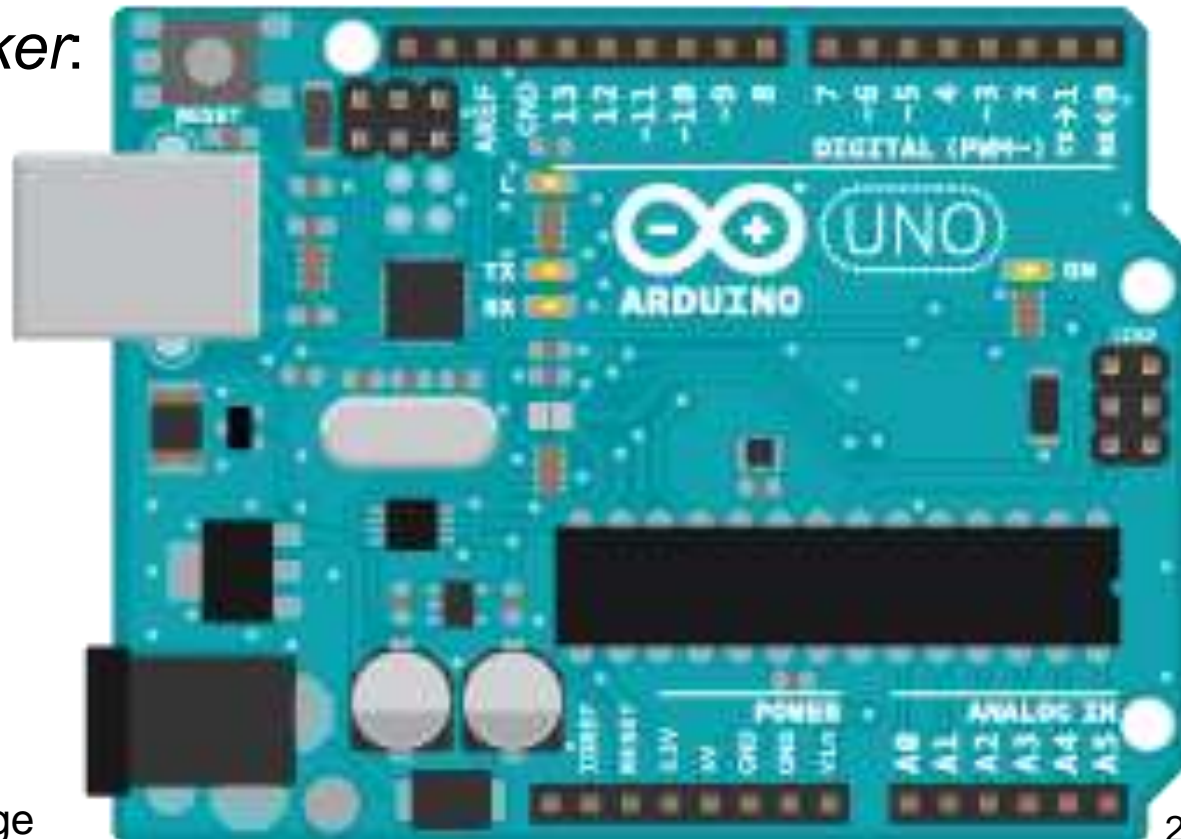


ARDUINO

Prof. Me. Pietro Martins de Oliveira

- **Arduino**

- Open-source (open hardware);
- Prototipação de projetos eletrônicos;
- Comunidade *maker*.
 - Artistas;
 - Designers;
 - Hobbistas;
 - Cientistas.
- www.arduino.cc



- **Tratamento de dados:**
 - Recebe, gera e interpreta dados;
 - Dados digitais e analógicos.
- **Automação de tarefas**
 - Controles residenciais;
 - Robótica;
 - Design interativo;
 - Projetos eletrônicos.

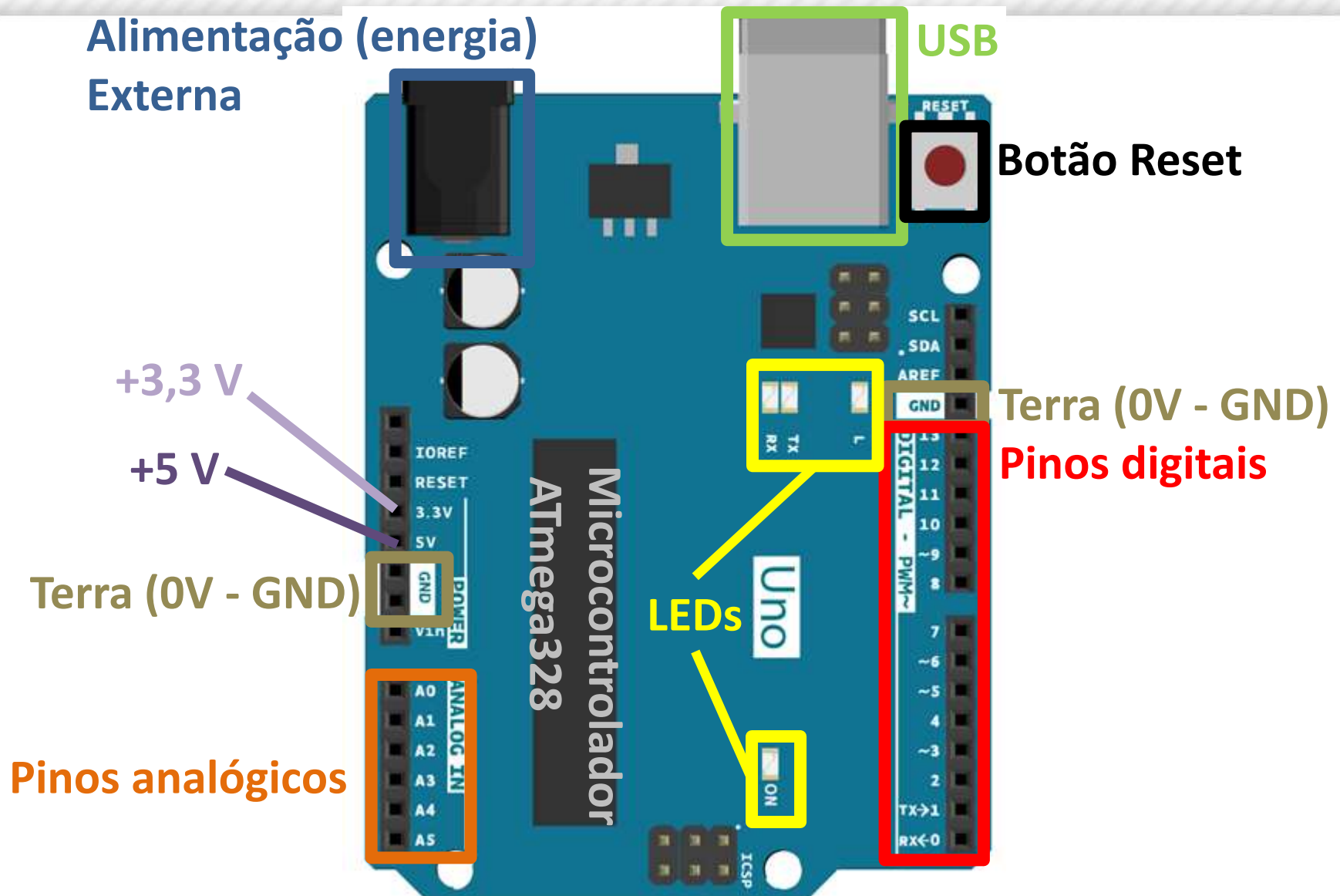
- **Dados analógicos:**

- O sinal é como uma onda contínua (não-discretos);
- Dados em formato original (não-tratados).

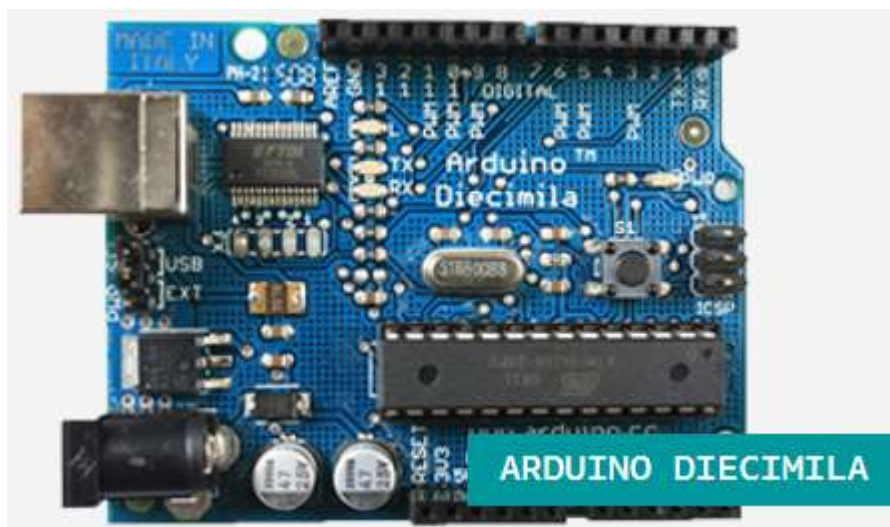
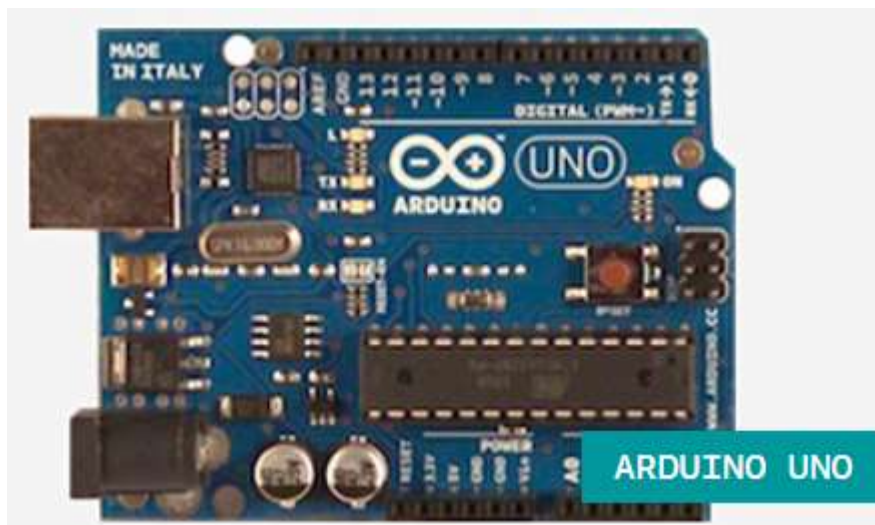
- **Dados digitais:**

- O sinal “convertido” em bits: 0 (zero) ou 1 (um);
- Onda analógica é utilizada para compor os dados digitais via amostragem;
- É possível fazer a conversão inversa: a partir dos dados digitais, gera-se uma onda analógica muito parecida com o sinal original (PWM).

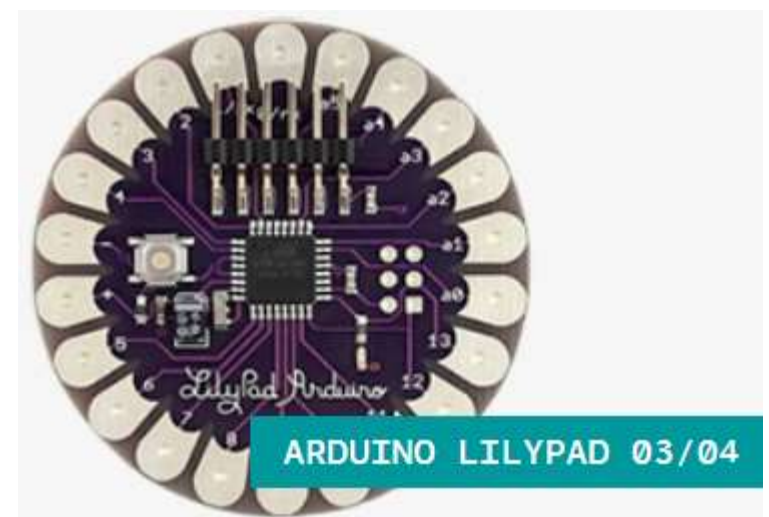
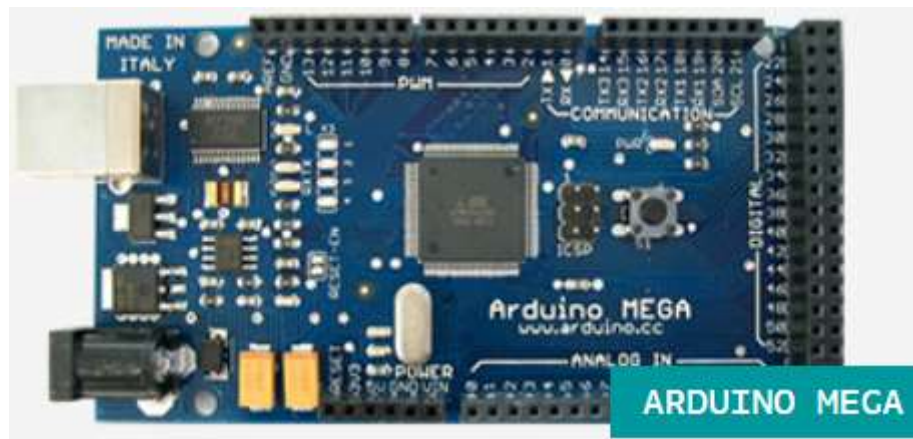
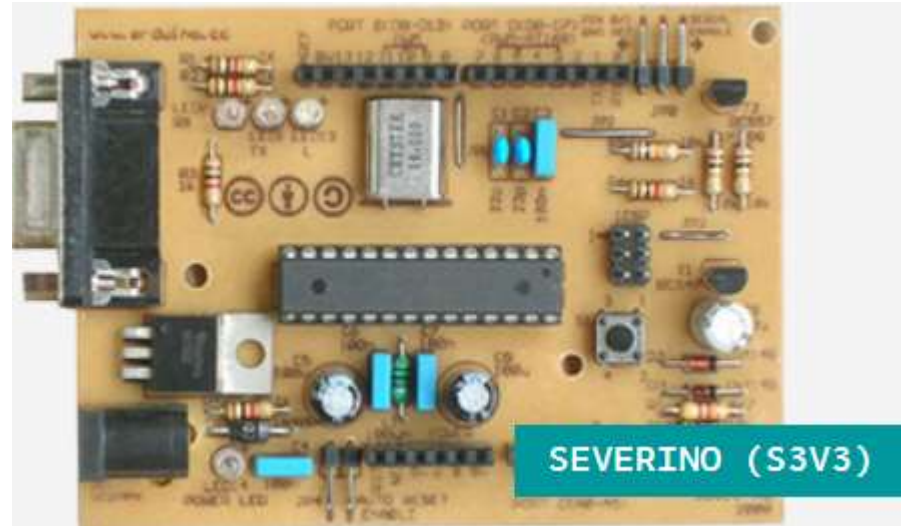
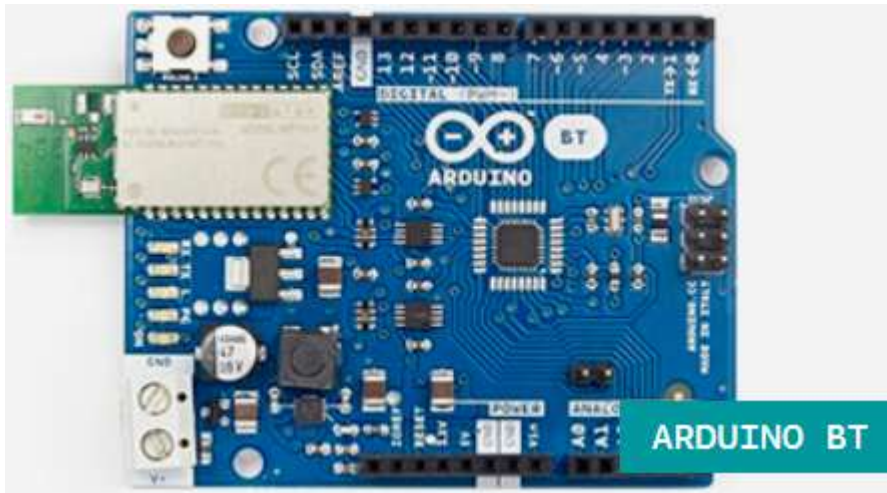
Detalhes do Hardware – Arduino Uno



Arduino - Versões

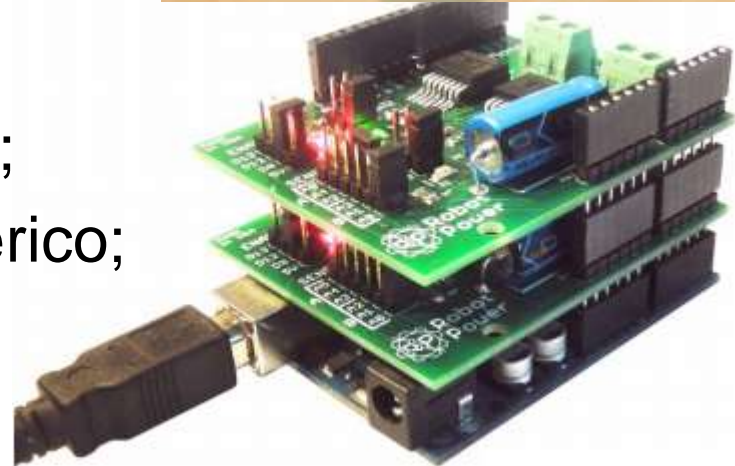


Arduino - Versões



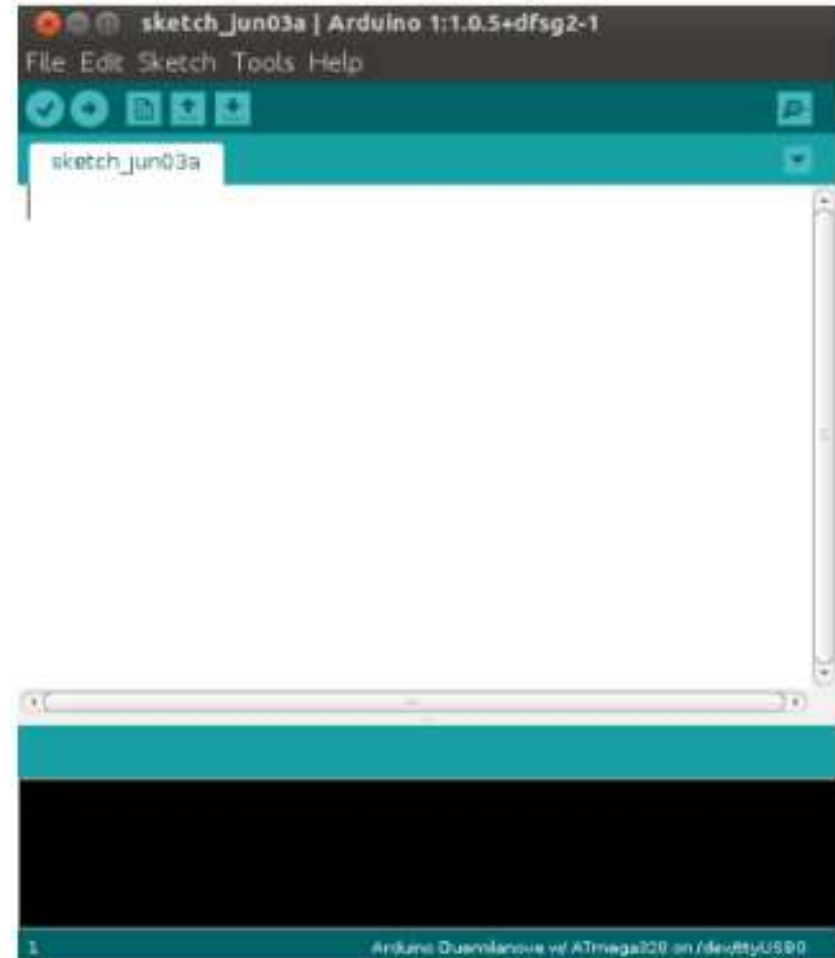
- **Shields são extensões para o Arduino:**

- Bluetooth;
- Conexão Ethernet;
- Ponte H (motores);
- Controladora USB;
- Protoboard;
- MicroSD/SD;
- MP3 player;
- Display: 7 segmentos / LCD;
- Teclado: numérico/alfanumérico;
- Sensores.



- **Ambiente de Desenvolvimento**

- C/C++;
- Corpo principal do programa:
 - Setup;
 - Loop.
- Funcionalidades:
 - Verify;
 - Upload;
 - Serial Monitor.



- Para mais detalhes e exemplos da utilização do Arduino:
 - MONK, Simon.
Programação com Arduino: começando com sketches. Porto Alegre: Bookman, 2013. 148 p. (Série Tekne) ISBN 978-85-8260-026-9



- ARDUINO HOMEPAGE. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 31 de agosto de 2018.
- MONK, Simon. **Programação com Arduino: começando com sketches**. Porto Alegre: Bookman, 2013.

ARDUINO

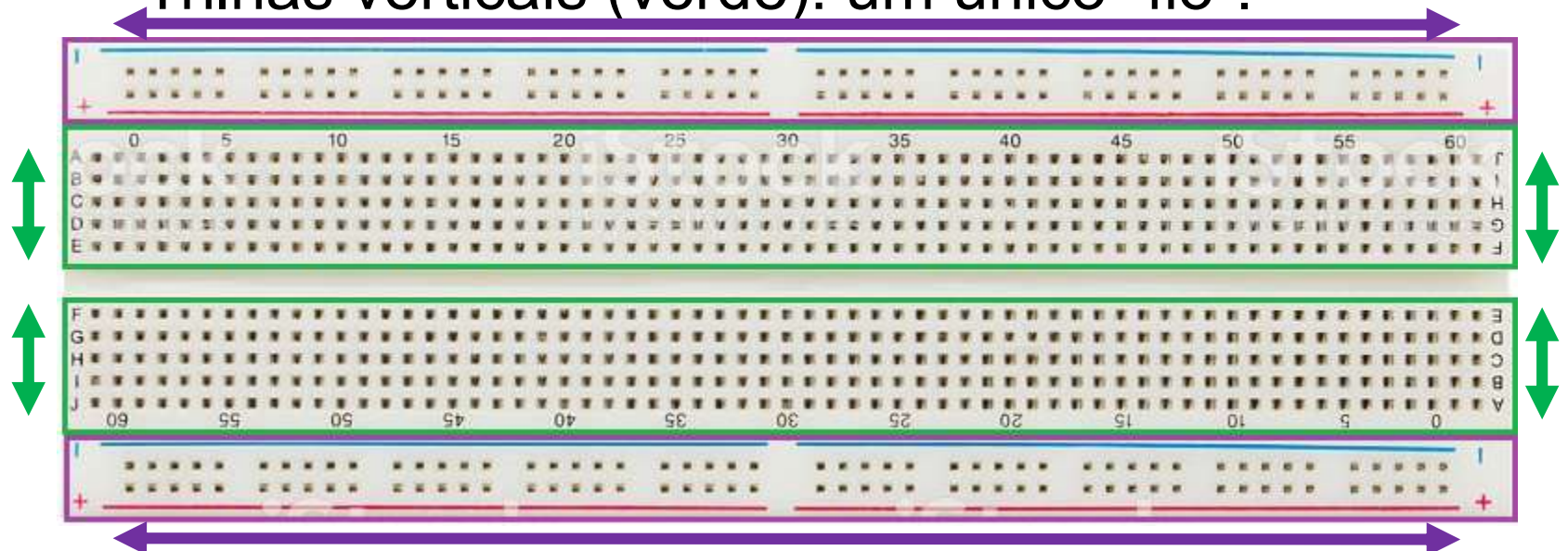
Prof. Me. Pietro Martins de Oliveira

INTRODUÇÃO À ELETRÔNICA BÁSICA

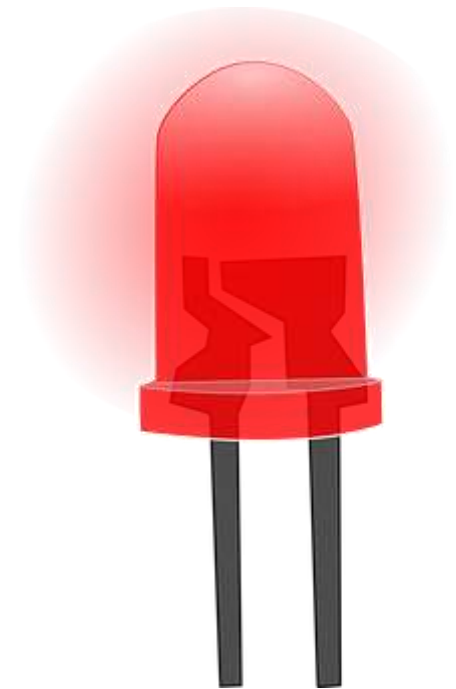
Prof. Me. Pietro Martins de Oliveira

- **Protoboard:**

- Emula o circuito final: elimina necessidade de soldar;
- Útil para conectar componentes e testar o circuito;
- Placa com conectores para criar protótipos:
 - Trilhas horizontais (roxo): um único “fio”;
 - Trilhas verticais (verde): um único “fio”.



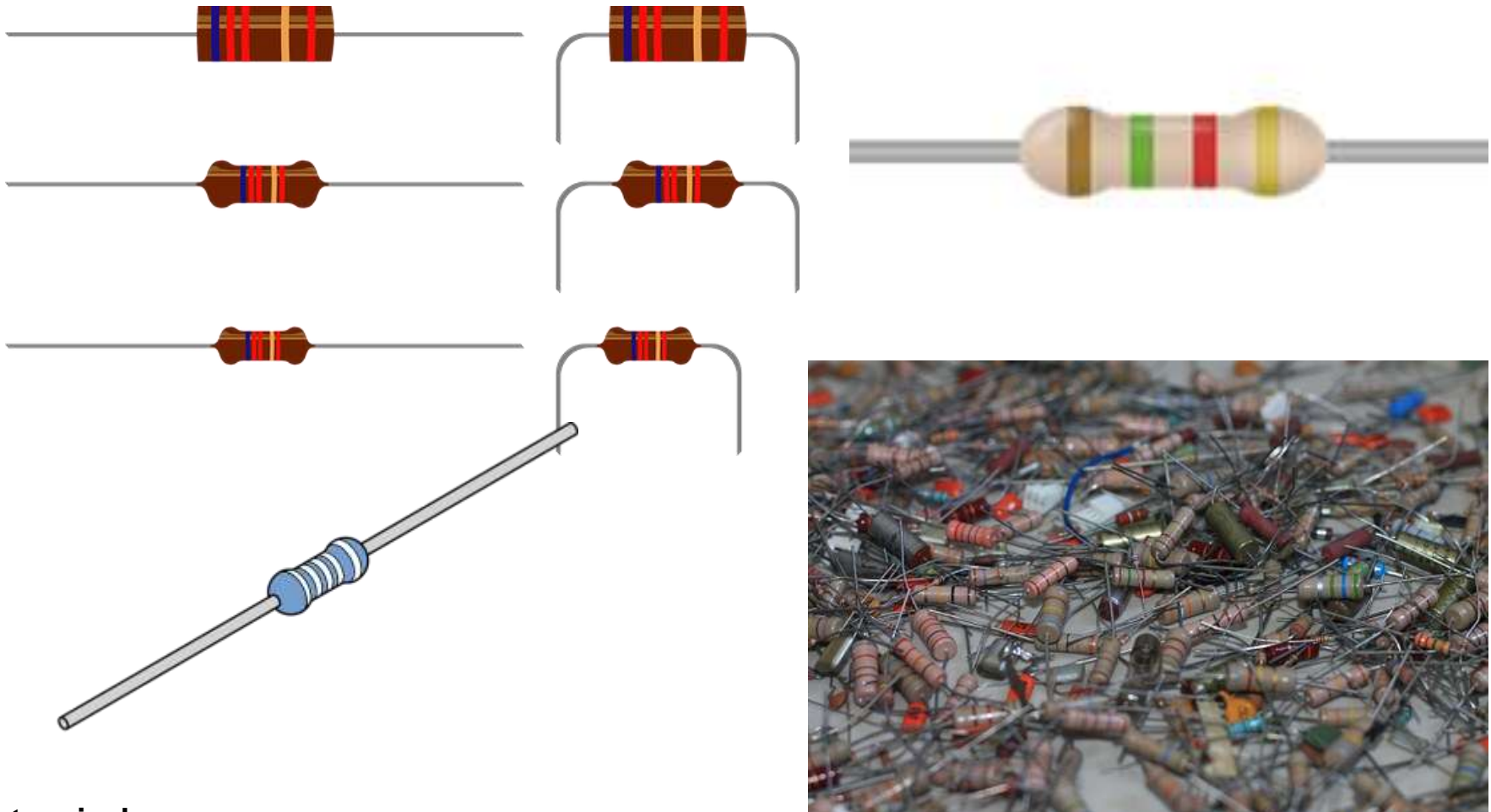
- **LEDs (*Light Emitting Diode*):**
 - Pequenas “lâmpadas”;
 - Várias cores / Tensões de alimentação variadas;
 - O acionamento de um LED depende das tensões aplicadas em seus terminais;
 - Só funciona em “um sentido”.



- **Resistência (*Resistor*):**

- Cria resistência na corrente elétrica, diminuindo a tensão em seus terminais;
- A unidade de medida desse componente é o OHM (letra grega ômega – Ω);
- Úteis para regular a tensão/corrente aplicada à um componente específico;
- Arduino: pinos digitais trabalham em 5 V (cinco volts) e 40 mA (quarenta miliampéres);
- É um componente que pode superaquecer caso sejam aplicadas corrente/voltagem muito altas em seus terminais.

- Resistências



- **Resistências:**

- Como saber qual resistor utilizar? Devemos saber quais são as especificações dos componentes que serão conectados ao resistor: corrente & voltagem;
- Pode-se empregar a seguinte fórmula para calcular qual é o resistor mais adequado:

$$R = \frac{Vf - Vc}{i}$$

- Onde ***R*** é o valor da resistência, ***Vc*** é a voltagem do componente que será conectado ao resistor, ***Vf*** é a voltagem fornecida e ***i*** é a corrente do componente.

- **Exemplo de Cálculo de Resistência**

- Considere que o componente é um LED verde que utiliza necessita de uma tensão de 2 V (dois volts) e 35 mA (trinta e cinco miliampéres). Sabendo que o Arduino oferece 5 V de tensão, temos:

$$R = \frac{V_f - V_c}{i}$$

$$R = \frac{5 - 2}{0.035}$$

$$R = 85,71 \, \Omega$$

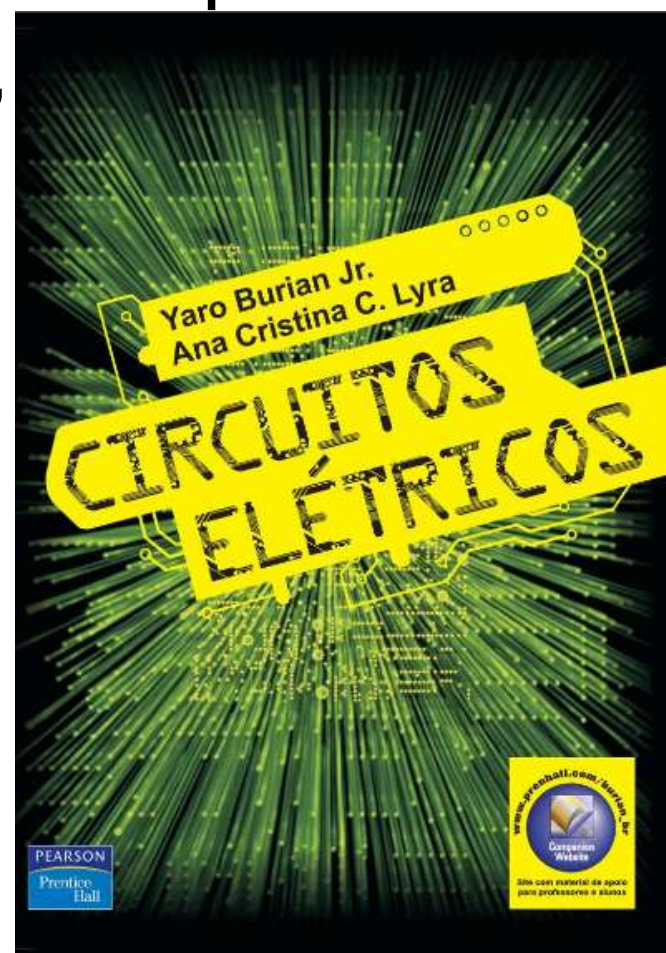
- **Exemplo de Cálculo de Resistência:**
 - Não existe um resistor cuja resistência mede exatamente $85,71\ \Omega$.
 - Devemos encontrar um resistor com valor de resistência próximo a $85,74\ \Omega$.
 - Devemos garantir que o LED não seja danificado por receber tensão ou corrente maiores que o especificado: escolhe-se um valor maior que $85,74\ \Omega$.
 - O mais próximo, para garantir segurança, é o resistor de $100\ \Omega$.

Protótipos com Componentes Eletrônicos

Cor	Primeira faixa	Segunda faixa	Terceira faixa (multiplicador)	Quarta faixa (tolerância)
Preto	0	0	$\times 10^0$	
Marrom	1	1	$\times 10^1$	$\pm 1\%$
Vermelho	2	2	$\times 10^2$	$\pm 2\%$
Laranja	3	3	$\times 10^3$	
Amarelo	4	4	$\times 10^4$	
Verde	5	5	$\times 10^5$	$\pm 0,5\%$
Azul	6	6	$\times 10^6$	$\pm 0,25\%$
Violeta	7	7	$\times 10^7$	$\pm 0,1\%$
Cinza	8	8	$\times 10^8$	$\pm 0,05\%$
Branco	9	9	$\times 10^9$	
Dourado			$\times 10^{-1}$	$\pm 5\%$
Prata			$\times 10^{-2}$	$\pm 10\%$
Nenhuma				$\pm 20\%$



- Para mais informações a respeito de circuitos elétricos e seus respectivos componentes:
 - BURIAN JUNIOR, Yaro; LYRA, Ana Cristina C. **Circuitos elétricos**. São Paulo: Prentice-Hall, 2006.
ISBN 9788576050728



- BURIAN JUNIOR, Yaro; LYRA, Ana Cristina C. **Circuitos elétricos**. São Paulo: Prentice-Hall, 2006.

INTRODUÇÃO À ELETRÔNICA BÁSICA

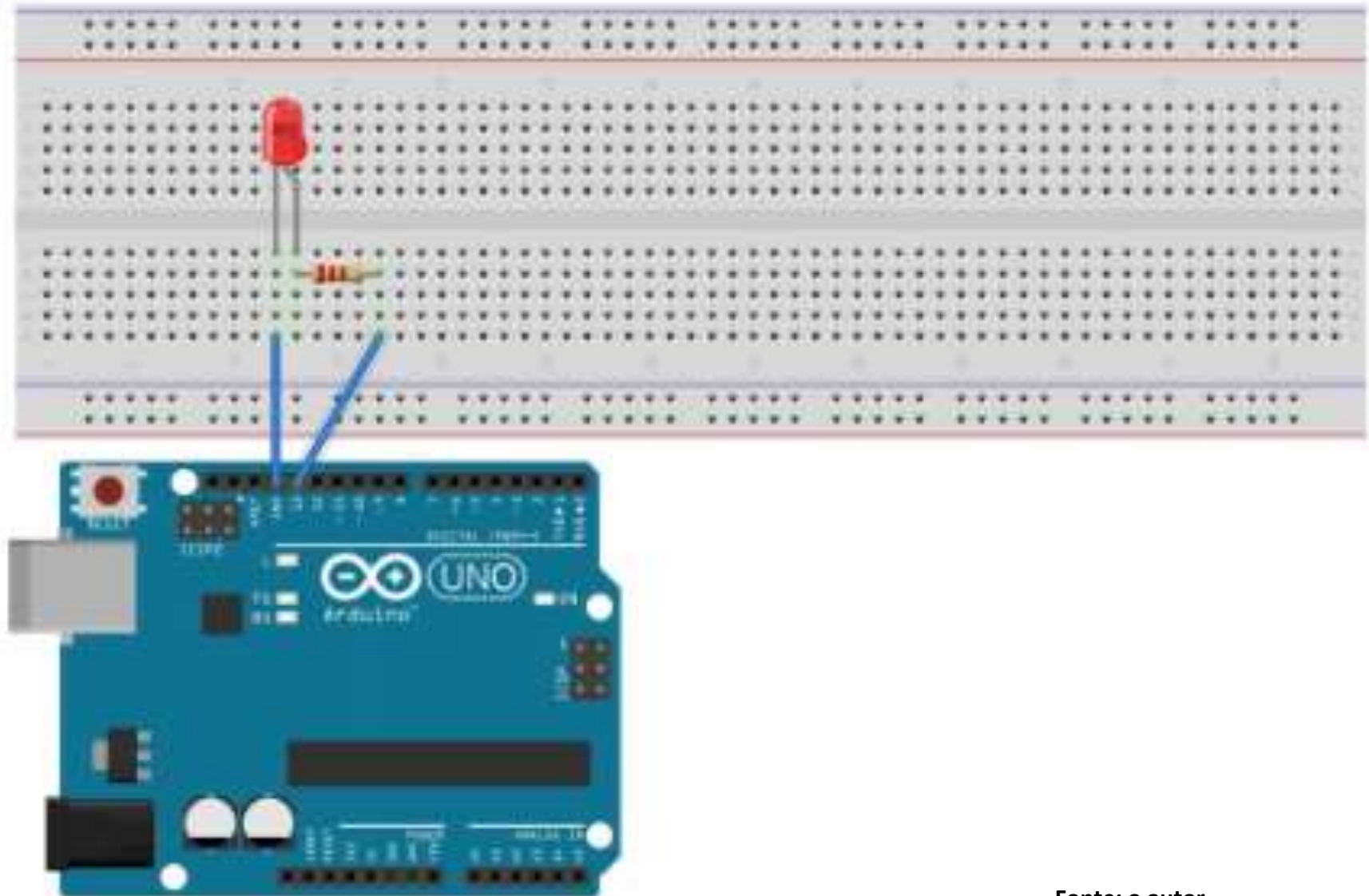
Prof. Me. Pietro Martins de Oliveira

EXEMPLOS: LED BLINK

Prof. Me. Pietro Martins de Oliveira

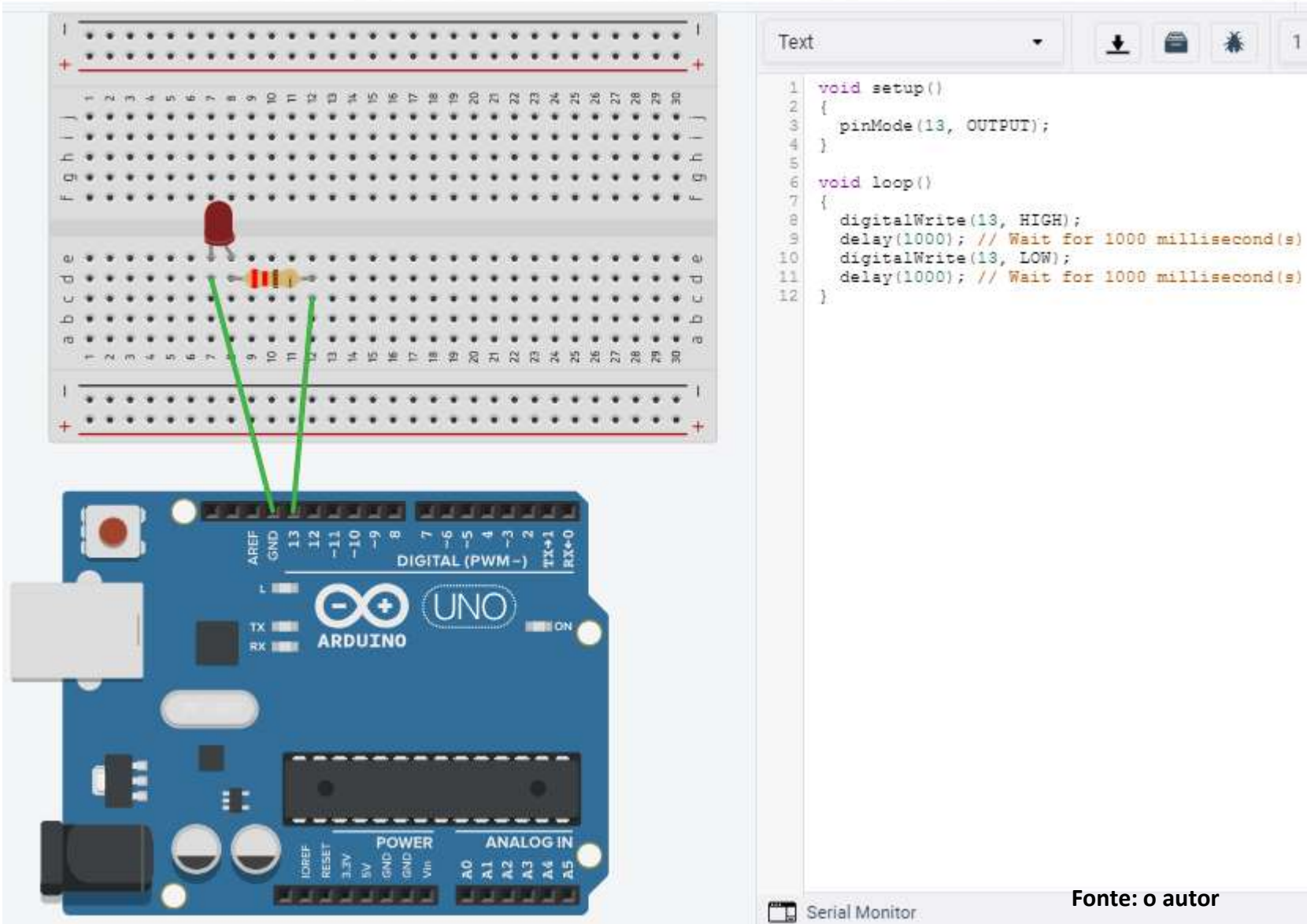
- **O exemplo clássico em microcontroladores:**
 - Piscar um LED (led blink);
 - Conectar o Arduino ao computador;
 - Abrir a IDE do Arduino;
 - Transcrever o código do exemplo para a IDE;
 - Ler o programa para interpretar os comandos;
 - Compilar o programa;
 - Enviar (upload) o programa da IDE para o Arduino;
 - Observar a execução (simulador).
 - <https://www.tinkercad.com>

Hardware – Exemplo “Pisca-pisca”



Código – Exemplo “Pisca-pisca”

```
/*  
  Blink  
  Acende o LED por um segundo e depois desliga por um segundo, repetidamente.  
*/  
  
// Essa variavel tem um led conectado e o pino utilizado sera o de numero 13.  
int led = 13;  
  
// O metodo setup e o estado inicial do programa, o primeiro a ser invocado:  
void setup() {  
  // inicia a variavel led como OUTPUT, ou seja, uma saida que enviara dados ao pino 13.  
  pinMode(led, OUTPUT);  
}  
  
// O metodo loop sera executado repetidamente ate que se encerre o programa:  
void loop() {  
  digitalWrite(led, HIGH);    // liga o led (HIGH e o nivel de voltagem)  
  delay(1000);                // espera por um segundo  
  digitalWrite(led, LOW);     // desliga o led tornando a voltagem LOW  
  delay(1000);                // espera por um segundo  
}
```

The image shows a digital circuit simulation environment. On the left, an Arduino Uno board is connected to a breadboard. A red LED is connected to digital pin 13 (anode) and a 220Ω resistor is connected to the same pin (cathode). The other end of the resistor is connected to ground. The code in the Serial Monitor is as follows:

```

1 void setup()
2 {
3   pinMode(13, OUTPUT);
4 }
5
6 void loop()
7 {
8   digitalWrite(13, HIGH);
9   delay(1000); // Wait for 1000 millisecond(s)
10  digitalWrite(13, LOW);
11  delay(1000); // Wait for 1000 millisecond(s)
12 }
  
```

At the bottom right, the text "Serial Monitor" is visible, and the source is attributed to "Fonte: o autor".

- ARDUINO HOMEPAGE. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 31 de agosto de 2018.
- MONK, Simon. **Programação com Arduino: começando com sketches**. Porto Alegre: Bookman, 2013.

EXEMPLOS: LED BLINK

Prof. Me. Pietro Martins de Oliveira

EXEMPLOS: SINAIS ANALÓGICOS

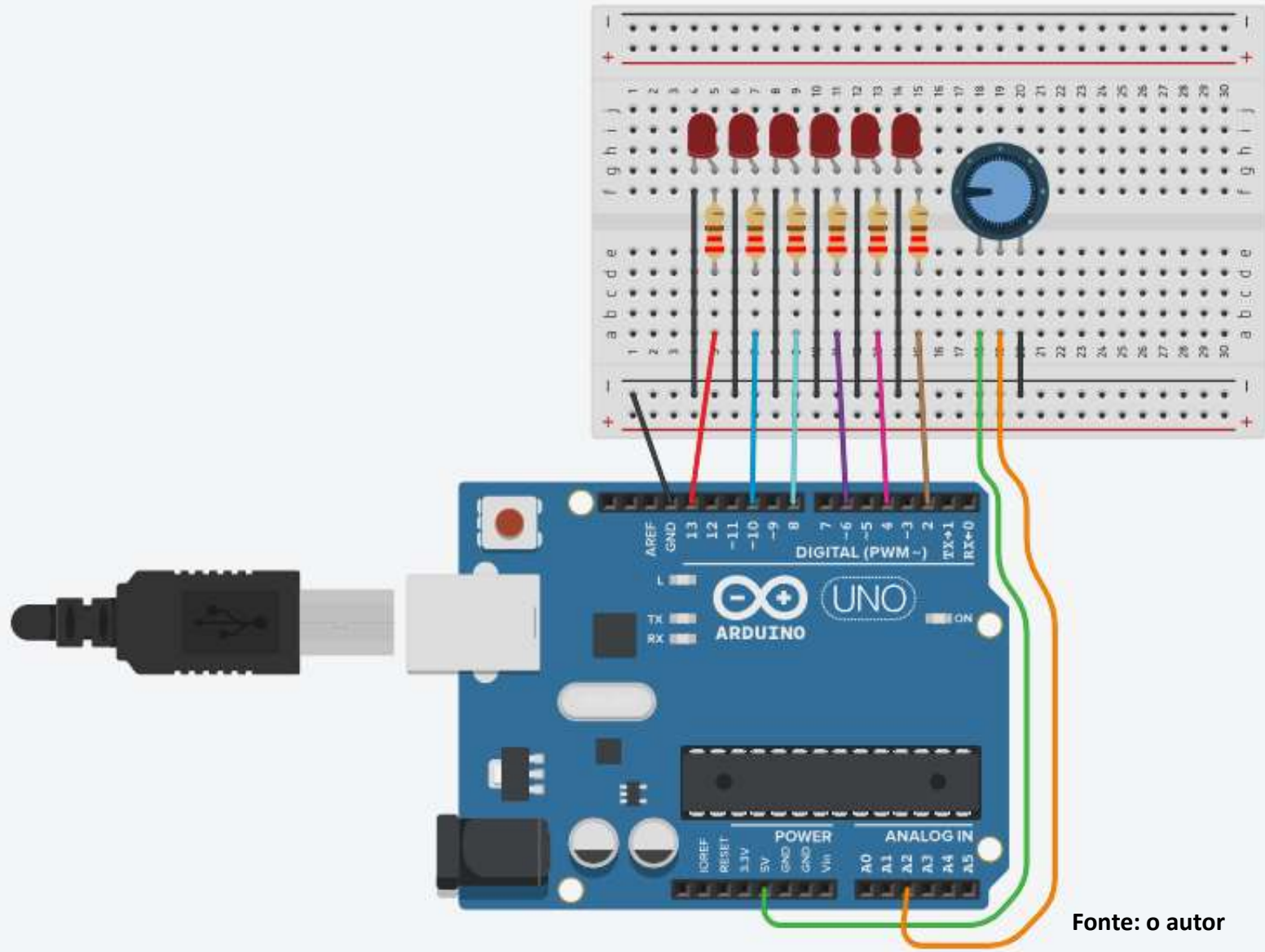
Prof. Me. Pietro Martins de Oliveira

Exemplo - Potenciômetro

- **Antes da eletrônica digital: sinais analógicos.**
- **Potenciômetro:**
 - Resistência variável;
 - Controla a corrente/tensão aplicada ao terminal central;
 - Regula um sinal de maneira analógica.
- **Exemplo:**
 - Controlar a velocidade de acionamento de um conjunto de LEDs.



Hardware Exemplo Potenciômetro



Fonte: o autor

```
byte ledPin[] = {2,4,6,8,10,13};  
int ledDelay;  
int potPin = 2;  
  
void setup(){  
    for(int x=0; x<6; x++){  
        pinMode(ledPin[x], OUTPUT);  
    }  
}
```

Software (continuação)

Exemplo Potenciômetro

```
void loop(){  
    ledDelay = analogRead(potPin);  
    for(int x=0; x<6; x++)  
    {  
        digitalWrite(ledPin[x], HIGH);  
        delay(ledDelay);  
        digitalWrite(ledPin[x], LOW);  
        delay(ledDelay);  
    }  
}
```

- ARDUINO HOMEPAGE. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 31 de agosto de 2018.
- MONK, Simon. **Programação com Arduino: começando com sketches**. Porto Alegre: Bookman, 2013.

EXEMPLOS: SINAIS ANALÓGICOS

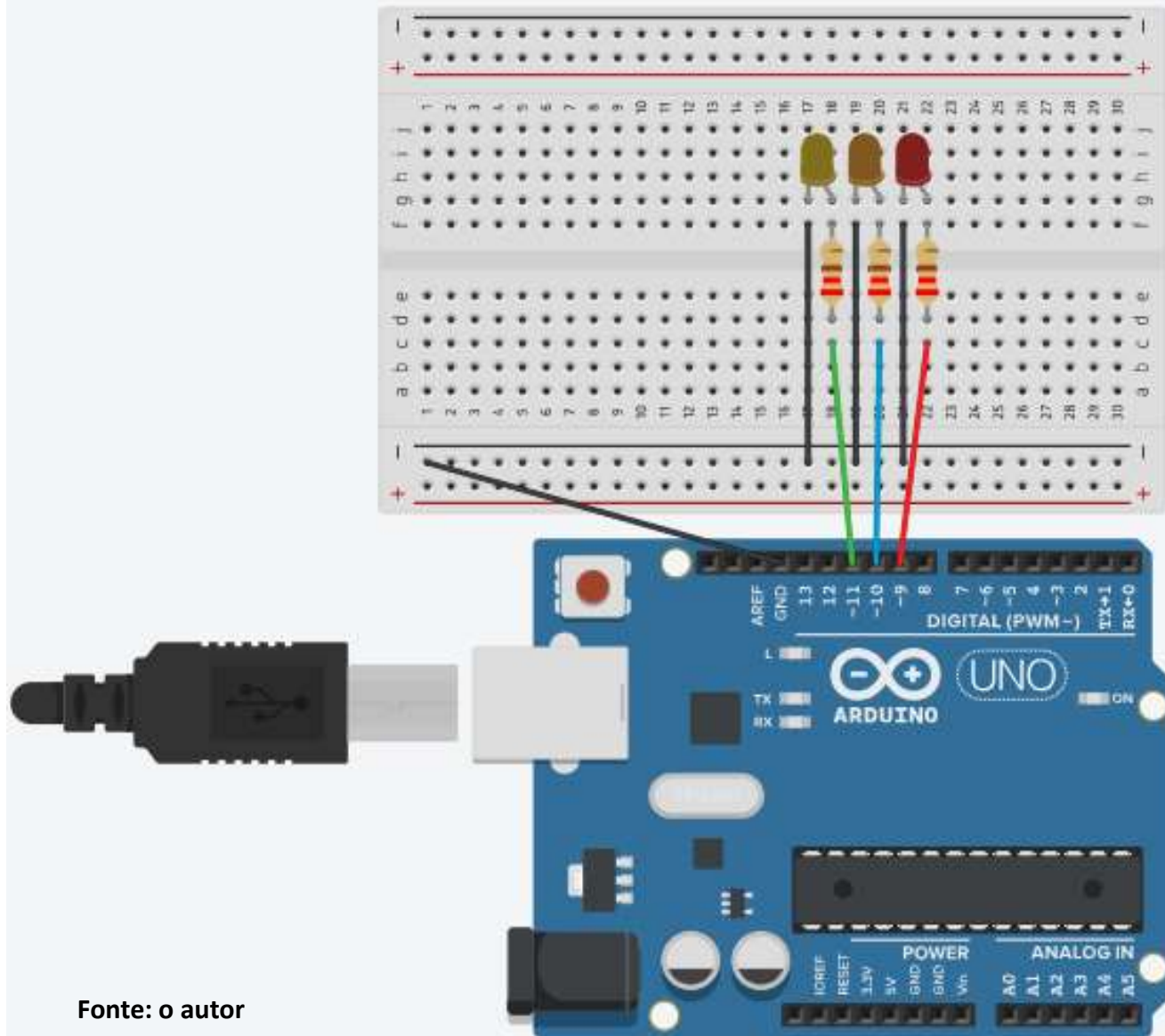
Prof. Me. Pietro Martins de Oliveira

EXEMPLOS: VALORES ALEATÓRIOS

Prof. Me. Pietro Martins de Oliveira

- **Valores aleatórios:**
 - Depende das circunstâncias incertas;
 - Difícil de prever;
 - Também conhecidos como valores randômicos.
- **Exemplo:**
 - Acionar três LEDs, cada um independente do outro;
 - Os LEDs terão suas intensidades alteradas aleatoriamente.

Hardware Exemplo Aleatório



Fonte: o autor

```
int ledPin1 = 9;  
int ledPin2 = 10;  
int ledPin3 = 11;  
  
void setup() {  
    pinMode(ledPin1, OUTPUT);  
    pinMode(ledPin2, OUTPUT);  
    pinMode(ledPin3, OUTPUT);  
}
```



```
void loop() {  
    analogWrite(ledPin1, random(256));  
    analogWrite(ledPin2, random(256));  
    analogWrite(ledPin3, random(256));  
    delay(random(100));  
}
```

- ARDUINO HOMEPAGE. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 31 de agosto de 2018.
- MONK, Simon. **Programação com Arduino: começando com sketches**. Porto Alegre: Bookman, 2013.

EXEMPLOS: VALORES ALEATÓRIOS

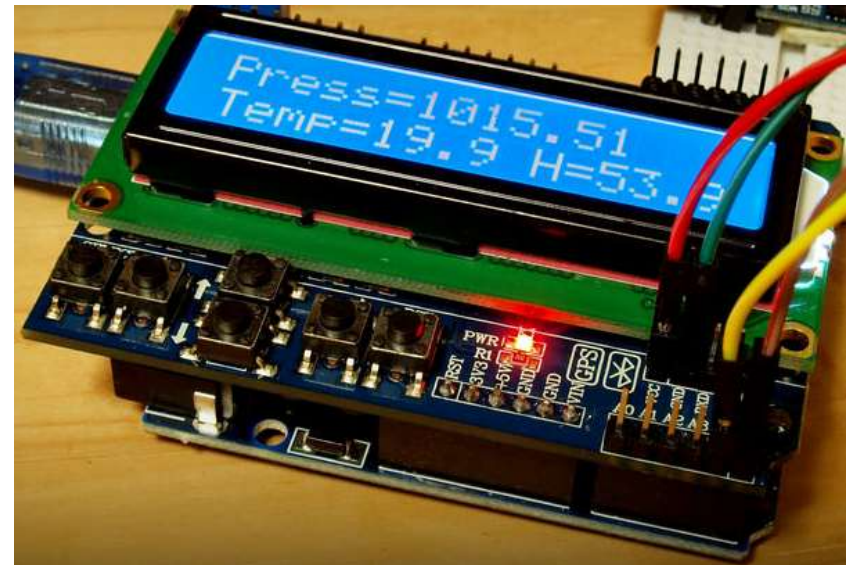
Prof. Me. Pietro Martins de Oliveira

EXEMPLOS: LIQUID CRYSTAL DISPLAY

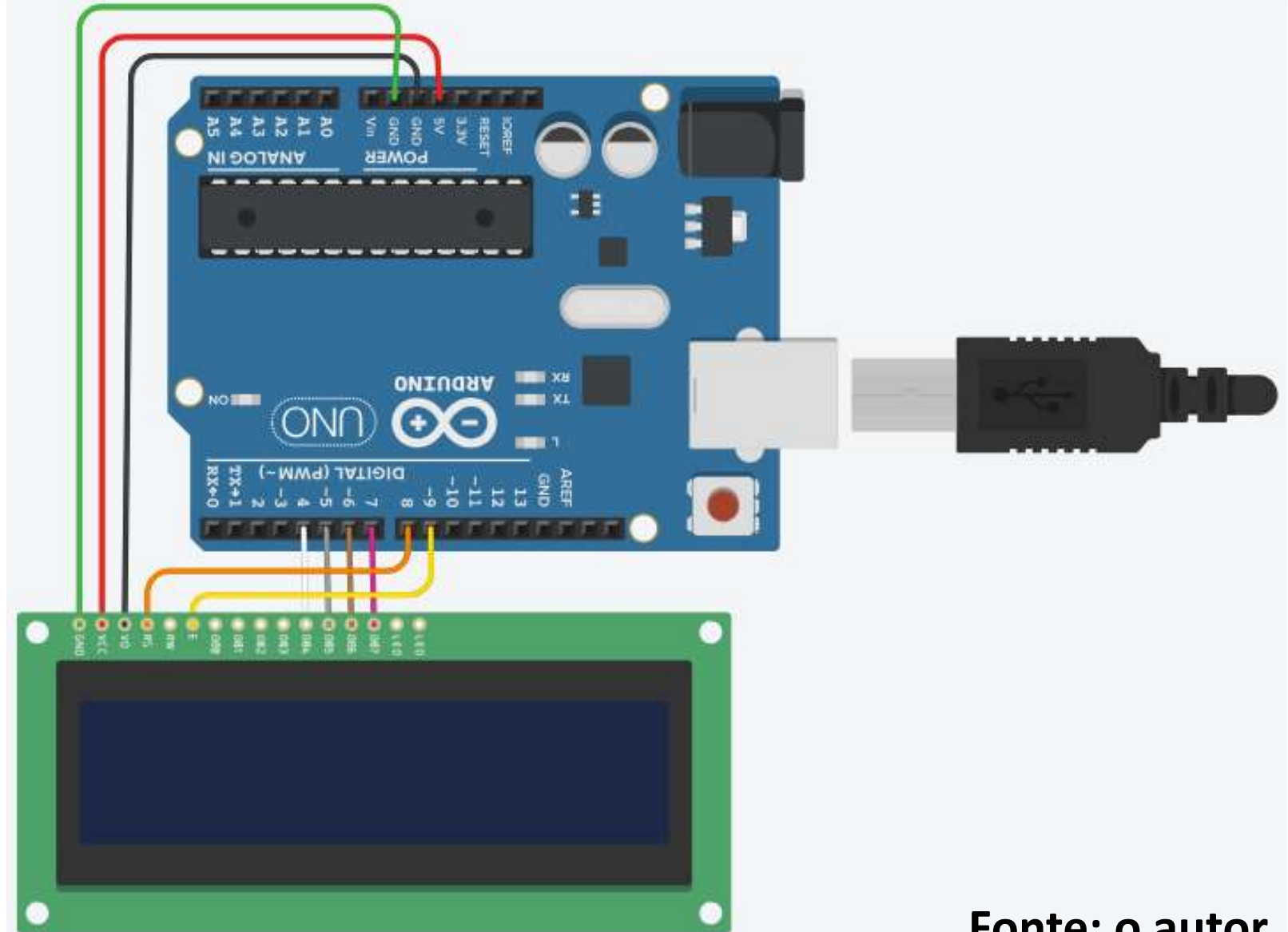
Prof. Me. Pietro Martins de Oliveira

- **Display LCD**

- Liquid Crystal Display;
- Existem diversos modelos/fabricantes p/ Arduino;
- Mais comum: 2 linhas × 16 colunas:
 - Dezesseis caracteres por linha.
- 4 bits ou 8 bits.
- Pinos básicos:
 - RS: *register select*;
 - E: *enable*;
 - D0 a D7: bits p/ representação do caractere.



Hardware Exemplo LCD



```
#include <LiquidCrystal.h> //Biblioteca LCD
//lcd(RS, E, D4, D5, D6, D7)
LiquidCrystal lcd(8,9,4,5,6,7);
int numRows = 2; //linhas
int numCols = 16; //colunas
int i = 0;
void setup(){
    lcd.begin(numRows, numCols); //Início
    lcd.clear(); //Limpa o LCD
}
```

```
void loop(){  
    lcd.setCursor(0,0); //Ajusta o cursor  
    lcd.print("Arduino"); //Imprime  
    delay(1000);  
    lcd.clear();  
    lcd.setCursor(0,1);  
    lcd.print("Rules");  
    delay(1000);  
    lcd.clear();  
}
```

- ARDUINO HOMEPAGE. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 31 de agosto de 2018.
- MONK, Simon. **Programação com Arduino: começando com sketches**. Porto Alegre: Bookman, 2013.

ARDUINO

Prof. Me. Pietro Martins de Oliveira