



NIVELAMIENTO DE ROBOTICA

DIA - 1

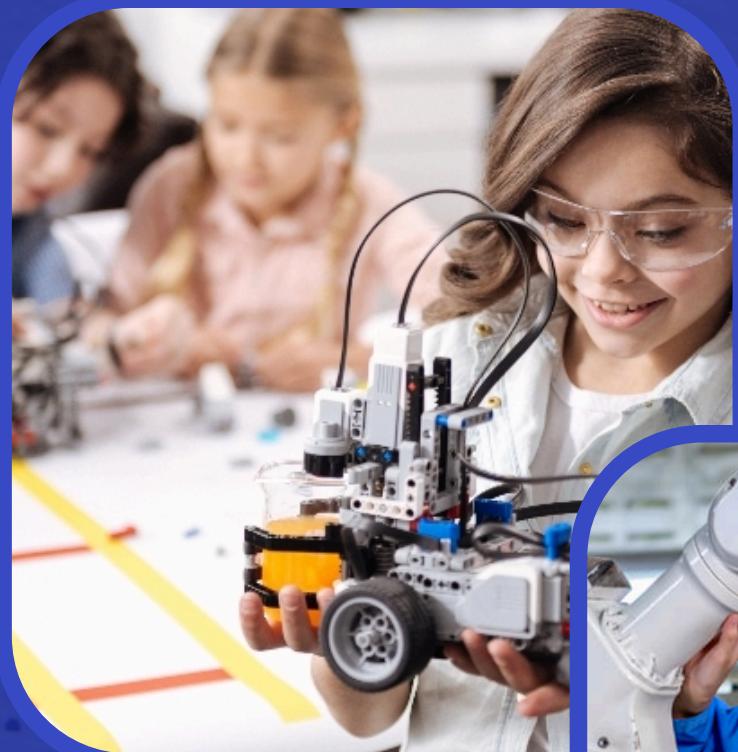
COMBU
MOKER



O QUE É ROBÓTICA?

COMBU
MOKER

CONCEITOS DA ROBÓTICA



eee

O que é Robótica?

A robótica é um campo interdisciplinar que combina engenharia, ciência da computação e outras áreas para projetar, construir, operar e aplicar robôs.

O que é um Robô?

Mecanismos automáticos que realizam trabalhos e/ou movimentos humanos

APLICAÇÕES



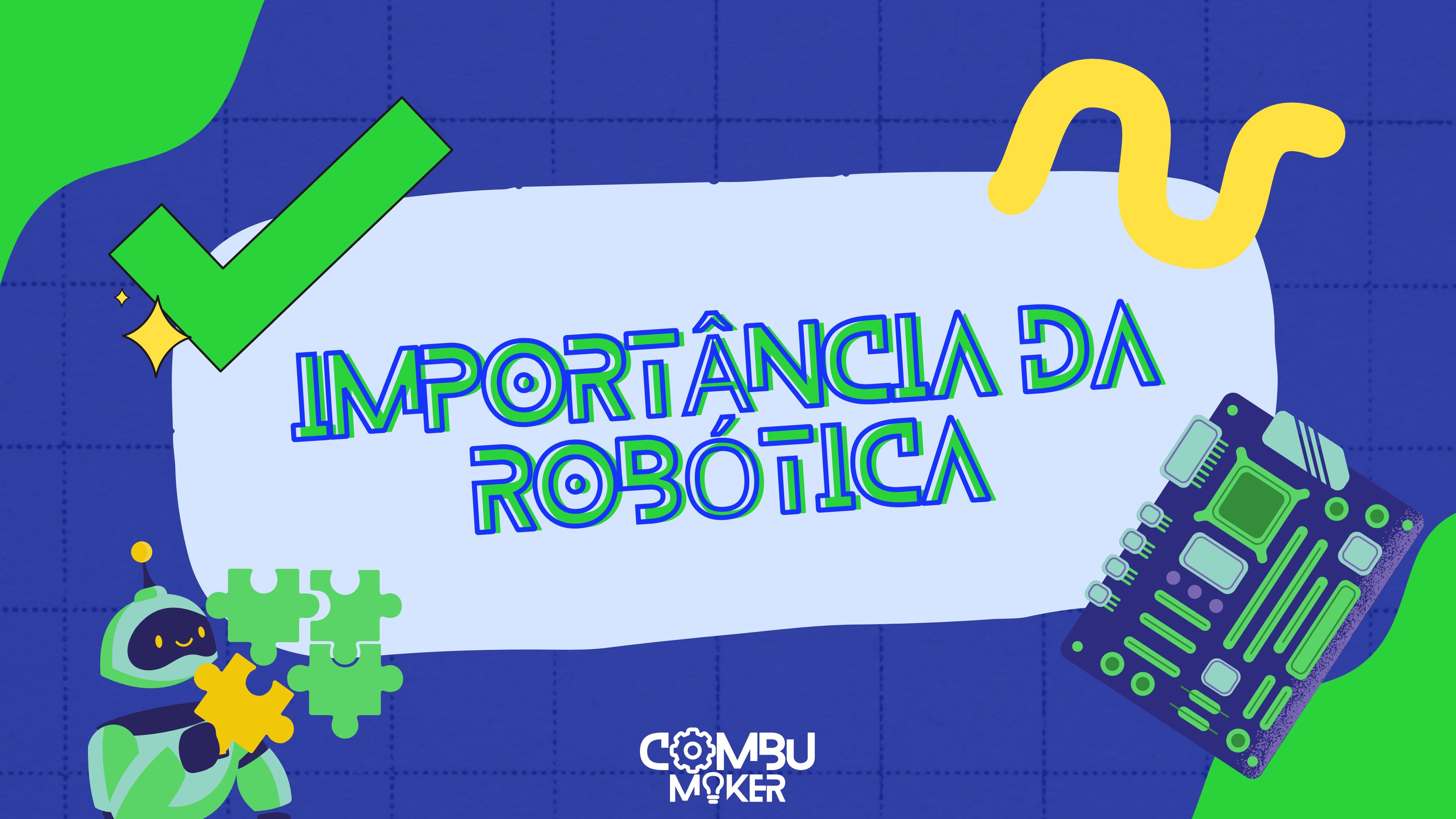
Braço robótico industrial:

Usado para soldar, pintar e montar peças em fábricas.

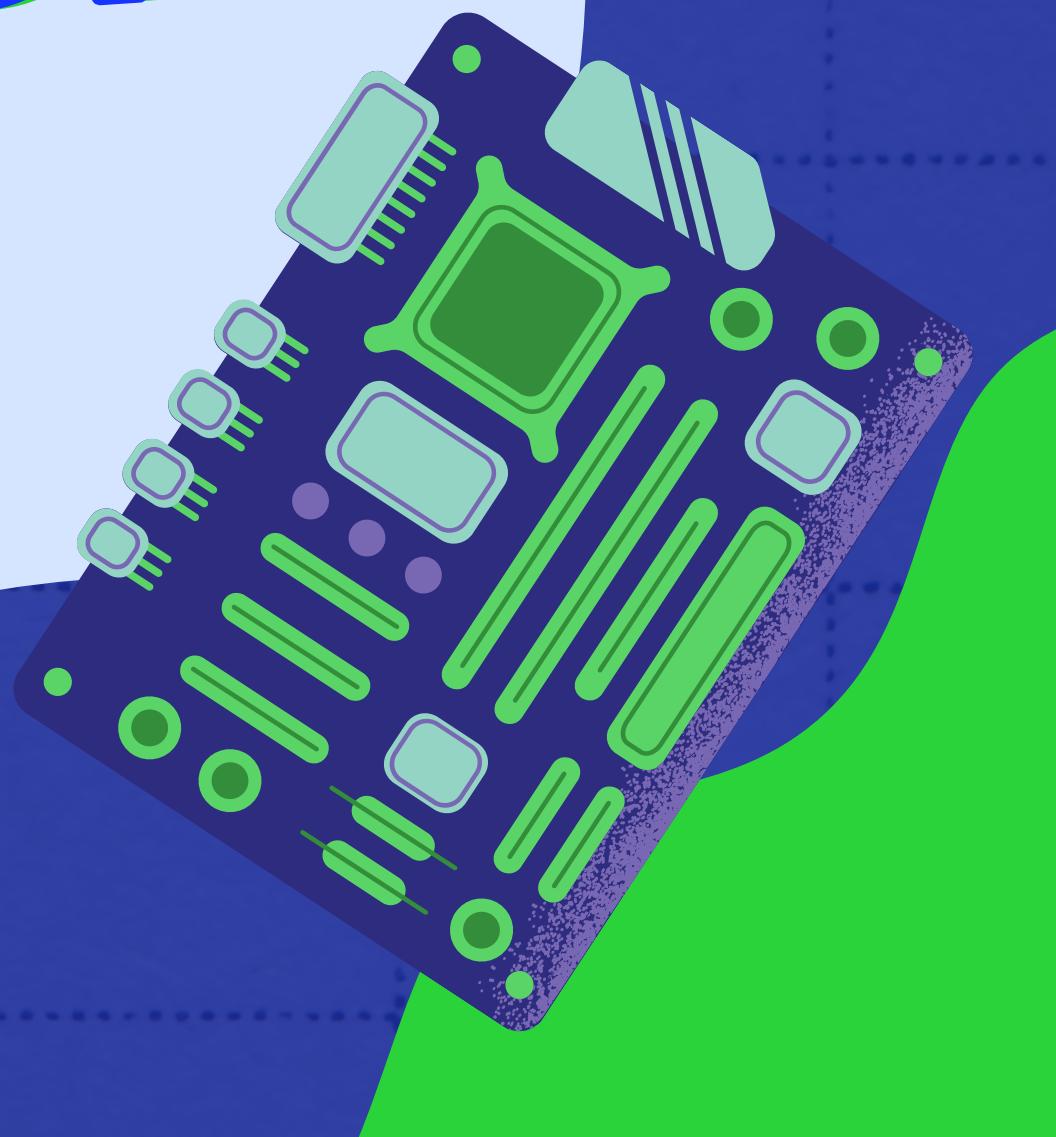
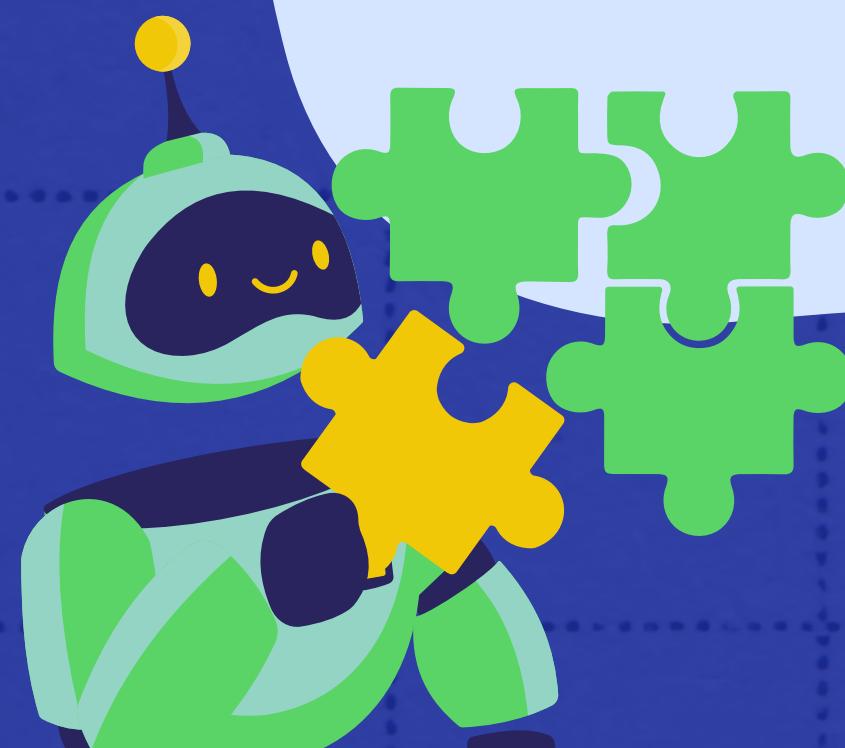
Robô cirúrgico: Usado para realizar cirurgias com maior precisão e menos invasivas.

Drone: Usado para entrega de produtos, mapeamento e fotografia aérea.

Carro autônomo: Um carro que pode dirigir sozinho, sem a necessidade de um motorista humano.

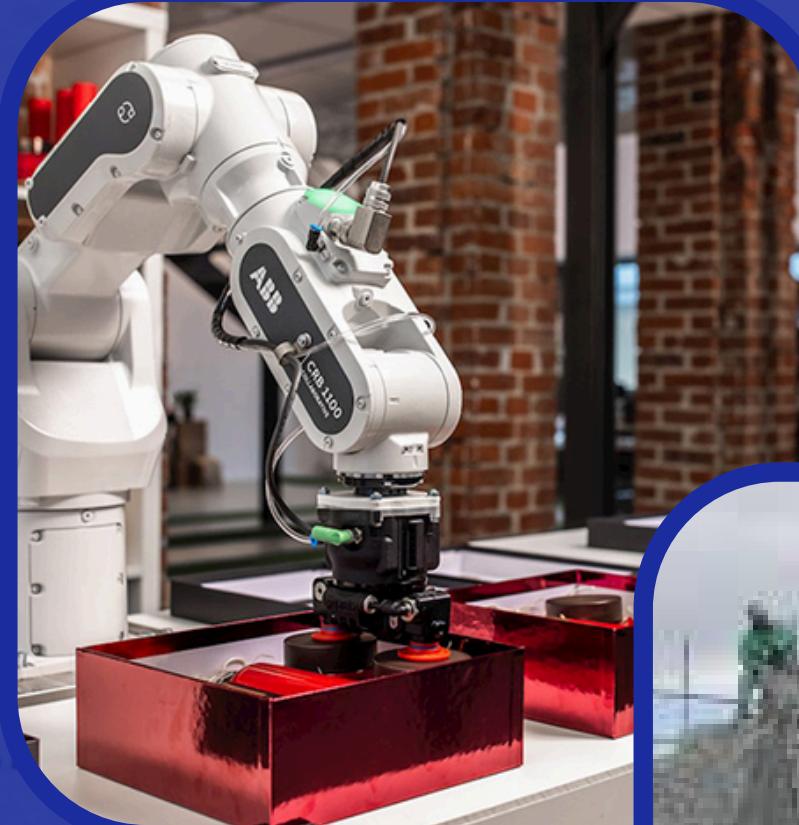


IMPORTÂNCIA DA ROBÓTICA



IMPORTÂNCIA DA ROBÓTICA

eee



Produtividade

A robótica automatiza tarefas repetitivas e perigosas, aumentando a produtividade e a eficiência das linhas de produção. Isso significa maior volume de produtos com menor tempo e custo, além de garantir a qualidade e a padronização dos resultados.

IMPORTÂNCIA DA ROBÓTICA



Assistência a idosos e pessoas com deficiência

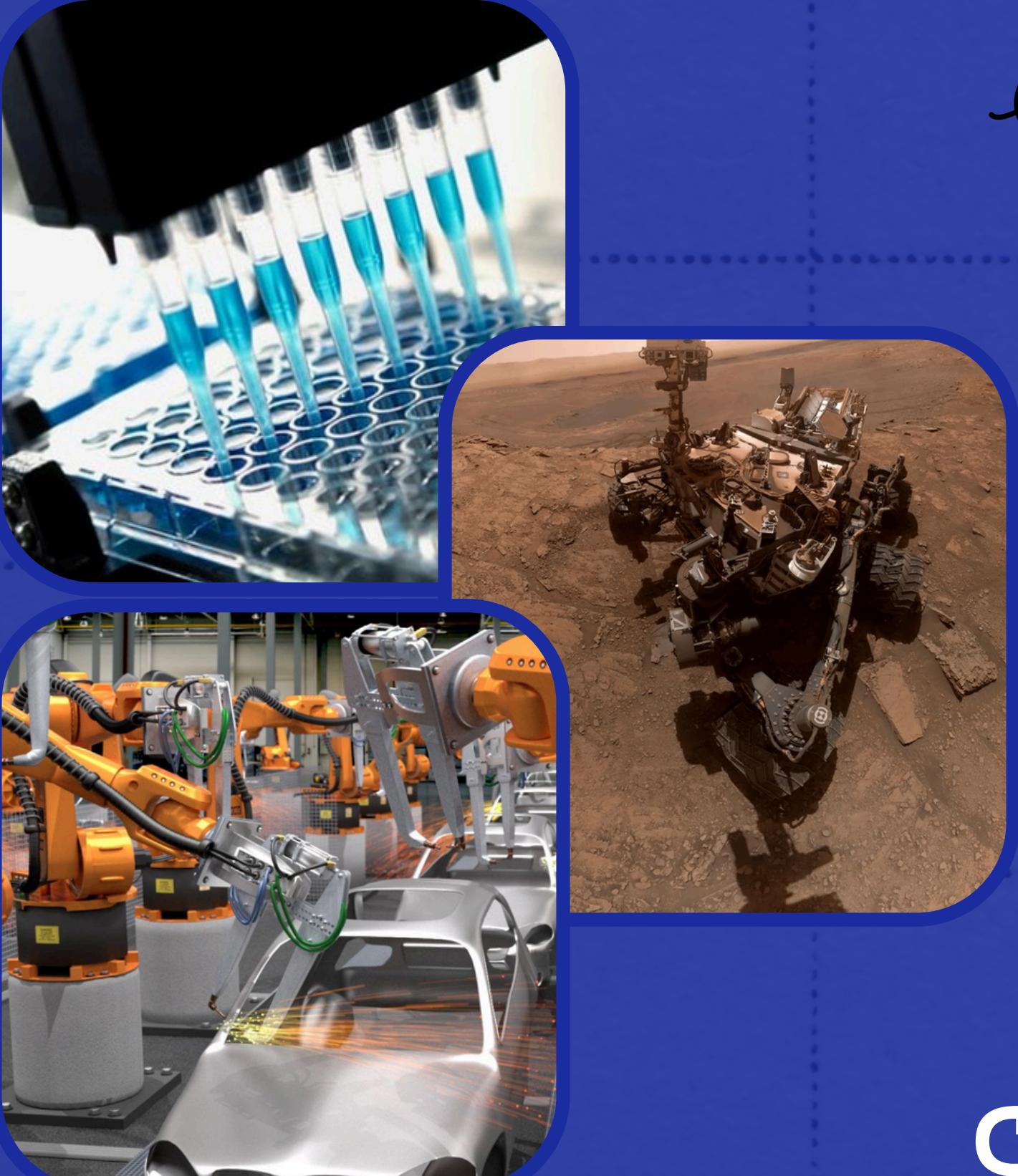
Robôs podem ser usados para auxiliar no cuidado de idosos e pessoas com deficiência, proporcionando-lhes companhia, segurança e apoio nas atividades diárias. Isso permite que essas pessoas vivam com mais independência e autonomia, reduzindo a necessidade de internação em asilos ou casas de repouso.

IMPORTÂNCIA DA ROBÓTICA

eee

Segurança

Os robôs assumem tarefas perigosas e insalubres, protegendo os trabalhadores de acidentes e doenças ocupacionais. Isso contribui para um ambiente de trabalho mais seguro e saudável, reduzindo custos com seguros e indenizações.

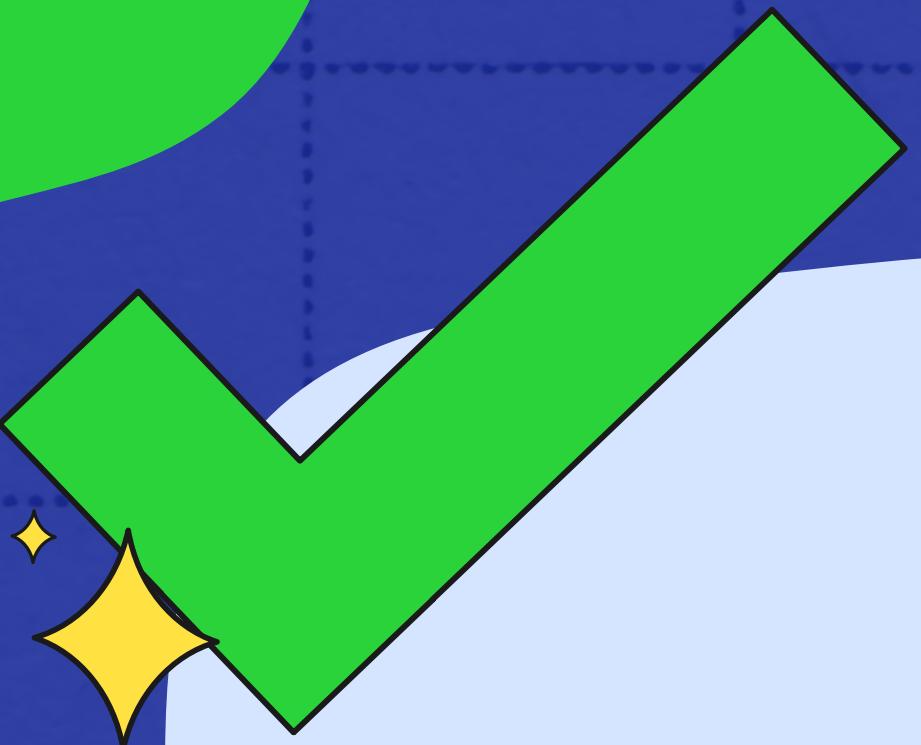


IMPORÂNCIA DA ROBÓTICA

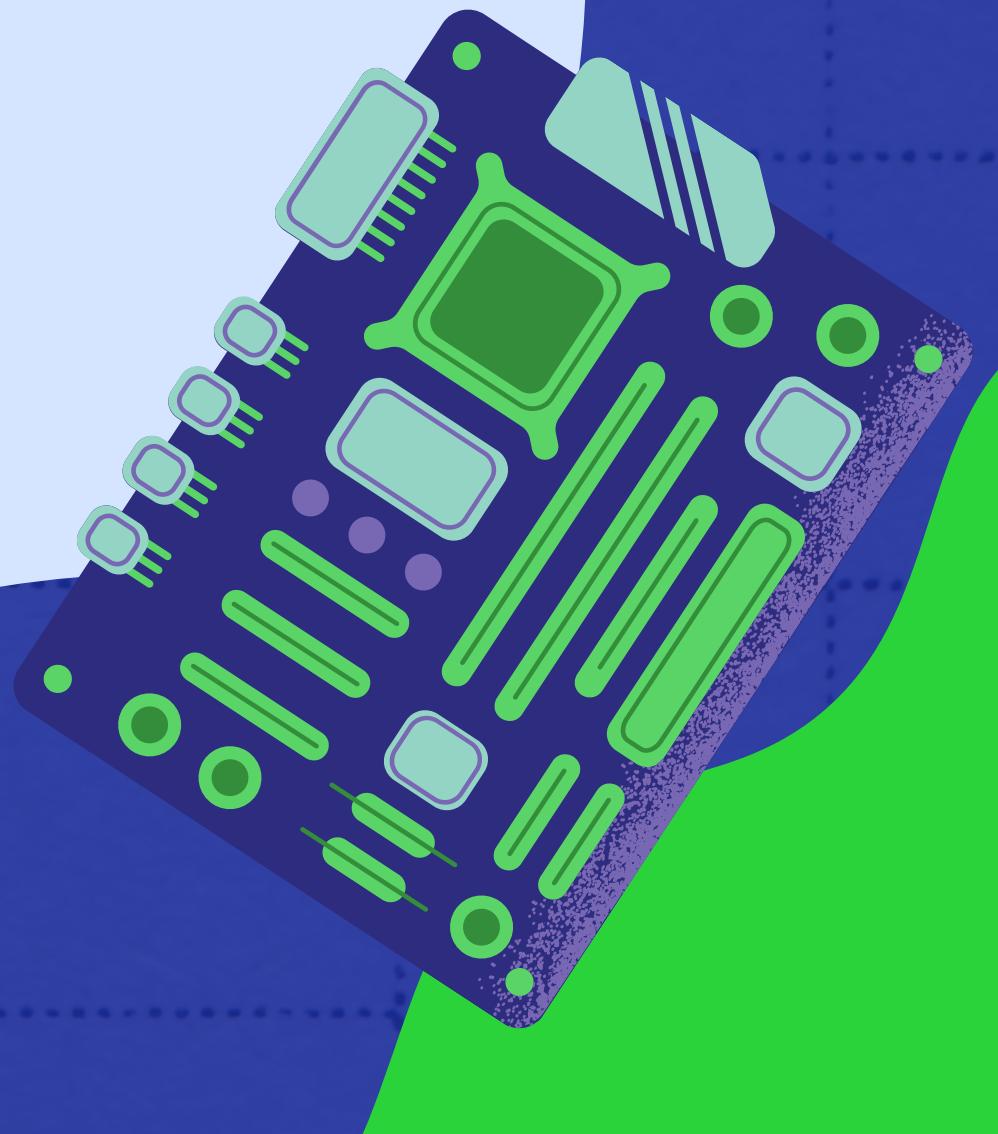
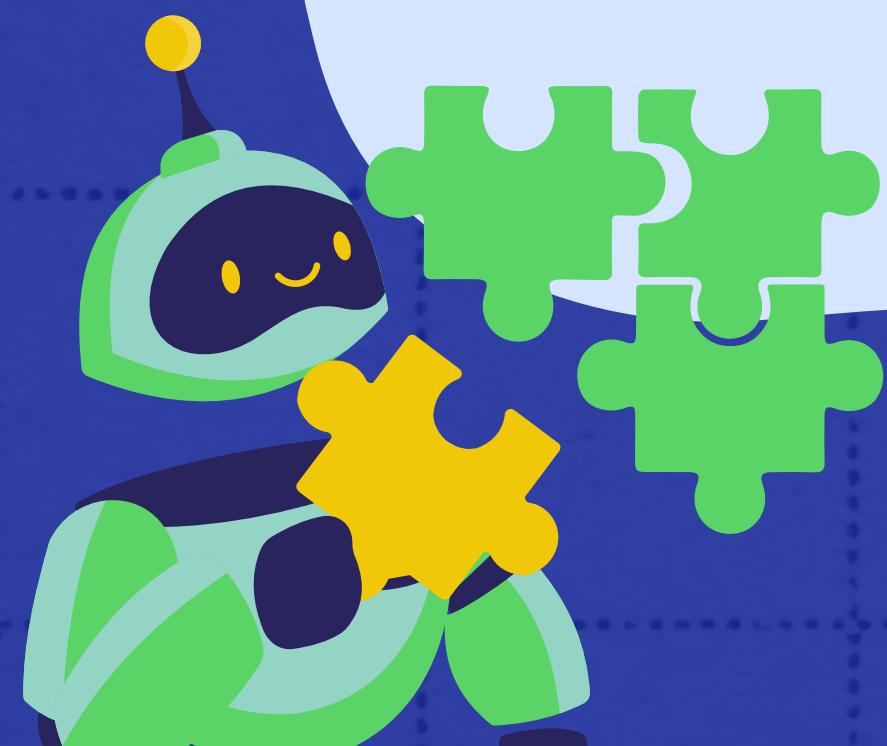


Desenvolvimento de habilidades

A robótica educacional ajuda os alunos a desenvolver habilidades importantes para o século 21, como trabalho em equipe, resolução de problemas, criatividade e pensamento crítico. Isso os torna mais preparados para o mercado de trabalho e para os desafios do futuro.



ARDUINO

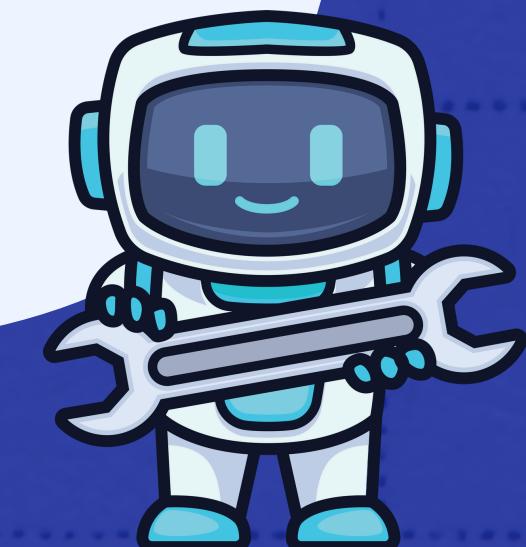


COMBU
MOKER

CONTEXTO HISTÓRICO



- Criado em 2005, com o objetivo de criar um dispositivo barato, funcional e fácil de programar;
- O Uno também introduziu o uso de um chip USB-to-Serial incorporado para facilitar a programação e a comunicação com o computador, tornando-o ainda mais amigável para iniciantes.

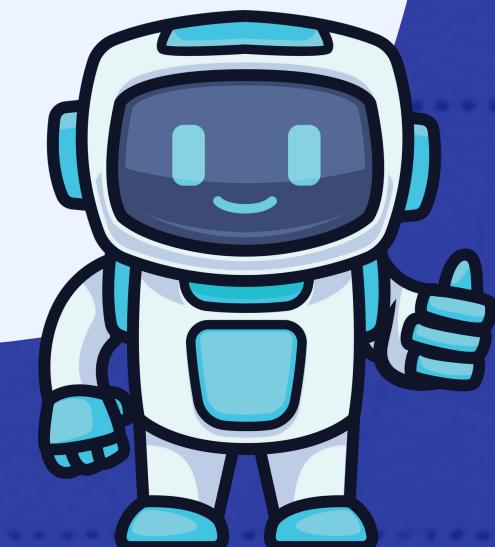


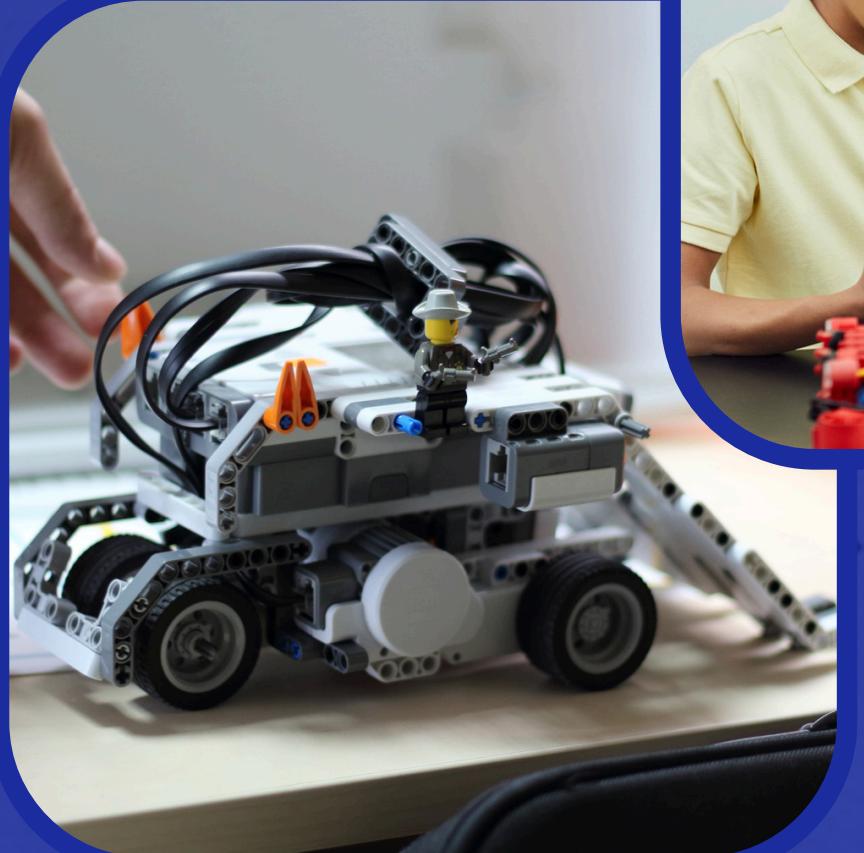
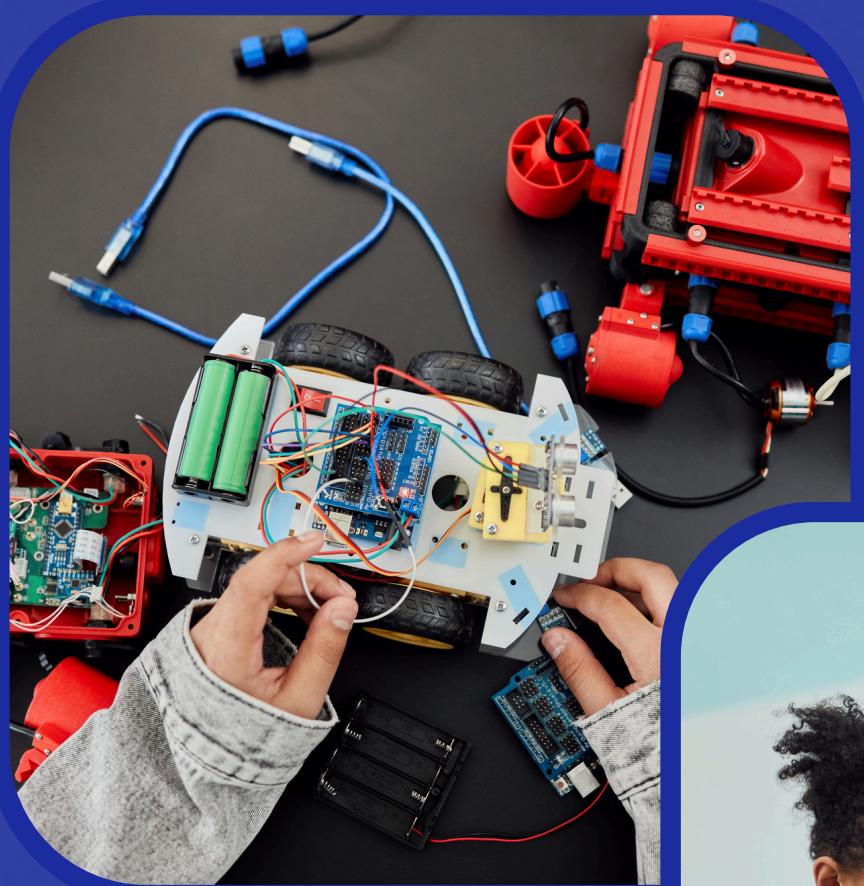
O ARDUINO



O Arduino é uma plataforma de prototipagem eletrônica de código aberto, composta por:

- Hardware
- Software
- Fácil de usar
- Versátil

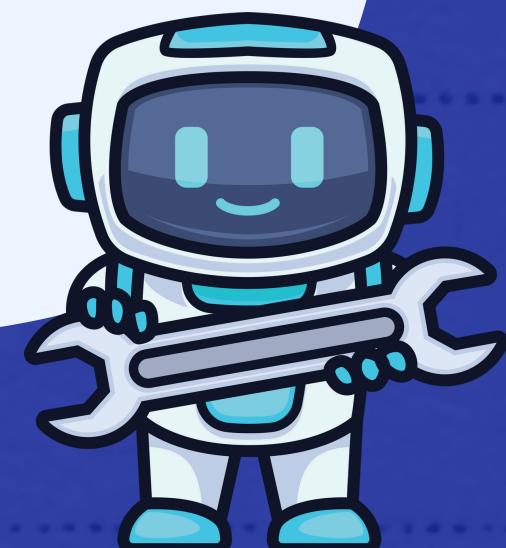




ARDUINO

eeeel

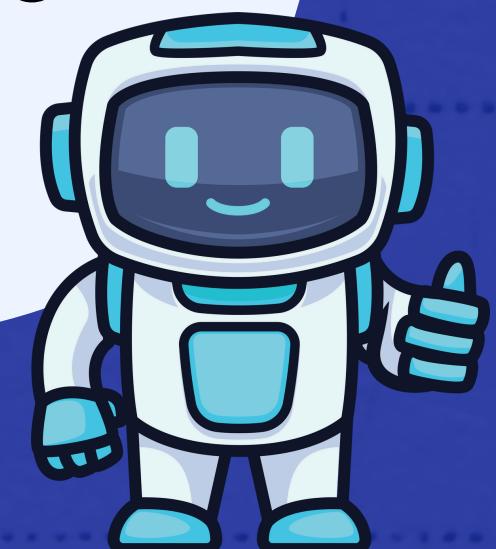
- Hardware:
 - Placa Arduino com microcontrolador;
 - Componentes eletrônicos (sensores, LEDs, motor, etc).
- Software:
 - IDE Arduino para escrever e carregar programas;
 - Linguagem C/C++.

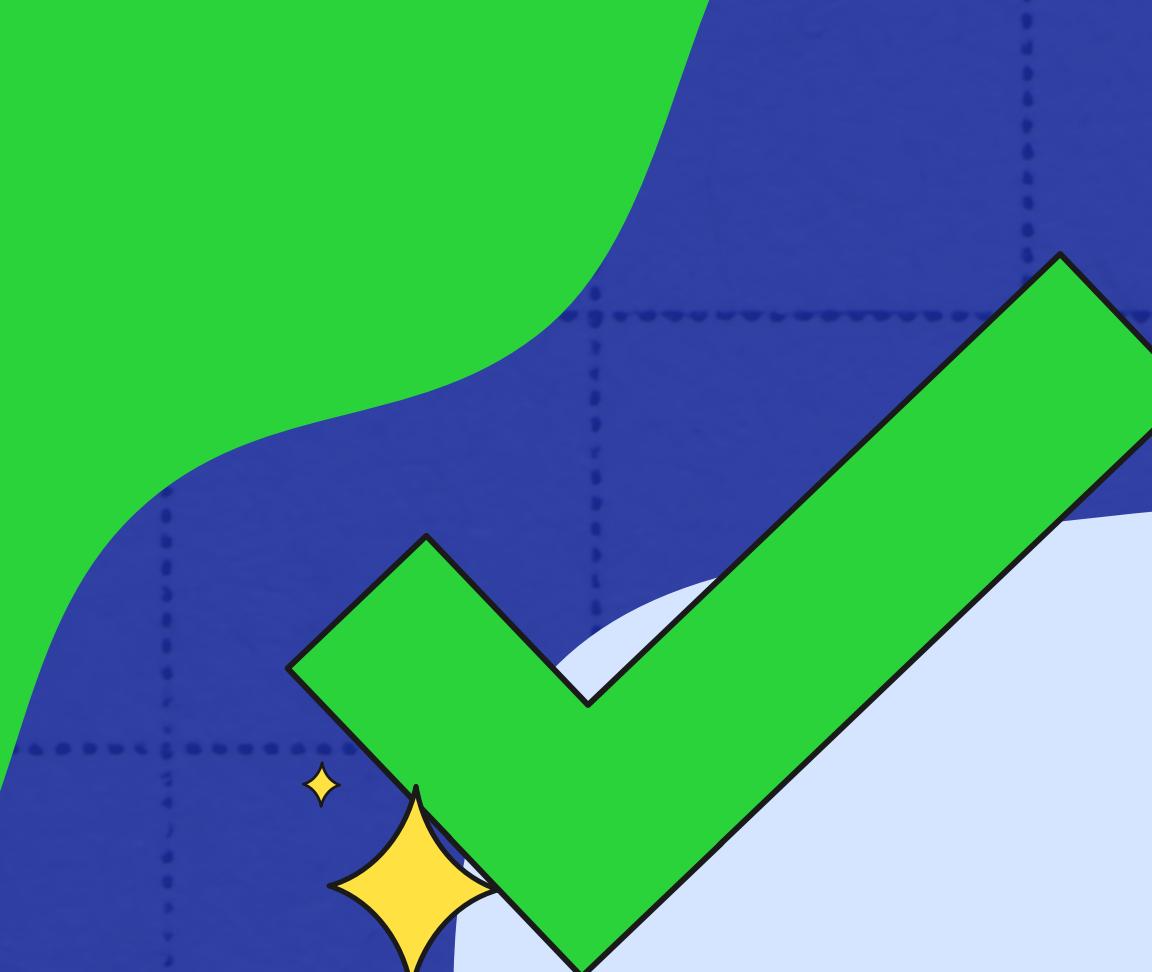


O ARDUINO

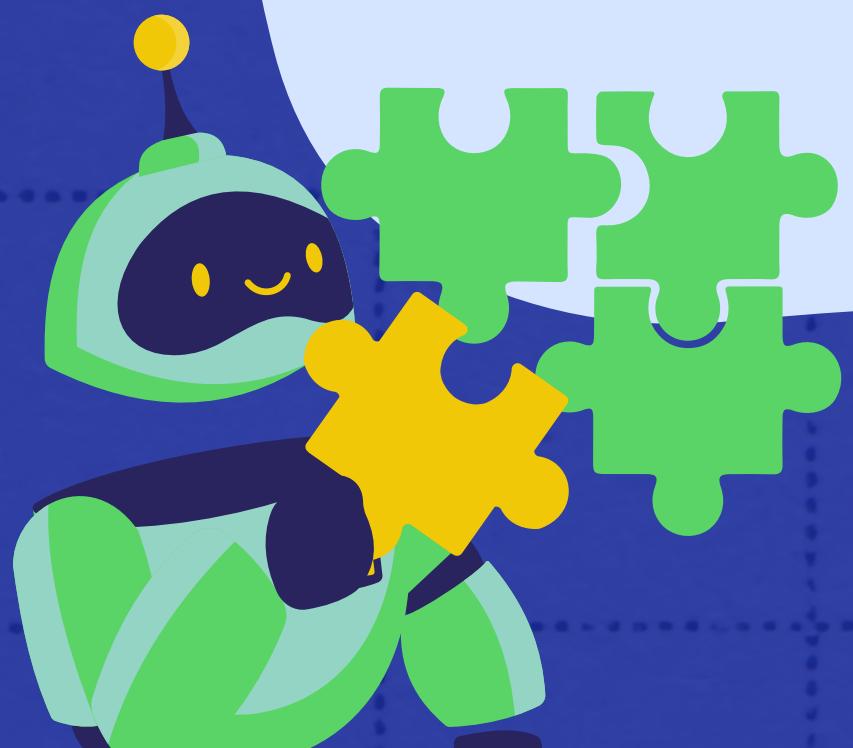


- Fácil de Usar:
 - A plataforma Arduino é amigável e fácil de aprender, mesmo para iniciantes em eletrônica.
- Versátil:
 - Pode ser usada para criar uma grande variedade de projetos, devido sua flexibilidade.



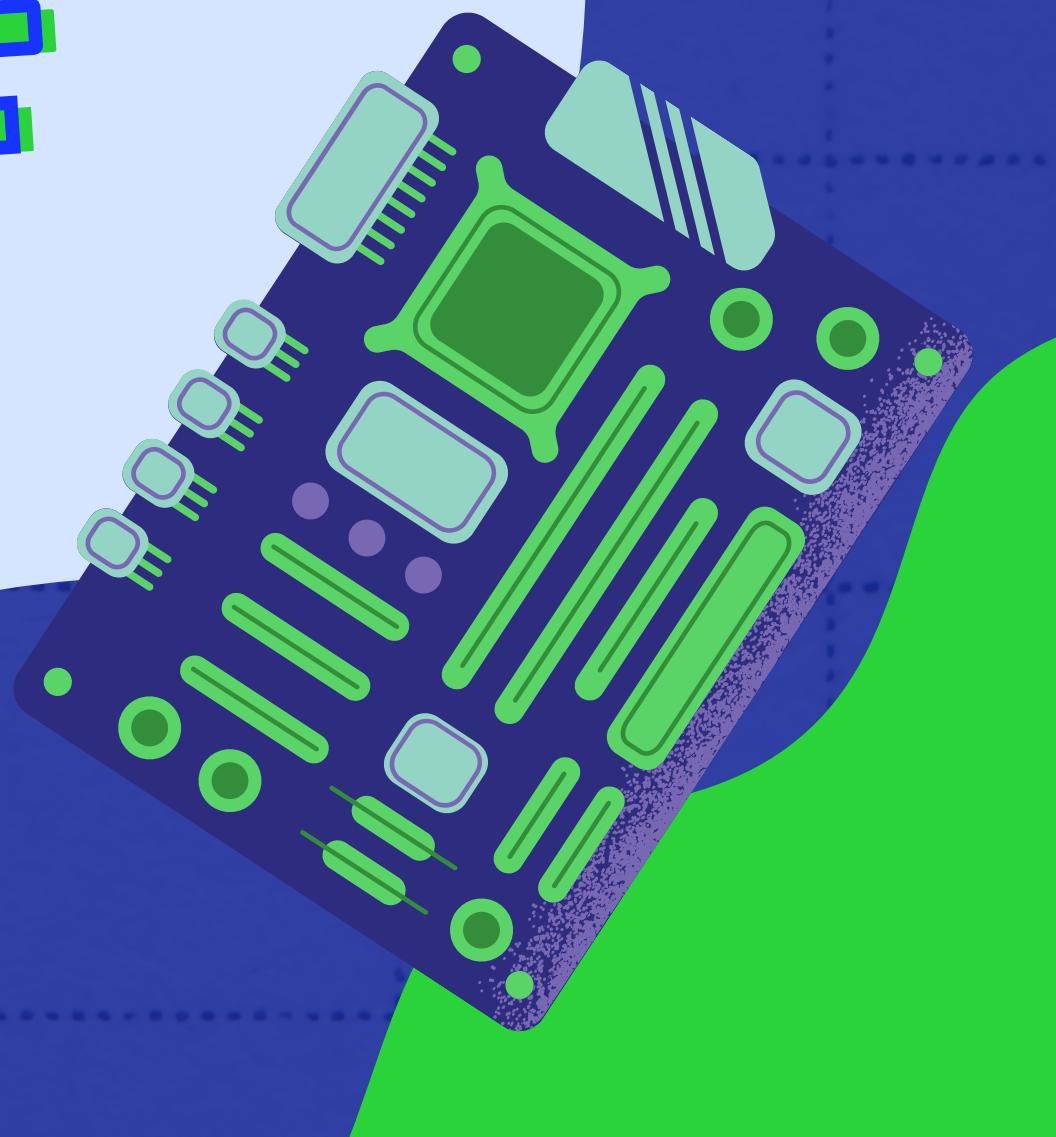


ARDUINO IDE



COMBU
MOKER

-∞+
ARDUINO



ARDUINO IDE

O Arduino IDE, ou Ambiente de Desenvolvimento Integrado Arduino, é um software fundamental para dar vida aos seus projetos eletrônicos com a plataforma Arduino. Ele oferece diversos recursos que facilitam a escrita, compilação e envio de código para as placas Arduino.



ARDUINO IDE

Funcionalidades:

- Editor de código: Permite escrever e editar código em C/C++ com destaque de sintaxe e autocompletar para facilitar a programação.
- Compilador: Converte o código escrito em linguagem C/C++ para o formato que o microcontrolador da placa Arduino pode entender.
- Monitor serial: Permite monitorar a comunicação serial entre a placa Arduino e o computador, útil para depuração e visualização de dados.

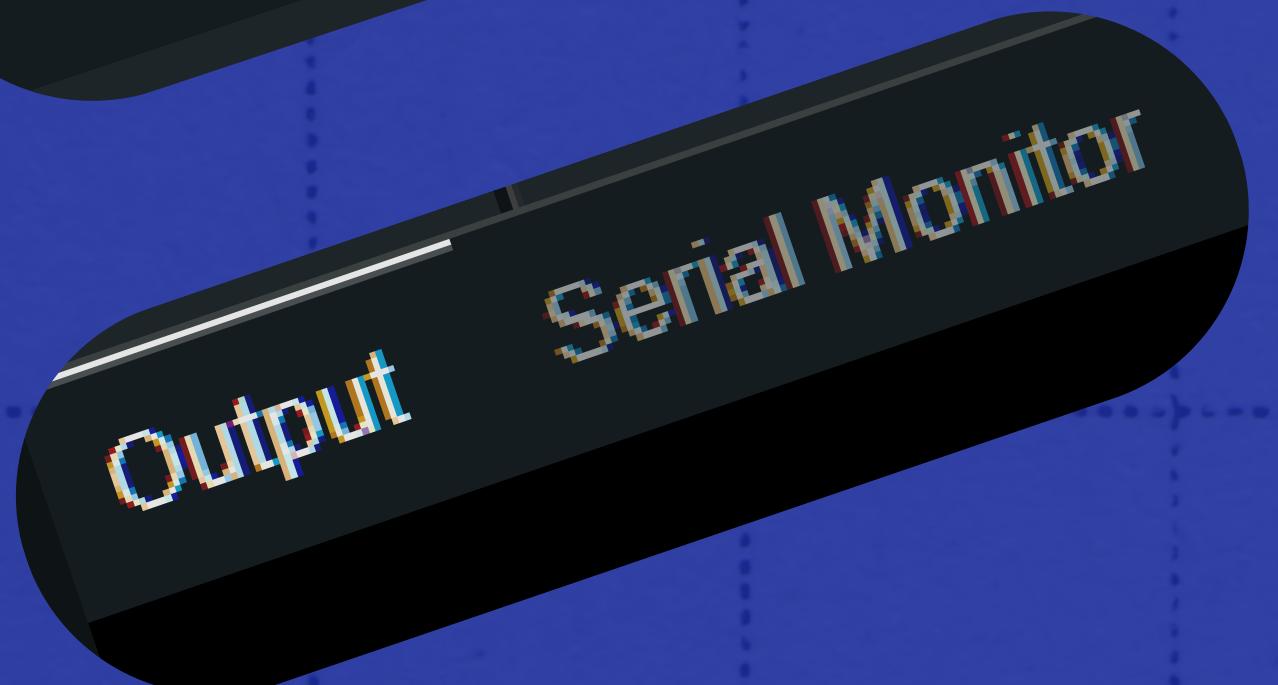
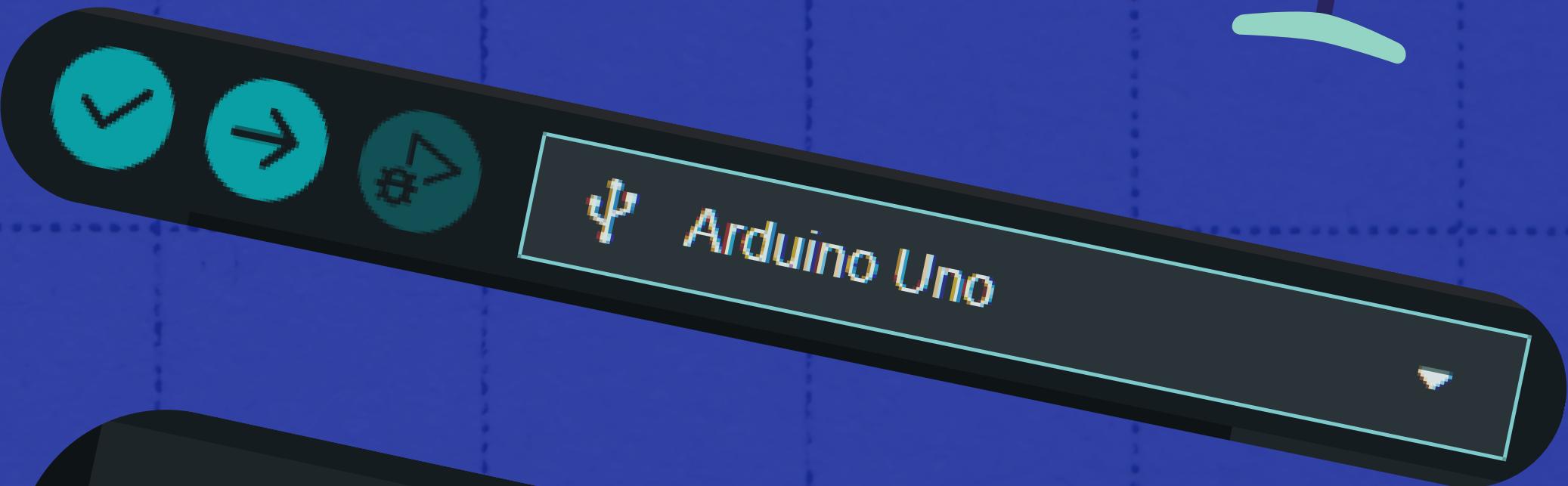
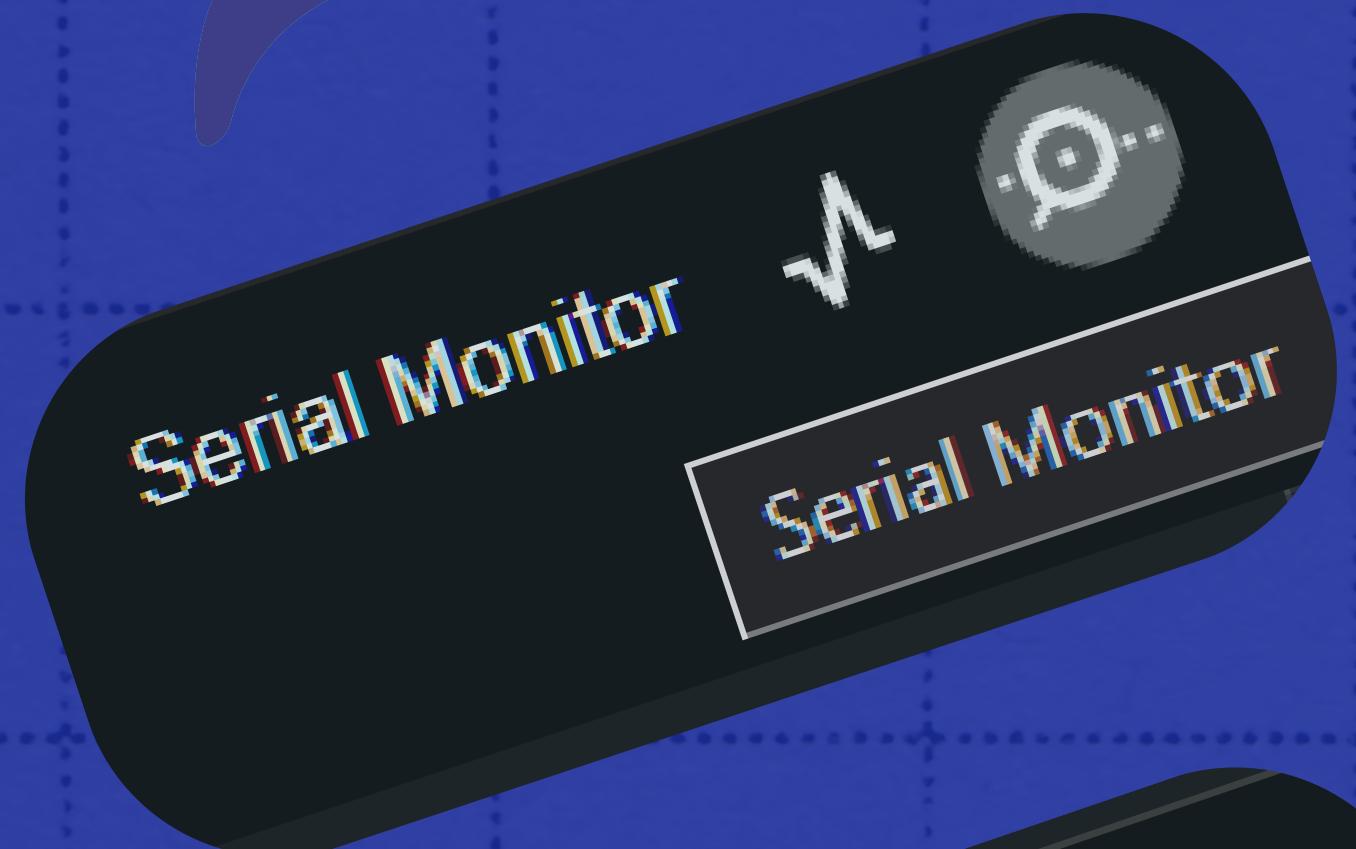
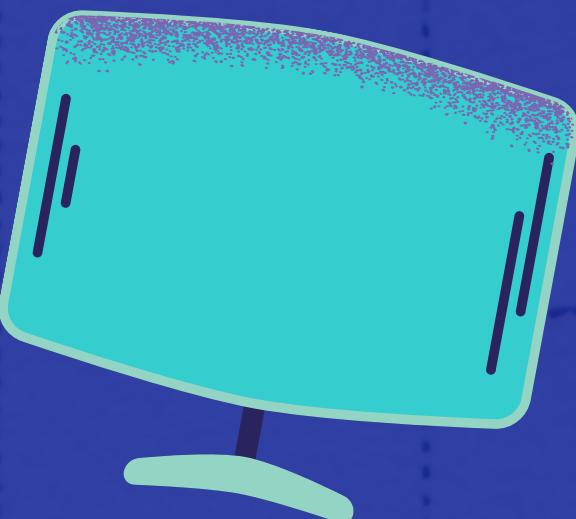
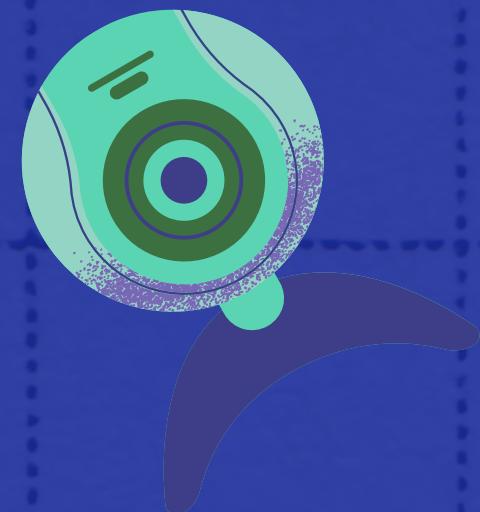
ARDUINO IDE

Gerenciador de placas: Permite instalar e gerenciar diferentes placas Arduino, incluindo modelos oficiais e de terceiros.

- Biblioteca de exemplos: Inclui diversos exemplos de código para diferentes tipos de projetos, como piscar LEDs, controlar motores, ler sensores e muito mais.

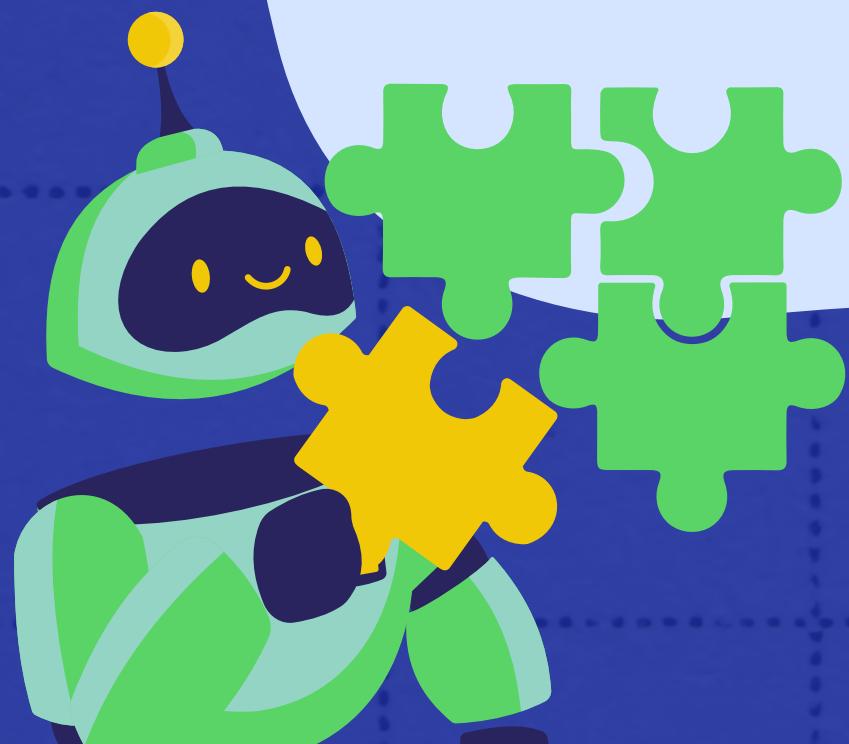


PYTHON

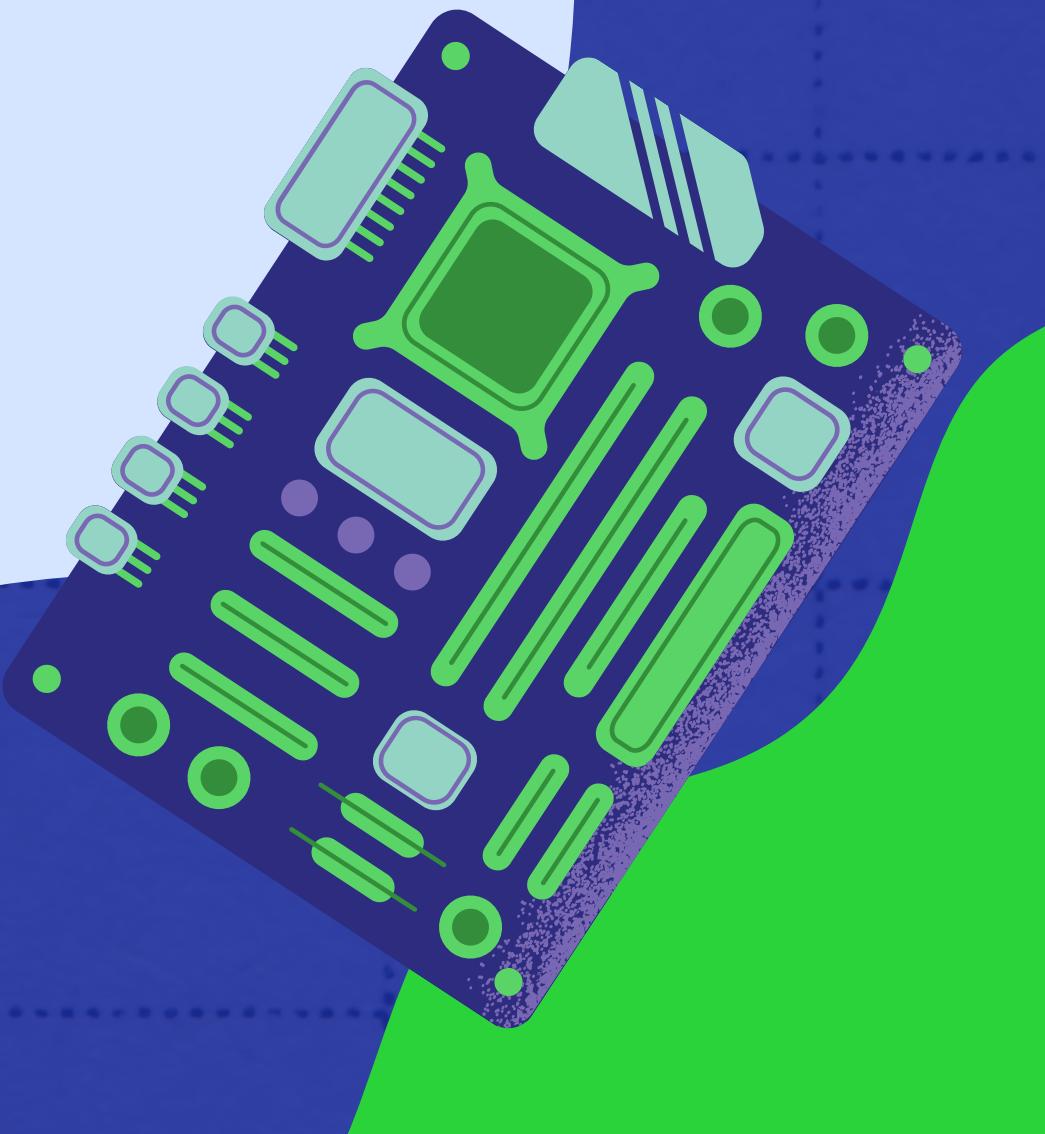




LINGUAGEM ARDUINO



COMBU
MOKER



LINGUAGEM ARDUINO

A linguagem de programação utilizada no Arduino é baseada na linguagem C/C++, embora seja simplificada e adaptada para facilitar o desenvolvimento de projetos de eletrônica, especialmente para iniciantes. Esta linguagem é comumente referida como "Arduino Sketch" ou "Código Arduino".



LINGUAGEM ARDUINO

Setup:

- **Execução:** É executada apenas uma vez, quando o Arduino liga ou reinicia.
- **Objetivo:** Configurar o hardware do Arduino, como:
 - Definir pinos como entrada ou saída
 - Inicializar bibliotecas
 - Configurar comunicação serial
 - Definir valores iniciais para variáveis



LINGUAGEM ARDUINO

Loop:

- **Execução:** É executada continuamente, em um loop infinito, enquanto o Arduino estiver ligado.
- **Objetivo:** Conter o código que define o comportamento do seu programa, como:
 - Ler dados de sensores
 - Controlar LEDs, motores e outros dispositivos
 - Fazer cálculos
 - Realizar comunicação serial



TIPOS DE DADOS

TIPOS DADOS

int: Um tipo de dado inteiro que armazena valores numéricos inteiros;

float: Um tipo de dado de ponto flutuante usado para representar números reais (números com casas decimais);

Boolean: Um tipo de dado que armazena valores lógicos, ou seja, true (verdadeiro) ou false (falso);

TIPOS DADOS

Char: É a unidade básica de texto em um sistema de codificação. Por exemplo, 'a', 'B', 'í', '!'.
eee

String: É uma sequência de caracteres. Uma coleção ordenada de caracteres que formam uma unidade de texto. Por exemplo, "Olá, mundo!".



OPERADORES

OPERADORES ARITMÉTICOS

- % (resto);
- * (multiplicação);
- + (adição);
- (subtração);
- / (divisão);
- = (operador de atribuição).



OPERADORES COMPARAÇÃO

- \neq (diferente de)
- $<$ (menor que)
- \leq (menor que ou igual a)
- \equiv (igual a)
- $>$ (maior que)
- \geq (maior que ou igual a)

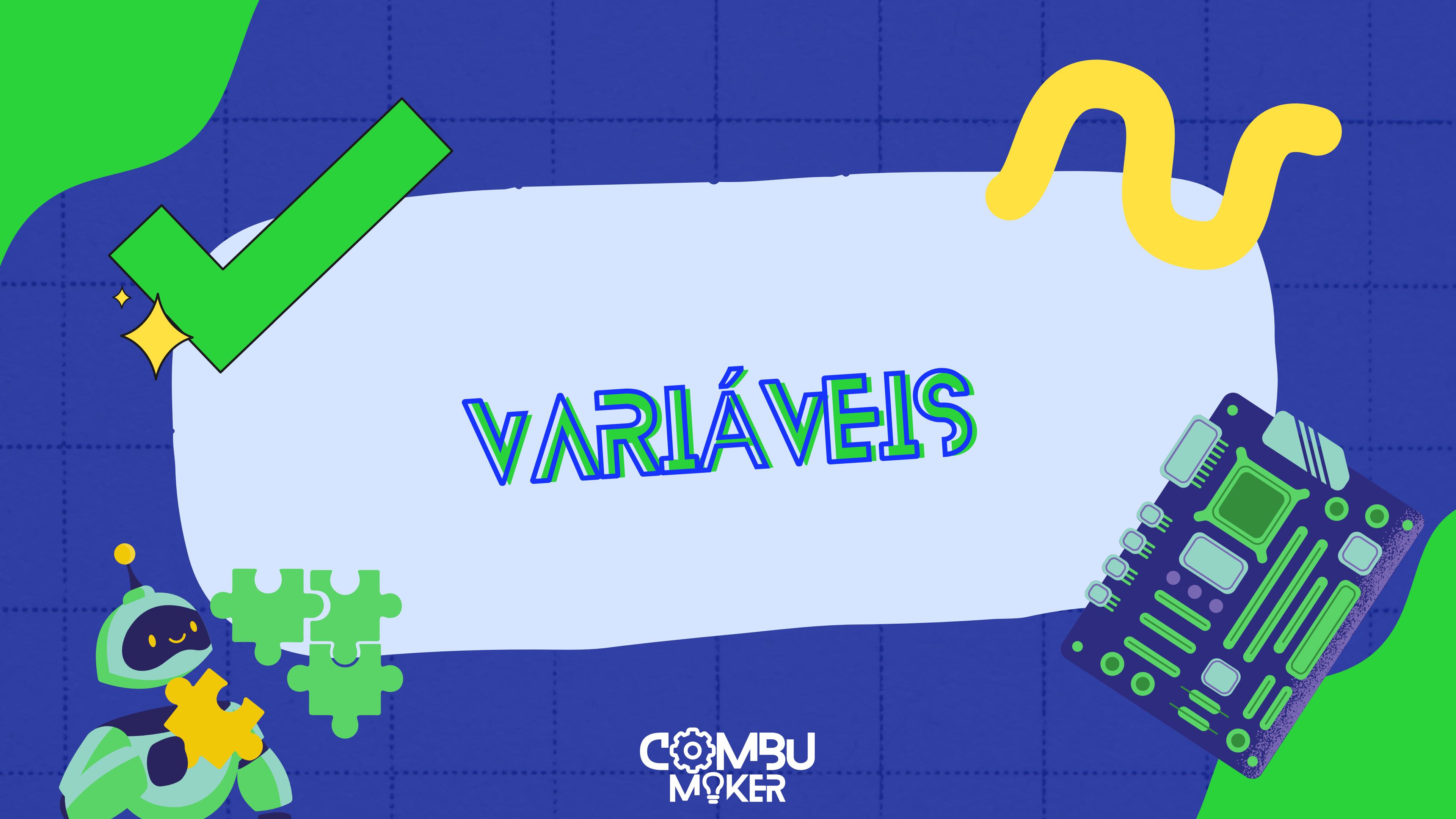
OPERADORES BOOLEANOS

! (NÃO lógico)

&& (E lógico)

|| (OU lógico)





VARIÁVEIS

COMBU
MOKER

VARIÁVEIS

Variáveis em Arduino funcionam de maneira semelhante às variáveis em outras linguagens de programação, como C/C++. Elas são utilizadas para armazenar dados temporários durante a execução do programa. Existem diferentes tipos de variáveis em Arduino, cada uma com seu próprio tamanho e faixa de valores. Aqui estão os tipos de variáveis mais comuns em Arduino:



VARIAVEIS



```
1 const int buttonPin = 2;  
2 const int ledPin = 13;
```

VARIAVEIS



```
1 int buttonPin = 2;  
2 int ledPin = 13;
```

VARIAVEIS



```
1 #define LED_PIN 13
```



CONDICIONALES

CONDICIONAIS

As condicionais são estruturas de controle que permitem tomar decisões com base em condições específicas. As condicionais permitem que o programa execute diferentes blocos de código dependendo se uma condição é verdadeira ou falsa.



CONDICIONAIS

IF

A estrutura if é usada para executar um bloco de código se uma condição especificada for verdadeira. Se a condição não for verdadeira, o bloco de código dentro do if não será executado.

Pode ser seguido opcionalmente por else ou else if para definir o que fazer quando a condição não for verdadeira.



CONDICIONAIS

ELSE IF:

A estrutura else if permite verificar múltiplas condições em sequência. Se a condição dentro do if for falsa, a condição dentro do else if será verificada.



CONDITIONALS



```
1 void loop() {  
2     int buttonState = digitalRead(buttonPin);  
3  
4     if (buttonState == LOW) {  
5         digitalWrite(ledPin, HIGH);  
6     } else {  
7         digitalWrite(ledPin, LOW);  
8     }  
9 }
```

CONDICIONAIS

SWITCH:

A estrutura switch é usada quando há várias condições possíveis para testar. Ela permite selecionar um dos muitos blocos de código para serem executados.



CONDICIONAIS

```
● ● ●  
1 switch (buttonState) {  
2   case HIGH:  
3     digitalWrite(ledPin, HIGH);  
4     Serial.println("LED ligado");  
5     break;  
6  
7   case LOW:  
8     digitalWrite(ledPin, LOW);  
9     Serial.println("LED desligado");  
10    break;  
11 }
```



LACOS DE REPEIÇÃO

LAÇOS

Os laços de repetição são estruturas que permitem executar um bloco de código várias vezes consecutivamente. Isso é útil quando você precisa realizar uma tarefa repetitiva sem escrever o mesmo código várias vezes.



LAÇOS

FOR:

O laço for é utilizado quando se sabe exatamente quantas vezes o código precisa ser repetido. Ele consiste em três partes: uma inicialização, uma condição de continuação e uma expressão de incremento (ou decremento). O bloco de código dentro do for é executado repetidamente enquanto a condição for verdadeira.



LAÇOS

5



```
1  for (int i = 0; i <= rodada; i++) {  
2      digitalWrite(sequencia[i], HIGH);  
3      delay(500);  
4      digitalWrite(sequencia[i], LOW);  
5      delay(500);  
6  }  
7
```

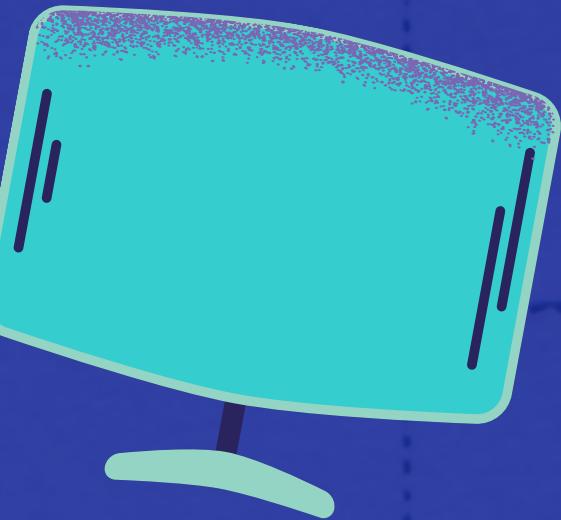
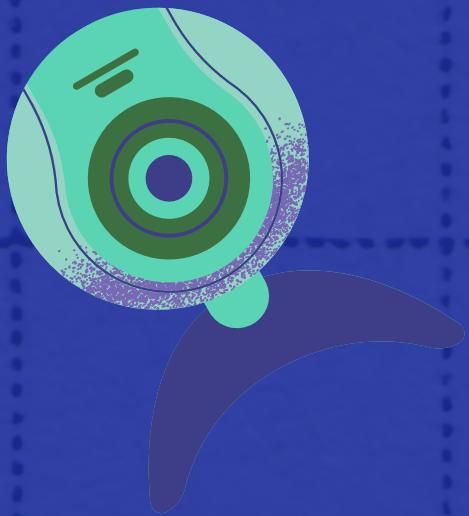
LAÇOS

WHILE:

O laço while é usado quando você precisa repetir um bloco de código enquanto uma condição específica for verdadeira. O bloco de código é executado repetidamente enquanto a condição for verdadeira.



LACOS

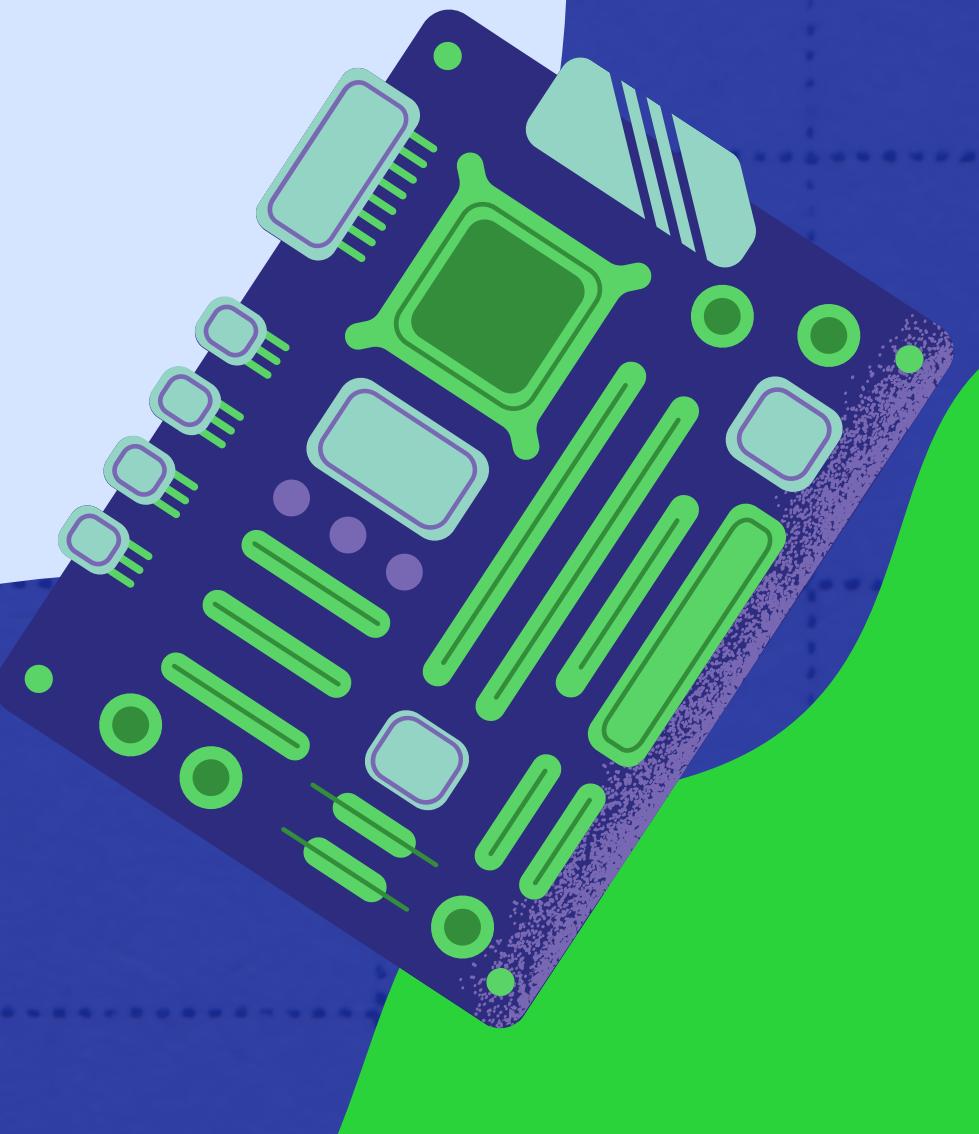
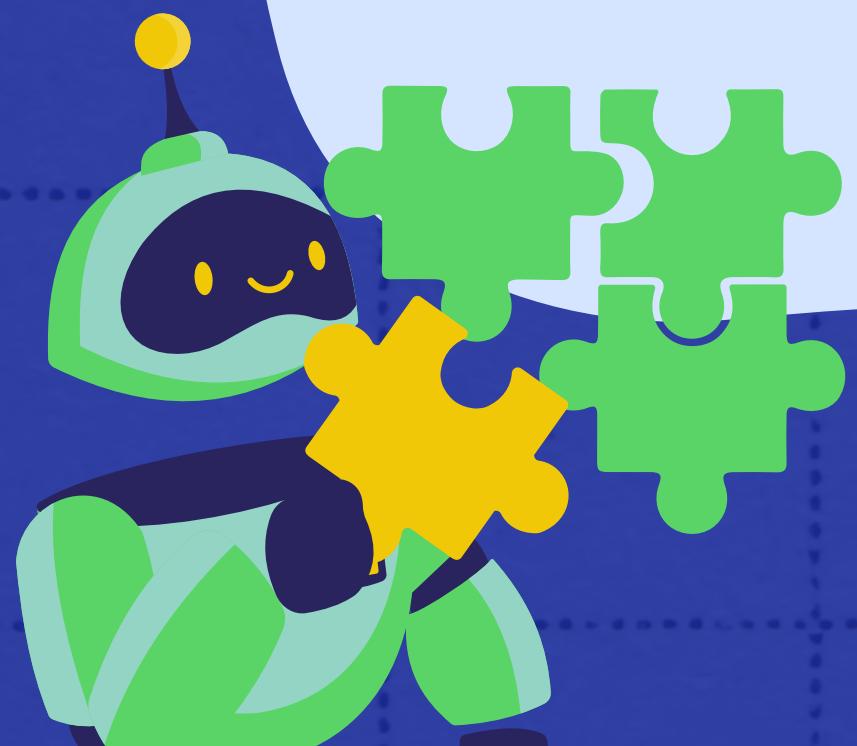


```
1 while (true) {  
2     // Liga o LED  
3     digitalWrite(LED_PIN, HIGH);  
4  
5     // Aguarda 1 segundo  
6     delay(1000);  
7  
8     // Desliga o LED  
9     digitalWrite(LED_PIN, LOW);  
10  
11    // Aguarda 1 segundo  
12    delay(1000);  
13 }
```





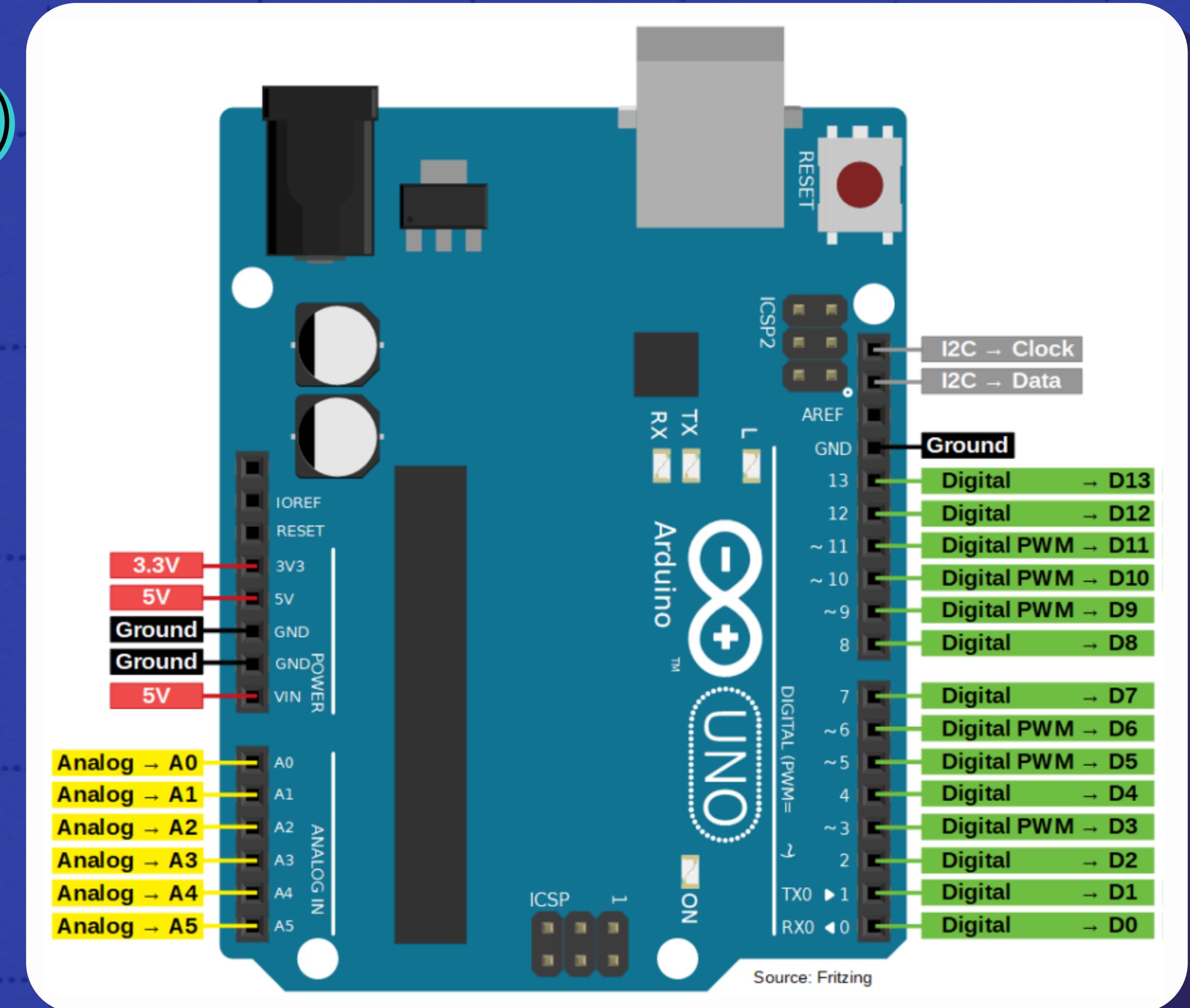
CONHECA ARDUINO

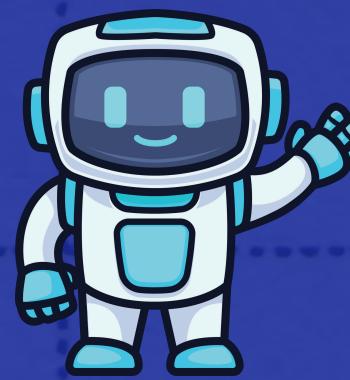


COMBU
MOKER

CONHECA O ARDUINO:

COMBU
MOKER





CONHEÇA O ARDUINO:

- Entradas e Saídas:

- Digitais:

O Arduino Uno possui 14 pinos digitais que podem ser usados como entradas ou saídas digitais.

- Analógicas:

Possui 6 pinos de entrada analógica, que podem ser usados para ler valores analógicos, como sensores de luz, temperatura e potenciômetros.

- Ground (GND):

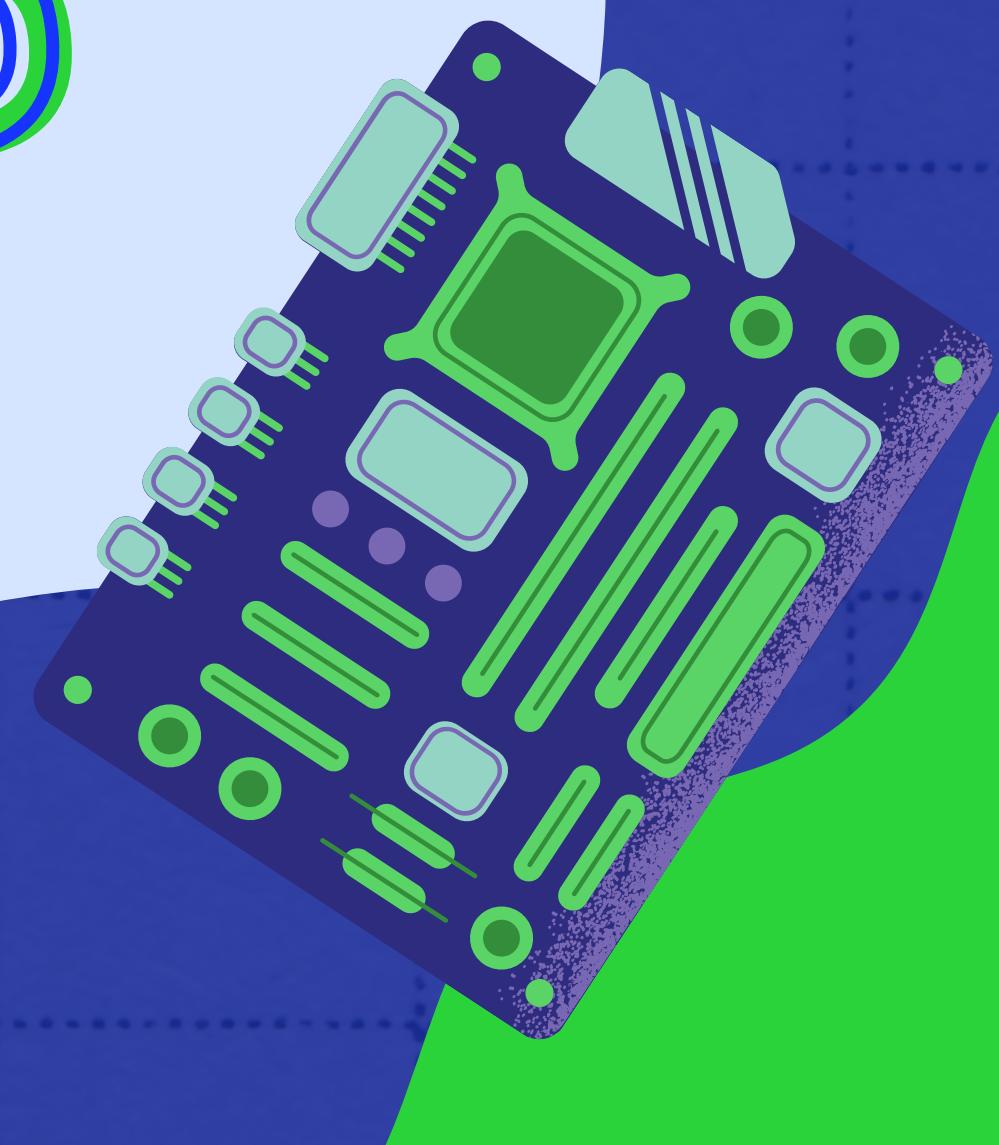
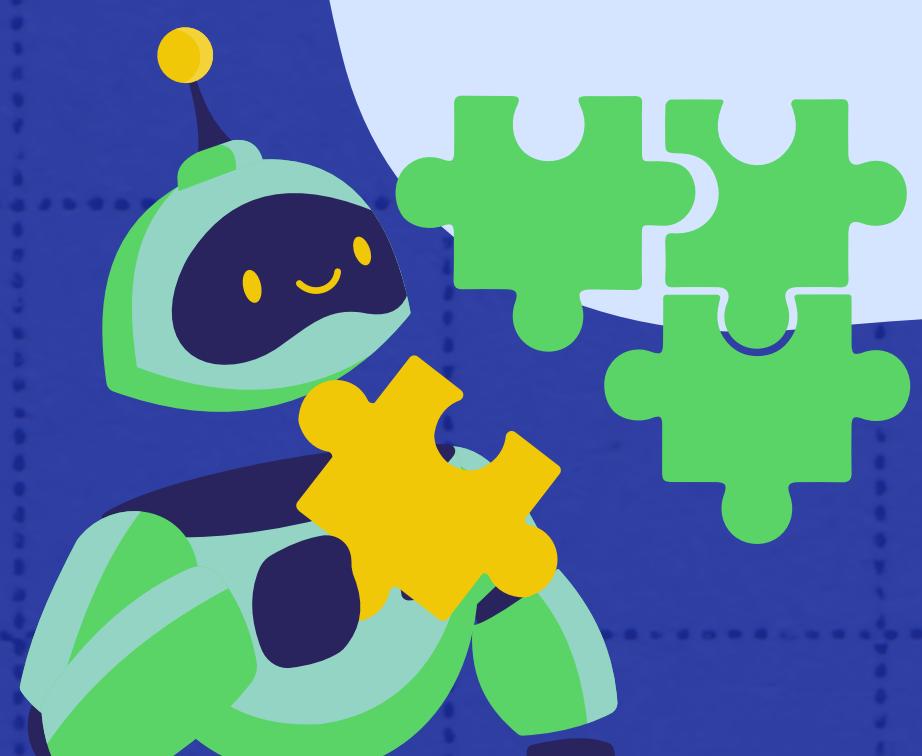
Representa a referência de 0 volts para o circuito e é o caminho de retorno para a corrente.

- Sinal PWM:

O PWM é uma técnica usada para controlar a intensidade de um sinal elétrico, como uma tensão ou corrente, variando a largura dos pulsos em um sinal digital.



PROTOBOARD



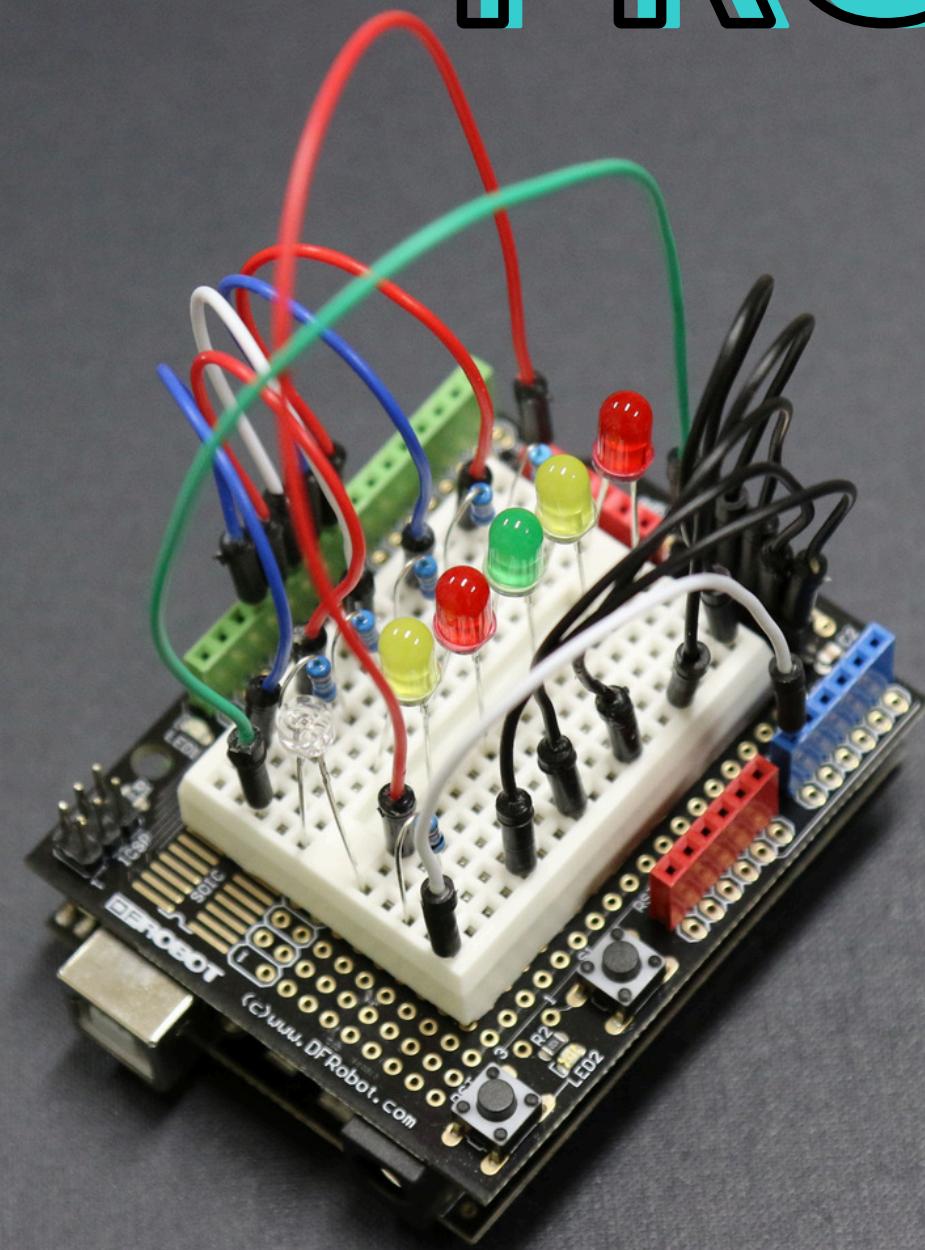
COMBU
MOKER

PROTOTOBARD

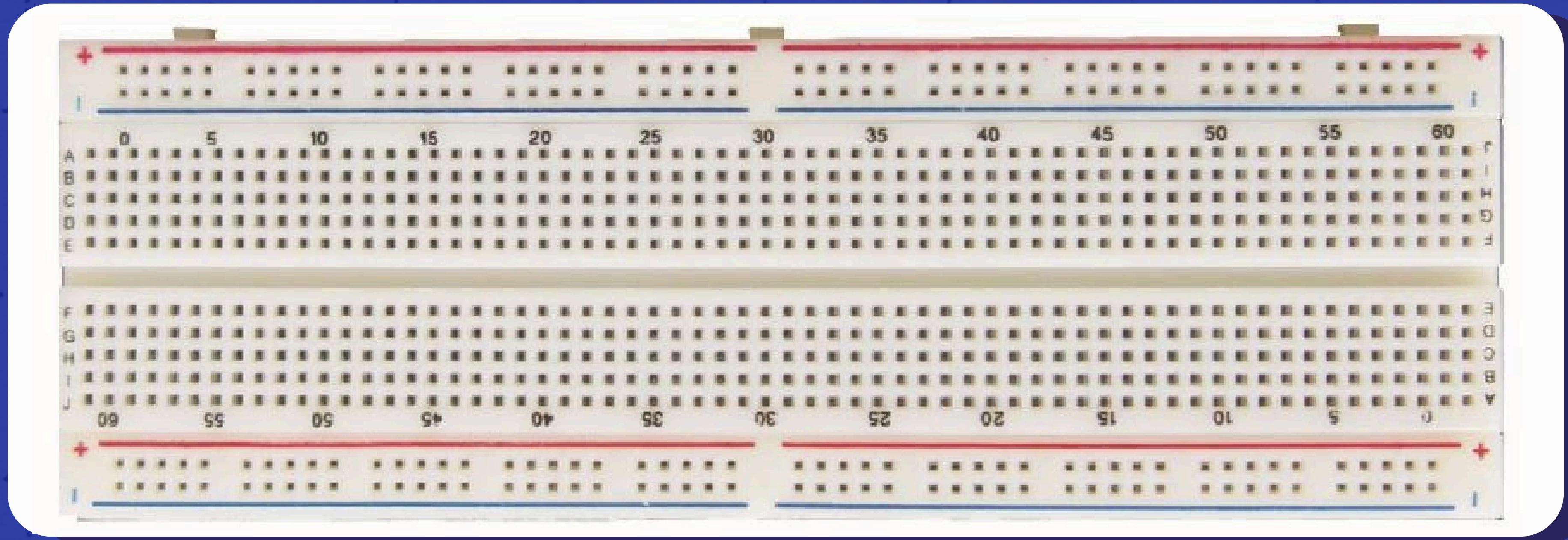
• O que é?

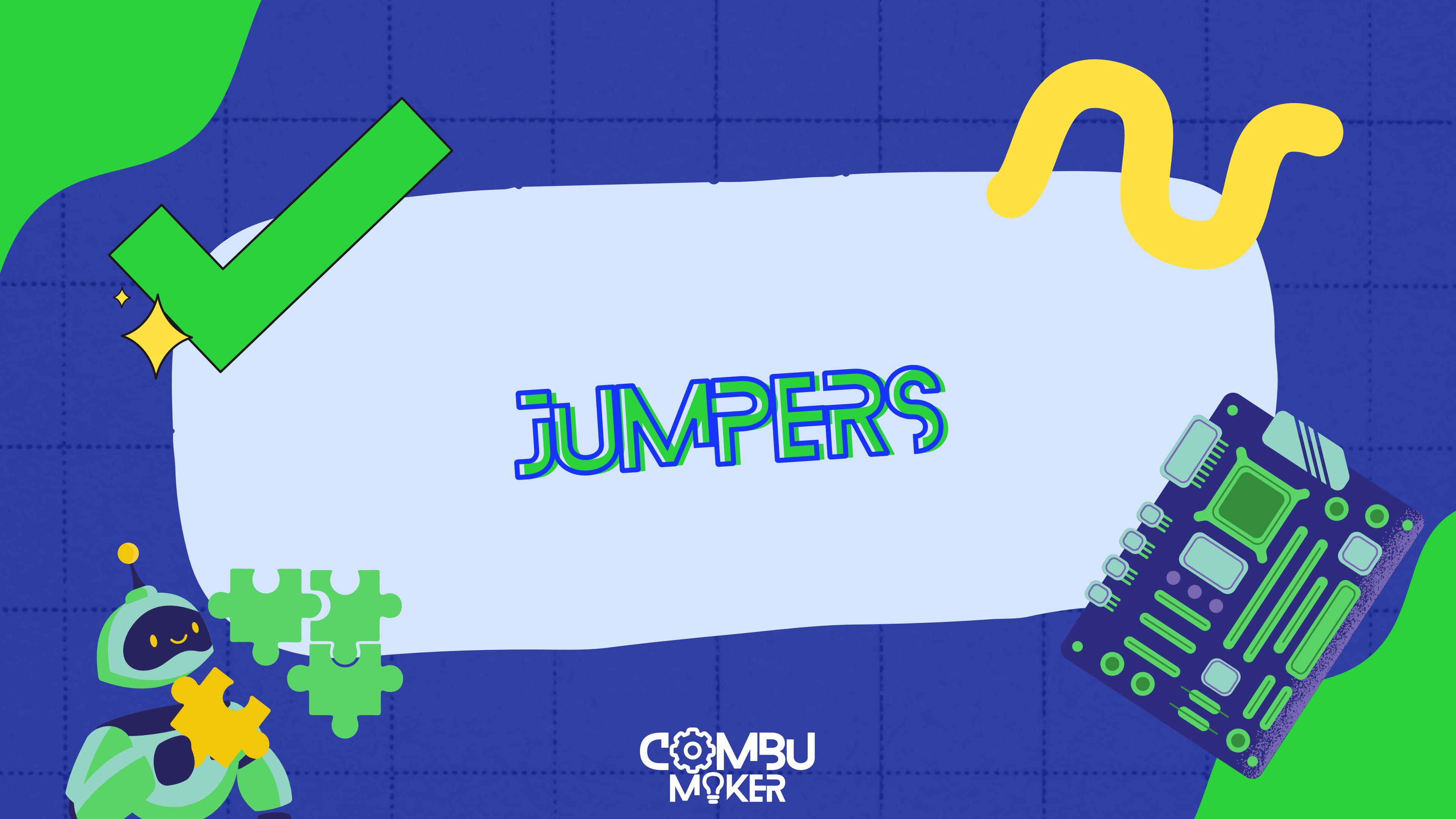
- É uma placa de circuito eletrônico sem soldagem, usada para criar protótipos de circuitos elétricos e eletrônicos.

- É uma ferramenta essencial para experimentar, testar e depurar circuitos.



CONHEÇA A PROTOBOARD:





JUMPERS

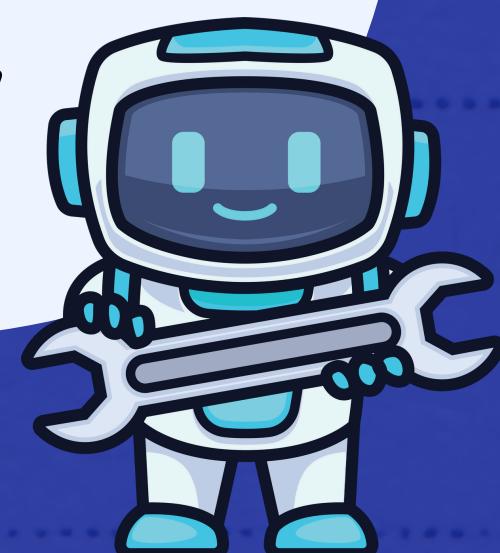
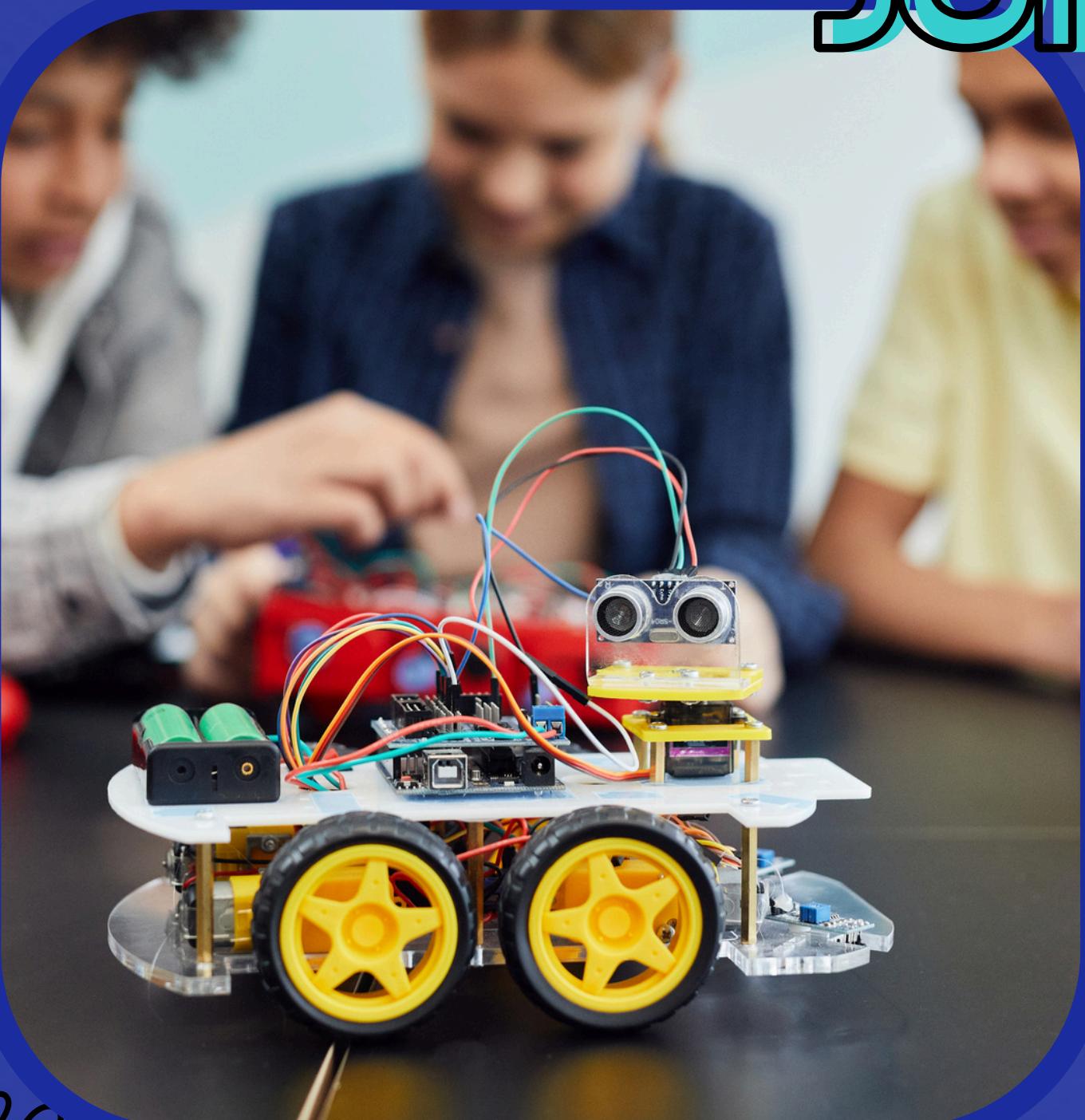
COMBU
MOKER

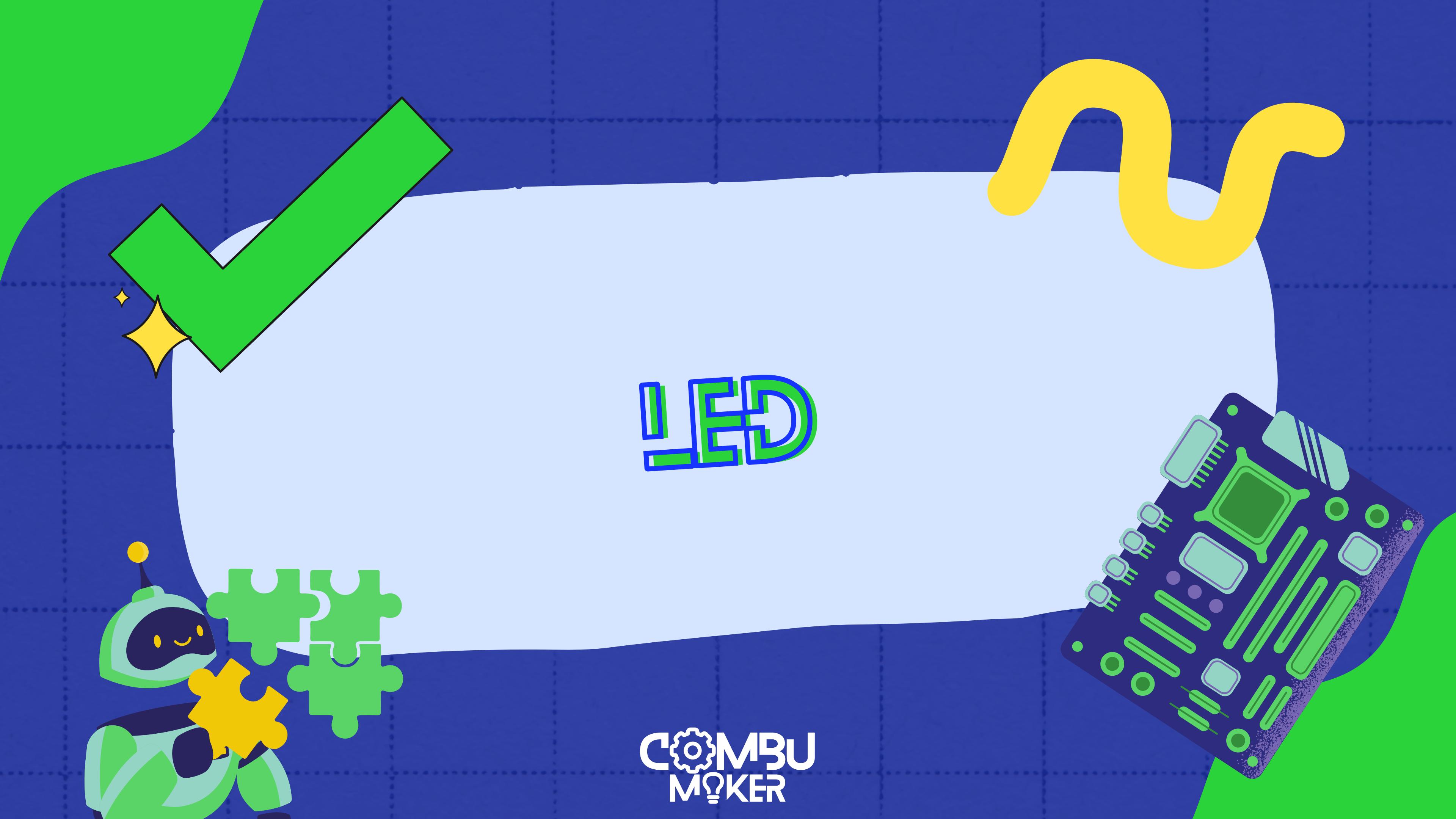
JUMPERS

eee

- O que é?

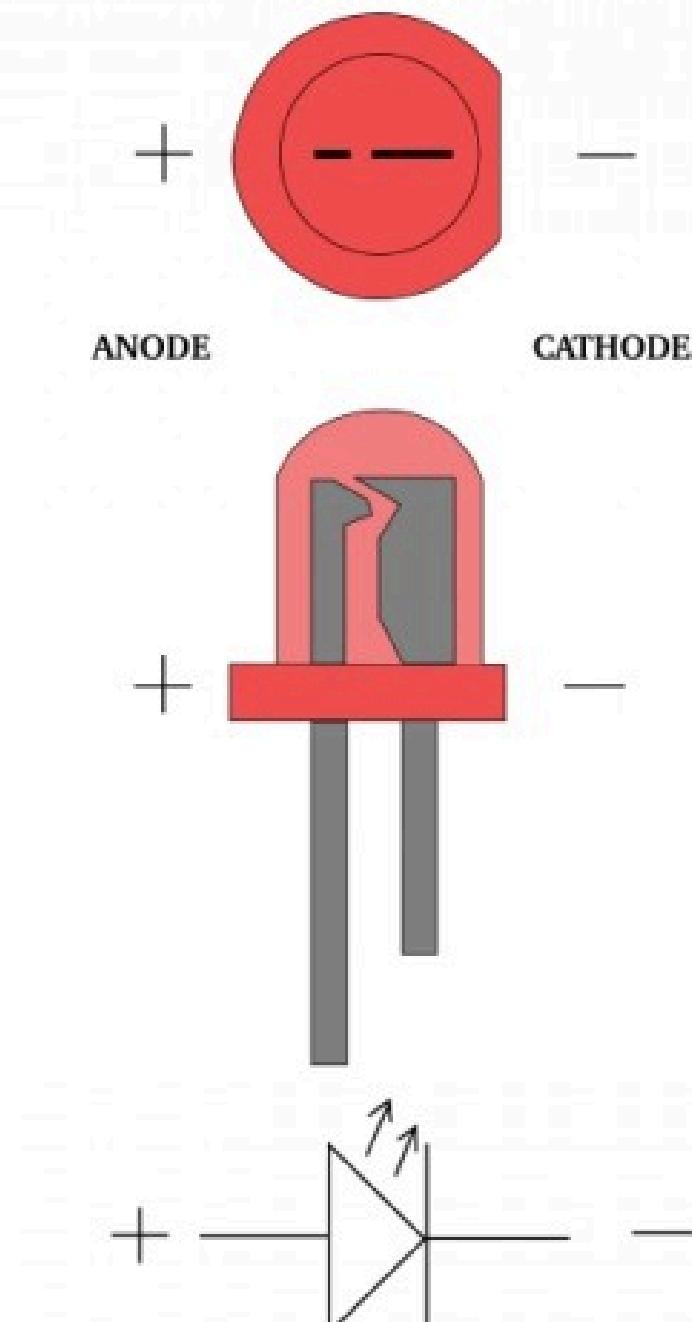
Um jumper é um pequeno componente eletrônico utilizado para fazer conexões temporárias entre diferentes partes de um circuito. No Arduino, os jumpers são frequentemente usados para fazer conexões entre os pinos do microcontrolador e outros componentes eletrônicos, como sensores, displays, ou módulos.





COMBU
MOKER

LED



- **O que é?**

O LED é um componente eletrônico que emite luz quando uma corrente elétrica passa por ele. No Arduino, os LEDs são frequentemente utilizados para indicar o status de um circuito, como ligado/desligado, ou para fornecer feedback visual.





RESISTORES

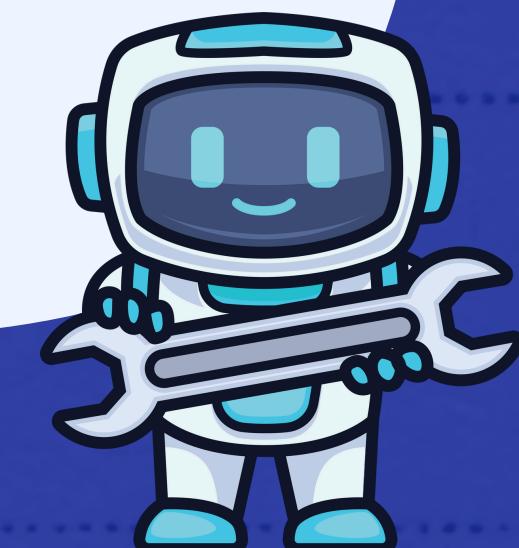
COMBU
MOKER

RESISTORES



• O que são?

Os resistores são componentes eletrônicos básicos e muito utilizados em projetos com Arduino. Eles funcionam como limitadores de corrente, protegendo os componentes e garantindo o funcionamento correto do circuito.

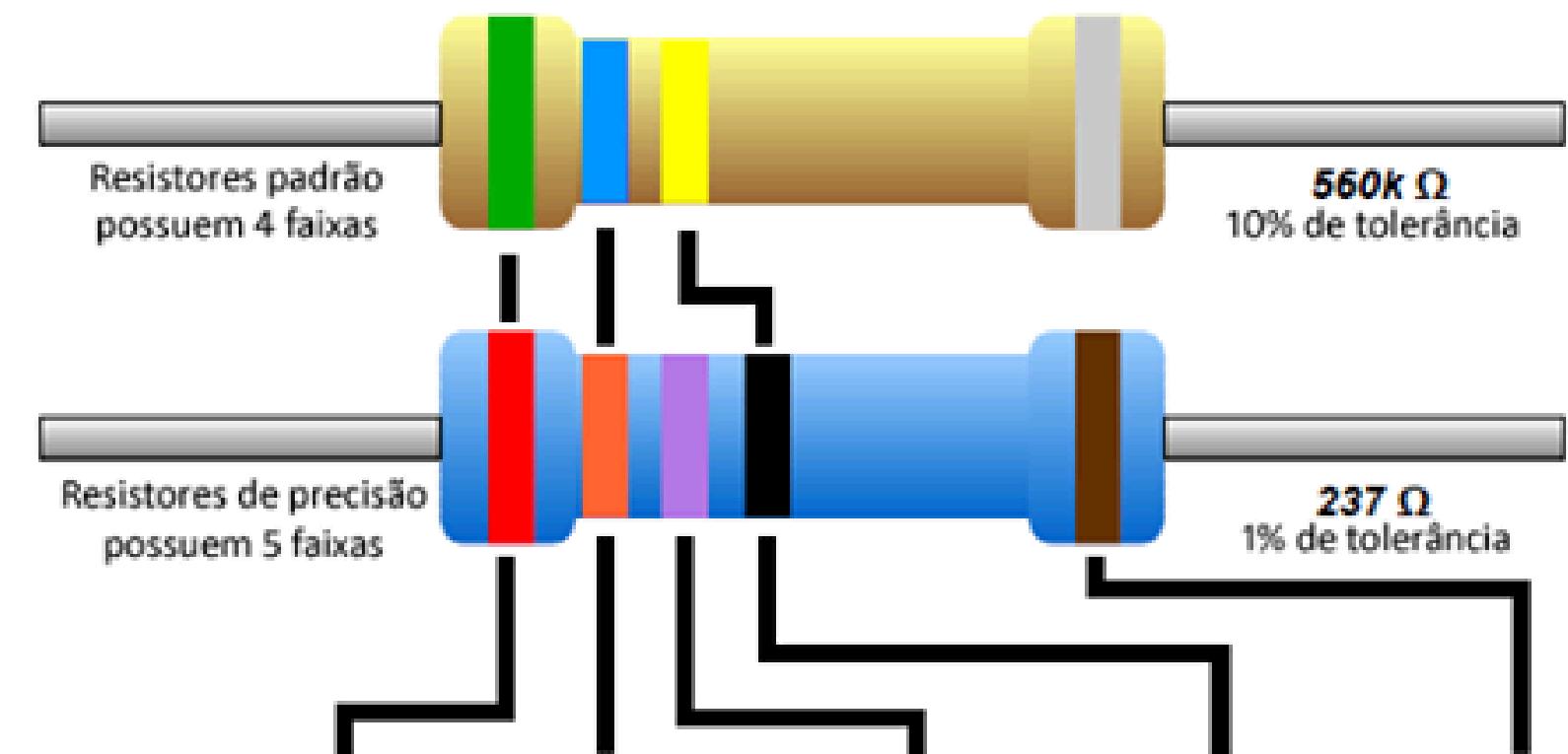


RESISTORES

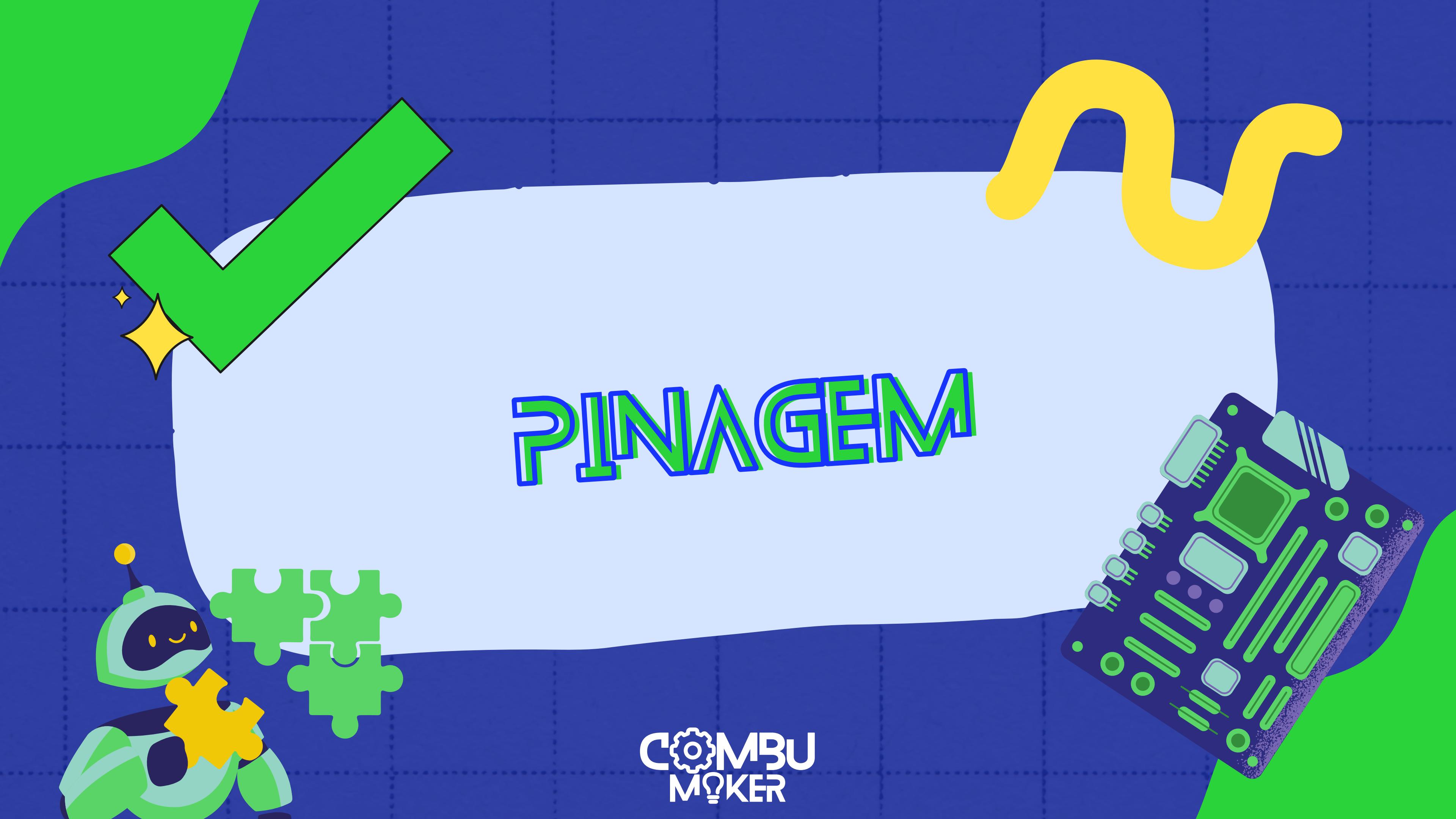
COMBU
MOKER

Código de Cores

A extremidade com mais faixas deve apontar para a esquerda



Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Preto	0	0	0	x 1 Ω	
Marrom	1	1	1	x 10 Ω	+/- 1%
Vermelho	2	2	2	x 100 Ω	+/- 2%
Laranja	3	3	3	x 1K Ω	
Amarelo	4	4	4	x 10K Ω	
Verde	5	5	5	x 100K Ω	+/- .5%
Azul	6	6	6	x 1M Ω	+/- .25%
Violeta	7	7	7	x 10M Ω	+/- .1%
Cinza	8	8	8		+/- .05%
Branco	9	9	9		
Dourado				x .1 Ω	+/- 5%
Prateado				x .01 Ω	+/- 10%



PINAGEM

COMBU
MOKER

LINGUAGEM ARDUINO



```
1 int pare = 8;  
2 int atencao = 9;  
3 int siga = 10;
```

Pinagem: Para definir as portas, basta declarar o tipo de variável em seguida o nome e em seguida atribuir a variável ao pino que será usado

LINGUAGEM ARDUINO



```
1 pinMode(pare, OUTPUT);  
2 pinMode(atencao, OUTPUT);  
3 pinMode(siga, OUTPUT);
```

Configuração dos pinos

(pinMode): Para começar a usar um pino específico, você precisa configurá-lo como entrada (input) ou saída (output) usando a função **pinMode();**

LINGUAGEM ARDUINO



```
1 void sinalVermelho() {  
2     digitalWrite(pare, HIGH);  
3     digitalWrite(atencao, LOW);  
4     digitalWrite(siga, LOW);  
5 }
```

Configuração dos pinos

(digitalWrite): É uma função usada para ativar o sinal digital de determinado pino, possui dois argumentos que são o pino e o tipo de sinal



DESAFIOS

COMBU
MOKER

DESAFIO

1. Acenda um LED;
2. Pisque o LED a cada 3 segundos;
3. Semáforo;
4. Semáforo com semáforo de pedestres;