



FACULDADE DE TECNOLOGIA E CIÊNCIAS DA BAHIA

Microcontroladores e Microprocessadores

• Professor Marcos Vinicius •

Softwares utilizados

- **Arduino**



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 

Linux 32 bits

Linux 64 bits

Linux ARM 32 bits

Linux ARM 64 bits

Mac OS X 10.10 or newer

[Release Notes](#)

[Checksums \(sha512\)](#)

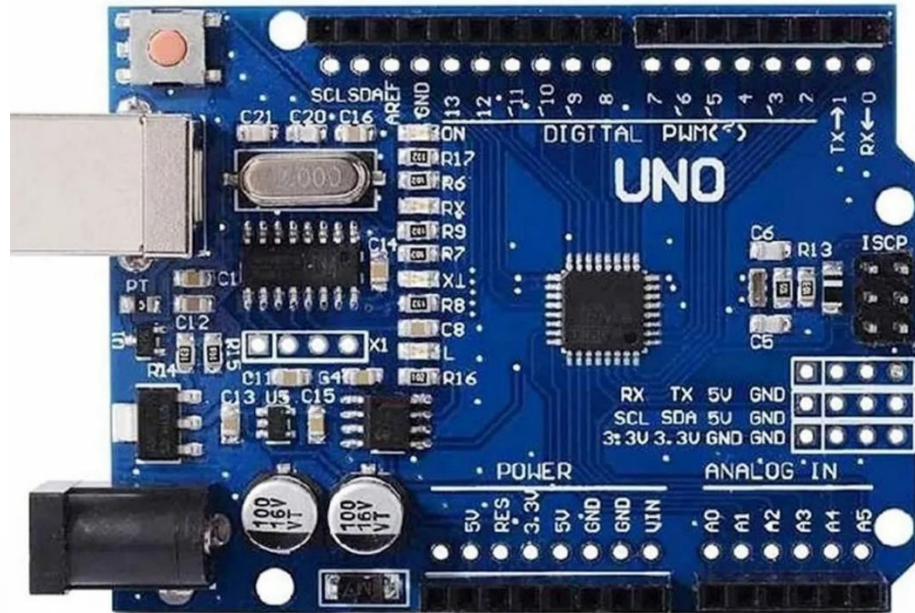
Softwares utilizados

- IDE para C

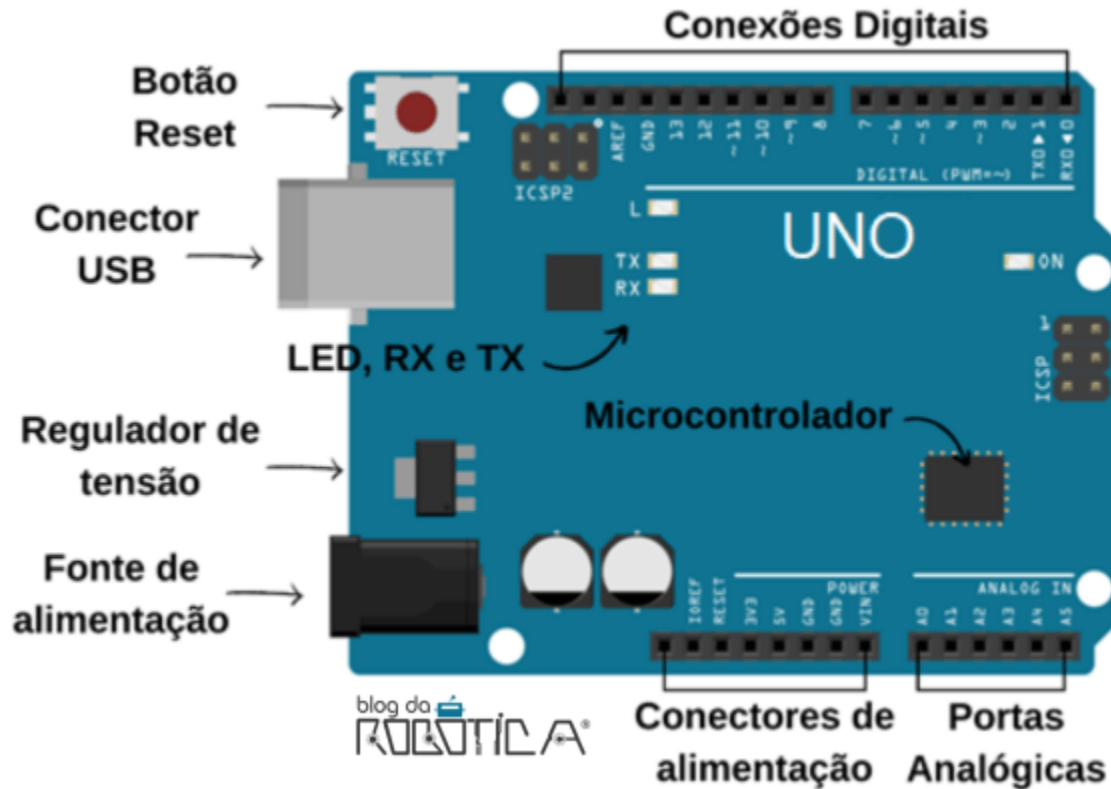


Hardware utilizado

Arduino UNO



Componentes de um Arduino UNO



Microcontrolador

- O coração de nosso Arduino é um **microcontrolador**.
- Praticamente os demais componentes se destinam ao fornecimento de energia e comunicação com o seu computador.

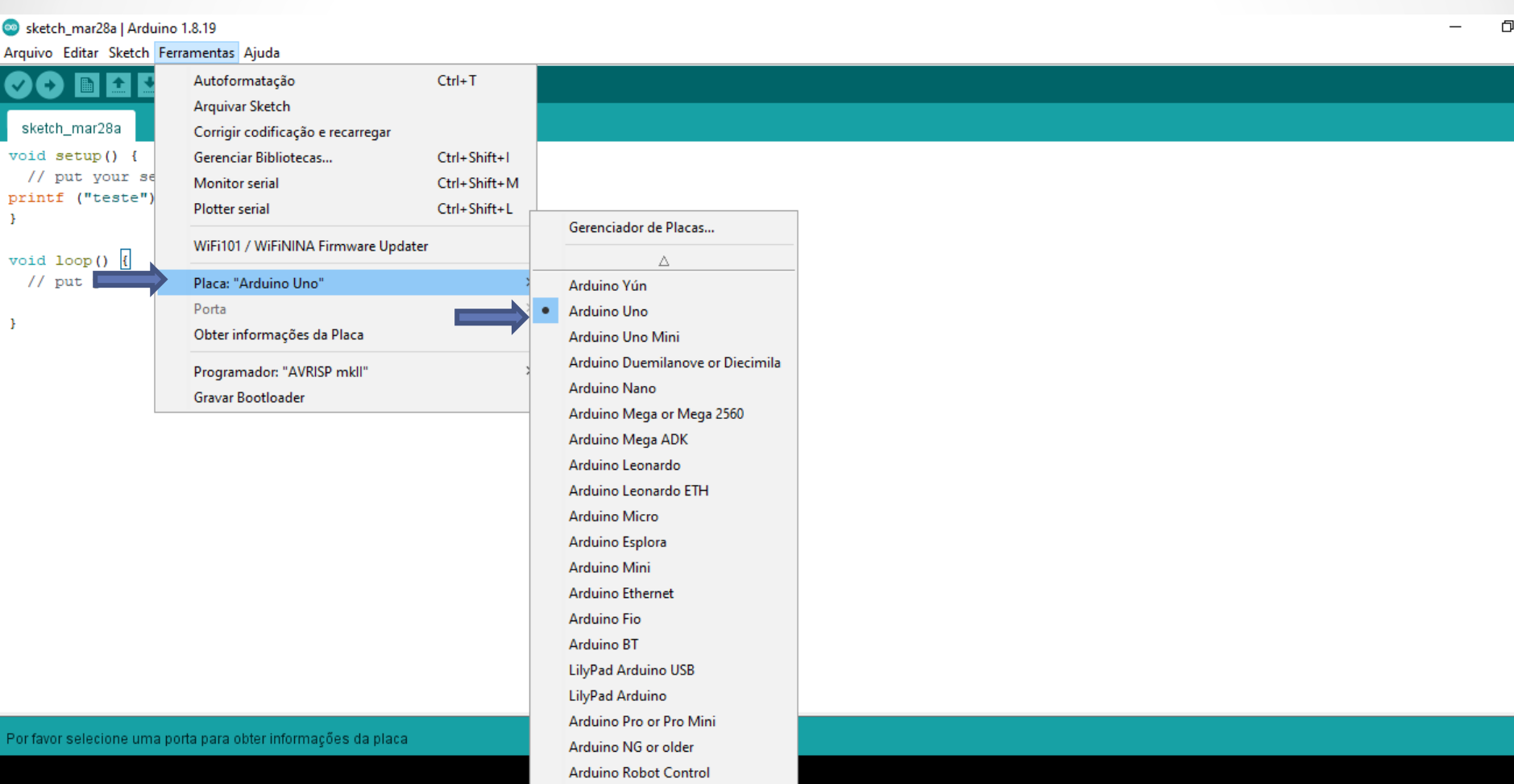
Hardware

- Pequeno computador em um chip.
 - Processador.;
 - 2 ou 2,5 Kb de memória RAM
 - 1 Kbyte de memória EPROM
 - Quilobytes de memória Flash.

Com essa configuração o Arduino não espera ser um Hardware de ponta, porém objetiva controlar e executar tarefas simples.

Conexão com o Software

- Definir a placa



The image shows the Arduino IDE interface. The 'Ferramentas' menu is open, and the 'Placa: "Arduino Uno"' option is selected. The 'Gerenciador de Placas...' dialog box is also open, showing a list of Arduino boards. The 'Arduino Uno' option is selected in the list. A blue arrow points from the 'Placa: "Arduino Uno"' option in the menu to the 'Arduino Uno' option in the dialog box. Another blue arrow points from the 'Obter informações da Placa' option in the menu to the 'Gerenciador de Placas...' dialog box.

```
sketch_mar28a | Arduino 1.8.19
Arquivo Editar Sketch Ferramentas Ajuda
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"
Porta
Obter informações da Placa
Programador: "AVRISP mkII"
Gravar Bootloader
```

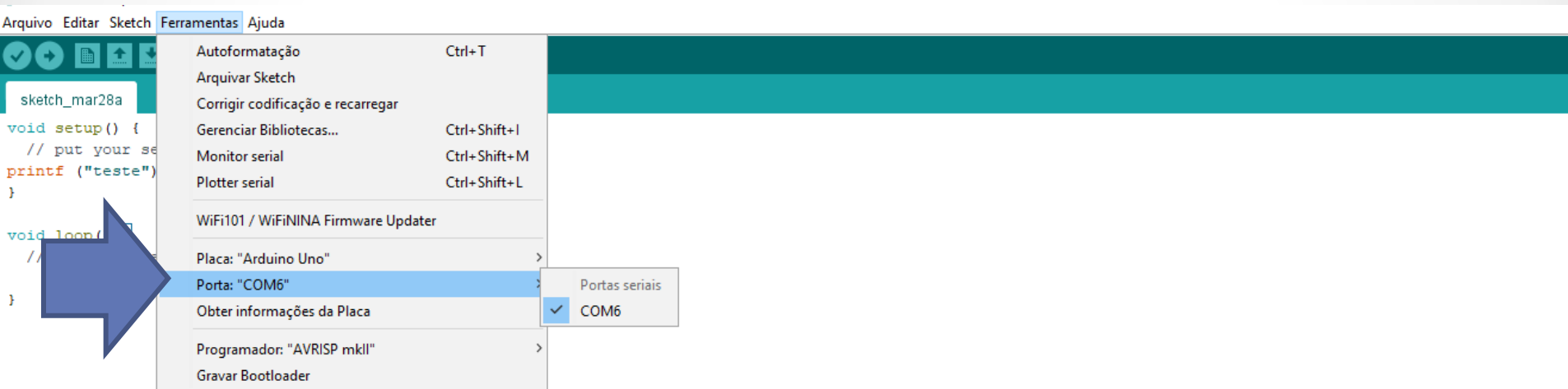
Gerenciador de Placas...

- Arduino Yún
- Arduino Uno
- Arduino Uno Mini
- Arduino Duemilanove or Diecimila
- Arduino Nano
- Arduino Mega or Mega 2560
- Arduino Mega ADK
- Arduino Leonardo
- Arduino Leonardo ETH
- Arduino Micro
- Arduino Esplora
- Arduino Mini
- Arduino Ethernet
- Arduino Fio
- Arduino BT
- LilyPad Arduino USB
- LilyPad Arduino
- Arduino Pro or Pro Mini
- Arduino NG or older
- Arduino Robot Control

Por favor selecione uma porta para obter informações da placa

Conexão com o Software

- Após selecionar a placa, escolha a porta de comunicação.



Sketchs

- Como já vimos os **sketchs** são os programas (códigos) que irão interagir com a placa do Arduino.
- Muitas linguagens podem ser utilizadas para programar microcontroladores, com o nosso conhecimento da **linguagem de programação C**, podemos praticar a programação.

Sketch 01: Pisca Led

sketch_may16a

```
//declaração que o led está conectado no pino 13
int led = 13;
//Função que é chamada apenas uma vez quando o arduino é ligado ou reiniciado
void setup() {

//A função configura o pino como entrada ou saída
pinMode (led,OUTPUT);
}

void loop() {
    // Atribui nível máximo ao pino, Acende o led
    digitalWrite (led, HIGH);

//define tempo de espera 1000 ms ou 1 seg
delay(1000);
    // Atribui nível mínimo ao pino, Apaga o led
    digitalWrite (led, LOW);
//define tempo de espera 1000 ms ou 1 seg
delay(1000);

}
```

Sketch 01: Pisca Led

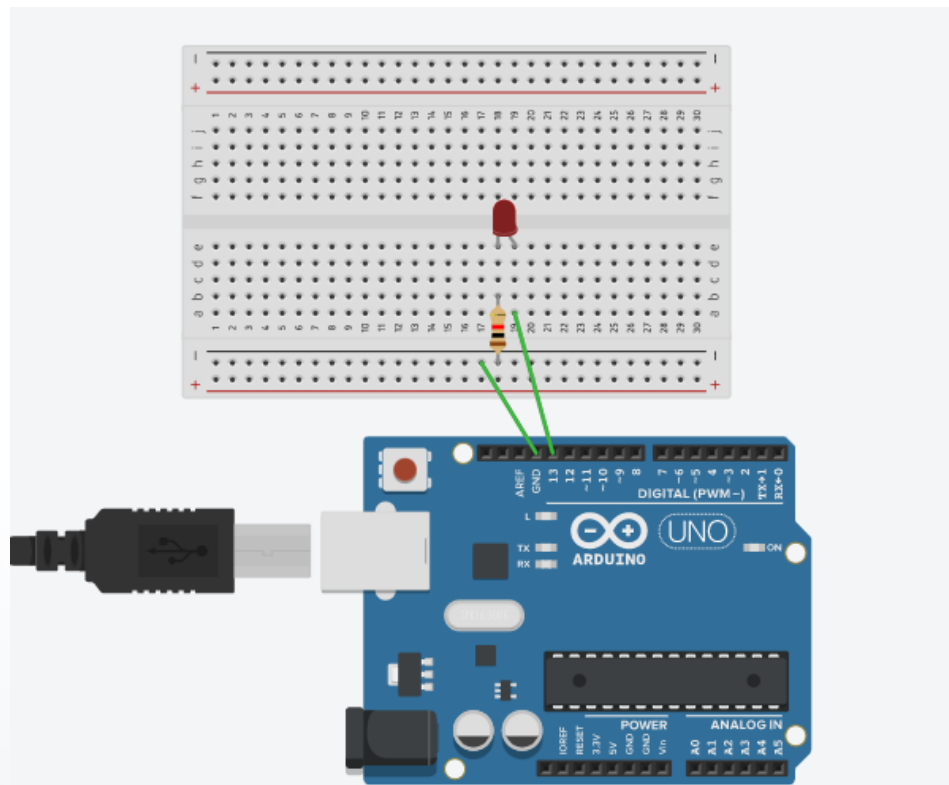
- No primeiro Sketch observamos um programa que tem como objetivo piscar o LED de 1 em 1 segundo (1000 ms);
- Para simplificar e indicar qual porta iríamos utilizar declaramos a **variável int led = 13**. Desta maneira, ao chamar a porta no código, podemos simplificar chamando a variável.
- Da mesma maneira podemos definir uma variável para definição de tempo. **Ex: int tempo = 1000;**

Conexão no Arduino

- Na imagem abaixo podemos observar o simulador online para Arduino, o [TinkerCad](#).
- A criação da conta é simples, bastando informar o e-mail ou conta Gmail.
- Com o **Tinkercad**, caso você não tenha a placa física poderá testar os seus Sketchs através do simulador.

Conexão no TinkerCad

- Utilizamos para esse projeto: 1 led, 1 resistor, 2 filamentos para conexão, protoboard e a placa de Arduino Uno.



Sketch 2

- Código Morse era um método vital de comunicação nos séculos XIX e XX.
- A codificação de letras na forma de uma sequência de pontos e traços significava que o código Morse poderia ser enviado através de fios telegráficos, enlaces de rádio e sinalização luminosa.
- As letras SOS (Salve nossas almas), neste projeto
- iremos emitir esta sequência repetidamente.

Sketch 2

- Sinalizador de SOS em código Morse

projeto_2

```
//declaração que o led está conectado no pino 13
int led = 13;
//Função que é chamada apenas uma vez quando o arduino é ligado ou reiniciado
void setup() {

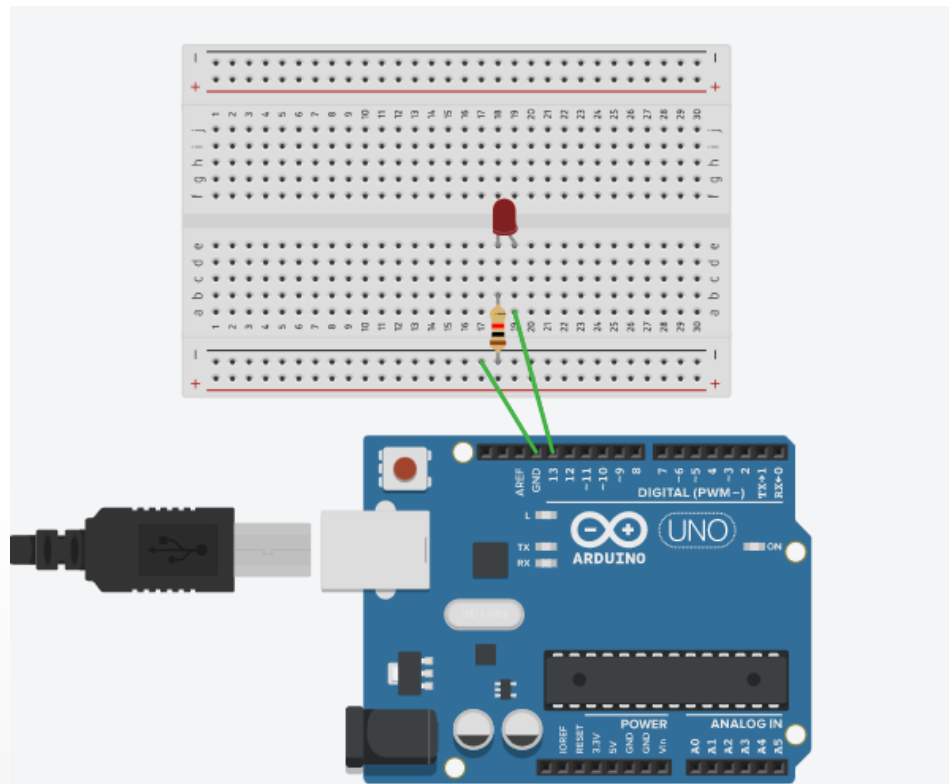
//A função configura o pino como entrada ou saída
pinMode (led,OUTPUT);
}

void loop() {
  flash(200);flash(200);flash(200); // S
  delay(300);
  flash(500);flash(500);flash(500); //O
  flash(200);flash(200);flash(200); //S
  delay(1000); // espera 1 seg
}

void flash(int duration){
  digitalWrite (led,HIGH);
  delay(duration);
  digitalWrite (led,LOW);
  delay(duration);
}
```

Conexão no TinkerCad

- Usamos a mesma configuração do projeto anterior: 1 led, 1 resistor, 2 filamentos para conexão, protoboard e a placa de Arduino Uno.



Estruturas para Sketch 2

- Para atualizar o sketch 2 e facilitar a execução iremos adicionar **estruturas de repetição e arrays** no nosso código;
- **Relembrando estruturas de repetição:**

```
for(int i = 0; i<100;i++){
```

```
flash(200);
```

```
}
```

ou

```
while (i <100)
```

```
flash(200);
```

```
i=i+1;
```

```
}
```

Arrays (Vetor)

- Os **arrays (vetores)** são uma forma de criar uma lista de valores;
- As variáveis como vimos até agora armazenavam um único valor, já o **array** é **uma estrutura** que pode armazenar um **conjunto de valores**;

P 0	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9
500	220	225	280	300	400	600	800	100	50

Arrays

- O array tem um conjunto de valores e que pode ser **acessado através do seu índice(posição)**;
- Exemplo: Se quiser exibir a 2ª posição do array, devemos chamar o nome dele e entre colchetes a sua posição: `durations [1]`;
- Na maioria das linguagens o **1º item começa com a posição 0** ao invés de 1, por isso a segunda posição no array é o `[1]`;

Atualizando o Sketch 2

projeto_2.1

```
//declaração que o led está conectado no pino 13
int led = 13;
int durations[] = {200,200,200,500,500,500,200,200,200};

//Função que é chamada apenas uma vez quando o arduino é ligado ou reiniciado
void setup() {

//A função configura o pino como entrada ou saída
pinMode (led,OUTPUT);
}

void loop() {
  // estrutura de repetição que irá repetir 10 vezes, exibindo os itens do array
  for (int i = 0; i<9; i++){
    flash (durations[i]);
  }
  delay(1000); // espera 1 segundo antes de iniciar novamente.
}

void flash(int duration){
  digitalWrite (led,HIGH);
  delay(duration);
  digitalWrite (led,LOW);
  delay(duration);
}
}
```

Exercício

- 1- Usando ainda o mesmo esquema dos dois primeiros projetos, crie um código para piscar de acordo a letra digitada. Exemplo A = 500 ms, B 800 ms e C = 1000ms;
- 2- Criar um código com 3 leds simulando um semáforo, onde o vermelho deve durar 15 segundos, amarelo 5 segundos e o verde deve durar 10 segundos.

Referências

- Monk, Simon. Editora Bookman, 30 projetos com Arduino.
- Monk, Simon. Editora Bookman, 2ª edição , Programação com com Arduino – começando com Sketches.

Links para Acesso e Download

- **Link Download Arduino:**

<https://www.arduino.cc/en/software>

- **Link Download Dev C:**

<https://sourceforge.net/projects/orwelldevcpp/>

- **Link TinkerCad:**

<https://www.tinkercad.com/dashboard?type=circuits&collection=designs>

- **Loja para compra de Kit Arduino e peças:**

<https://www.institutodigital.com.br/>

Contato:

Marcos Vinicius

marcosvcsantos@gmail.com