

ELAS.NET APRESENTA:

Curso Internet das Coisas

Instrutoras: Bruna Sanches e Maria Eduarda Carmona



Revisão Módulo 4: Programação no Arduino

Revisão

Módulo 4: Programação no Arduino

Arduino e linguagem de
programação

Variáveis e Tipos de dados

Funções

Operadores

Comandos de Decisão e
Repetição

Sensores e Atuadores

Bibliotecas

Tarefa

Arduino e Linguagem de Programação

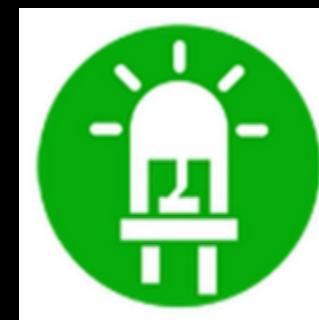


Arduino e Linguagem de Programação

Arduino: ONDE CONSULTAR?



Apostilas Eletrogate
Desde iniciante até avançado



Brincando com Ideias

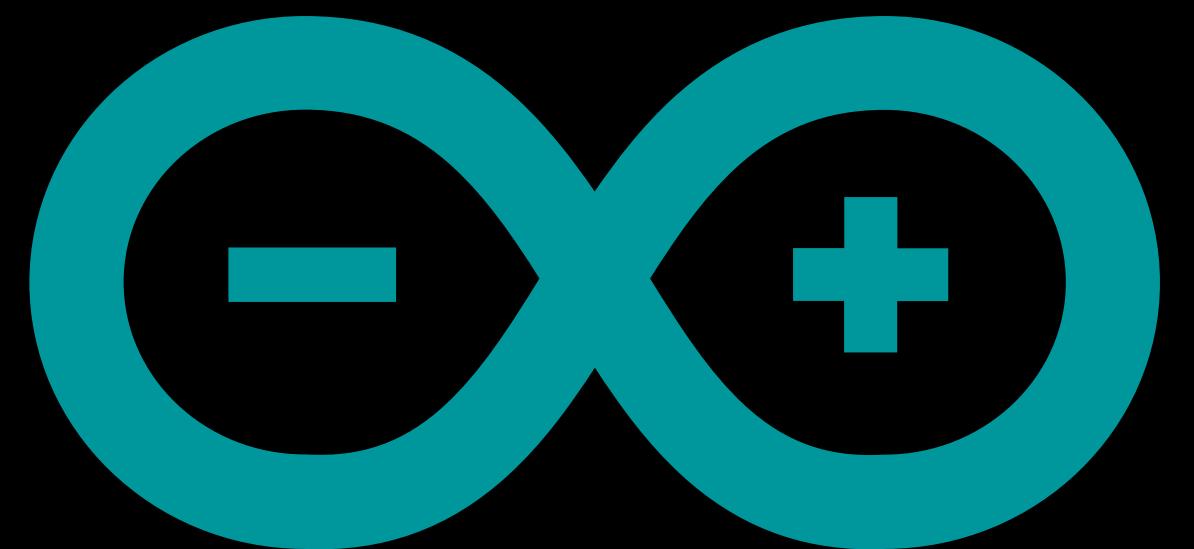
Pensou em Arduino ? Internet das Coisas ? Está no lugar certo! Aqui você vai aprender sobre Arduino e tudo o que está relacionado. Desde a base, passando pelos conceitos básicos da eletrônica e lógica de programação...

YouTube

Canal Brincando com Ideias

Arduino e Linguagem de Programação

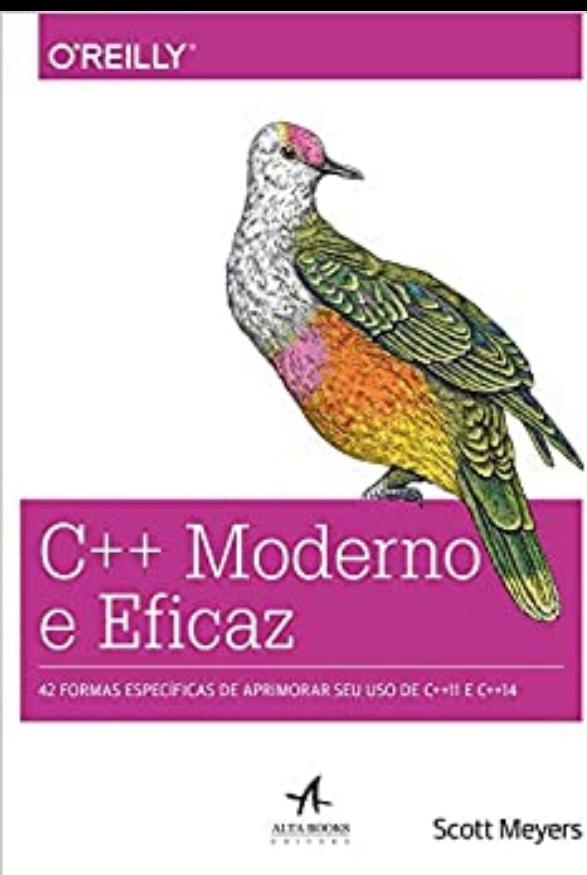
Arduino: ONDE CONSULTAR?



ARDUINO
Documentação

Arduino e Linguagem de Programação

C++: ONDE CONSULTAR?



mReviews - As Melhores e Mais divertidas Reviews da Internet

O mREVIEWS é um site reviews, dicas, seleções e avaliações não muito convencionais, de marcas, produtos, aplicativos, serviços e...

mreviews.com.br

<https://www.mreviews.com.br/2169/top-10-melhores-livros-para-aprender-a-programar-na-linguagem-c>

Variáveis

Tipos de dados básicos

```
palavra-chave nomeVariavel = valor;
```

char Caractere

int Numérico Inteiro

float Numero com ponto flutuante simples

double Numero com ponto flutuante duplo

bool Valores true ou false

Pergunta: Variáveis e Tipos de dados básicos

Qual das alternativas apresenta a forma correta de se declarar e inicializar uma variável do tipo inteiro?

- A soma = 16
- B float soma = 16
- C int soma = 16;
- D int soma = 16



Pergunta: Variáveis e Tipos de dados básicos

Qual das alternativas apresenta a forma correta de se declarar e inicializar uma variável do tipo caractere?

- A char = 'a'
- B char letra = 'a';
- C char letra = a;
- D char letra = "a";
- E char letra = 'ab';



Variáveis

Tipos de dados básicos

```
palavra-chave nomeVariavel = valor;
```

Exemplos:

```
char pontoExclamacao = ' ! ';  
char mensagem[ 4 ] = "Ola";  
int idade = 17;  
float salário = 2500.6;  
double pi = 3.14159265359;
```

●	LEMBRETES:
●	Valores char são escritos dentro de aspas simples
●	
●	Strings são cadeias de caracteres e devem ser criados como um vetor
●	e são escritos dentro de aspas duplas
●	

Funções

Uma função é um trecho de código que faz alguma tarefa específica e pode ser chamado de qualquer parte do programa quantas vezes desejarmos.

- Sintaxe

```
tipo de retorno nome_da_função (1º parâmetro,...){  
    //código  
    //se a função possui retorno  
return valor_de_retorno;  
}
```

- Retorno

- Parâmetros

- Chamada de função

Exemplo:

```
int tres(){  
    return 3;  
}
```

Funções

Uma função é um trecho de código que faz alguma tarefa específica e pode ser chamado de qualquer parte do programa quantas vezes desejarmos.

- Sintaxe
- **Retorno**
 - Frequentemente, uma função faz algum tipo de processamento ou cálculo e precisa devolver o resultado desse procedimento.
 - Usa-se a palavra `return`
 - Exemplo:

```
int tres(){  
    return 3;  
}  
  
int numero = tres();  
//numero = 3
```
- Parâmetros
- Chamada de função

Funções

Uma função é um trecho de código que faz alguma tarefa específica e pode ser chamado de qualquer parte do programa quantas vezes desejarmos.

- Sintaxe

Valor que é fornecido à função quando ela é chamada. Eles funcionam como variáveis locais.

- Retorno

Exemplo:

```
int soma(int num1, int num2){  
    return num1 + num2;  
}  
  
int resultado2 = soma(10, 20);  
//resultado2 = 30
```

- Parâmetros

- Chamada de função

Funções

Uma função é um trecho de código que faz alguma tarefa específica e pode ser chamado de qualquer parte do programa quantas vezes desejarmos.

Sintaxe

```
nome_da_função (arg1, ...);
```

Ou

```
variavel = nome_da_função (arg1, ...);
```

Exemplo:

```
int soma(int num1, int num2){  
    return num1 + num2;  
}
```

```
int resultado2 = soma(10, 20);  
//resultado2 = 30
```

Parâmetros

Chamada de função

Pergunta: Funções

Qual das alternativas apresenta a forma correta de chamar a função abaixo e o valor retornado?

```
int geraIdade(int anoNascimento, int anoAtual){  
    return anoAtual-anoNascimento;  
}
```

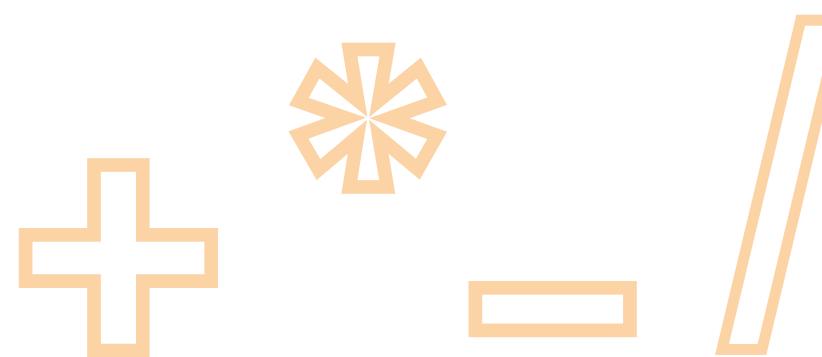
- A idade = geraIdade(2004,2021); //retorno: 17
- B idade = geraIdade(2021,2004); //retorno: 17
- C idade = geraIdade(); //não retorna nenhum valor
- D idade = geraIdade(2021,2004); //não retorna nenhum valor



Operadores

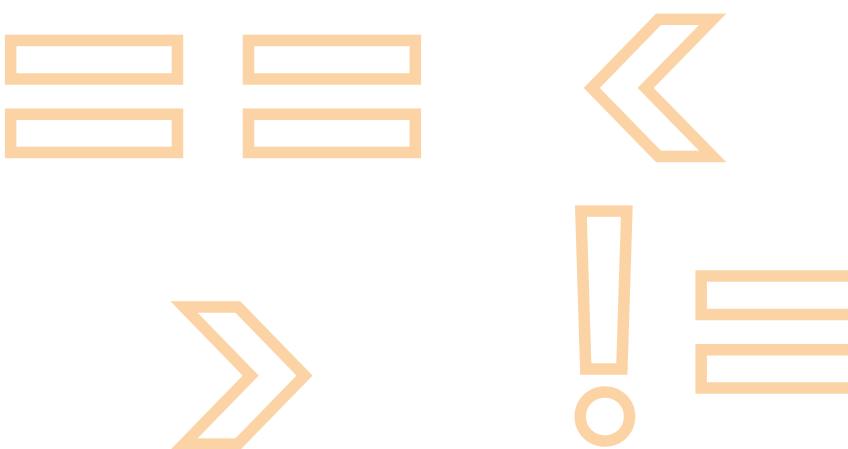
Matemáticos

Operadores matemáticos são os já conhecidos comandos para fazermos contas em linguagens de programação.



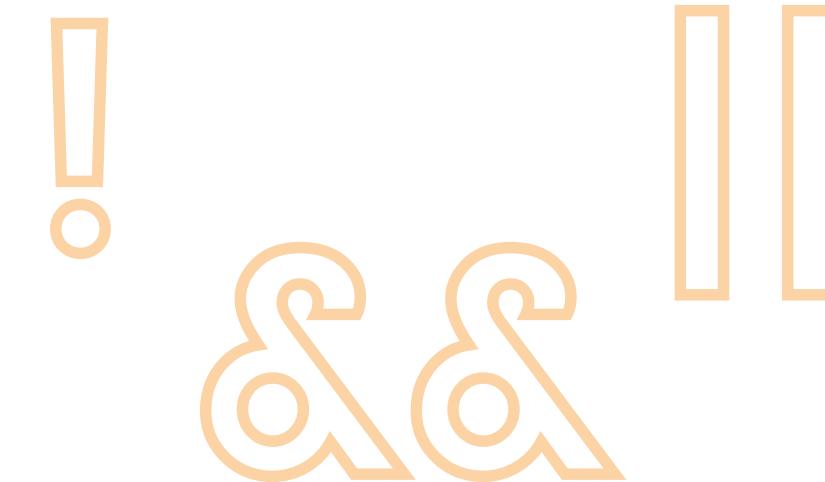
Relacionais

Operadores de comparação que analisam dois itens e retornam um valor verdadeiro ou falso (os booleans) caso a comparação seja atendida ou não.



Lógicos

São utilizados em conjunto com os nossos operadores condicionais para decidir o fluxo de um programa. Eles também retornam verdadeiro ou falso caso a premissa seja atendida ou não.



Operadores Matemáticos

Operação em C++	Operador Aritmético em C++	Expressão Algébrica	Expressão em C++
Adição	+	$a+b$	$a + b$
Subtração	-	$a-b$	$a - b$
Multiplicação	*	ab ou $a.b$	$a * b$
Divisão	/	a/b ou $a \div b$	a / b
Módulo	%	$a \text{ mod } b$	$a \% b$

Operadores

Relacionais

Símbolo	Nome do Operador	Exemplo	Significado
<code>></code>	Maior que	$x > y$	x é maior que y ?
<code>>=</code>	Maior ou igual	$x >= y$	x é maior ou igual a y ?
<code><</code>	Menor que	$x < y$	x é menor que y ?
<code><=</code>	Menor ou igual	$x <= y$	x é menor ou igual a y ?
<code>==</code>	Igualdade	$x == y$	x é igual a y ?
<code>!=</code>	Diferente de	$x != y$	x é diferente de y ?

Operadores Lógicos

Operador	Significado
Operador &&	Operador lógico and
Operador	Operador lógico or
Operador !	Operador lógico not

altura > 1,50 && idade > 12

altura > 1,50 || idade > 12

! true = false

! false = true

Pergunta: Operadores

Qual das alternativas apresenta corretamente o valor retornado pelas expressões?

- A `10+5 > 3*5 && 'a' == 'a'` //valor retornado: true
- B `!("Ana" == "ana")` //valor retornado: false
- C `5 % 2 == 1 && (9-3) >= 6` //valor retornado: false
- D `1+1 == 2 || 'A' == 'a'` //valor retornado: false



Comandos

Decisão e Repetição

Decisão

Estrutura lógica condicional onde uma decisão é tomada baseada numa condição específica cujo resultado sempre será Verdadeiro ou Falso.

if/else

Repetição

Comandos para repetir uma sequência de instruções. Estas estruturas de repetição, também conhecidas como laços (do inglês loops).

for
while

Comandos

Decisão

```
if (condição){  
    <instrução>;  
}  
  
if (condição){  
    <instrução>;  
} else if (condição2) {  
    <instrução2>;  
} else {  
    <intrução3>;  
}
```

```
if (nome == "Bruna"){  
    printf("Instrutora");  
}  
  
if (nome == "Bruna"){  
    printf("Instrutora");  
} else if (nome == "Duda"){  
    printf("Instrutora");  
} else {  
    printf("Aluna");  
}
```

Comandos

Decisão

```
if (condição){  
    <instrução>;  
}
```

```
if (condição){  
    <instrução>;  
} else if (condição2) {  
    <instrução2>;  
} else {  
    <intrução3>;  
}
```

```
if (nome == "Bruna"){  
    printf("Instrutora");  
}
```

```
if (nome == "Bruna"){  
    printf("Instrutora");  
} else if (nome == "Duda"){  
    printf("Instrutora");  
} else {  
    printf("Aluna");  
}
```

Pergunta: Comando de Decisão

Qual das alternativas apresenta corretamente o valor que a variável resultado assumirá ao final da execução?

```
if (!("Ana" == "ana")){
    resultado = 0;
} else {
    resultado = 1;
}

if (resultado == 1){
    resultado += 2;
}

if (resultado > 0){
    resultado--;
}
```

- A 0
- B 1
- C 2
- D 3



Comandos

Repetição



for (variável; condição; atualização){
 <instrução>;
}



for (i = 0; i < 3; i++){
 soma = soma + i
}



while (condição){
 <instrução>;
}

while (i < 3){
 i = i + 1;
 //ou i += 1 ou i++
}

Pergunta: Comando de Repetição

Qual das alternativas apresenta corretamente o valor que a variável soma assumirá ao final da execução?

```
int soma = 0;  
  
for (i = 1; i <= 5; i+2){  
    soma = soma + i  
}
```

- (A) 0
- (B) 4
- (C) 9
- (D) 15



Pergunta: Comando de Repetição

Qual das alternativas apresenta corretamente a quantidade de vezes que o laço while repetirá até parar?

```
bool verificador = true;  
int contador = 0;  
  
while(verificador == true){  
    contador++;  
  
    if(contador > 5){  
        verificador = false;  
    }  
}
```

A 0

B 1

C 5

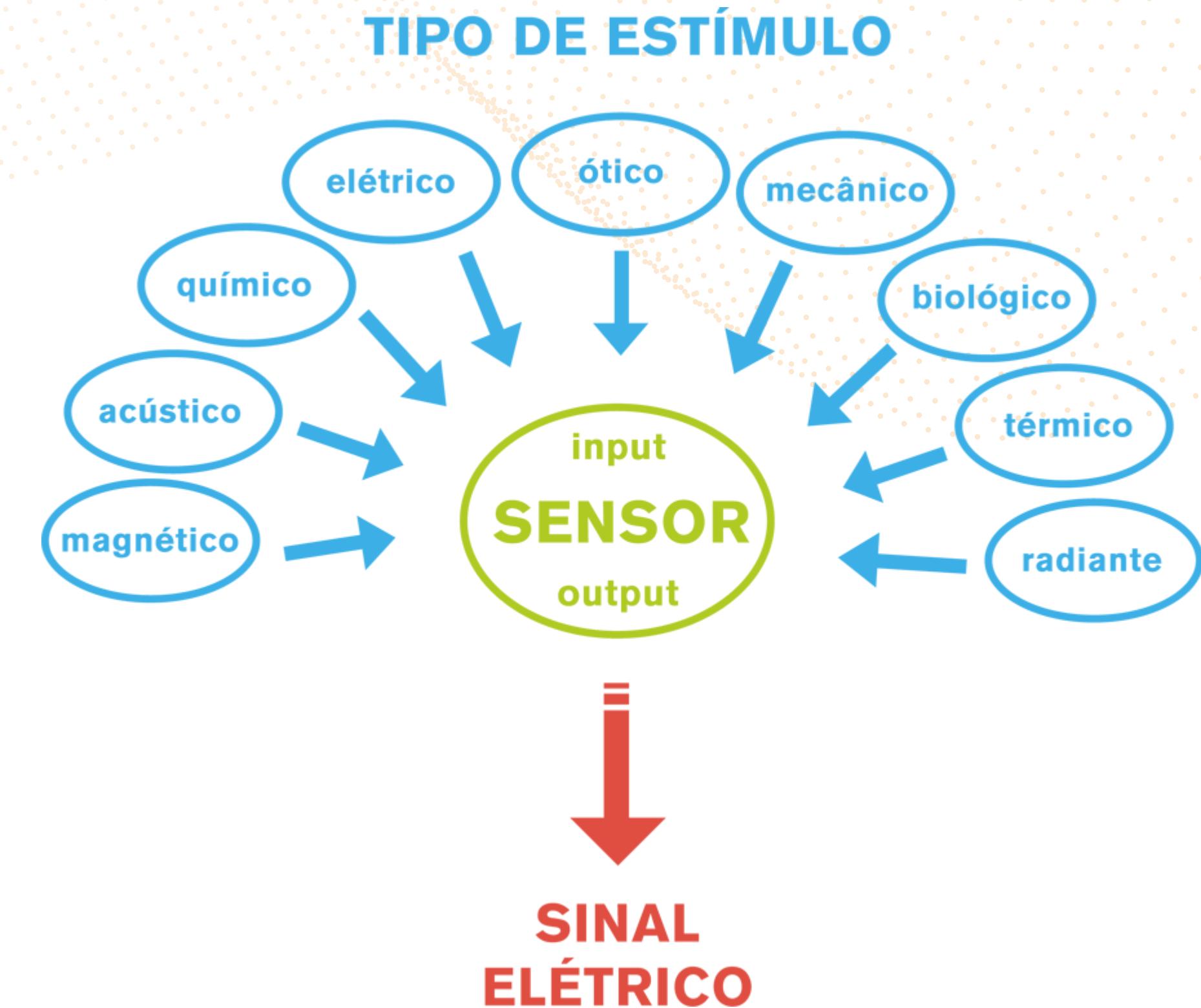
D 6

E o laço não para



Sensores

Um sensor é geralmente definido como um dispositivo que recebe e responde a um estímulo ou um sinal.



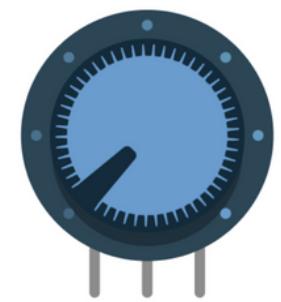
Atuadores

São componentes que executam determinadas ações no ambiente.



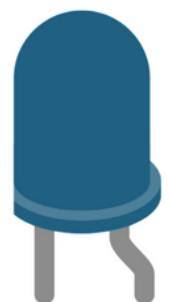
Sensor/atuador no circuito...

É um sensor ou um atuador?



Potenciômetro

Resistência regulável de
deixa passar mais ou
menos corrente.



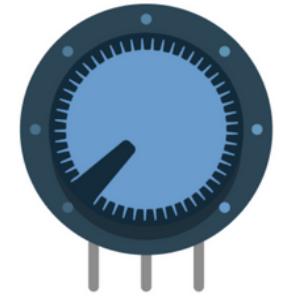
LED



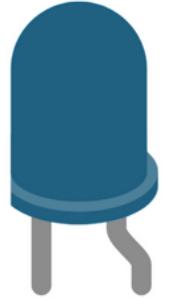
É diodo emissor de luz que
transforma energia elétrica
em luz.

Sensor/atuador no circuito...

É analógico ou digital?



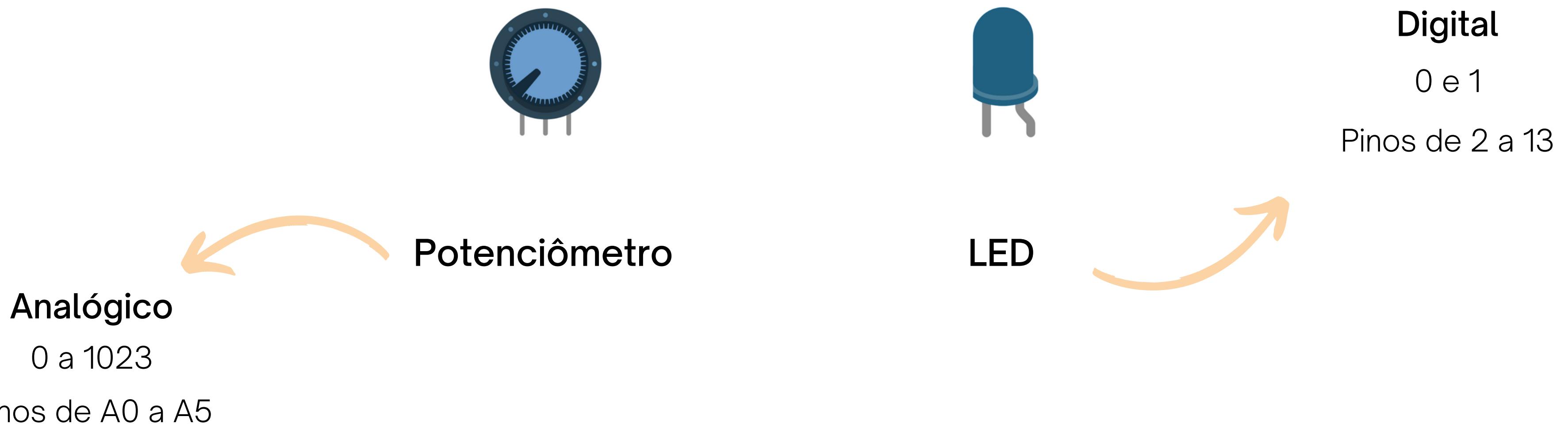
Potenciômetro



LED

Sensor/atuador no circuito...

É analógico ou digital?



Pergunta: Sensor/atuador no circuito...

Um circuito faz a leitura de um valor do Potenciômetro e a partir dele, diminui ou aumenta a intensidade da luz do LED. Quem é a entrada e a saída?

- A LED é a entrada e Potenciômetro saída
- B LED é a saída e Potenciômetro entrada



Pergunta: Sensor/atuador no circuito...

Qual é o código que inicializa o Potenciômetro como entrada e o LED como saída?

- A `pinMode(OUTPUT, pino);` e `pinMode(pino, INPUT);`
- B `pinMode(pino, INPUT);` e `pinMode(OUTPUT, pino);`
- C `pinMode(pino, OUTPUT);` e `pinMode(pino, INPUT);`
- D `pinMode(pino, INPUT);` e `pinMode(pino, OUTPUT);`



Bibliotecas

Uma biblioteca é um trecho de software que fornece uma funcionalidade específica a um programa.

```
#include <nome da biblioteca>
```

Exemplo: Servo Motor

```
#include <servo.h>
```

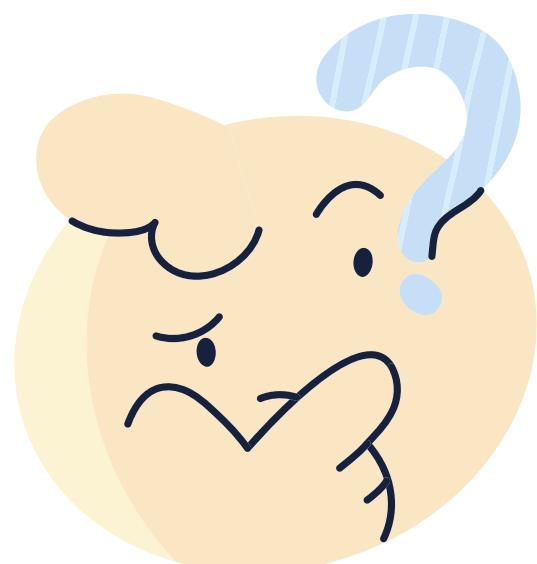
Methods

- `attach()`
- `write()`
- `writeMicroseconds()`
- `read()`
- `attached()`
- `detach()`

Pergunta: Bibliotecas

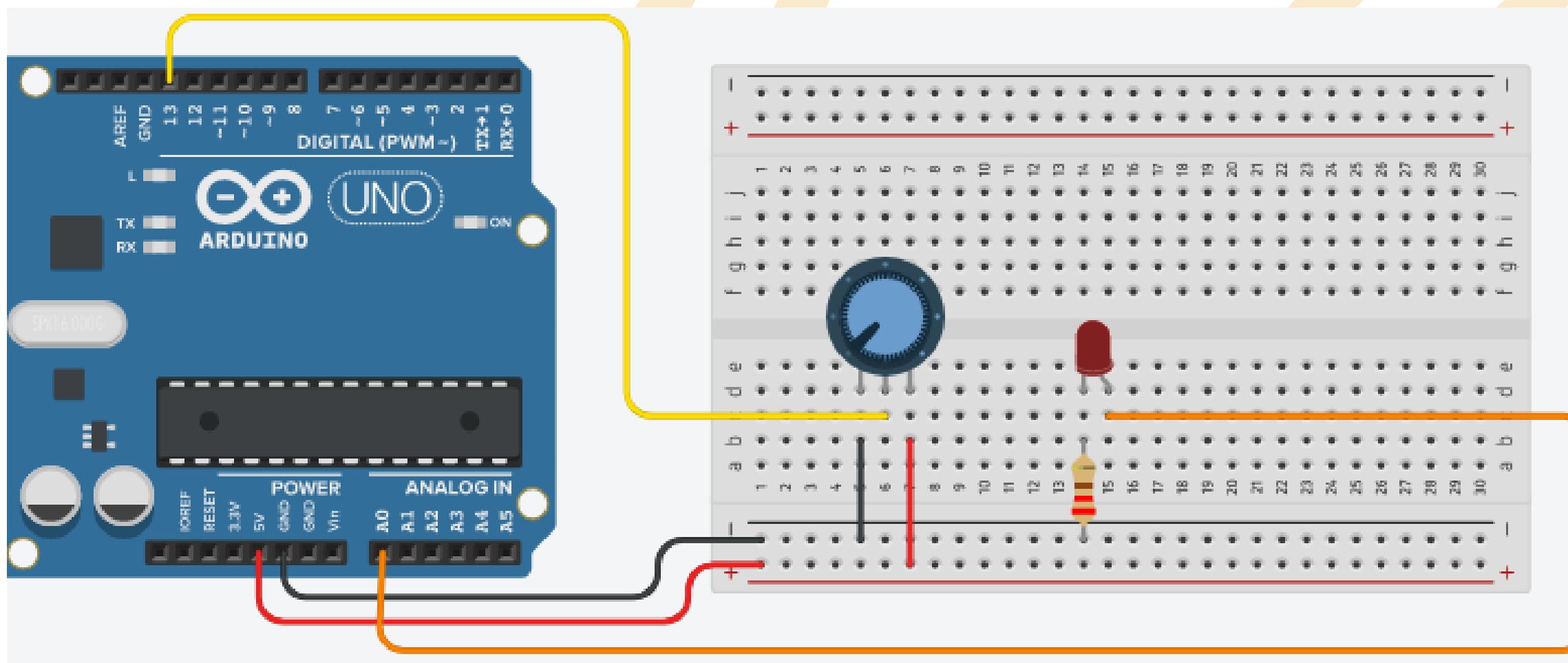
Para que o Servo Motor gire um determinado ângulo, qual comando da biblioteca **servo.h** eu devo usar?

- A attach()
- B write()
- C read()
- D white()



Pergunta: Circuitos

O circuito do Potenciômetro com o LED não está funcionando corretamente. Qual é o erro?



```
void setup()
{
    pinMode(A0, INPUT);
    pinMode(13, OUTPUT);
}

void loop()
{
    int valor = 0;

    valor = map(analogRead(A0), 0, 1023, 0, 255);
    analogWrite(13, valor);
}
```

Pergunta: Códigos

O código não está funcionando corretamente. Qual é o erro?

```
1 #include <Servo.h> // incluindo biblioteca
2 Servo servo_10; // iniciando objeto servo_10
3 void setup()
{
    servo_10.attach(10); //indicando objt no pino 10
    pinMode(13, INPUT); // pinos como entrada (botao)
    pinMode(12, INPUT);
    pinMode(11, INPUT);

}
11
12 void loop()
13 {
    // botao pino 13 - 0°
    //botao pino 12 - 90°
    //botao pino 11 - 180°

18    int pressionado = 1
19
20    // criando variaveis dos botões
21    int botao1 = 0;
22    int botao2 = 0;
23    int botao3 = 0;
24
25    // lendo o valor dos botoes
26    botao1 = digitalRead(13);
27    botao2= digitalWrite(12);
28    botao3= digitalRead(10);
```

Dica: 4 erros

Tarefa 2

Módulo 4:

Programação no Arduino

- **Links dos circuitos:**
 1. Circuito com Buzzer
 2. Servo Motor e 3 Botões
 3. PWM - Variar luminosidade do LED
 4. Sensor Ultrassônico e LED
 5. Motor DC com botão
 6. Motor DC com Potenciômetro
 7. Circuito com sensor de temperatura e display LCD
- **Comentários nos códigos;**
- **Descrição dos circuitos e códigos no documento.**



Alguma
questão?

Esperamos que tenham gostado e aprendido algo novo.



Bruna bcpsaches@gmail.com

Maria Eduarda dudacarmona1802@gmail.com

@elasnetcefetmg