

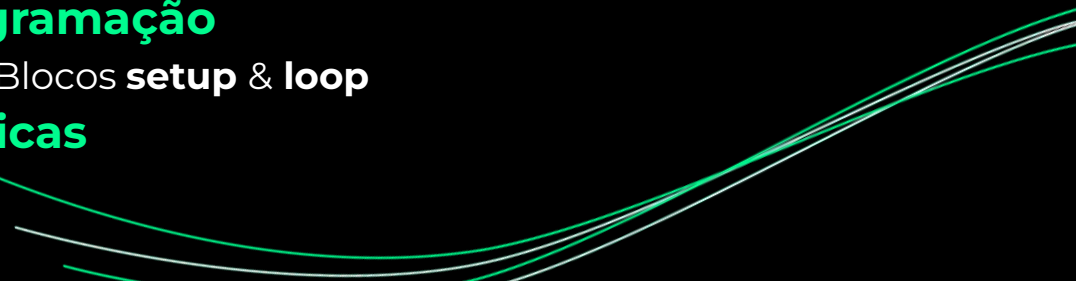


Introdução a Eletronica básica (IoT)

Por: Marcos Augusto Campagnaro Mucelini



Nesta aula veremos os conceitos de:


- **O que é Arduino**
 - arquitetura
 - usos do arduino
 - microcontroladores
 - **IDE**
 - Elementos gerais
 - Instalação
 - Dicas
 - **Programação**
 - Blocos **setup** & **loop**
 - **Praticas**
- 





O que é Arduino?

Uma **placa** para prototipagem de projetos eletrônicos



Uma plataforma que consiste de um **circuito** contendo um **microcontrolador** configurado para facilitar a **programação** e controle de **entradas e saídas**.

Criado para **facilitar** o uso da eletrônica por artistas.

Open hardware - Circuito pode ser montado e vendido sem pagar royalties.



Arquitetura do Arduino

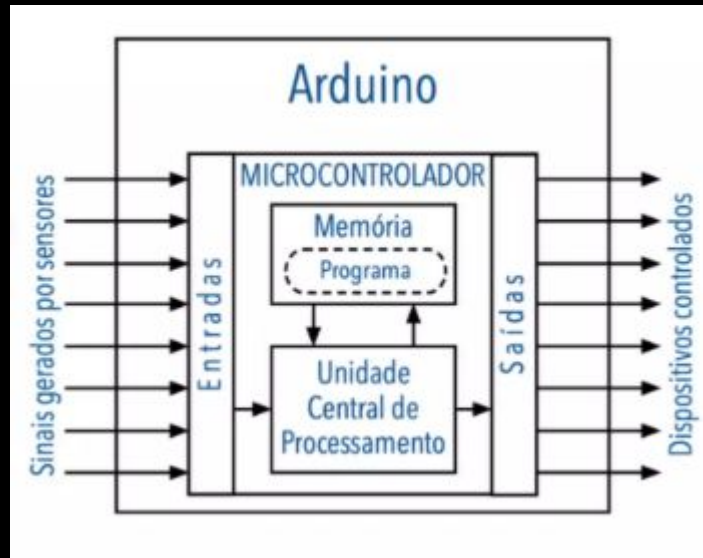
Entradas

(sensores, chaves, geradores de sinais, etc)



Processamento

(Plataforma de processamento baseada em microcontroladores)



Saídas

(dispositivos, mecanismos, processadores de sinais)





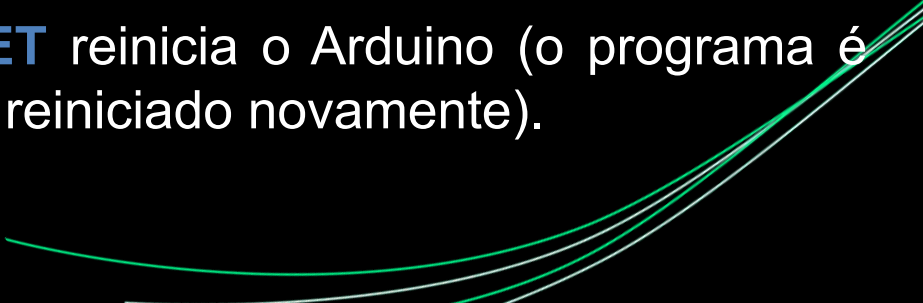
Uso do Arduino

Depois que um programa é transferido para a memória do Arduino, ele executará automaticamente sempre que for conectado a uma fonte de energia.



Um programa geralmente é construído para executar continuamente, lendo os sinais recebidos de sensores e ou mecanismos de acionamento.

O botão **RESET** reinicia o Arduino (o programa é interrompido e reiniciado novamente).

A decorative graphic element in the bottom right corner consisting of several thin, curved lines in light blue and white, sweeping upwards and to the right.

Microcontroladores

Um microcontrolador é um pequeno computador contido num unico chip

O Arduino é um cricuito que existe para dar suporte ao microcontrolador (que faz todo o trabalho).

Arduinos utilizam o microncontrolador da arquitetura AVR fabricadas pela ATMel. Os mais utilizados são da série ATMega e ATTiny.

ATMega328 SMD



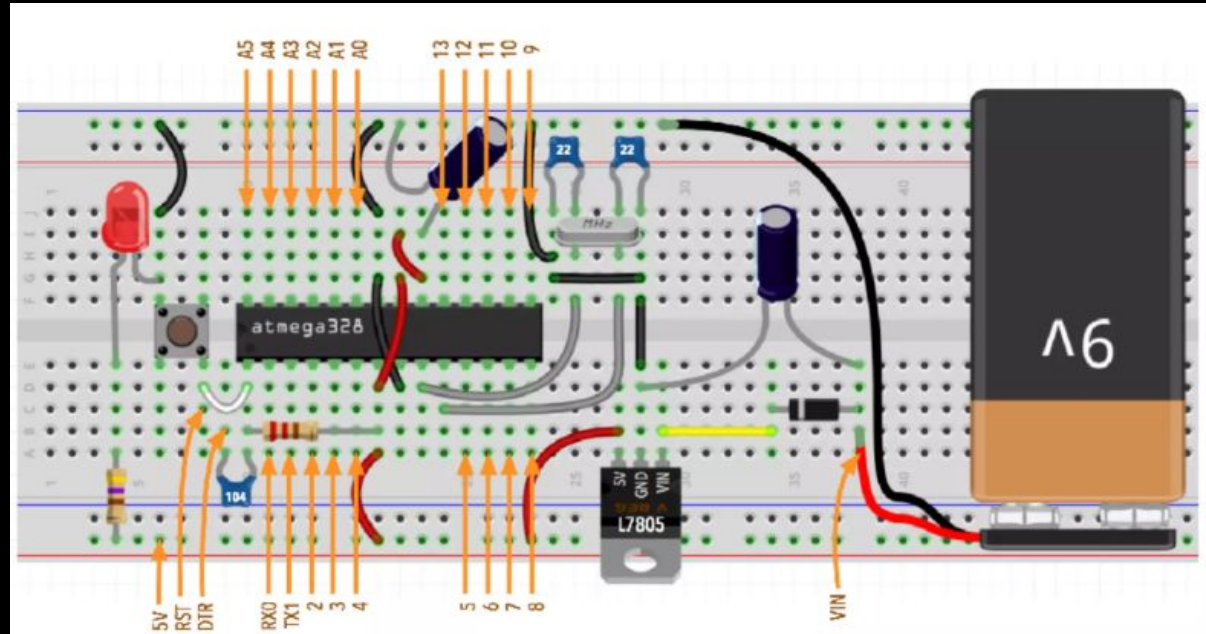
ATMega168 DIL



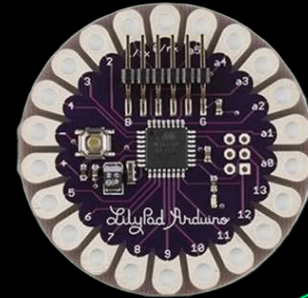
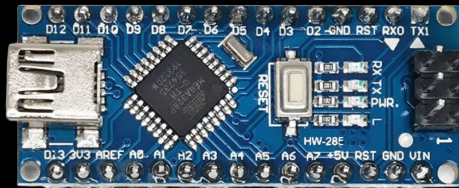
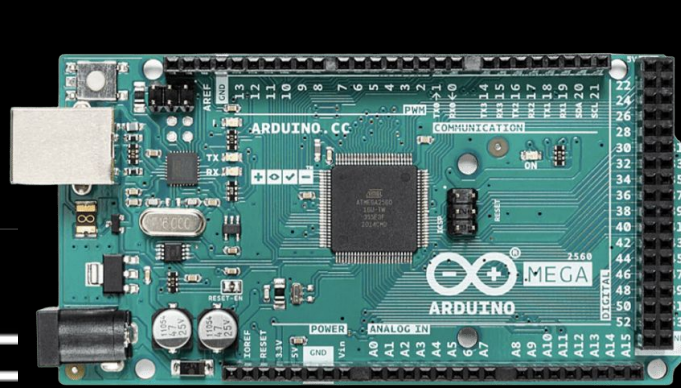
ATTiny85



Exemplo de circuito Arduino utilizando um ATmega328



Microcontroladores



Pinagem Arduino

Botão de
RESET

Pinos digitais enumerados de 0 à 13

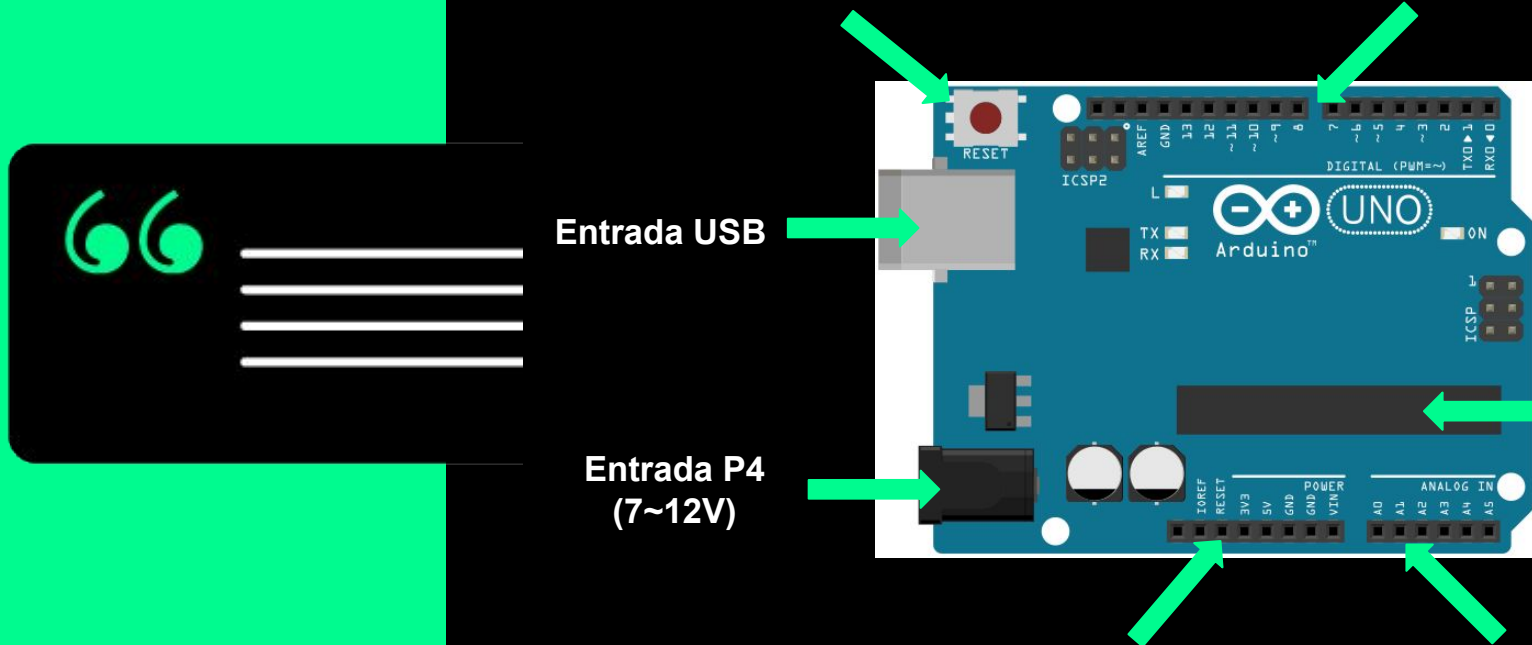
Entrada USB

Entrada P4
(7~12V)

microcontrolador

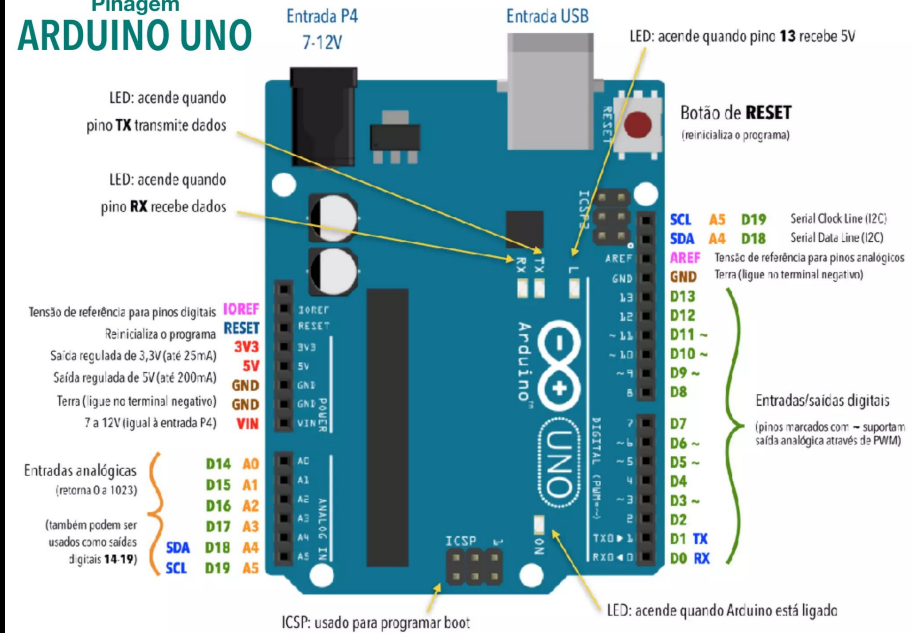
Pinos de alimentação

Entradas analógicas



Pinagem Arduino

Pinagem ARDUINO UNO





Instalação

Para programar o Arduino é necessário usar um **ambiente de desenvolvimento** (IDE) para compilar o código (traduzir para linguagem de máquina do microcontrolador) e transferir via USB.

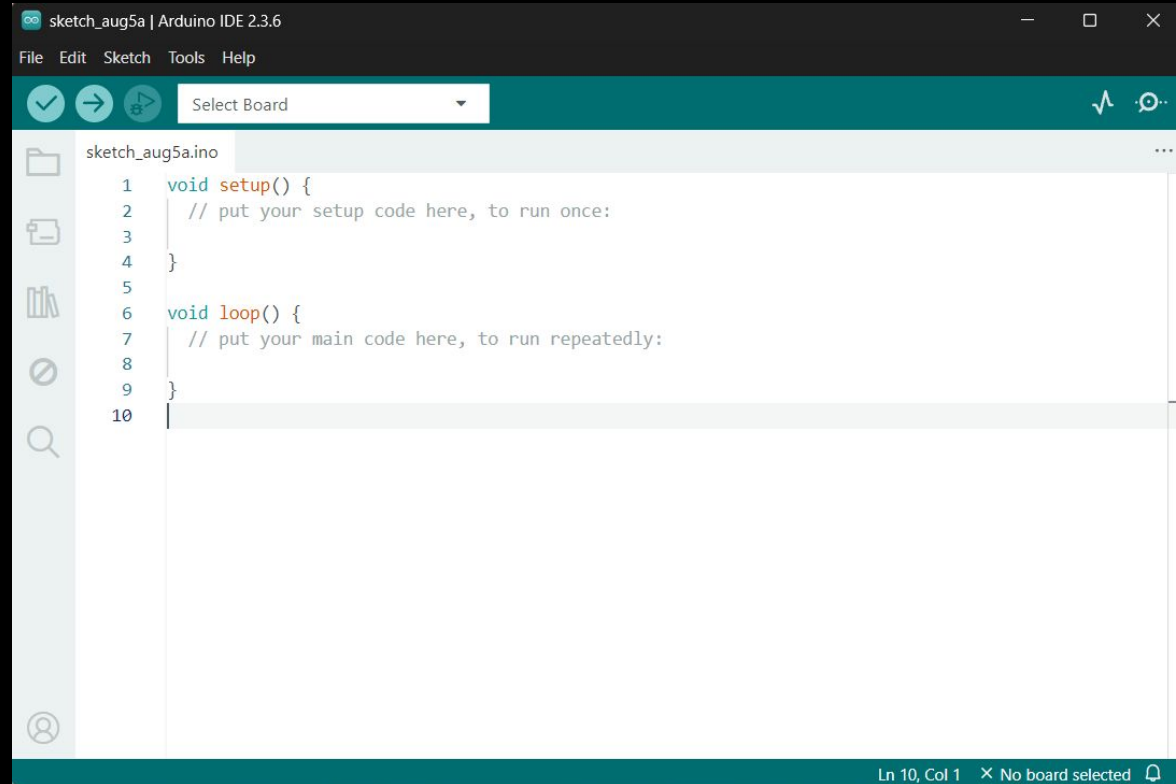


Alguns microcontroladores muitas vezes necessitam de drivers externos para que o computador possa reconhecê-los via USB (Precisa ser baixado no site do fabricante normalmente).

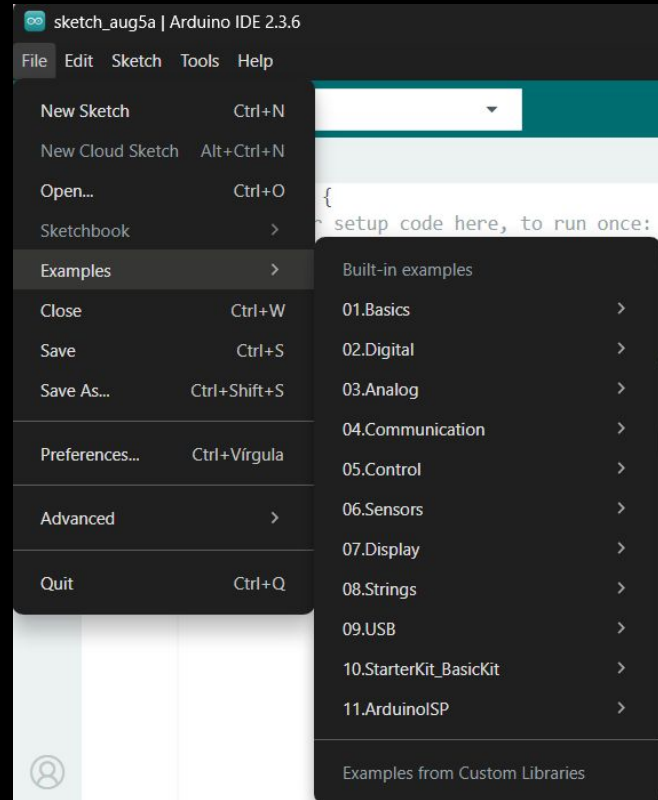
A IDE do arduino pode ser encontrada no site <https://www.arduino.cc/en/software>

A decorative graphic element in the bottom right corner consisting of several thin, curved, light blue lines.

IDE

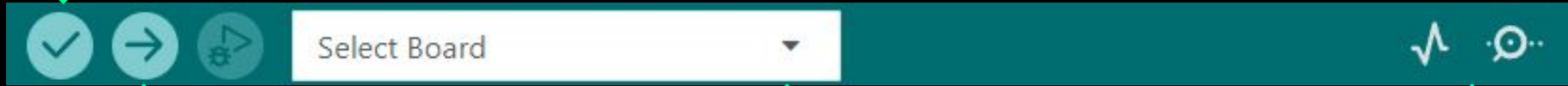


IDE - Exemplos



IDE - Exemplos

Compila o programa
(Checagem de erros)



**Transfere o programa
para o Arduino**



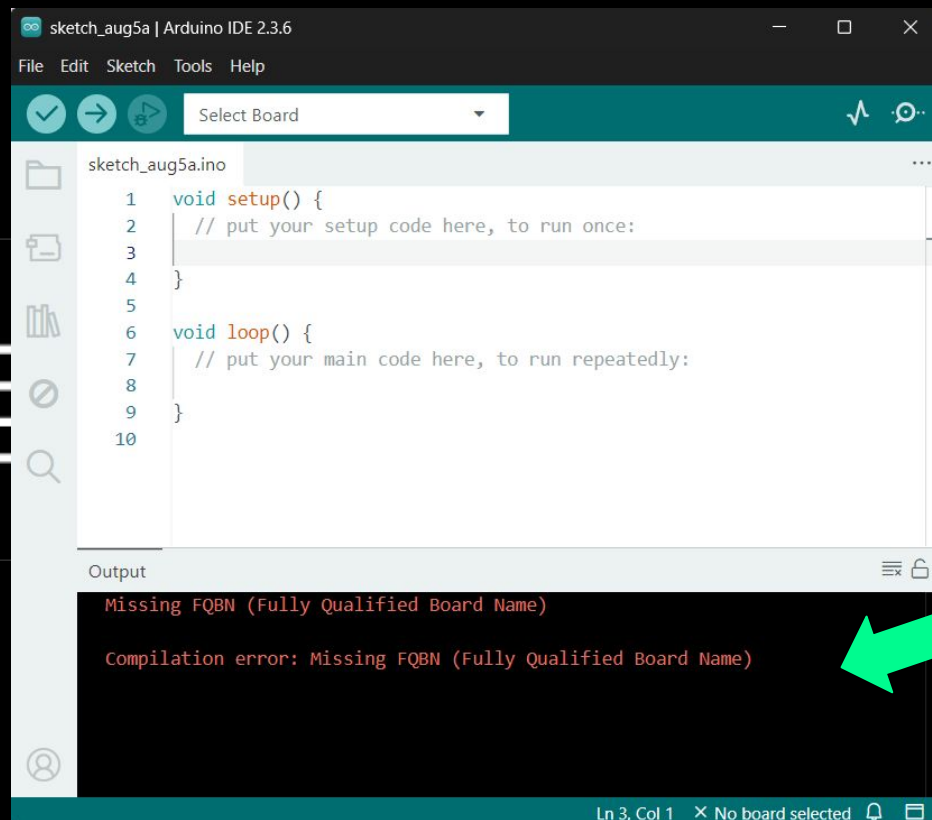
**Seleção de placas
Arduinos**
(É necessário informar
para a IDE qual é o
tipo de placa que
iremos enviar o código)



**Monitoramento e
Plotagem serial**
(Utilizado para visualizar
as informações que
estão sendo
enviadas/recebidas pelo
Arduino)



Erros



The screenshot shows the Arduino IDE 2.3.6 window. The top menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for checking, running, and uploading, along with a 'Select Board' dropdown menu. The main editor area displays a sketch named 'sketch_aug5a.ino' with the following code:

```
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
```

At the bottom, the 'Output' window shows a red error message: 'Missing FQBN (Fully Qualified Board Name)'. A red arrow points from the text on the right to this error message. The status bar at the bottom right indicates 'Ln 3, Col 1' and 'No board selected'.

Erros aparecerão aqui

(Junto do erro aparece uma mensagem de erro, indicando o que pode ser e onde aconteceu)

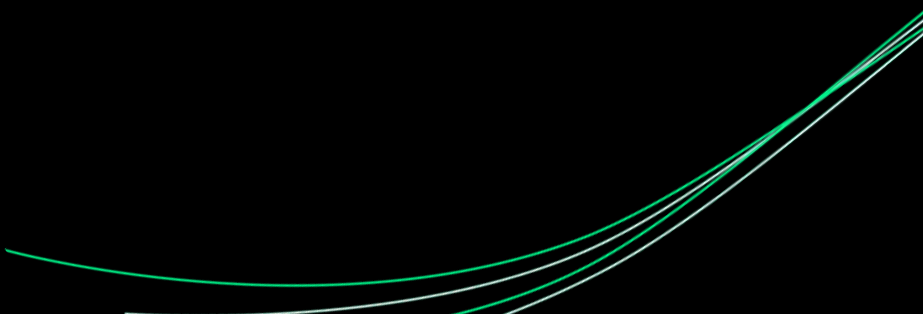


Programação

O arduino opera com duas funções principais, sendo elas: **setup** e **loop**. Elas são delimitadas pelas chaves {...}



Dentro de cada uma delas deverão conter as instruções que o arduino irá executar *ad. nauseam*

Three curved, wavy lines in light blue, dark blue, and white, sweeping from the bottom left towards the bottom right of the slide.

Programação

setup()

- **Executar uma unica vez** quando a placa é ligada ou reiniciada.
- **Objetivo:** inicializar variáveis, configurar pinos (entrada/saída) e iniciar comunicação com periféricos.



```
void setup() {  
  pinMode(13, OUTPUT); // Confirgura o pino 13 para o modo de saída.  
}
```

Programação

loop()

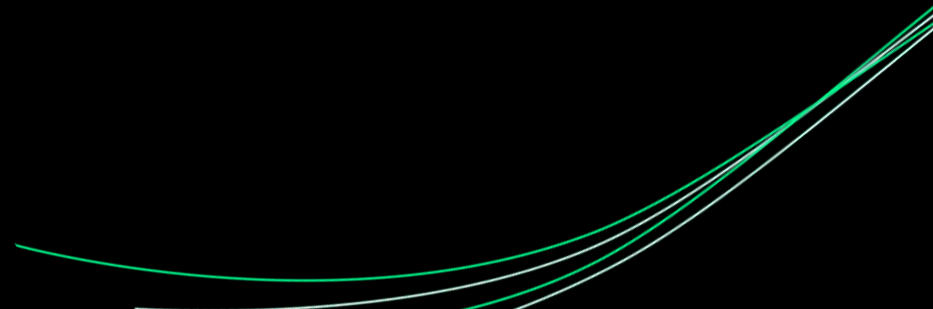
- Executada continuamente após a função setup().
- **Objetivo:** manter o funcionamento do programa, atualizando ações e leituras.



```
void loop() {  
  digitalWrite(13, HIGH); // Liga LED  
  delay(1000);             // Aguarda 1 segundo  
  digitalWrite(13, LOW);  // Desliga LED  
  delay(1000);             // Aguarda 1 segundo  
}
```



Pratica



Por onde começar

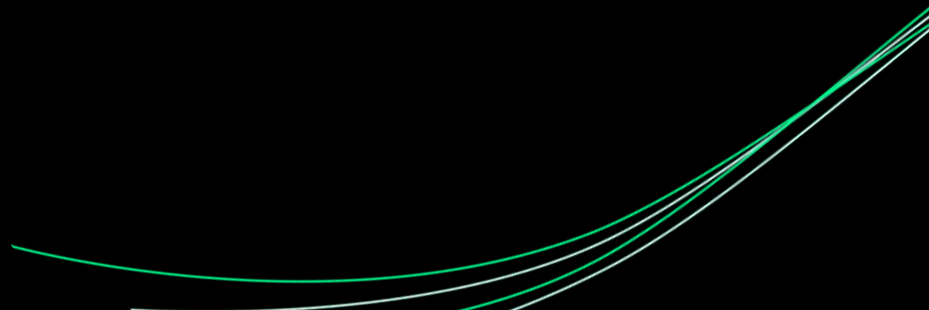


Acessem o site do <https://www.tinkercad.com> e criem sua conta, ou peçam para que o professor adicione na sala de aula.

Após estar na sala de aula, crie um novo projeto de circuitos, e aguarde até que todos estejam neste passo.



Blink





Pratica - Blink

Toda tecnologia tem seu “**Hello World**”, o Arduino não seria diferente.

Objetivo: Fazer com que um LED acenda e apague a cada 0.5s.



Circuito

Precisamos montar o seguinte circuito. Para isto precisamos dos seguintes componentes:

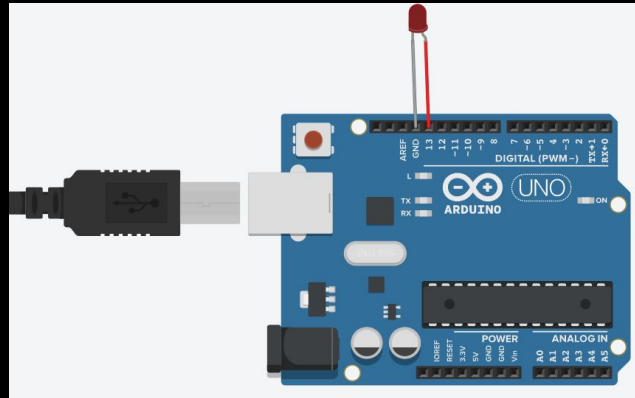


1x - LED de qualquer cor
1x - Arduino

Obs.: Não saia conectando os pinos do LED com o arduino ligado, pode queimar o LED

Perna mais curta vai no GND

Diagrama & Programação



Obs.: Conectar o LED diretamente no pino 13 faz com que ele receba 5V diretamente, gerando um aviso no Tinkercad, porém, neste contexto podemos ignorá-lo.

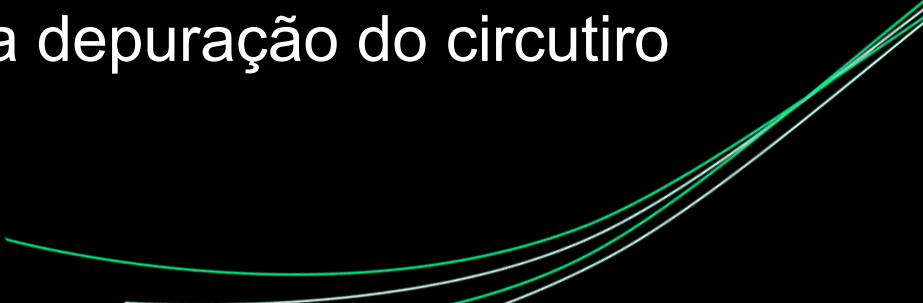




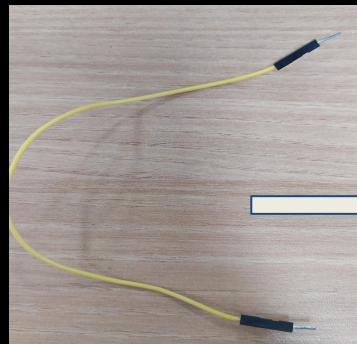
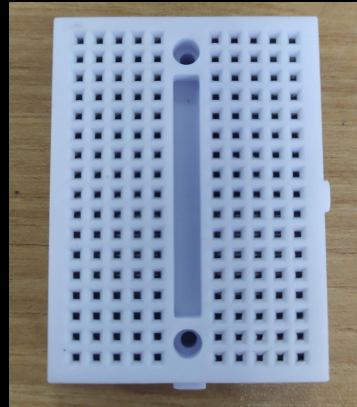
Placa de prototipação

Protoboard



- Dispositivo usado para a construção de circuitos temporários (sem soldagem)
 - Usada para testar se o circuito projetado funciona corretamente
 - Permite a depuração do circuito
- 

Protoboard



Para energizar a protoboard utilizamos **JUMPERS** (Fios de energia). Ao fazer isso a linha toda até o meio da protoboard funciona da mesma maneira que o pino do arduino está ligado ao JUMPER.

JUMPER: Fio de cobre envolto numa camada de plastico isolante. Podendo conter extremidades **macho** (“agulha” de metal) e(ou) **fêmea** (encaixe para o pino macho).



Pratica - LEDs em série

Nesta prática vamos fazer com que os LEDs sejam ligados em série, e vamos observar o que ocorre com eles.



Objetivo: Entender como funciona um circuito simples em série, e qual a implicação prática disso.

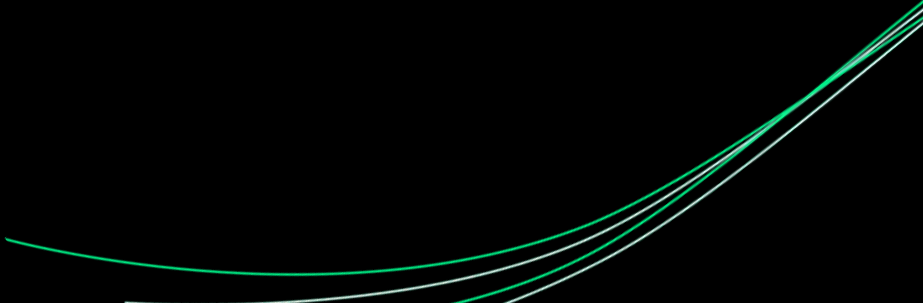
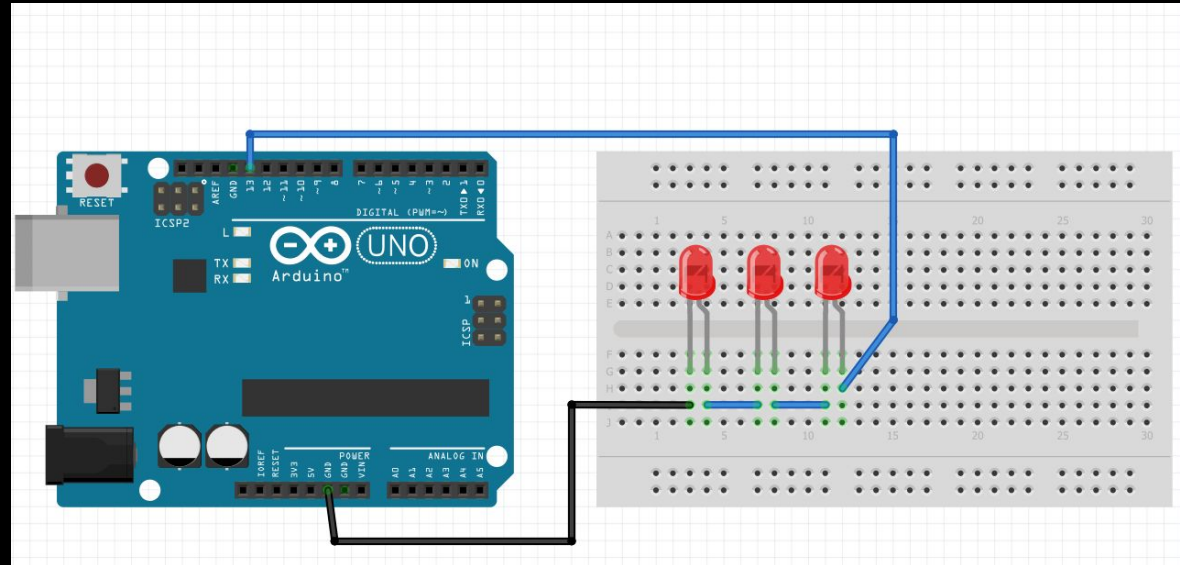


Diagrama & Programação



Obs.: A programação é a mesma que a da prática anterior

Circuito

Precisamos montar o seguinte circuito. Para isto precisamos dos seguintes componentes:



3x - LED de qualquer cor

1x - Arduino

1x - Proto board

2x - Jumper

Obs.: Não saia conectando os pinos do LED com o arduino ligado, pode queimar o LED

Perna mais curta vai no GND



Atividade

Considerando a prática de LEDs em série, agora, seu objetivo é montar um semaforo que alterna entre os sinais de siga (**Verde**), atenção (**Amarelo**) e pare (**Vermelho**).



Materiais:

- Arduino UNO R3
- 3x LEDs simples

Obs.: Lembre-se que o tempo do amarelo não é igual aos demais!

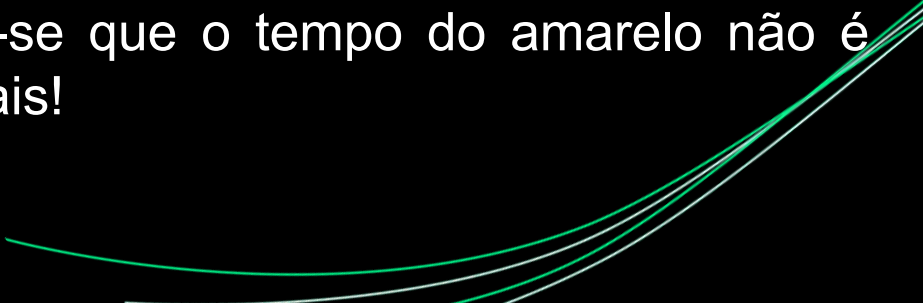
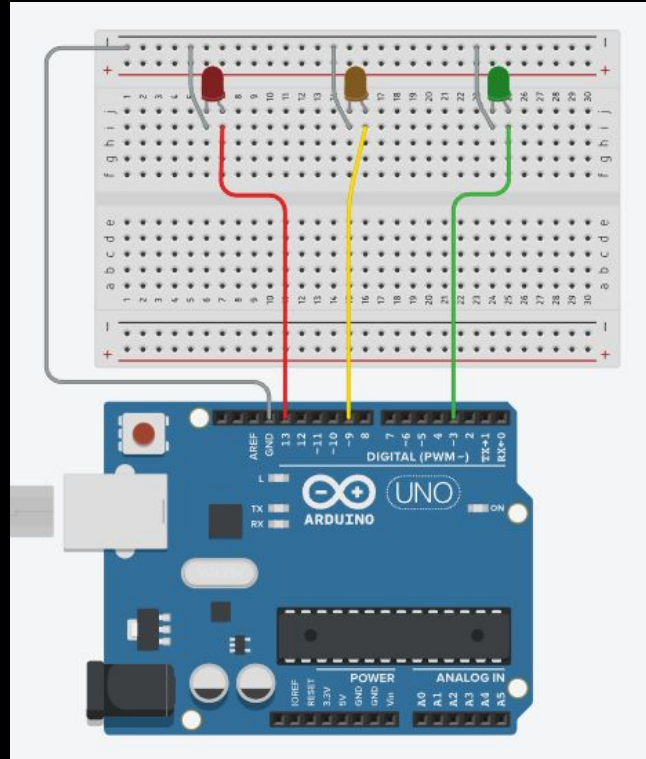


Diagrama & Programação

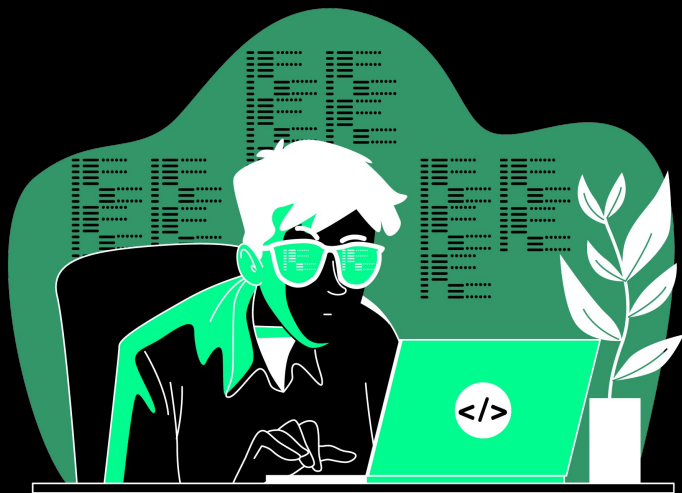


no início

```
definir pino 13 como BAIXO
definir pino 9 como BAIXO
definir pino 3 como BAIXO
```

para sempre

```
definir pino 13 como ALTO
aguardar 2 s
definir pino 13 como BAIXO
definir pino 9 como ALTO
aguardar 1 s
definir pino 9 como BAIXO
definir pino 3 como ALTO
aguardar 2 s
definir pino 3 como BAIXO
```



Obrigado!