

Semana 1: Arduino, Movimento Maker and DIY

Prof. Irineu Lopes Palhares Junior

FCT/UNESP,
irineu.palhares@unesp.br



Informações sobre o curso de programação para Arduino.

- **Aulas:** terças e quintas-feiras das 10:00 as 12:00
- **Duração do curso:** de 08/03/2022 a 13/05/2022
- **Critério de aprovação:** frequência acima de 50%.
- **Avaliações:** não há. Porém, no último encontro os alunos apresentarão um mini-projeto.
- **Acesso às aulas gravadas e aos slides:**
<https://drive.google.com/drive/folders/1-Xm2DSn1Rda1u2EzQSyjc4LYAeQgyUo4?usp=sharing>

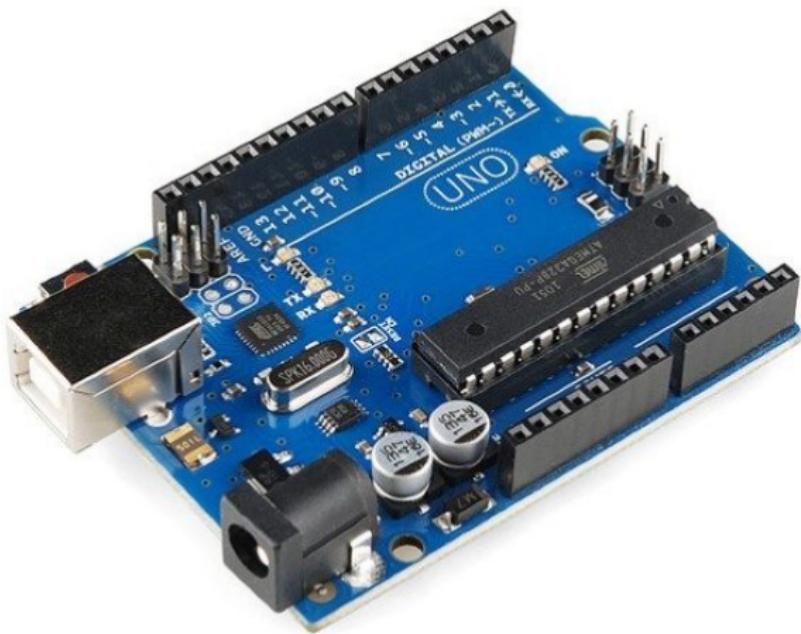
O que é Arduino?

- Arduino = **Hardware** (componentes elétricos) + **software** (IDE)



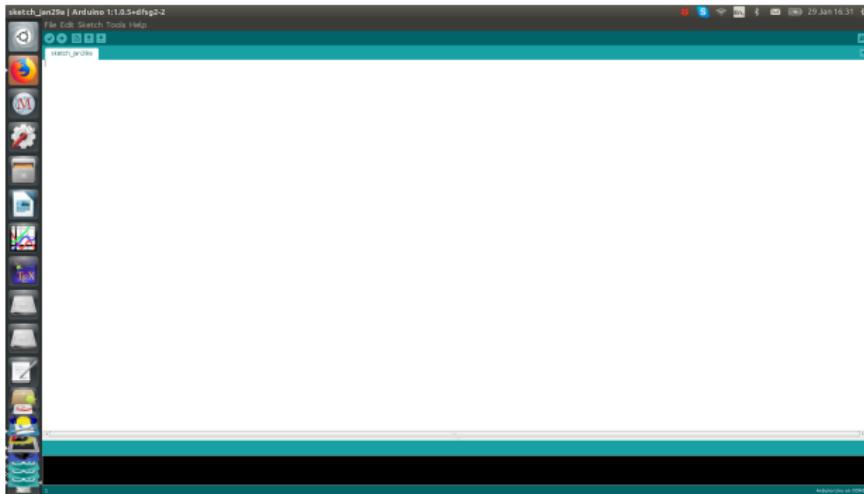
Figura 1: Logo do arduino.

Objeto chamado arduino! Placa de circuito elétrico com algumas componentes.



Sofware Arduino: ambiente de programação chamado IDE

Esse é o editor oficial do arduino, contendo as bibliotecas com as funções prontas para uso.



Mais detalhes sobre o Arduino

- Criado em 2005 por um grupo de 5 pesquisadores italianos.
- Objetivos do dispositivo: **barato**, **funcional** e de fácil programação



- Hardware livre/Open source
- Programação via *IDE* (*Integrated Development Environment*, ou *Ambiente de Desenvolvimento Integrado*) utilizando linguagem baseada em C/C++

Para que serve?

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.

Alguns projetos interessantes:

- Cubo de leds que formam imagens
- Tenis que amarram sozinhos



- Robo que desvia de obstáculos
- Abrir tranca por sequência sonora de batidas

Exemplo de uso do Arduino

Imagens Isncs



Tipos de arduino

O que muda de um arduino para outro é o microcontrolador e a quantidade de pinos disponíveis.



Figura 2: (a) Arduino UNO; (b) Arduino MEGA; (c) Arduino MINI.

Componentes do arduino

- Microcontrolador Atmel (“Cerebro do arduino”): controle de entrada e saída, armazenamento e execução do código



- Porta USB: energização e comunicação.
- Porta de alimentação: energização via fonte ou bateria (Tensão: 7 V a 12 V).
- Regulador de voltagem: controla a voltagem de entrada.
- *reset*
- Pinos de entradas e saídas digitais/analógicas.
- LEDs TX e RX: piscam rapidamente, o que indica que a comunicação e a transmissão do código foram estabelecidas.

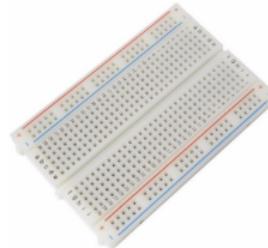
Componentes extras (parte 1)

- Sensores (modulos)



Figura 3: (a) Sensor de distância ultrassônico HC-SR04; (b) Sensor de luminosidade LDR; (c) Buzzer

- *Protoboard*



Componentes extras (parte 2)

- *Shields*: placa adicional para aumentar a funcionalidade do arduino



Instalação do Arduino IDE no computador:

- **Windows:** Entrar no site arduino.cc e fazer o download do “instalador” e na sequência executá-lo.



- **Linux:** Linha de comando no terminal: `sudo apt-get install arduino`





O que é
**MOVIMENTO
MAKER**
e como criar a
CULTURA MAKER?



O movimento maker representa a cultura conhecida como “faça você mesmo” (DIY – do it yourself em inglês), ou ainda “mão na massa”



A proposta é promover o estímulo de pessoas comuns para que explorem sua criatividade e possam desenvolver soluções. Dale Dougherty é creditado com o criador do movimento com o lançamento do Make Magazine em 2005. A cultura maker se nutri da iniciativa de construir, consertar, modificar ou fabricar suas próprias coisas. E os adeptos deste movimento são conhecidos como makers ou fazedores. Estes já inventaram soluções que celebram melhorias em muitos campos – ciência, tecnologia, arte, engenharia, design, música, entre outros.



- **O que é o Arduino?**

- <https://www.youtube.com/watch?v=sv9dDtYnE1g>
- <https://www.filipeflop.com/blog/o-que-e-arduino>

- **Download e instalação do Arduino**

- www.arduino.cc

- **Projetos com o Arduino**

- Cubo de leds que formam imagens:

<https://www.youtube.com/watch?v=6mXM-oGggrM>

- Tenis que amarram sozinhos:

<https://www.youtube.com/watch?v=L2RkeikLrPY>

- Robo que desvia de obstaculos:

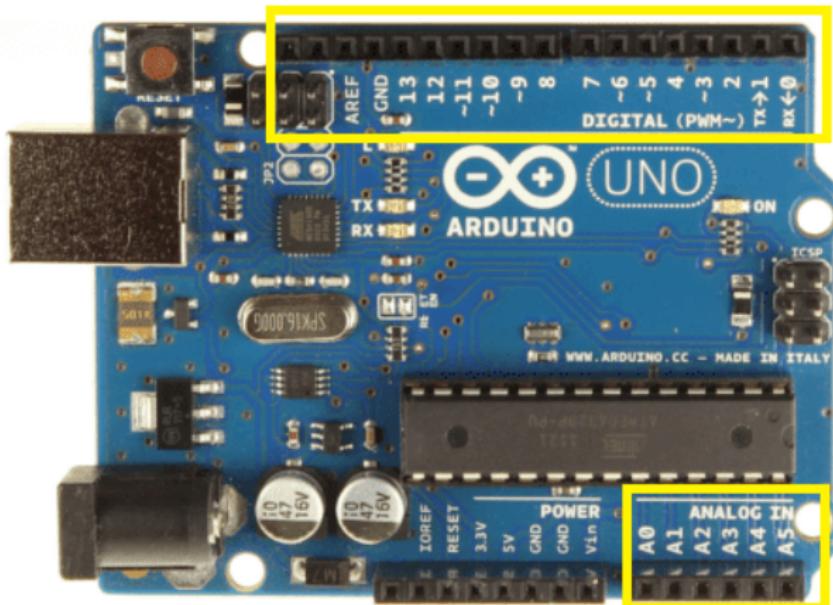
<https://www.youtube.com/watch?v=utSVBXwpsg8>

- Abrir tranca por sequência sonora de batidas:

<https://www.youtube.com/watch?v=PnmQ8pK8I6U>

1.2 Entradas e saídas do Arduino

Entradas / Saídas Digitais



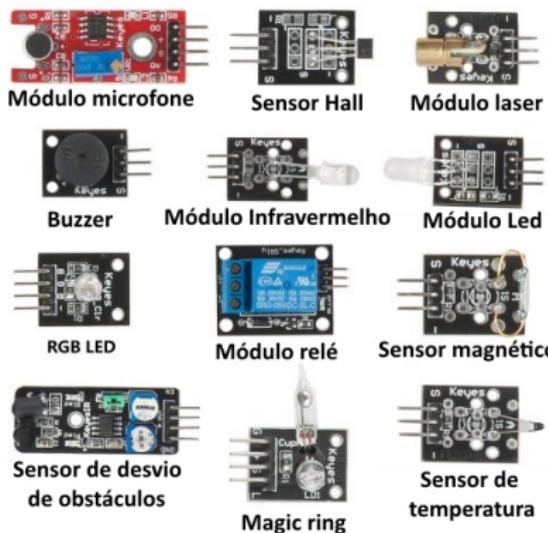
**Entradas
Analógicas**

Detalhes sobre os pinos do Arduino

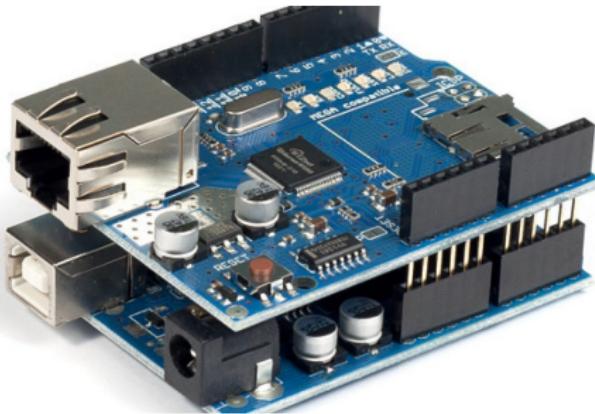
- **IOREF** – Fornece uma tensão de referência para que shields possam selecionar o tipo de interface apropriada, dessa forma shields que funcionam com a placas Arduino que são alimentadas com 3,3V. podem se adaptar para ser utilizados em 5V. e vice-versa.
- **RESET** – pino conectado a pino de RESET do microcontrolador. Pode ser utilizado para um reset externo da placa Arduino.
- **3,3 V** – Fornece tensão de 3,3V. para alimentação de shield e módulos externos. Corrente máxima de 50 mA.
- **5 V** – Fornece tensão de 5 V para alimentação de shields e circuitos externos.
- **GND** – pinos de referência, terra.
- **VIN** – pino para alimentar a placa através de shield ou bateria externa. Quando a placa é alimentada através do conector Jack, a tensão da fonte estará nesse pino.

1.3 Módulos

Os módulos são peças que podem aumentar o poder do Arduino, dando ao seu projeto maior robustez. Estas pequenas placas contém sensores, resistores, capacitores ou led's e são utilizadas para determinadas funções quando utilizada junto ao Arduino.

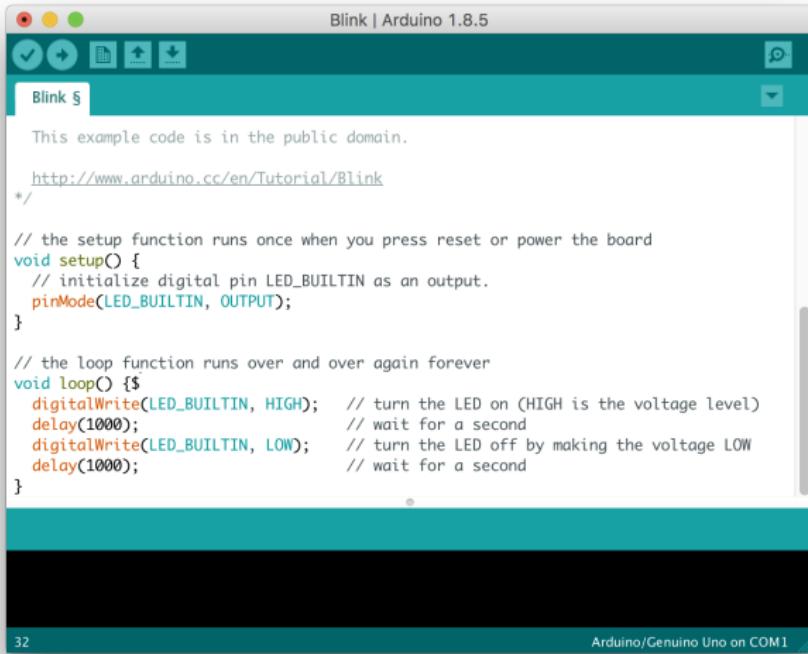


Os *shields* são placas que se encaixam ao Arduino para acrescentar funcionalidades. Um exemplo de *Shield* que é bastante utilizado em projetos é o *Shield Ethernet*, cuja função é permitir que a pessoa tenha acesso à rede (Ethernet ou Internet) através da conexão de um cabo entre o *Shield* e o roteador.



1.4 IDE

Integrated Development Environment (IDE): é a plataforma do Arduino onde são escritos os códigos.



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.8.5". The main window displays the "Blink" example code. The code is a standard "Blink" sketch that alternates the state of the built-in LED. It includes comments explaining the setup and loop functions. The Arduino board dropdown at the bottom right shows "Arduino/Genuino Uno on COM1".

```
This example code is in the public domain.  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
    // initialize digital pin LED_BUILTIN as an output.  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {$  
    digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)  
    delay(1000);                      // wait for a second  
    digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW  
    delay(1000);                      // wait for a second  
}
```

Website: tinkercad.com

- Acessar a Curso: Programação para Arduino
- Código da turma:

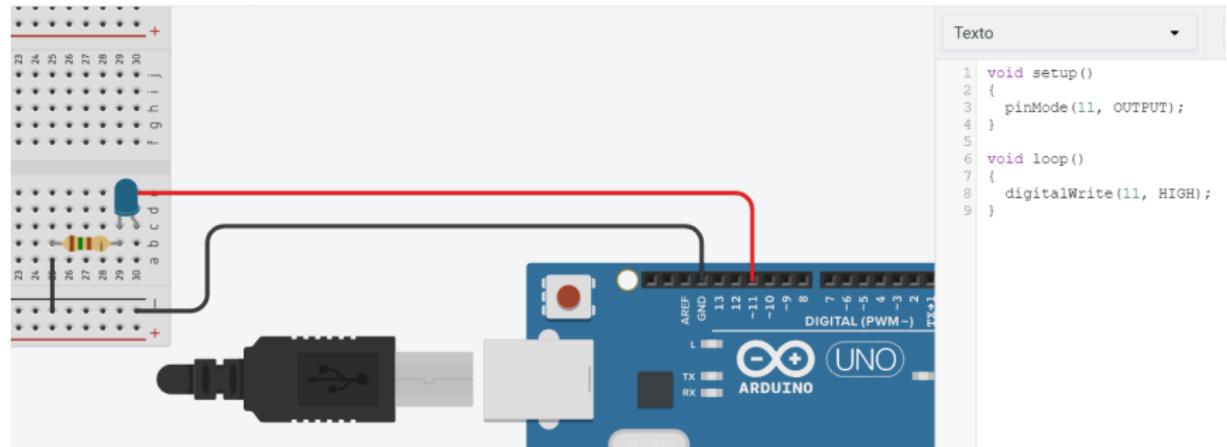


Principais funções do Arduino IDE:

- **setup()**: esta função é responsável por definir os pinos de entrada e saída, bem como inicializar todo o Arduino.
- **loop()**: função de repetição, onde os comandos serão construídos.

Exemplo de uso das funções setup e loop

Acendendo uma LED

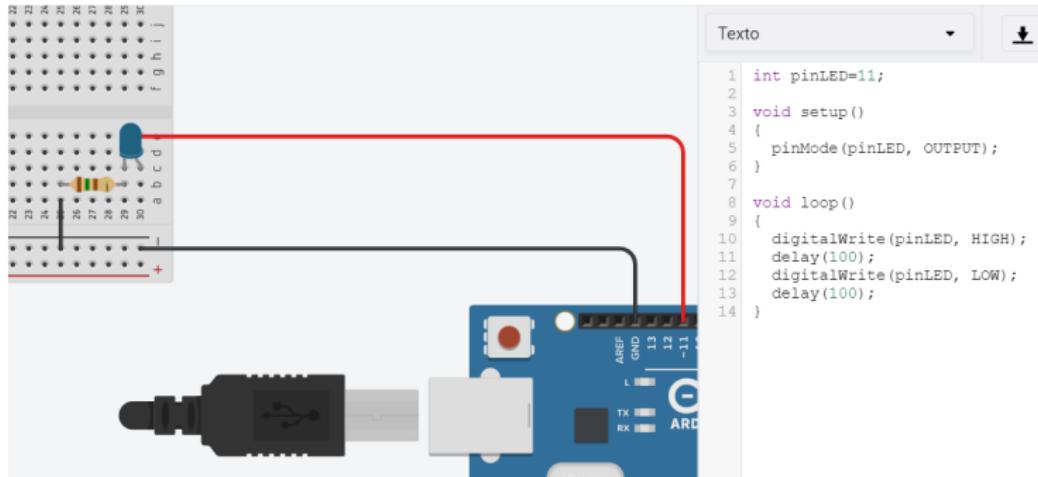


Conceitos básicos

- **Operadores aritméticos:** +, -, *, /
- **Operadores lógicos:** &&(e), ||(ou), == (Comparação de igualdade), = (atribuir valor)
- **Operadores condicionais:** if, else, while, for
- **Comandos terminados em ";"**
- **Sobre as funções:** inicia com "{" e termina com "}"

1.7 Programa "Hello World"

Construção do circuito e código para piscar uma LED.



Atividade (para o lar) - Construção de um semáforo

Com base no que vimos nesta aula, você deverá simular um semáforo utilizando algumas LEDs e o Arduino.

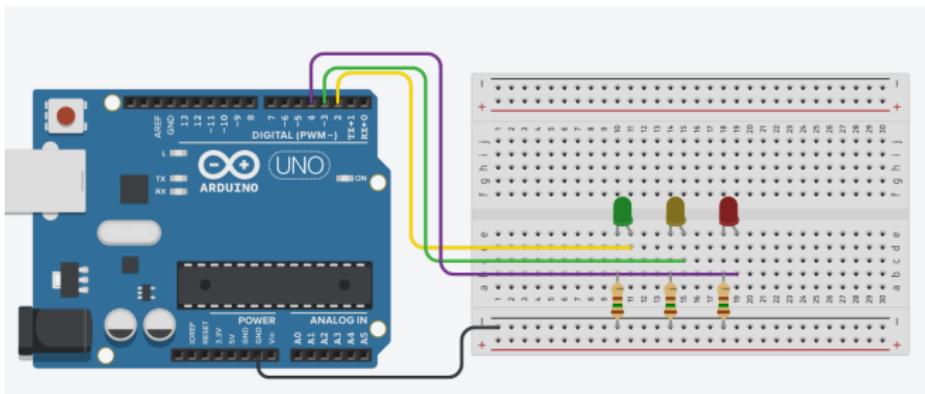


- Para iniciar o serial monitor precisamos do seguinte comando (dentro da função void setup()): **Serial.begin(9600);**
- Imprimir mensagem no serial monitor:
Serial.print("mensagem");
- Imprimir número inteiro no serial monitor:
Serial.println(numero);
- Verificar se algo foi digitado no serial monitor:
Serial.available() retorna valor 1
- Lendo um número a partir do serial monitor:
Serial.parseInt();
- Lendo uma String: **Serial.readString();**
- Lendo o byte mais recente do buffer: **Serial.read();**

Introdução aos Operadores Condicionais

- **for:** O comando *for* é usado para repetir um bloco de código envolvido por chaves. Um contador de incremento é geralmente utilizado para terminar o loop. O comando *for* é útil para qualquer operação repetitiva, e é usado frequentemente com vetores para operar em coleções de dados ou pinos.
- **if - else**
- **while**
- **switch**

Círcuito do Semáforo



Código das LEDs

```
const int pinG=2;
const int pinY=3;
const int pinR=4;
int aux=0;

String palavra;

void setup()
{
    pinMode(pinG, OUTPUT);
    pinMode(pinY, OUTPUT);
    pinMode(pinR, OUTPUT);

    Serial.begin(9600); //Iniciar o serial monitor
}

void loop()
{
    //Verificar se algo foi digitado no serial monitor
    if (Serial.available()>0)
    {
        palavra = Serial.readString();
    }

    //verificar o que é para ser feito
    if (palavra == "verde")
    {
        digitalWrite(pinG, HIGH);
        digitalWrite(pinY, LOW);
        digitalWrite(pinR, LOW);
    }
    else if (palavra == "amarelo")
    {
        digitalWrite(pinG, LOW);
        digitalWrite(pinY, HIGH);
        digitalWrite(pinR, LOW);
    }
    else if (palavra == "vermelho")
    {
        digitalWrite(pinG, LOW);
        digitalWrite(pinY, LOW);
        digitalWrite(pinR, HIGH);
    }
}
```