rexlab | Desde 1997

Introdução ao Arduino

Agenda

- Apresentação do Arduino
- Conhecendo a IDE do Arduino
- Primeiro Programa no Arduino
- Software Simulador

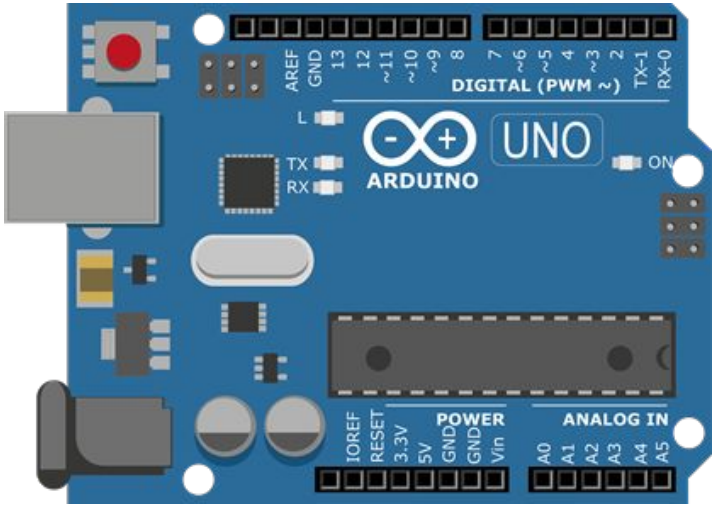
Apresentação do Arduino

Apresentação do Arduino

- O Arduino é uma plataforma utilizada para **prototipação de circuitos eletrônicos**
- É composto por uma placa com **microcontrolador** Atmel AVR e um ambiente de programação baseado em Wiring e C++
- Tanto o hardware como o ambiente de programação do Arduino são livres, ou seja, qualquer pessoa pode modificá-los e reproduzi-los.

Apresentação do Arduino

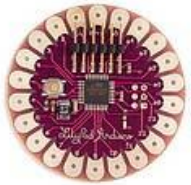
- Se baseia na facilidade de uso de hardware e software
- No hardware temos um microcontrolador capaz de ler entradas (inputs) e fornecer saídas (outputs)



Arduino Uno

Arduino Leonardo

Arduino Mega 2560



Arduino LilyPad

Arduino Mega ADK

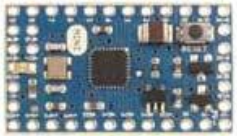
Arduino Fio

Arduino Ethernet

Arduino Pro

Arduino BT

Arduino Nano



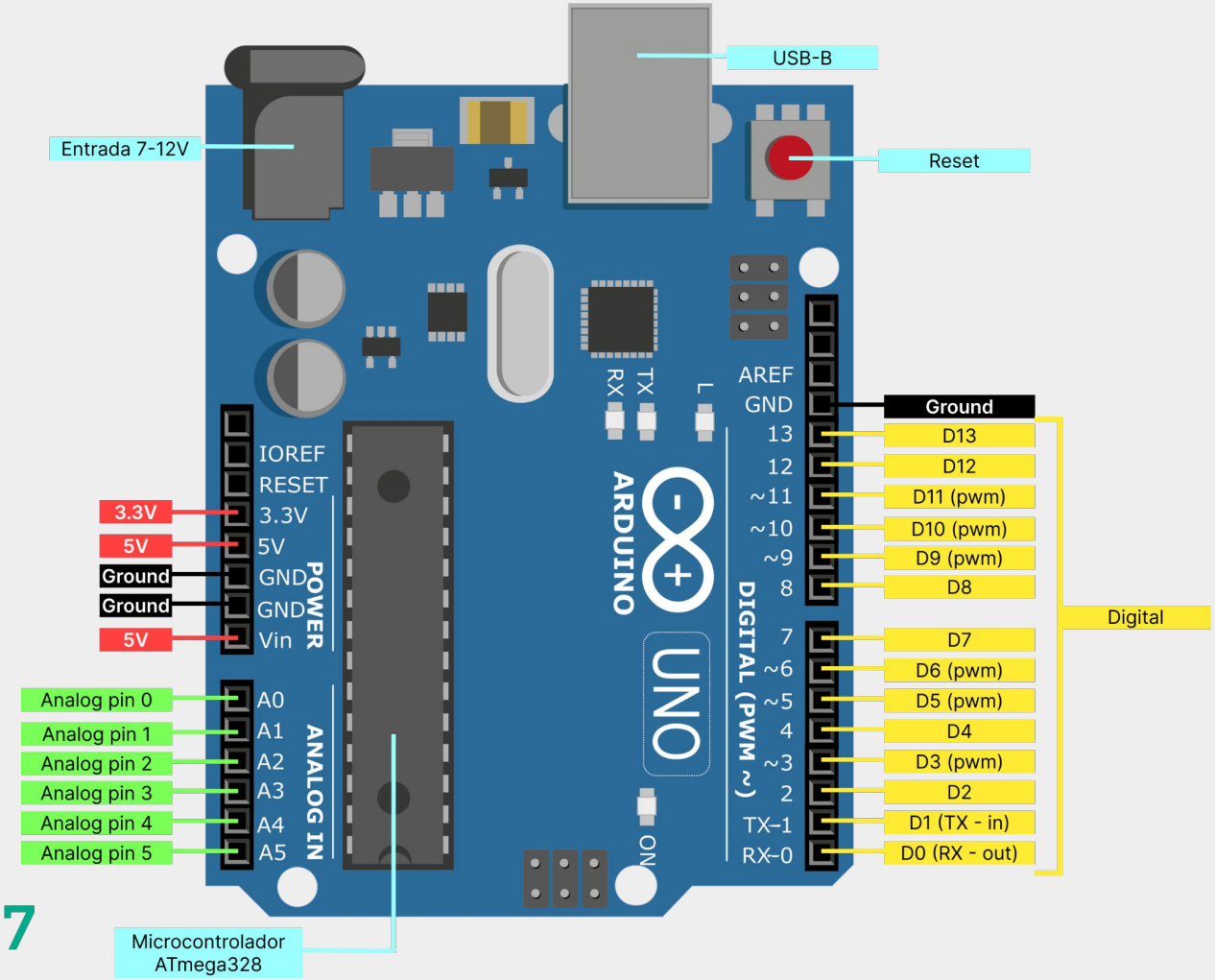
Arduino Mini

Arduino Pro Mini

Apresentação do Arduino

- Possui conectores de entrada e saída para controle de dispositivos e um circuito que facilita sua conexão com um computador via porta USB

- Além da placa, existe todo um ecossistema em volta do Arduino, como software de programação e desenvolvimento (IDE), bibliotecas, tutoriais, fóruns, comunidade, hardwares adicionais...



Apresentação do Arduino

- As **entradas digitais** só podem assumir dois estados, **HIGH** e **LOW**, ou seja, 0 V ou 5 V. Sua lógica é similar a um interruptor de dois estados.

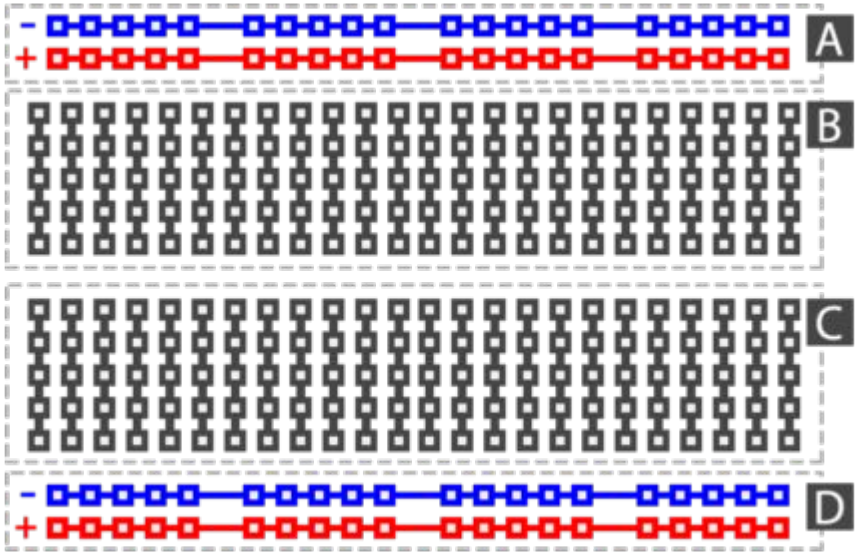
```
digitalWrite(porta, estado); - escrita
digitalRead(porta); - leitura
```

- Muitas situações a variação das grandezas envolvidas acontece de forma **analógica**, ou seja, variam continuamente em relação ao tempo e **podem assumir infinitos valores dentro de uma faixa**.

```
analogWrite(porta, estado); - escrita no pino PWM
analogRead(porta); - leitura do pino analógico
```

Protoboard

Uma protoboard serve para prototipagem de circuitos eletrônicos. É de fácil utilização e funciona da seguinte forma:



rexlab | Desde 1997

Programação

Programação

A linguagem de programação do Arduino é baseada em C++, com algumas modificações para facilitar o uso.

Todo programa Arduino é dividido em duas **funções** principais:

setup(): Esta função é executada apenas uma vez, na inicialização do Arduino. Define-se as configurações dos pinos, bibliotecas e outras variáveis importantes para o projeto.

loop(): Esta função é executada repetidamente, criando um ciclo infinito. Escreve-se o código que controla o comportamento do projeto, como ligar e desligar LEDs, ler sensores e enviar dados para outros dispositivos.

Programação

sketch_apr23a.ino

```
1   void setup() {
2     // put your setup code here, to run once:
3
4   }
5
6   void loop() {
7     // put your main code here, to run repeatedly:
8
9   }
10
```

Programação

Variáveis: Armazenam informações que podem ser usadas durante o programa. Elas podem ser números, textos, valores booleanos, entre outros.

Constantes: Valores fixos que não podem ser alterados durante o programa.

Funções: Blocos de código reutilizáveis que executam tarefas específicas.

Estruturas de controle: Permitem controlar o fluxo do programa, como loops, condicionais e instruções de salto.

Bibliotecas: Coleções de funções prontas que facilitam a interação com diversos componentes e sensores.

Manipulação de Estados

Apresentação do Arduino

A manipulação de estados no Arduino é o processo de **alterar o estado de um pino digital** de entrada ou saída
O estado de um pino digital pode ser **alto (HIGH ou 1)** ou **baixo (LOW ou 0)**

HIGH representa um nível lógico **alto**, geralmente **5V**
LOW representa um nível lógico **baixo**, geralmente **0V**

A forma mais comum de manipular o estado do pino é pela função **digitalWrite()**

```
1  // Define o pino 8 como alto
2  digitalWrite(8, HIGH);
3
4  // Define o pino 8 como baixo
5  digitalWrite(8, LOW);
6
```

rexlab | Desde 1997

Apresentação do Arduino

```
1   // Define o pino 8 como alto
2   digitalWrite(8, HIGH);
3
4   // Define o pino 8 como baixo
5   digitalWrite(8, LOW);
6
```

O primeiro argumento é o número do pino digital, e o segundo argumento é o estado desejado (HIGH ou LOW).

Apresentação do Arduino

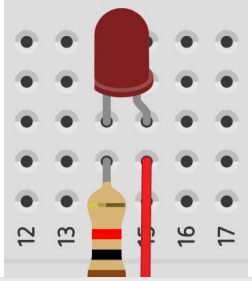
Já para ler o estado de um pino digital usa-se **digitalRead()**. O argumento é o número do pino.

```
// Define pin 2 como entrada (sensor)
pinMode(2, INPUT);

// Define pin 13 como saída (LED)
pinMode(13, OUTPUT);

void loop() {
  // Read (lê) o estado da porta digital 2
  int sensorValue = digitalRead(2);

  // If porta 2 é HIGH, liga o LED
  if (sensorValue == HIGH) {
    digitalWrite(13, HIGH);
  } else {
    // If porta 2 é LOW, desliga o LED
    digitalWrite(13, LOW);
  }
}

```

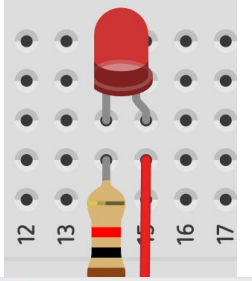
Apresentação do Arduino

Os sinais high e low também são usados para controlar dispositivos externos.

Por exemplo, um LED pode ser ligado ou desligado enviando um sinal high ou low para o pino do LED.




```
1  void setup()
2  {
3    pinMode(8, OUTPUT);
4  }
5
6  void loop()
7  {
8    digitalWrite(8, LOW);
9
10 }
11
```




```
1  void setup()
2  {
3    pinMode(8, OUTPUT);
4  }
5
6  void loop()
7  {
8    digitalWrite(8, HIGH);
9
10 }
11
```

Apresentação do Arduino

Temos também a função **analogWrite()**, que é **usada para definir a largura de pulso** dos pinos **PWM**
A largura de pulso é um valor entre 0 e 255

255 representa um ciclo de onda completo em **nível alto**
0 representa um ciclo de onda completo em **nível baixo**

```
1   // Define a largura do pulso do sinal PWM
2   {
3     analogWrite(9, 255);
4     delay(1000);
5     analogWrite(9, 127);
6     delay(1000);
7     analogWrite(9, 0);
8     delay(1000);
9   }
10
```

significa modulação por largura de pulso, controla a potência na carga

Apresentação do Arduino

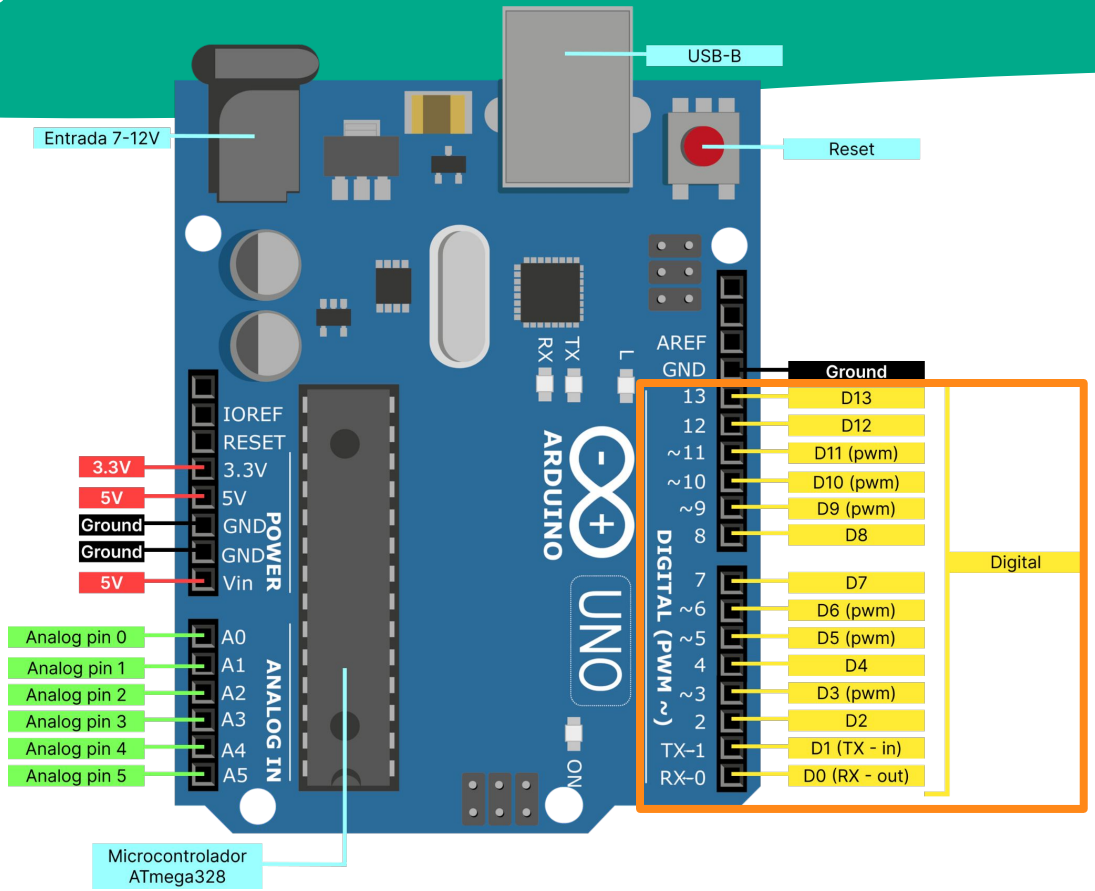
Principal **diferença** entre escrita **digital** e **analógica**:

- Sinais **digitais** só podem assumir dois valores, **0 ou 1**
- Sinais **analógicos** podem assumir um **número infinito** de valores dentro de um intervalo

A **escrita digital** é usada para controlar dispositivos que só podem funcionar com **dois estados**, como um motor **ligado ou desligado**.

A **escrita analógica** é usada para controlar dispositivos que podem funcionar com uma **variedade de níveis de potência**, como um LED que pode ser ajustado para **diferentes níveis** de brilho.

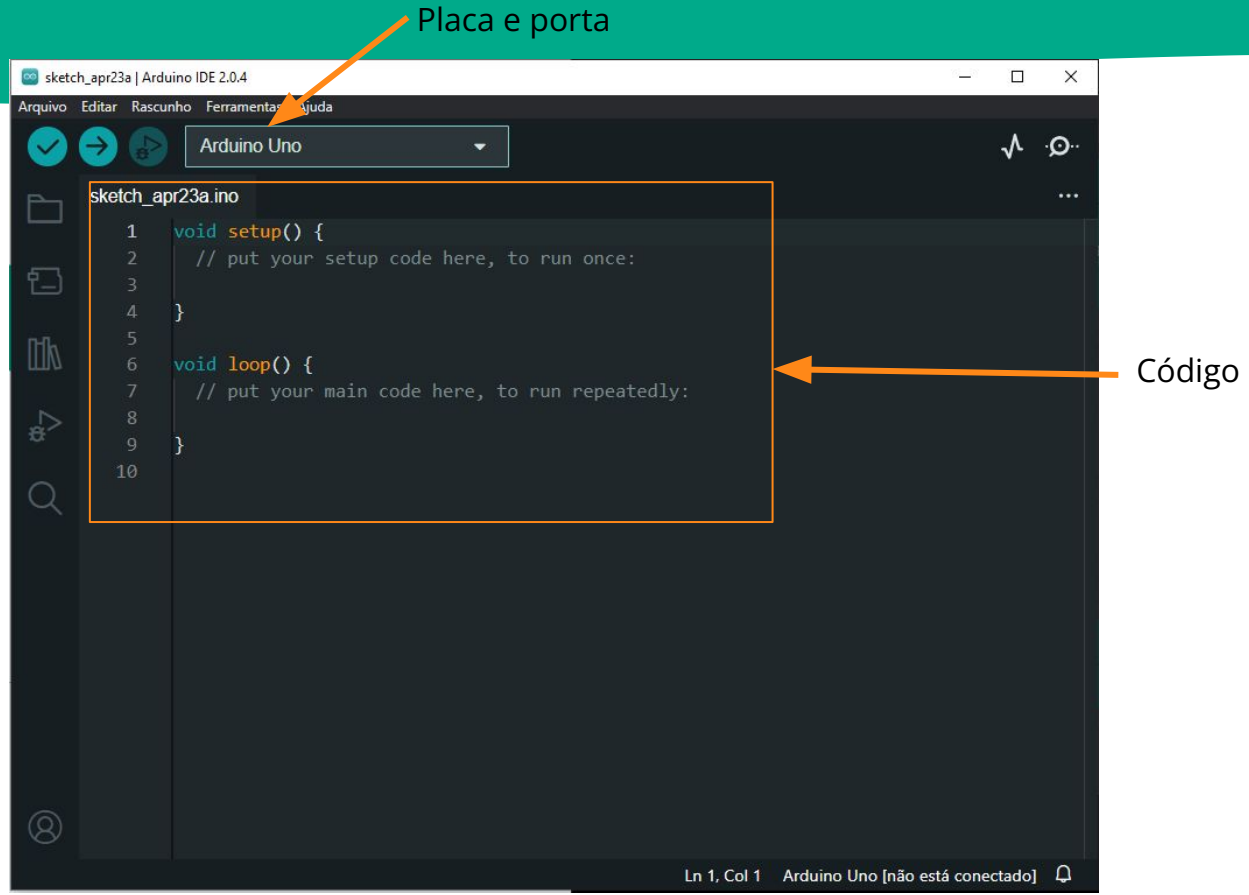
Apresentação do Arduino



rexlab | Desde 1997

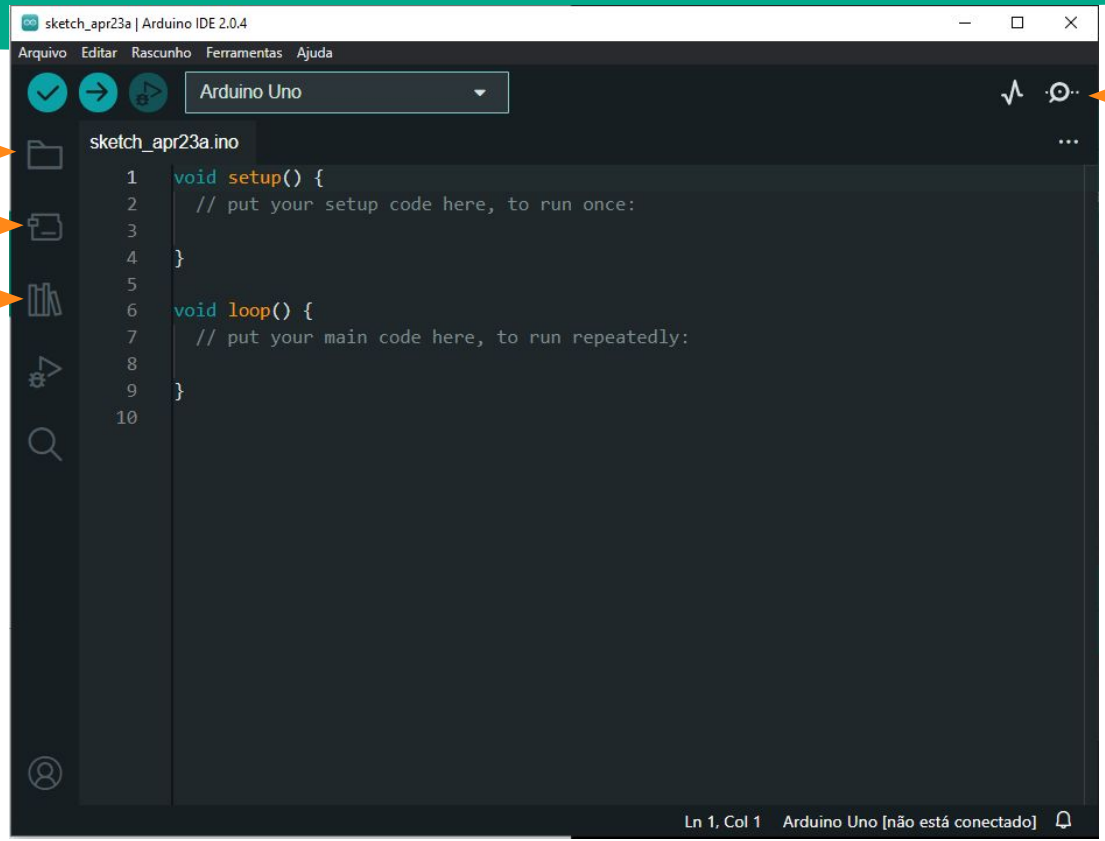
Conhecendo a IDE

Conhecendo a IDE



rexlab Desde 1997

Conhecendo a IDE



Sketchs

Placas

Bibliotecas

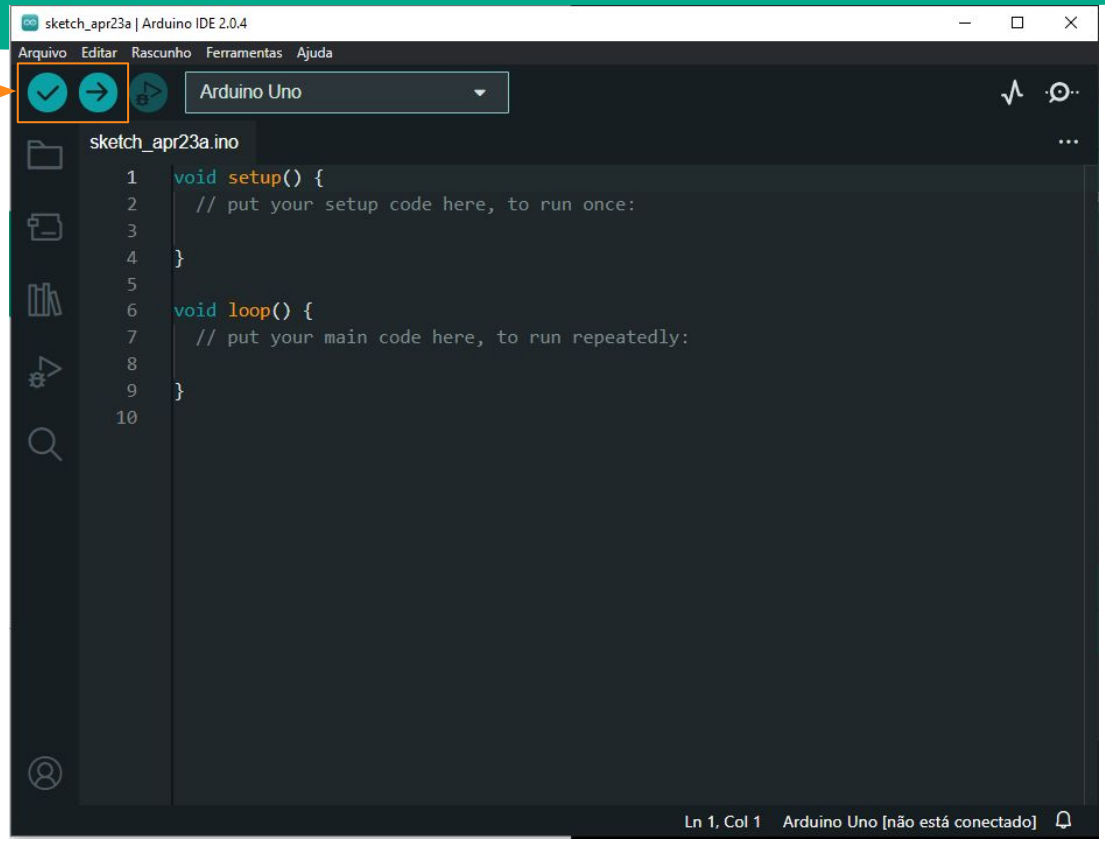
Monitor Serial

rexlab

Desde 1997

Conhecendo a IDE

Compilar e Enviar



rexlab Desde 1997

LEDs

Acionamento de LEDs

Os LEDs são dispositivos semicondutores que emitem luz quando uma corrente elétrica passa por eles

Possuem **dois terminais**, um **ânodo** e um **cátodo**

O **ânodo** é o terminal **positivo** (+)
O **cátodo** é o terminal **negativo** (-)

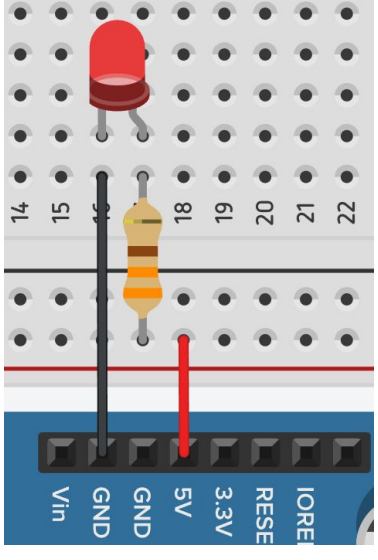
A polaridade do LED é importante e deve ser respeitada ao conectar o LED a uma fonte de alimentação!

rexlab | Desde 1997

Acionamento de LEDs

Existem vários métodos de acender um LED

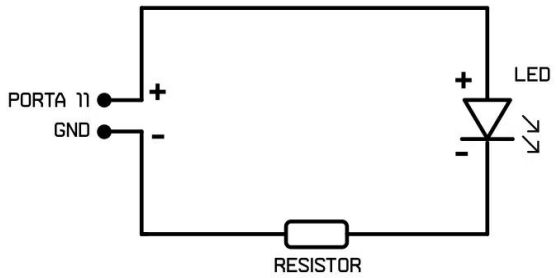
O método mais simples é conectar o LED diretamente a uma fonte de alimentação que forneça a tensão necessária para o LED (lembrar do resistor em série!)



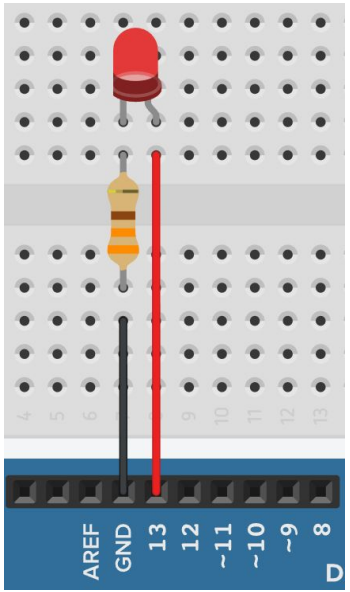
Acionamento de LEDs

Existem vários métodos de acender um LED

Para controlar o LED, pode-se utilizar também uma porta digital da placa microcontroladora, e criar um código para o acionamento, com a função digitalWrite()




```
#define led 13

void setup()
{
  pinMode(led, OUTPUT);
}

void loop()
{
  digitalWrite(led, HIGH);

}
```



Push Button

PUSH-BUTTON

Um push-button (botão) é um tipo de interruptor que é ativado pressionando-o

Podem ser usados em uma ampla variedade de aplicações, desde simples interruptores de luz até complexos painéis de controle

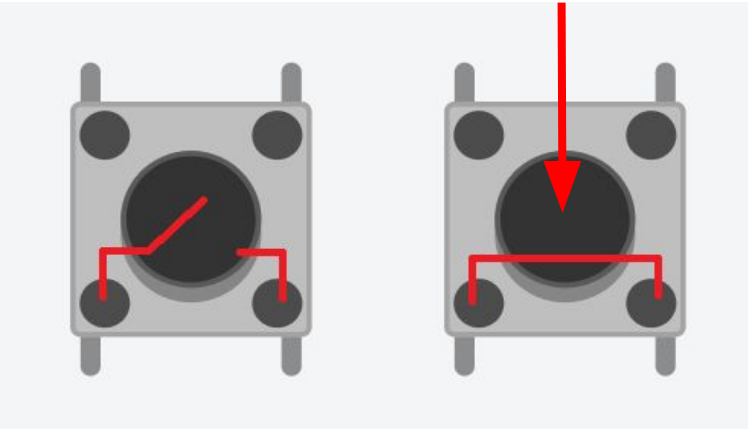
Eles são frequentemente usados em conjunto com outros componentes eletrônicos, como LEDs e relés

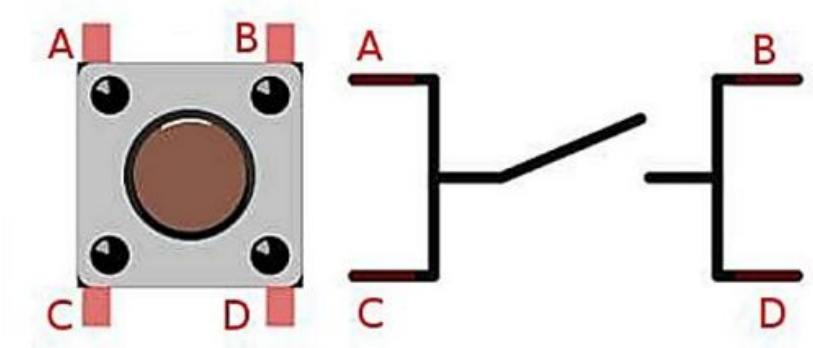
(Podem ser chamadas também de chave táctil)

rexlab | Desde 1997

PUSH-BUTTON

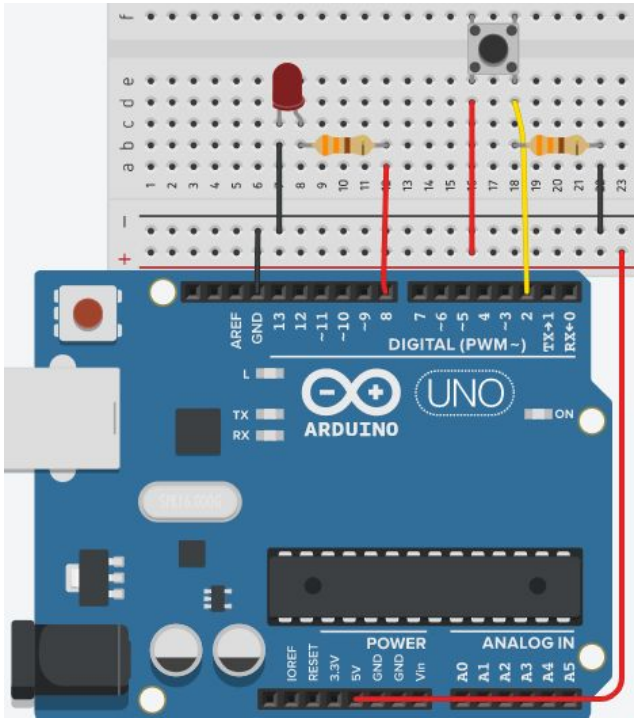
Um push-button (botão) é um tipo de interruptor que é ativado pressionando-o





Utilizando o Push Button

```ino
bool estadoLed = LOW;
bool leituraAnterior;
bool leituraAtual;

void setup() {
  pinMode(13, OUTPUT);
  pinMode(12, INPUT);
  leituraAtual = digitalRead(12);
}

void loop() {
  leituraAnterior = leituraAtual;
  leituraAtual = digitalRead(12);

  if (leituraAnterior == HIGH && leituraAtual == LOW) {
    estadoLed = !estadoLed;
    digitalWrite(13, estadoLed);
  }
}
```



Potenciômetro

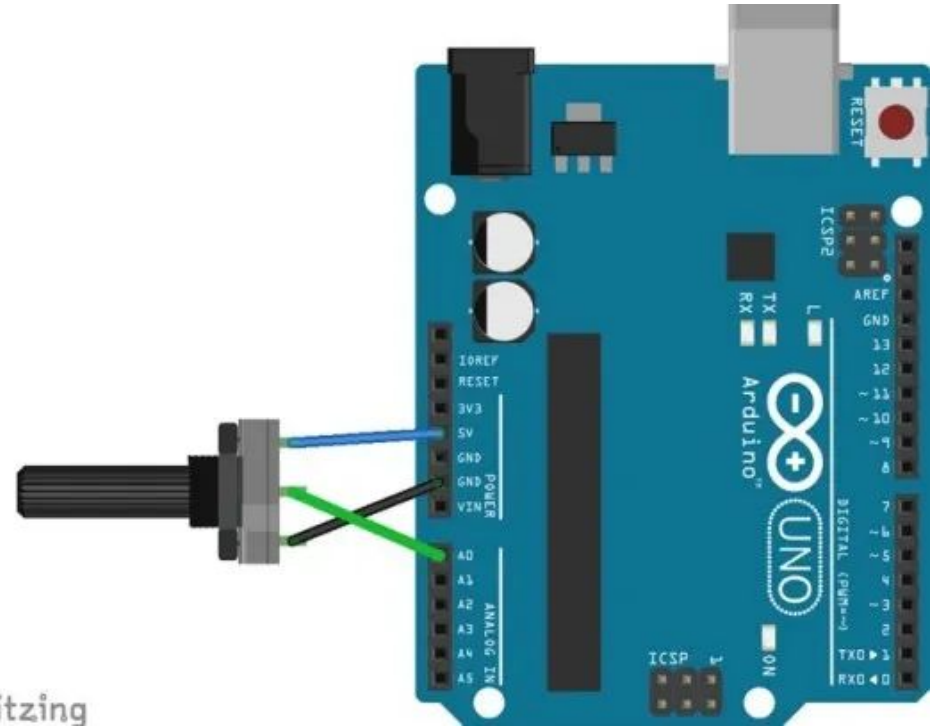
Como funciona o Potenciômetro

Um potenciômetro é um dispositivo eletrônico que ajusta a resistência elétrica entre dois terminais, permitindo o controle da voltagem ou corrente em um circuito ao girar seu eixo. Isso é alcançado através do movimento relativo de um contato deslizante em uma trilha resistiva, alterando a divisão da voltagem no circuito.

rexlab | Desde 1997

Potenciômetro na Prática

```
void setup() {
  Serial.begin(115200);
}

void loop() {
  int potValue = analogRead(A0);
  Serial.println(potValue);
}
```

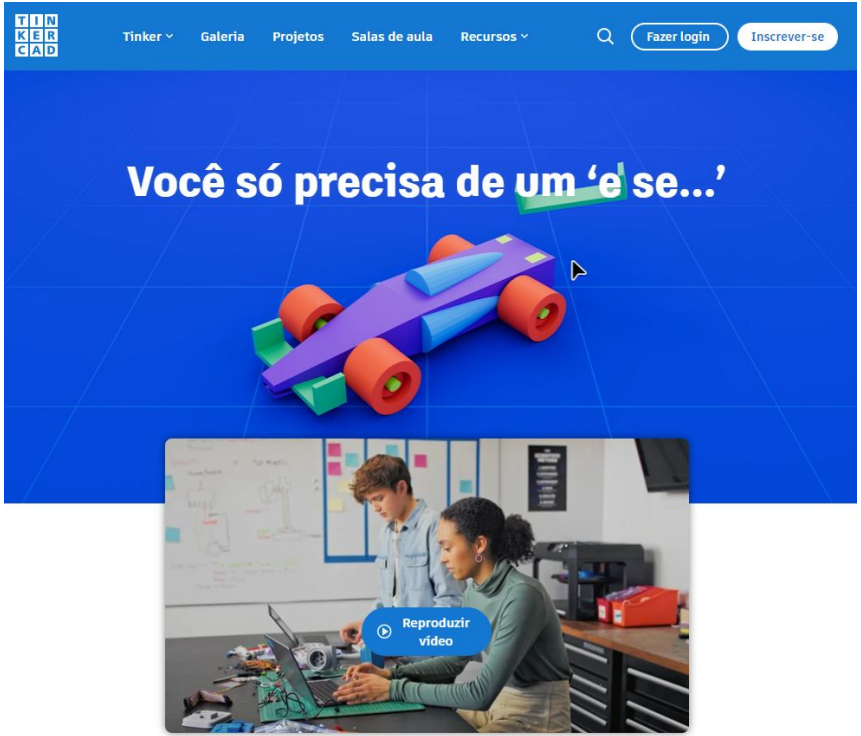


Simuladores

Simuladores

Simulador TINKERCAD

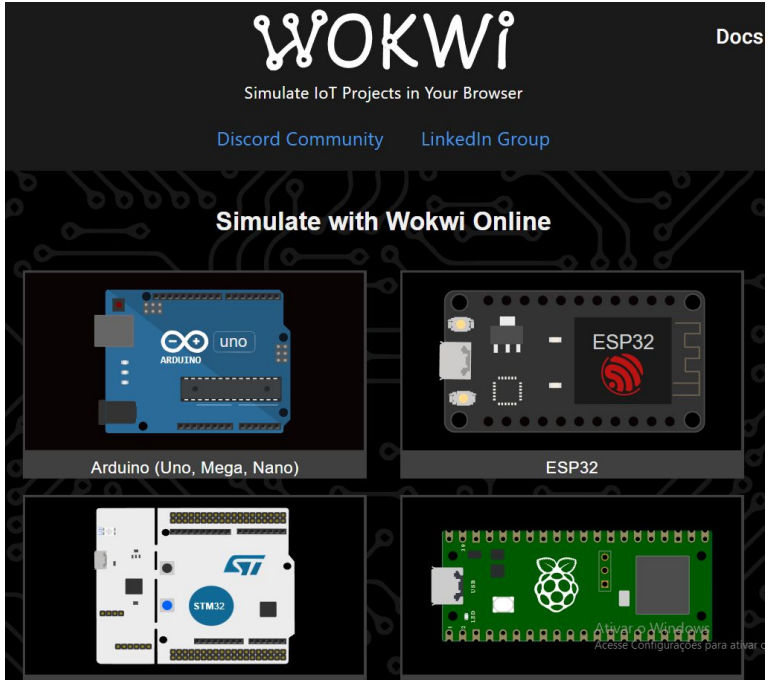
https://www.tinkercad.com/



Simuladores

Simulador Wokwi

https://wokwi.com/



Exercícios

Vamos praticar?

EX 1
‘Crie um circuito com 3 LEDs de cores diferentes.
Todos iniciam apagados.
Cada LED deve acender e apagar, um por vez’

Vamos praticar?

EX 2

‘Conecte um potenciômetro ao Arduino e use sua leitura para ajustar a intensidade de um LED conectado a um pino PWM, onde o giro do potenciômetro varia a luminosidade do LED de completamente apagado a totalmente aceso.

Dica: Use a função map(valor, deMin, deMax, paraMin, paraMax) para converter a faixa de leitura do potenciômetro (por exemplo, 0-1023) para ajustar a intensidade do LED (0-255).’

Vamos praticar?

EX 3

‘Crie um circuito com dois LEDs e um push-button.
Inicialmente um dos LEDs deve estar apagado e o outro aceso.
Crie um código que leia o estado do botão.
Ao clicar no botão o LED aceso deve apagar, e o apagado acender.
Use como parâmetro de acionamento o estado do botão.’