

# Curso de Arduino

## Unidade 03

**Cristiano Mota**  
**Lucas Weynars**  
**Marcelo Ribeiro**

# Cronograma de Aula

- Dúvidas
- Introdução a programação(parte 2)
- Introdução a eletrônica(parte 2)
- Projetos



\*

\*

# Dúvidas



\*

\*

# O que é Algoritmo?

O algoritmo é uma sequência de passos necessários para a execução de uma tarefa.

\*

\*

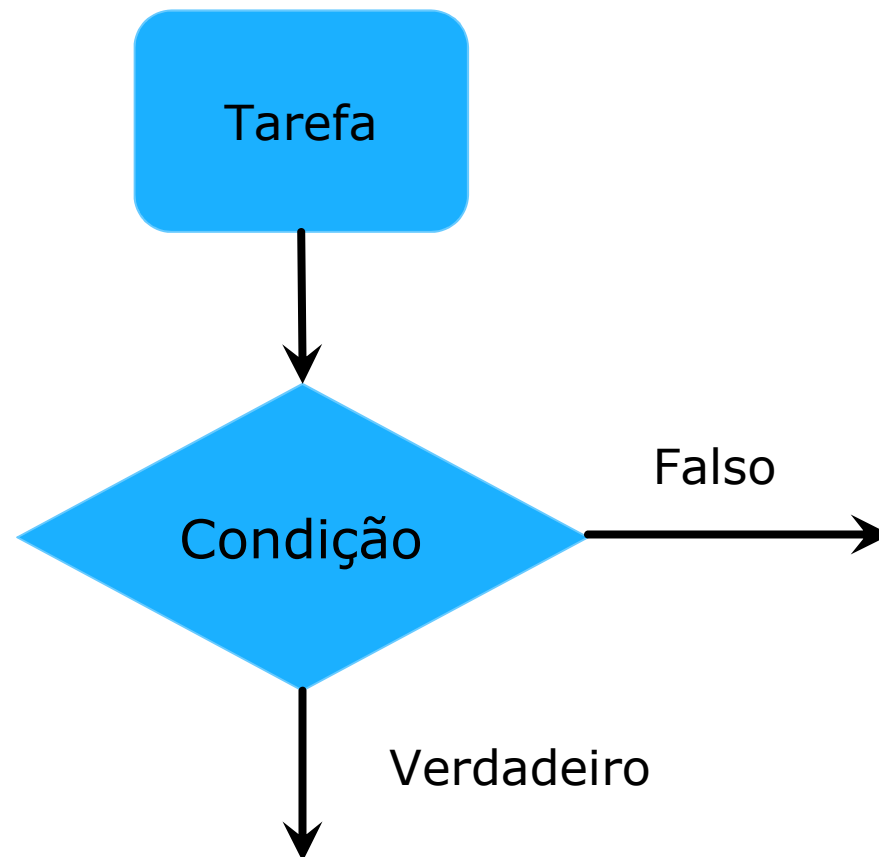
# O que é Estrutura de Controle?

É a ordem em que as instruções, expressões e chamadas de função são executadas ou avaliadas no algoritmo.

\*

\*

# Estruturas de controle



# Estruturas de controle

- Estrutura sequencial
- Estrutura de seleção
- Estrutura de repetição

\*

\*

# Estrutura Sequencial

Um estrutura sequencial realiza um conjunto pré-determinada de comandos de forma sequencial, na ordem em que foram declarados no código fonte.

\*

\*



```
void setup()
{
    pinMode(led, OUTPUT);
    digitalWrite(led,HIGH);
}
```

\*

\*

# Estrutura de Seleção

Esta estrutura realiza diferentes ações dependendo se a seleção (ou condição) é verdadeira ou falsa.

\*

\*

Divide-se em:

- If
- If ... Else
- Switch...Case

\*

\*

# If

Se é a estrutura de decisão que decide entre duas sequências de comandos que serão ou não executados.

\*

\*

```
if( variável>n)
{
    //executa as linhas de códigos
}
```

// Se a **variável** for maior que n então o código será executado

# If...else

Estende-se a estrutura `if()` permitindo que o programa escolha entre dois ou mais blocos de código.

\*

\*

```
if( variável>n)
{
    //executa a sequencia de comandos 1
}
else
{
    //executa a sequencia de comandos 2
}

// Se a variável for maior que n então o código
executa a sequencia de comandos 1 senão
executa a sequencia de comandos 2.
```

# Switch...case

Funciona como uma estrutura if, porém é mais conveniente usar quando existe mais de 3 alternativas. São aceitos somente inteiros e caracteres.

\*

\*



```
Int num = 1;

Void setup(){

Serial.begin(9600);

Switch(num){

Case 0:

Serial.println("Zero");//não executa

Break;

Case 1:

Serial.println("Um");//imprime 'Um'

Default:

Serial.println("nenhum dos dois");

Break;

}
```

# Estrutura de Repetição

Uma estrutura de repetição realiza e repete diferentes ações dependendo se é verdadeira ou falsa, condição essa que é uma expressão processada e transformada em um valor booleano.

\*

\*

## For( ; ; )

Utilizado para repetir um bloco de instruções.

```
For( i=0; i<n;i++)  
{  
    soma = soma +1;  
}
```

\*

\*

# While

Controla uma sequencia de repetições. A estrutura while executa uma série de declarações continuamente enquanto a expressão é verdadeira.

\*

\*

```
int x = 0;

void setup()
{
    while(x<100)//repete o bloco até que a
    condição se torne falsa
    {
        delay(50); // espera 50 ms

        x=x+1;
    }
}
```

\*

\*

## Do ... While

A função do...while funciona de maneira semelhante a função While, mas ao invés de checar a condição no início do loop o teste é realizado no final do laço.

\*

\*

```
int x = 0;
void setup()
{
    do {
        delay(50); // espera 50 ms
        x=x+1;
    }while (x<100); //esse código executa continuamente,
    até o valor ser maior que 100
```

\*

\*

# Entradas Analógicas

Converte a tensão de entrada, de 0 a 5 volts, a um valor digital entre 0 e 1023. Isto é feito por um circuito no interior do Arduino chamado um **conversor analógico-para-digital** ou **ADC**.

\*

\*



# analogRead()

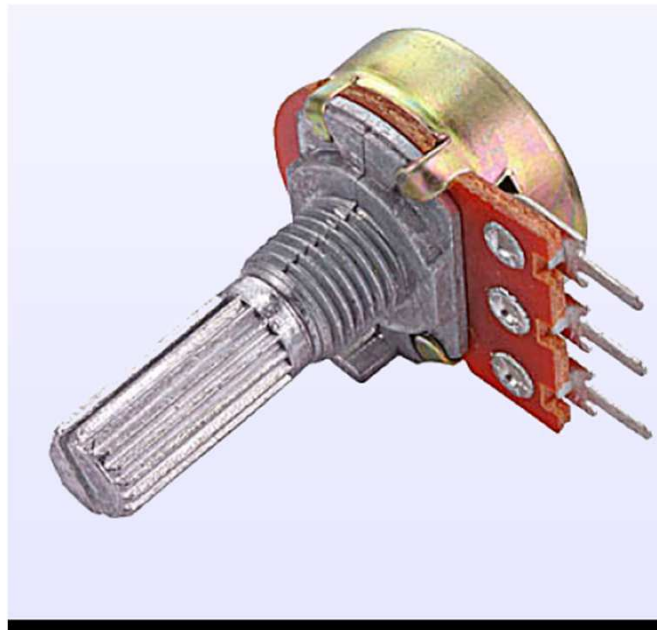
Lê o valor do pino analógico especificado. Isto significa que ele vai mapear tensões de entrada entre 0 e 5 volts em valores inteiros entre 0 e 1023. O intervalo de entrada e de resolução pode ser alterada utilizando [analogReference\(\)](#).

\*

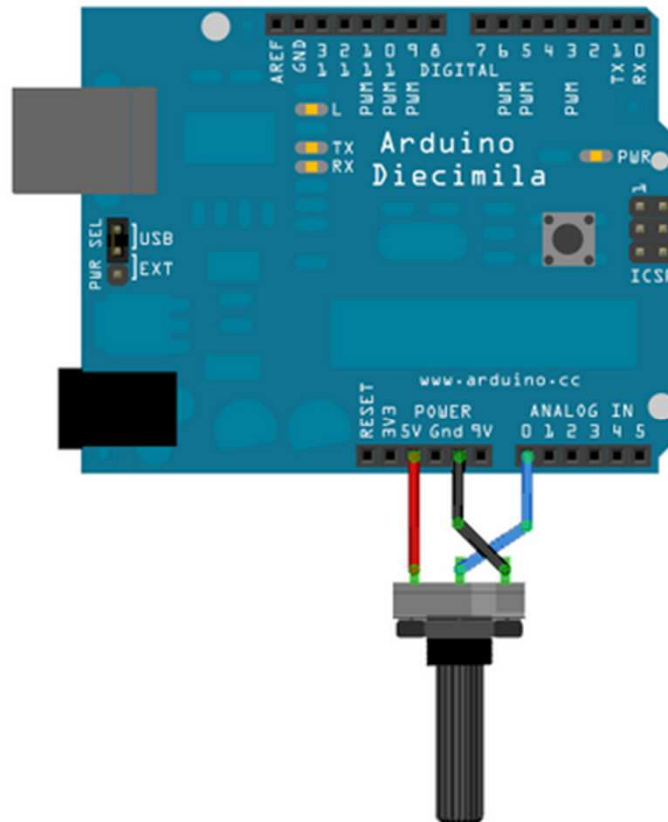
\*

# Potenciômetro

- Resistor variável



# Exemplo com potenciômetro



\*

\*

```
int potPin = A0; // selecione o pino de entrada ao potenciometro
int ledPin = 13; // selecione o pino ao LED
int val = 0;     // variavel para guardar o valor proveniente do sensor
```

```
void setup() {
  pinMode(ledPin, OUTPUT); // declarar o pino ledPin como saída
}
```

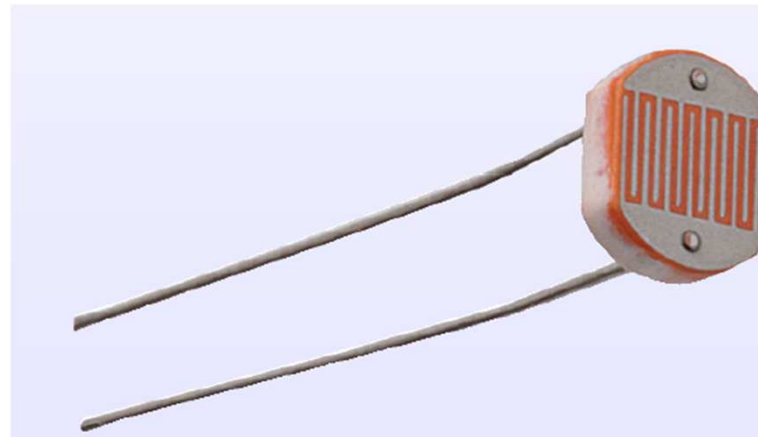
```
void loop() {
  val = analogRead(potPin); // ler o valor do potenciometro
  digitalWrite(ledPin, HIGH); // ligar o ledPin
  delay(val);                 // pausar o programa por algum tempo
  digitalWrite(ledPin, LOW); // desligar o ledPin
  delay(val);                 // pausar o programa por algum tempo
}
```

\*

\*

# LDR

- Resistor Dependente de Luz
- Converte a luz em valores de resistência
- Varia resistência conforme intensidade de Luz



# Projetos

1. Ligar LED - Exercício da unidade 02
2. Sensor de Luz LDR - Desafio

\*

\*