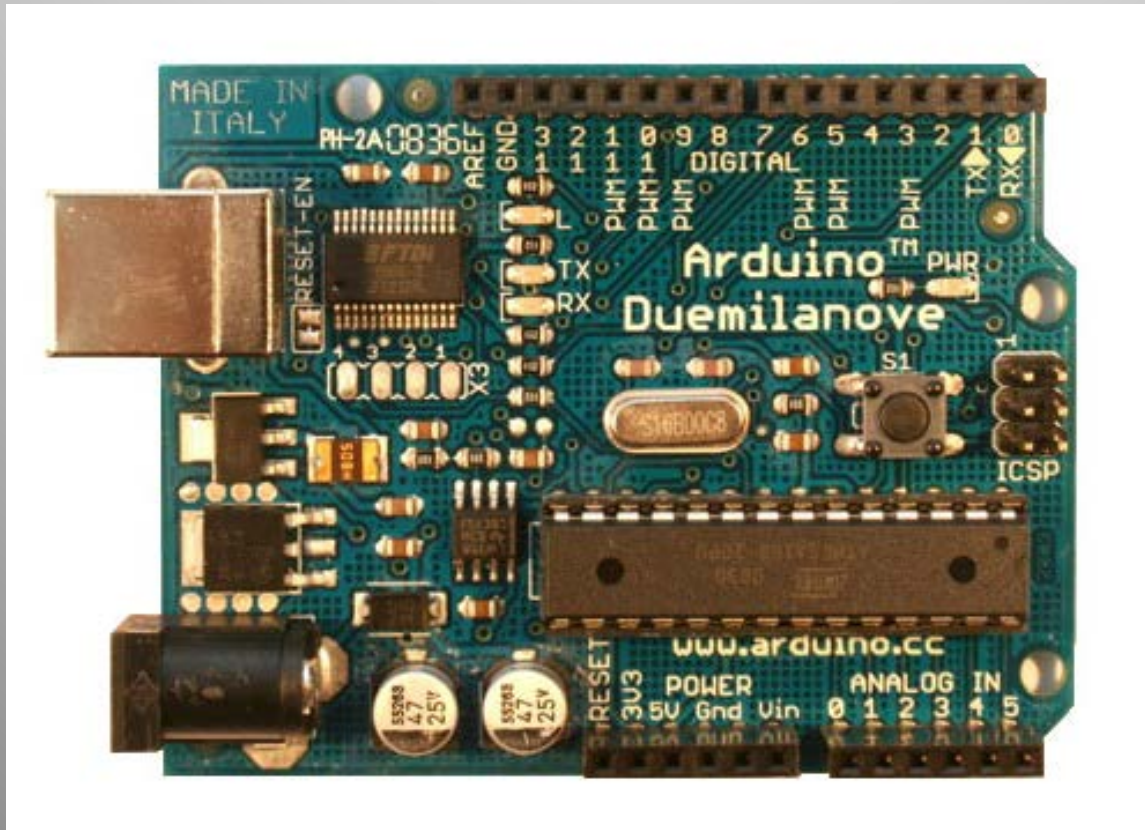


# **ARDUINO**

**Um tutorial inicial**

# Arduino

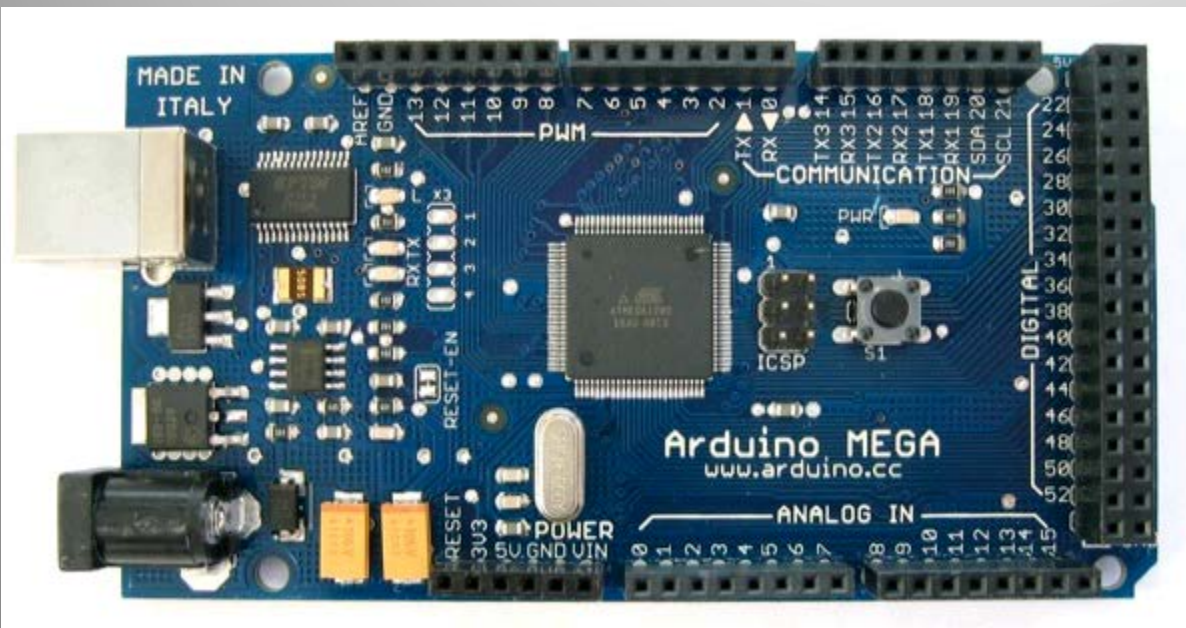
- ▶ Placa desenvolvida na Itália em 2005 (open source);
- ▶ Facilitar o desenvolvimento através de Shields;



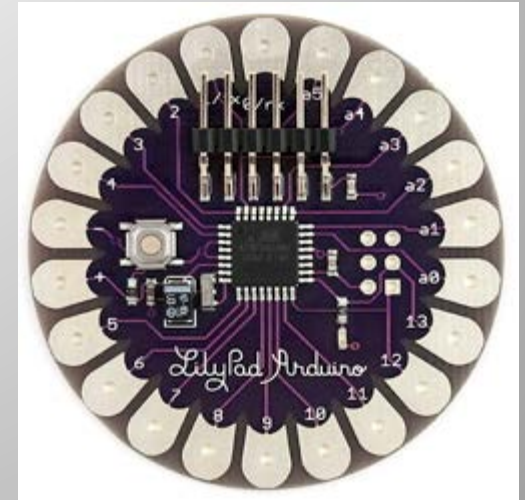
# Mas é só essa plaquinha ?

Não!

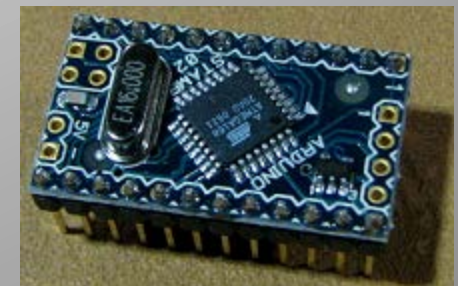
Temos varios modelos para aplicações diversas.



- Arduino Mega- 126k (Flash Memory)
- 54 Pinos de I/O



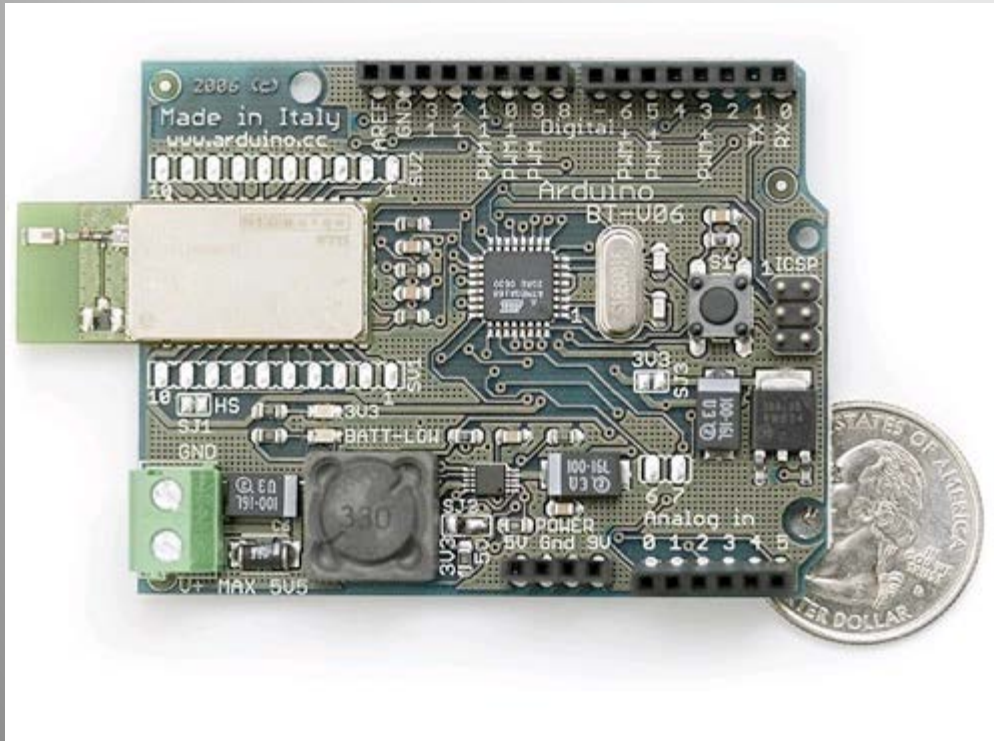
➤ LilyPad Arduino



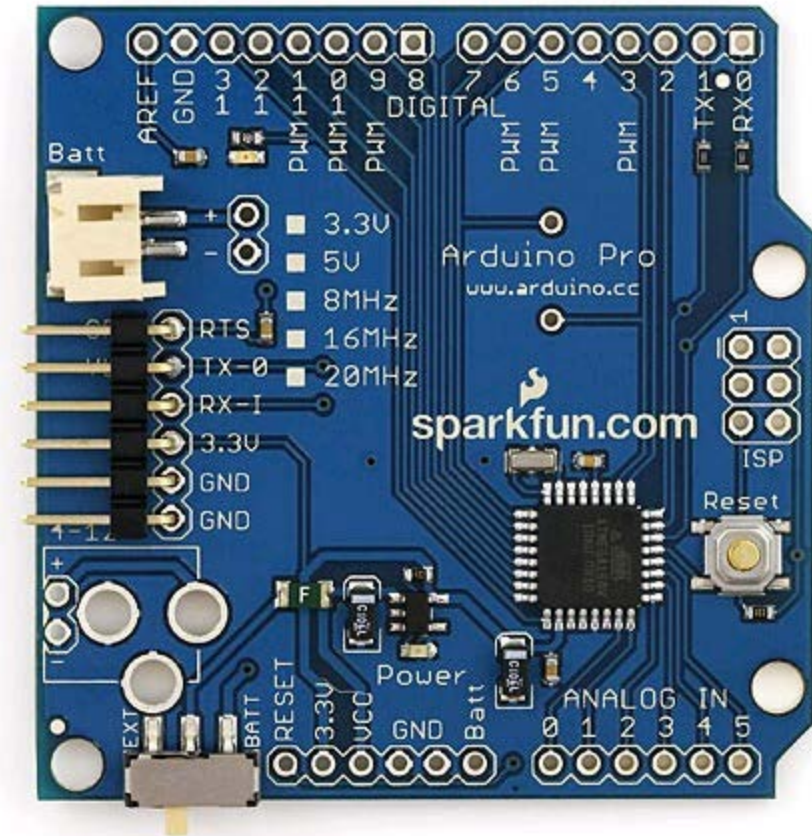
➤ Arduino Mini



# Mais ...



➤ Arduino BT (Bluetooth)

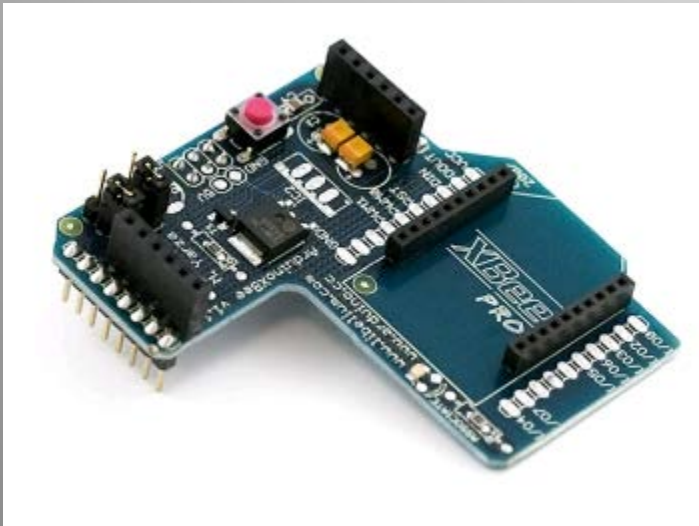


➤ Arduino Pro

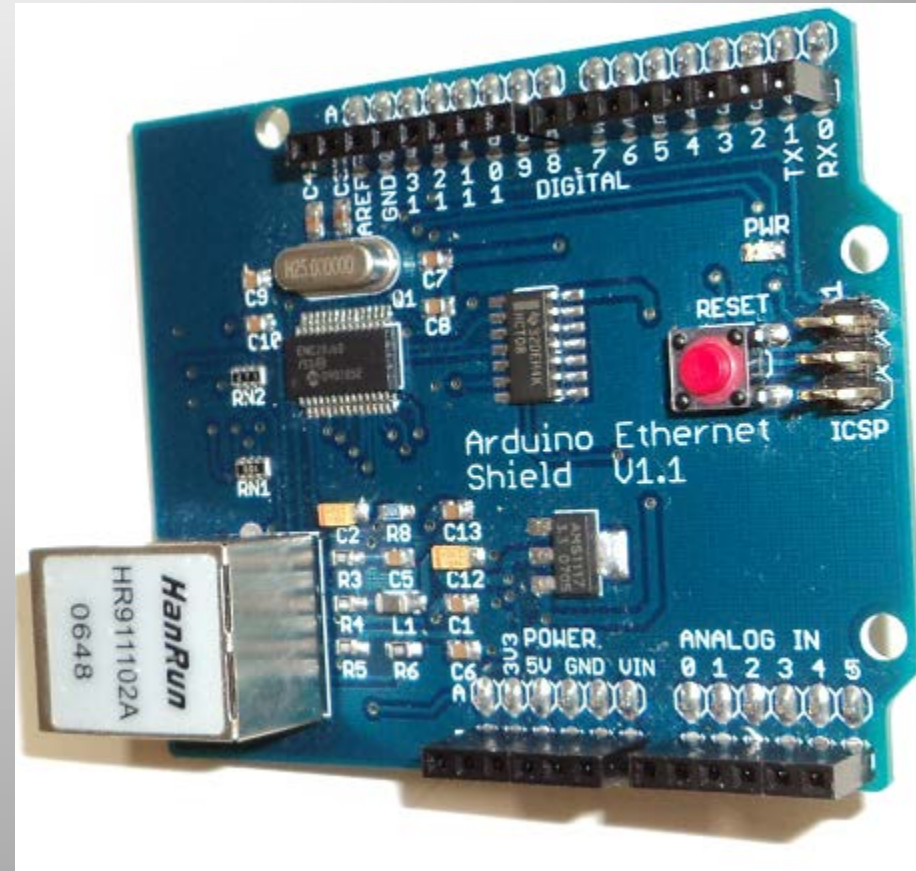
# E ainda tem os SHIELDS ...

## O que são Shields?

“Escudos”. Placas adicionais com conexões ao arduino e que permitem interagir com tecnologias diversas e com facilidade.



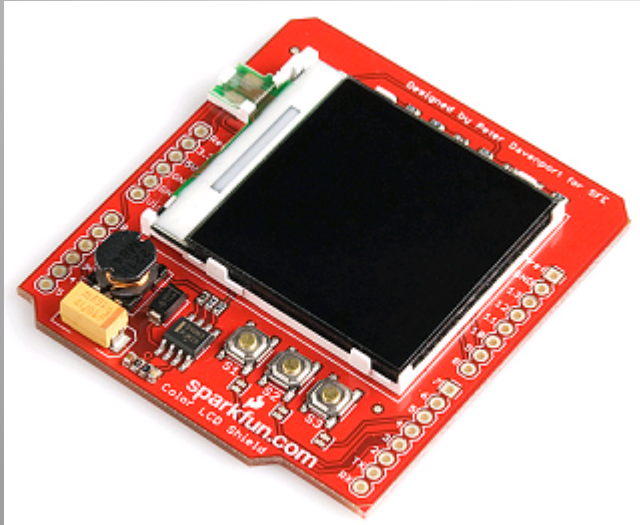
➤ Xbee shield



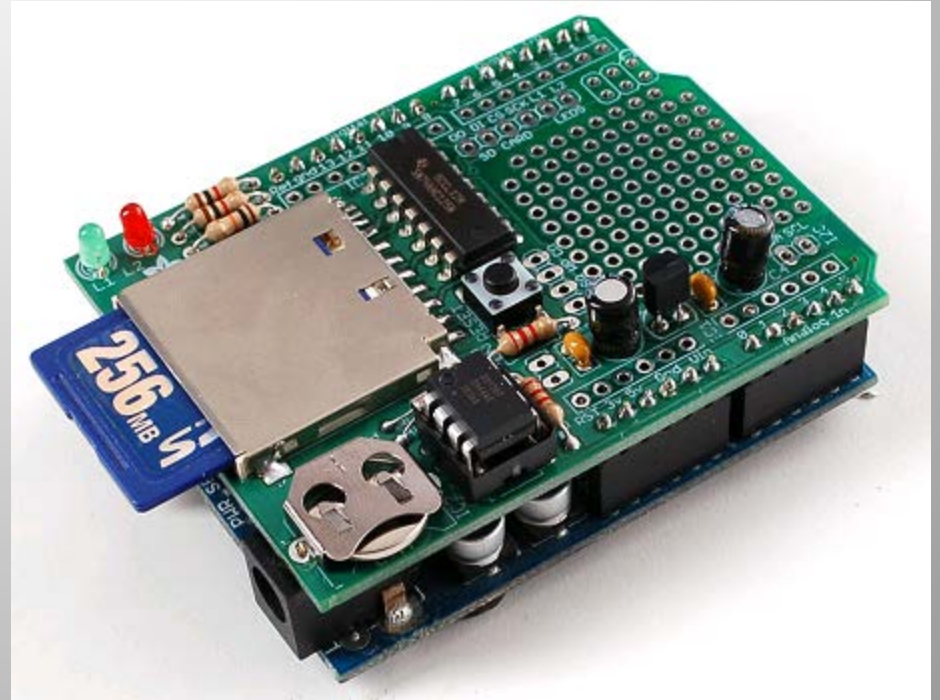
➤ Ethernet shield



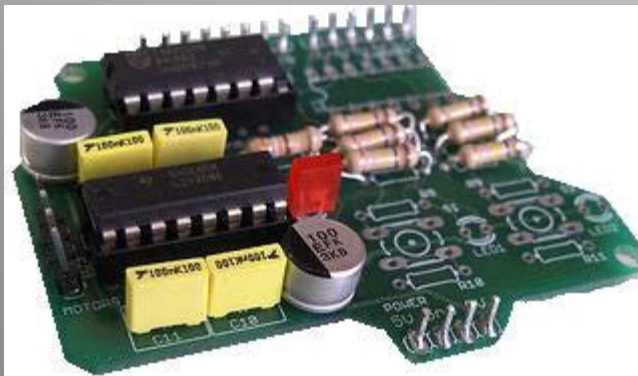
# Mais ...



➤ LCD shield



➤ SD shield

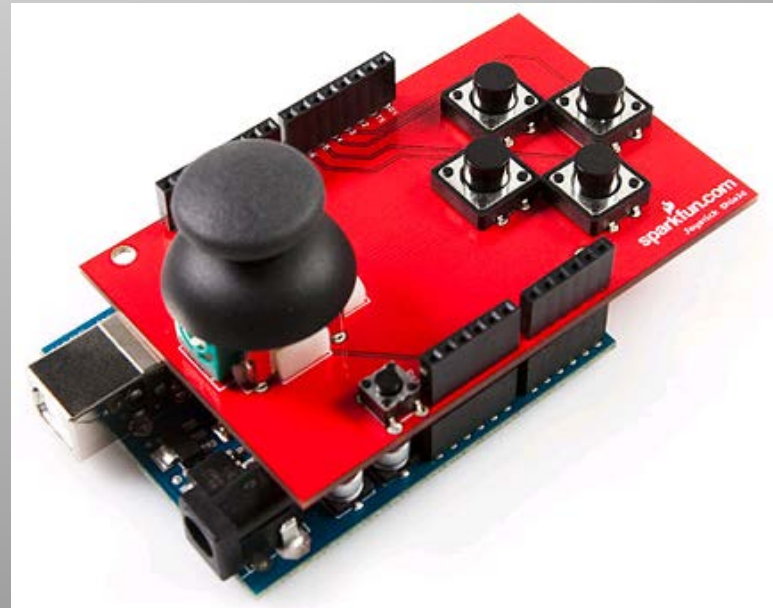


➤ Motor shield

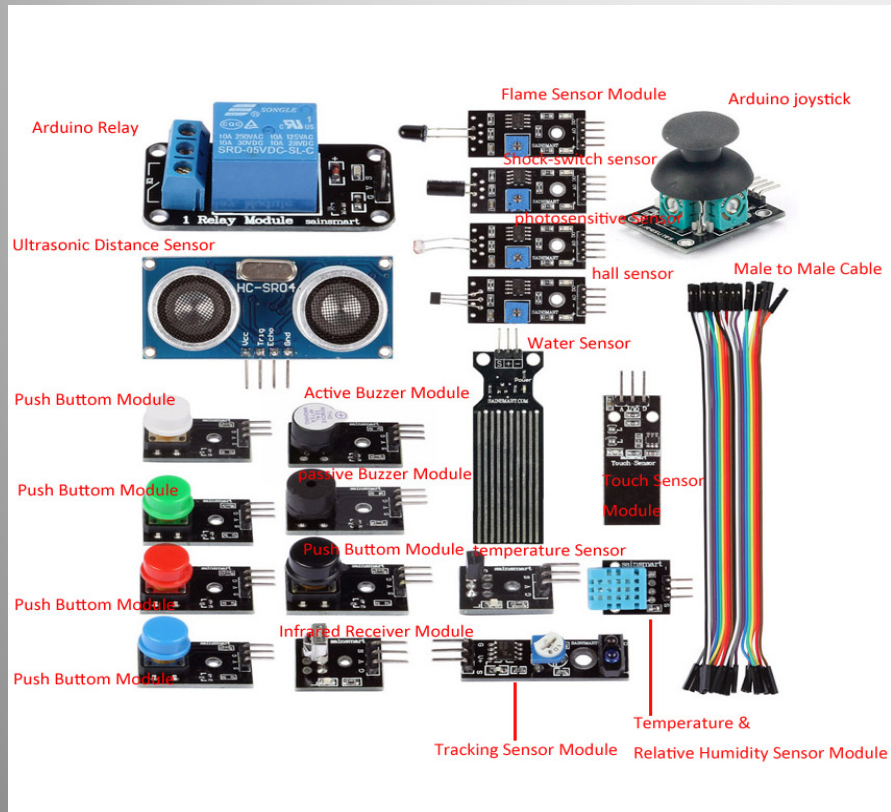
# Como liga ?

## Como usar os Shields?

Conecta-se sobre o Arduino .

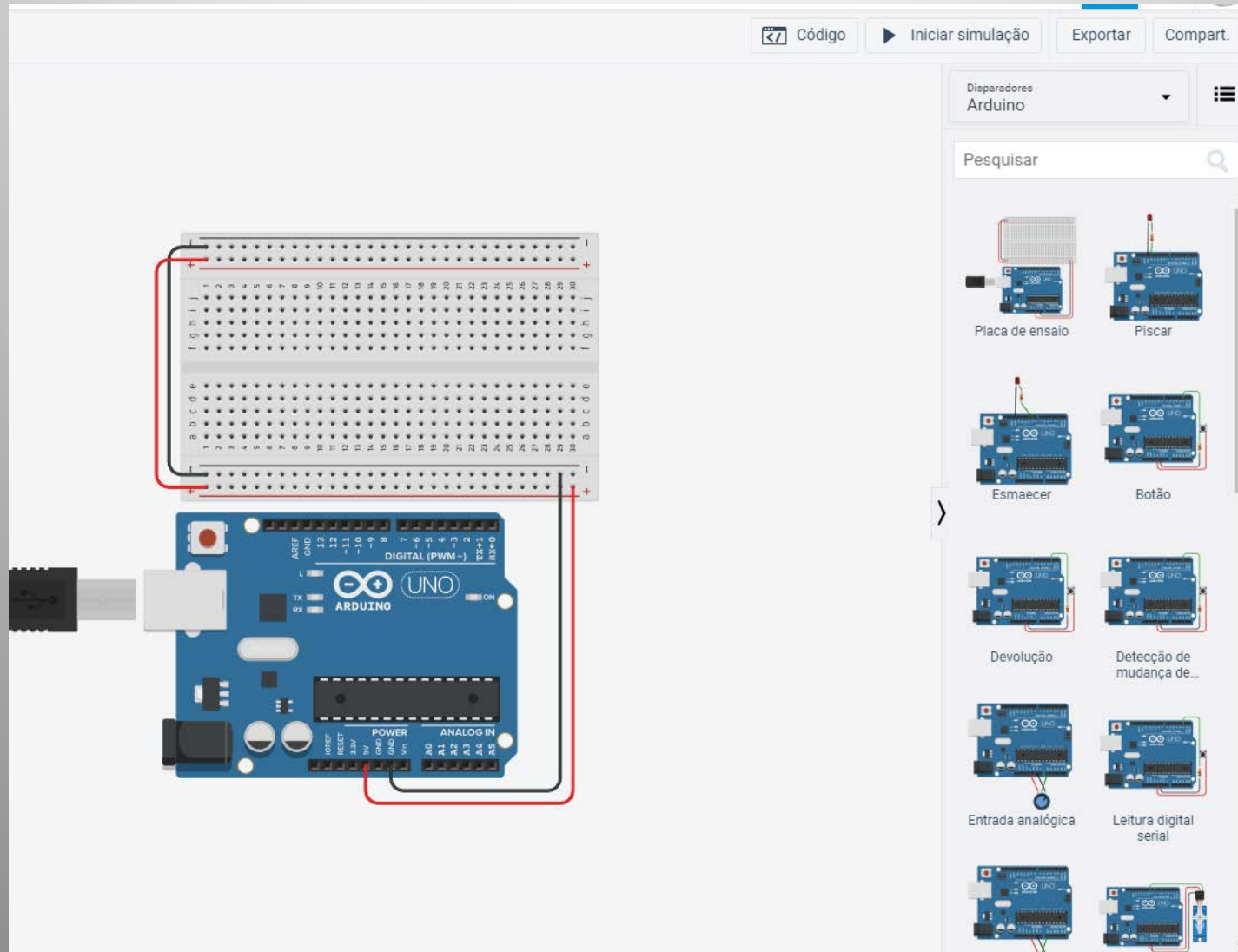


# E ainda os sensores e atuadores

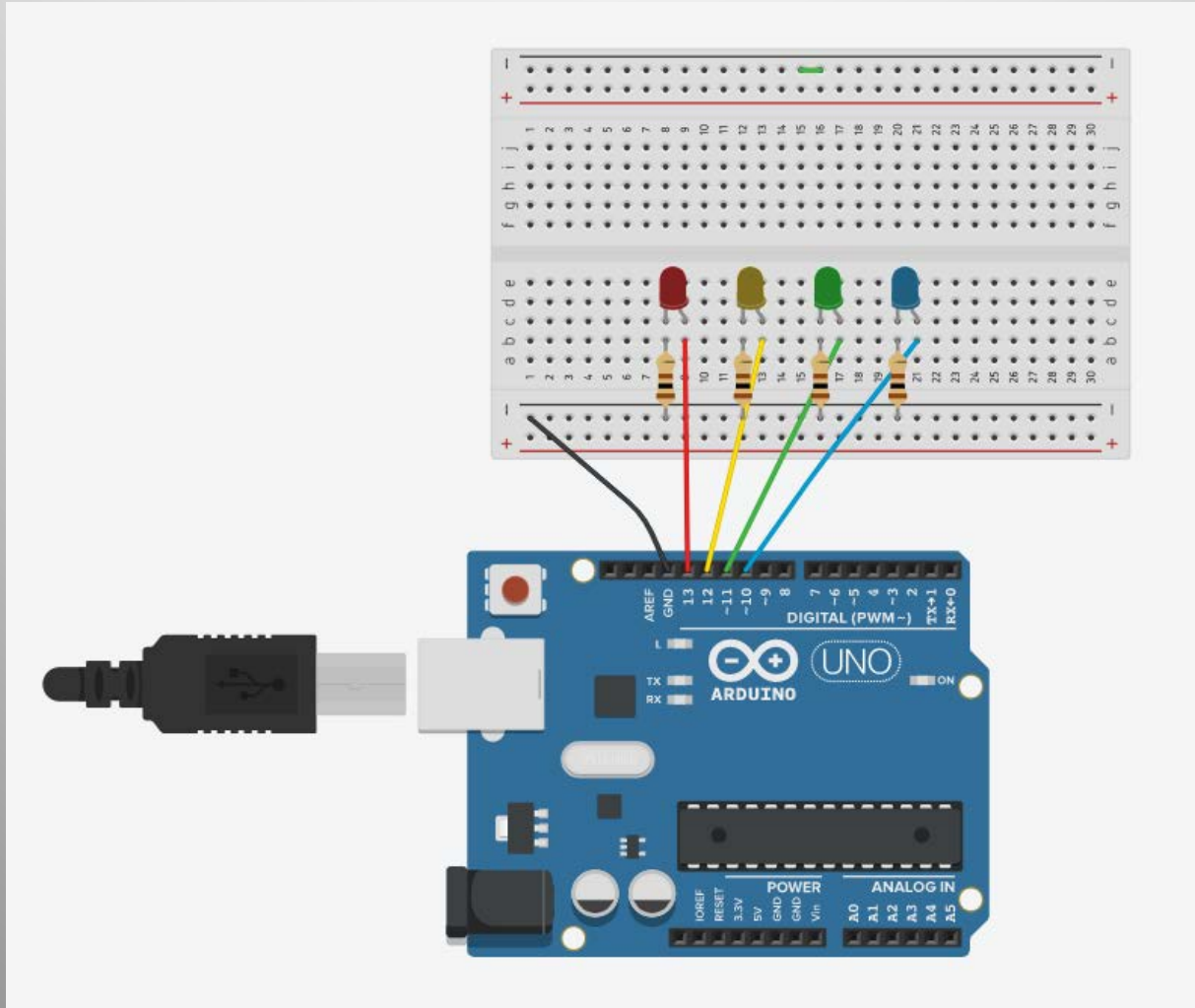




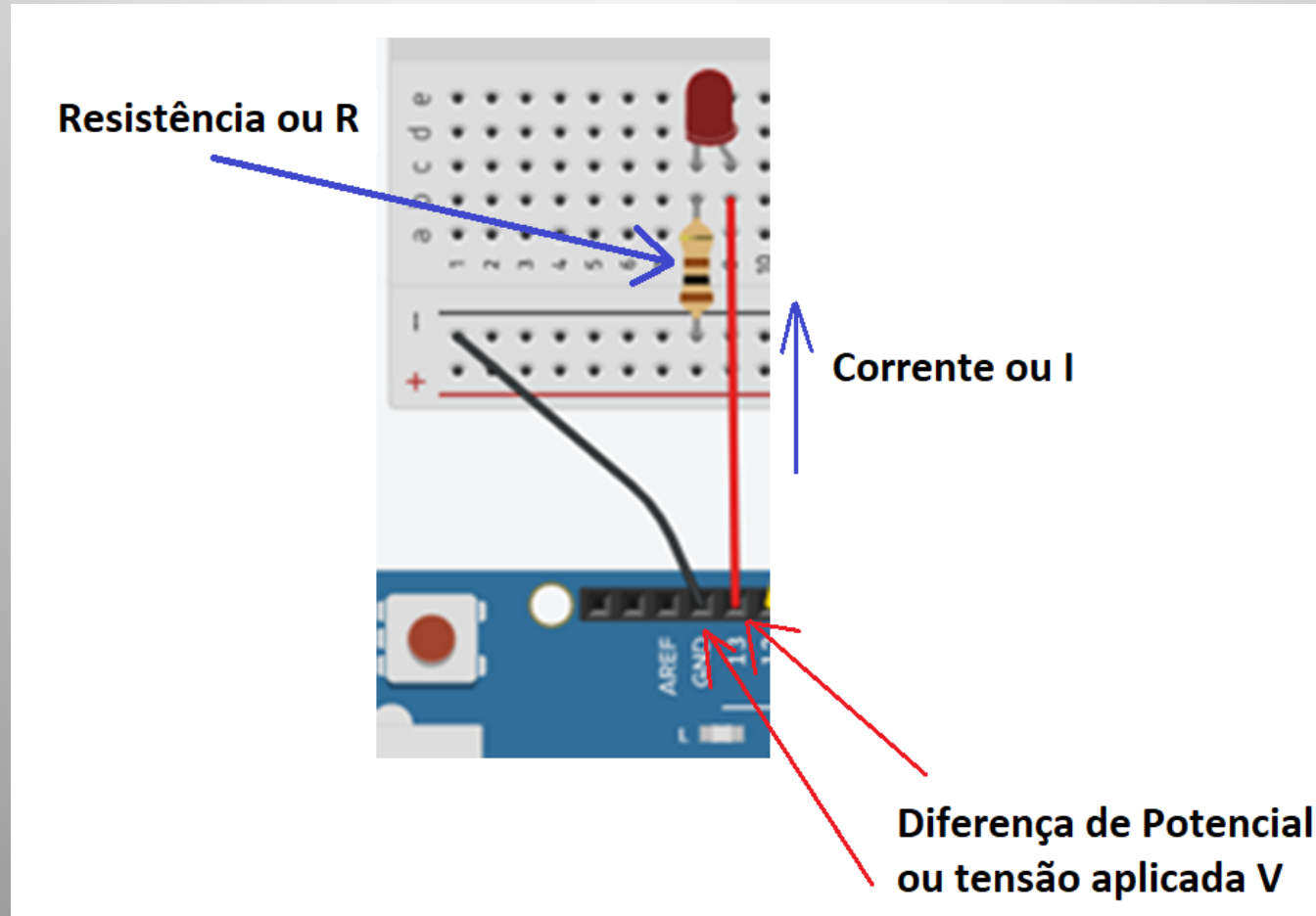
# Qual será o nosso circuito?



# Qual será o nosso circuito?



# O que é a montagem?

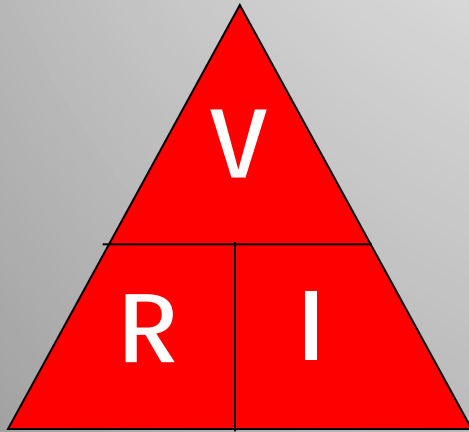




# Lei de OHM

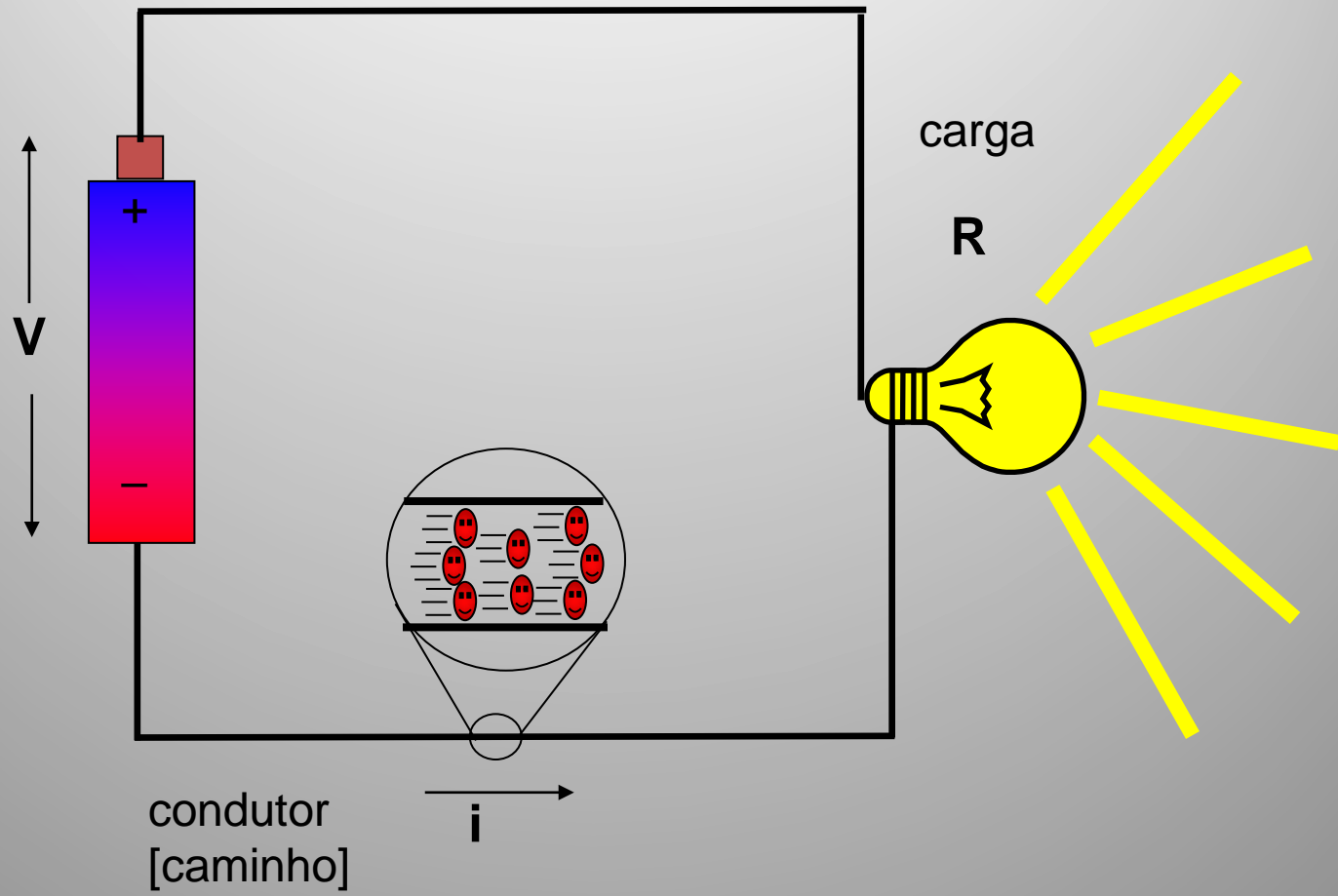
$$V = R \times I \text{ ou}$$

$$I = V / R$$

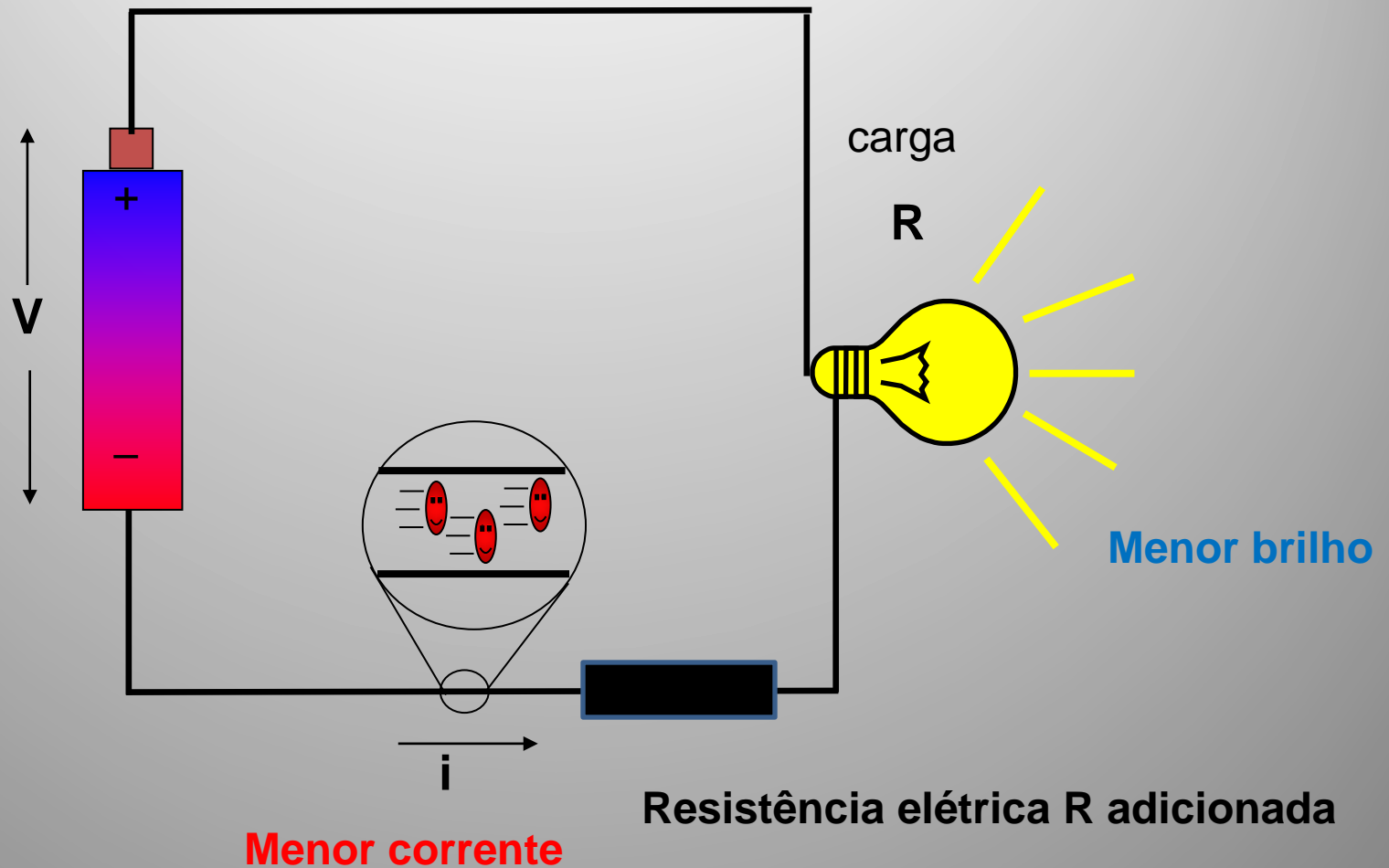


Se uma tensão  $V$  for aplicada a um circuito, um elemento  $R$  poderá diminuir a corrente que circulará no mesmo

# Um circuito elétrico



# Um circuito elétrico





# Um primeiro Programa

int led;      ← Definimos uma variável **led**

void setup() {  
    ← Configurações iniciais  
}

void loop() {      ← Loop sem fim

    ← Ações que serão  
    continuamente executadas  
}

# Um primeiro Programa

```
int led = 13;
void setup() {
  pinMode(led, OUTPUT);
}
void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

Definimos uma variável **led**,  
no caso será o nosso pino 13

Avisamos que será uma  
saída, pois iremos ligar  
um led.

Saida digital led recebe 1

Saida digital led recebe 0

Aguardamos 1 segundo

# Entendendo o Programa

- comandos básicos da linguagem
  - pinMode ( pino, modo )

Esta define um pino com entrada ou saída. O arduino possui 20 pinos disponíveis, 14 digitais e 6 analógicos (0 a 13 e 14 a 19). Esse comando deverá estar preferencialmente na função setup ().

Exemplo:

```
pinMode (13, OUTPUT);  
pinMode (10, INPUT);
```



# Entendendo o Programa

- **digitalWrite ( pino, valor )**

Liga ou desliga uma saída digital, apenas caso o pino seja definido como OUTPUT.

Os valores podem ser HIGH ou LOW (1 ou 0).

Exemplo:

```
digitalWrite (13, HIGH);  
digitalWrite (13, LOW);
```

# Entendendo o Programa

- `delay( ms)`

Aguarda o tempo passado como argumento em ms.

Exemplo:

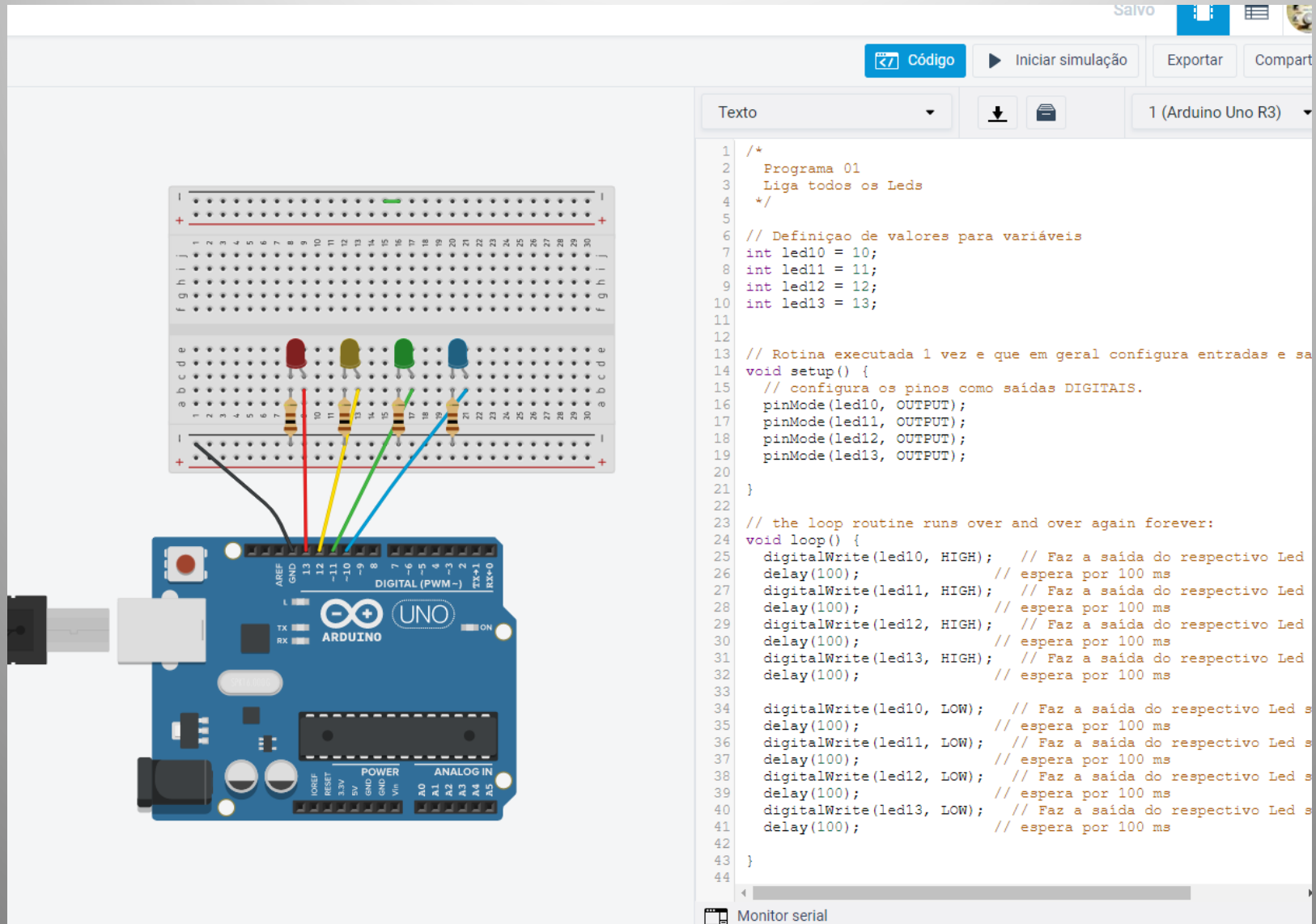
```
delay (1000) ;
```

# Um pouco mais elaborado:

```
arduino_01 $  
~/  
// Definição de valores para variáveis  
int led10 = 10;  
int led11 = 11;  
int led12 = 12;  
int led13 = 13;  
|  
// Rotina executada 1 vez e que em geral configura entradas e saídas  
void setup() {  
    // configura os pinos como saídas DIGITAIS.  
    pinMode(led10, OUTPUT);  
    pinMode(led11, OUTPUT);  
    pinMode(led12, OUTPUT);  
    pinMode(led13, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(led10, HIGH); // Faz a saída do respectivo Led ser alta ou High)  
    delay(100);                // espera por 100 ms  
    digitalWrite(led11, HIGH); // Faz a saída do respectivo Led ser alta ou High)  
    delay(100);                // espera por 100 ms  
    digitalWrite(led12, HIGH); // Faz a saída do respectivo Led ser alta ou High)  
    delay(100);                // espera por 100 ms  
    digitalWrite(led13, HIGH); // Faz a saída do respectivo Led ser alta ou High)  
    delay(100);                // espera por 100 ms  
    digitalWrite(led10, LOW);  // Faz a saída do respectivo Led ser baixa ou Low)  
    delay(100);                // espera por 100 ms  
    digitalWrite(led11, LOW);  // Faz a saída do respectivo Led ser baixa ou Low)  
    delay(100);                // espera por 100 ms  
    digitalWrite(led12, LOW);  // Faz a saída do respectivo Led ser baixa ou Low)  
    delay(100);                // espera por 100 ms  
    digitalWrite(led13, LOW);  // Faz a saída do respectivo Led ser baixa ou Low)  
    delay(100);                // espera por 100 ms  
}
```



# Executando ...



The image shows a screenshot of the Arduino IDE interface. On the left, a breadboard circuit is connected to an Arduino Uno R3. Four LEDs are connected to digital pins 10, 11, 12, and 13. The positive terminal of each LED is connected to its respective pin, and the negative terminal is connected to ground. The Arduino board is connected to a USB cable. On the right, the Arduino IDE code editor displays a C++ program. The code is a simple program to turn all four LEDs on. The code is as follows:

```
1 /*
2   Programa 01
3   Liga todos os Leds
4 */
5
6 // Definição de valores para variáveis
7 int led10 = 10;
8 int led11 = 11;
9 int led12 = 12;
10 int led13 = 13;
11
12
13 // Rotina executada 1 vez e que em geral configura entradas e sa
14 void setup() {
15   // configura os pinos como saídas DIGITAIS.
16   pinMode(led10, OUTPUT);
17   pinMode(led11, OUTPUT);
18   pinMode(led12, OUTPUT);
19   pinMode(led13, OUTPUT);
20 }
21
22
23 // the loop routine runs over and over again forever:
24 void loop() {
25   digitalWrite(led10, HIGH); // Faz a saída do respectivo Led
26   delay(100); // espera por 100 ms
27   digitalWrite(led11, HIGH); // Faz a saída do respectivo Led
28   delay(100); // espera por 100 ms
29   digitalWrite(led12, HIGH); // Faz a saída do respectivo Led
30   delay(100); // espera por 100 ms
31   digitalWrite(led13, HIGH); // Faz a saída do respectivo Led
32   delay(100); // espera por 100 ms
33
34   digitalWrite(led10, LOW); // Faz a saída do respectivo Led s
35   delay(100); // espera por 100 ms
36   digitalWrite(led11, LOW); // Faz a saída do respectivo Led s
37   delay(100); // espera por 100 ms
38   digitalWrite(led12, LOW); // Faz a saída do respectivo Led s
39   delay(100); // espera por 100 ms
40   digitalWrite(led13, LOW); // Faz a saída do respectivo Led s
41   delay(100); // espera por 100 ms
42
43 }
44
```

The IDE interface includes a toolbar at the top with buttons for "Código", "Iniciar simulação", "Exportar", and "Compart". The "Monitor serial" window is visible at the bottom.

# Comunicação Serial

- `available()`    Obtém o número de bytes disponíveis para leitura na porta serial.
- `begin()`    Configura a taxa de dados em bauds para transmissão serial de dados
- `end()`    Desabilita a comunicação serial, permitindo que os pinos TX e RX (0 e 1) sejam usados para entrada e saída geral de sinais.
- `print()`    Imprime dados na porta serial em formato legível por humanos (texto ASCII)
- `println()`    Idem a `print()`, porém adicionando um caractere de retorno de carro (ASCII 13) e um de nova linha (ASCII 10) no final da string.
- `read()`    Lê dados a partir da porta serial.
- `readBytes()`    Lê caracteres a partir da porta serial em um buffer até que uma determinada quantidade de caracteres tenha sido lida.
- `write()`    Escreve dados binários na porta serial

# Comunicação Serial 1

arduino\_02

```
// Testando a saída Serial

int numero=0;

void setup() {
  Serial.begin (9600);
  Serial.println("Gerando valores sequenciais");
}

void loop() {
  Serial.print ("Valor: ");
  Serial.println(numero);
  delay(500);
  numero++;
}
```

# Entendendo o Programa

- **Serial.begin ( velocidade )**

Essa função abre uma comunicação serial na velocidade passada como argumento. A função usa os pinos 0 e 1 para recepção e transmissão.

Exemplo:

```
Serial.begin ( 9600 ) ;
```



# Entendendo o Programa

- `Serial.println ( valor )`

Essa função transmite os caracteres ascii do arduino para o computador.

Exemplo:

```
Serial.println ( " Alo, tudo bem? " ) ;
```

# Comunicação Serial 1

00:02

Código

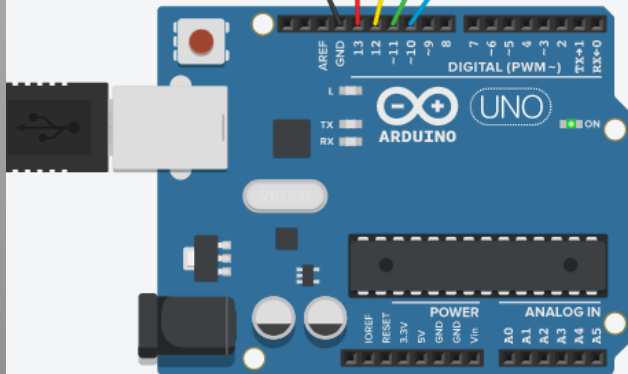
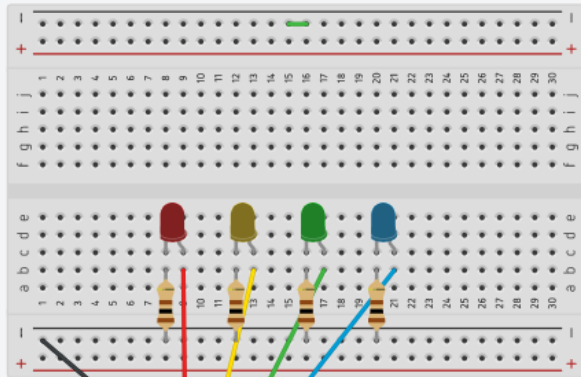
Parar simulação

Exportar

Cor



1 (Arduino Uno R3)



```
1 // Testando a saída Serial
2
3 int numero=0;
4
5 void setup() {
6   Serial.begin (9600);
7   Serial.println("Gerando valores sequenciais");
8 }
9
10 void loop(){
11   Serial.print ("Valor: ");
12   Serial.println(numero);
13   delay(500);
14   numero++;
15 }
16
```

Monitor serial

Gerando valores sequenciais  
Valor: 0  
Valor: 1  
Valor: 2  
Valor: 3  
Valor: 4  
Valor: 5

Env.

Apag.

# Exemplo 03:

arduino\_03 \$

```
char entrada = 'c';
int led = 13;
int i;
void setup() {
    Serial.begin(9600);    // abre a porta serial a 9600 bps
    pinMode(led, OUTPUT);
}

void loop() {

    // verifica se existem dados a serem lidos
    if (Serial.available() > 0)
    {
        // lê o dado
        entrada = Serial.read();
        if (entrada != '\n')
        {
            if (entrada == 'a')
                digitalWrite(led, HIGH);
            if (entrada == 'b')
                digitalWrite(led, LOW);
            Serial.println(entrada);
        }
    }
}
```

# Entendendo o Programa

- `Serial.read ( )`

Essa função lê um byte recebido pelo arduino.

Exemplo:

```
int recebido;  
recebido = Serial.read ( ) ;
```



# Entendendo o Programa

- **Serial.available ( )**

Essa função retorna o número de bytes disponíveis para leitura na porta serial.

Exemplo:

```
int recebido;  
if (Serial.available () > 0 )  
    recebido = Serial.read ( ) ;
```

# Exemplo 04:



arduino\_04

```
int entrada = 0;
int led = 13;

void setup() {
    Serial.begin(9600);
    pinMode(led, OUTPUT);
}

void loop() {

    if (Serial.available() > 0) {
        entrada = Serial.parseInt();

        if (entrada == 1)
            digitalWrite(led, HIGH);
        if (entrada == 2)
            digitalWrite(led, LOW);

        Serial.print(entrada);
    }
}
```

# Exemplo 05:

```
arduino_05 $  
int entrada1 = 0;  
int entrada2 = 0;  
int saida;  
int led = 13;  
void setup() {  
    Serial.begin(9600);  
    pinMode(led, OUTPUT);  
}  
void loop() {  
    if (Serial.available() > 0) {  
        entrada1 = Serial.parseInt();  
        entrada2 = Serial.parseInt();  
        if (Serial.read() == '\\n')  
        {  
            saida = soma(entrada1, entrada2);  
            Serial.print(saida);  
            if (saida == 1)  
                digitalWrite(led, HIGH);  
            if (saida == 2)  
                digitalWrite(led, LOW);  
        }  
    }  
}  
  
int soma(int a, int b)  
{  
    return(a+b);  
}
```

# Exemplo 06:

arduino\_06 \$

```
int entrada1 = 0;
int entrada2 = 0;
int entrada3 = 0;
int saida;
int led = 13;
void setup() {
    Serial.begin(9600);
    pinMode(led,OUTPUT);
}
void loop() {
    if (Serial.available() > 0) {
        entrada1 = Serial.parseInt();
        entrada2 = Serial.parseInt();
        entrada3 = Serial.parseInt();

        if (Serial.read()=='\n')
        {
            saida = soma(entrada1,entrada2,entrada3);
            Serial.print(saida);
            if (saida == 1)
                digitalWrite(led,HIGH);
            if (saida == 2)
                digitalWrite(led,LOW);
        }
    }
}
```

```
int soma(int a, int b, int c)
{
    return (a+b+c);
}
```



# Exemplo 07:

```
void loop() {  
    if (Serial.available() > 0) {  
        entrada1 = Serial.parseInt();  
        entrada2 = Serial.parseInt();  
        Serial.print("entrada1= ");  
        Serial.print(entrada1);  
        Serial.println();  
        Serial.print("entrada2= ");  
        Serial.print(entrada2);  
        Serial.println();  
        saida = portaxor(entrada1, entrada2);  
        Serial.print("xor= ");  
        Serial.print(saida);  
        Serial.println();  
        mostra(saida);  
        saida = portaor(entrada1, entrada2);  
        Serial.print("or= ");  
        Serial.print(saida);  
        Serial.println();  
        saida = portaand(entrada1, entrada2);  
        Serial.print("and= ");  
        Serial.print(saida);  
        Serial.println();  
        saida = portanot(entrada1);  
        Serial.print("not entrada1= ");  
        Serial.print(saida);  
        Serial.println();  
    }  
}
```

```
int entrada1 = 0;  
int entrada2 = 0;  
int led1 = 13;  
int saida;  
  
void setup() {  
    Serial.begin(9600);  
    pinMode(led1, OUTPUT);  
}
```

```
int portaxor(int a, int b)  
{  
    return(a^b);  
}  
  
int portaor(int a, int b)  
{  
    return(a|b);  
}  
  
int portaand(int a, int b)  
{  
    return(a&b);  
}  
  
int portanot(int a)  
{  
    return(~a);  
}  
  
int mostra (int a)  
{  
    if (a == 1)  
        digitalWrite(led1, 1);  
    else digitalWrite(led1, 0);  
}
```