

CONCEITOS
BÁSICOS DE
LÓGICA DE
PROGRAMAÇÃO E
ALGORITMOS

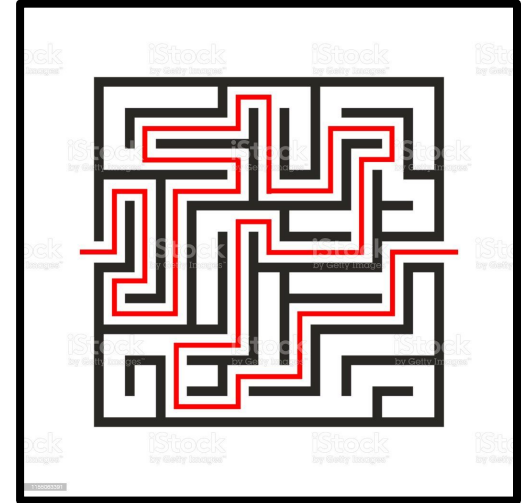
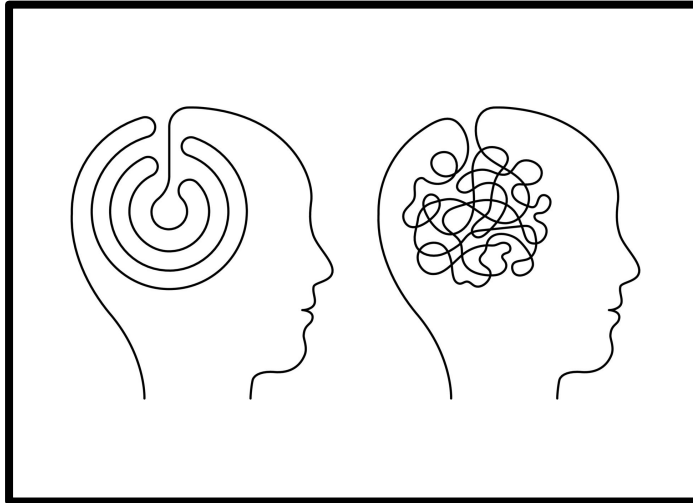
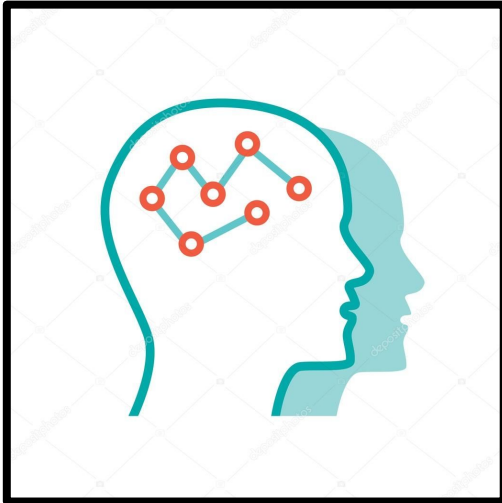
```
b.push(a[c]); } return b;
/\n|\r)/gm, " "), b = replace
for (var b = 0; b < a.length; b++)
push(inp_array[a]), b.push(inp_array[a]);
(a, " "); -1 < b
-1 < b && a.split(b) {
function use_array(a, b) {
for (var c = 0; c < a.length; c++)
= -1, d = a[c];
```

Plano de Apresentação

3	O que é Lógica
4	O que é Algoritmo
5	Exemplos de Algoritmo
9	Construindo Programas
10	Variáveis
11	Tipos mais utilizados de variáveis
13	Formação de identificadores (Nomes)
14	Pseudocódigos
15	Estrutura fundamental de um Pseudocódigo
17	Exemplo de um Pseudocódigo de Somatório
18	Pseudocódigos x Linguagem de Programação (C++)
22	Construindo um Programa - Exemplo 1 - Divisão
23	Construindo um Programa - Exemplo 2 - Média aritmética entre 2 números
24	Referências

O que é Lógica?

Lógica é uma maneira de nos direcionar a organizar o pensamento de forma ordenada, ou seja, é uma maneira correta de pensar, a qual segue uma linha sequencial de eventos.



Exemplos de Algoritmo

Exemplo 1: Apontar o lápis.

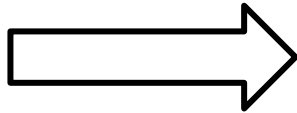
1º Passo: Pegar o lápis;

2º Passo: Pegar o apontador;

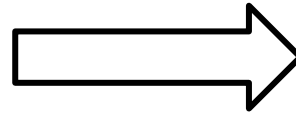
3º Passo: Apontar o lápis.



1º Passo



2º Passo



3º Passo

Exemplo 2: Beber refrigerante.

1º Passo: Abrir a geladeira;

2º Passo: Pegar a garrafa de refrigerante;

3º Passo: Fechar a geladeira;

4º Passo: Pegar um copo;

5º Passo: Colocar a garrafa de refrigerante em cima da mesa;

6º Passo: Colocar o copo em cima da mesa;

7º Passo: Abrir a garrafa de refrigerante;

8º Passo: Despejar o conteúdo da garrafa no copo;

9º Passo: Fechar a garrafa de refrigerante;

10º Passo: Beber o refrigerante do copo.

Exemplo 2: Beber refrigerante.

1º Passo: Abrir a geladeira;

2º Passo: Fechar a geladeira;

3º Passo: Pegar a garrafa de refrigerante;

4º Passo: Pegar um copo;

5º Passo: Colocar a garrafa de refrigerante em cima da mesa;

6º Passo: Colocar o copo em cima da mesa;

7º Passo: Abrir a garrafa de refrigerante;

8º Passo: Despejar o conteúdo da garrafa no copo;

9º Passo: Fechar a garrafa de refrigerante;

10º Passo: Beber o refrigerante do copo.

Exemplo 2: Beber refrigerante.

1º Passo: Abrir a geladeira;

2º Passo: Fechar a geladeira;

3º Passo: Pegar a garrafa de refrigerante;

4º Passo: Pegar um copo;

5º Passo: Colocar a garrafa de refrigerante em cima da mesa;

6º Passo: Colocar o copo em cima da mesa;

7º Passo: Abrir a garrafa de refrigerante;

8º Passo: Despejar o conteúdo da garrafa no copo;

9º Passo: Fechar a garrafa de refrigerante;

10º Passo: Beber o refrigerante do copo.

Analisando a nova ordem de eventos, podemos perceber que, no processo de construção de um algoritmo, se essa ordem estiver incorreta, a tarefa se torna impossível de ser realizada.

Nessa nova versão do exemplo, é impossível atingir o objetivo proposto (beber refrigerante), pois é impossível pegar a garrafa de refrigerante com a geladeira fechada.

Construindo Programas

Para construir um programa, é necessário saber:

> **O QUE** precisa ser feito;

- Entender a **lógica**, ou seja, por meio de uma sequência correta de ações, será planejado tudo aquilo que o sistema precisa executar.

> **COMO** isso será feito;

- Saber como aplicar sua lógica, por meio da construção de um **algoritmo**, em uma determinada linguagem de programação.

Variáveis

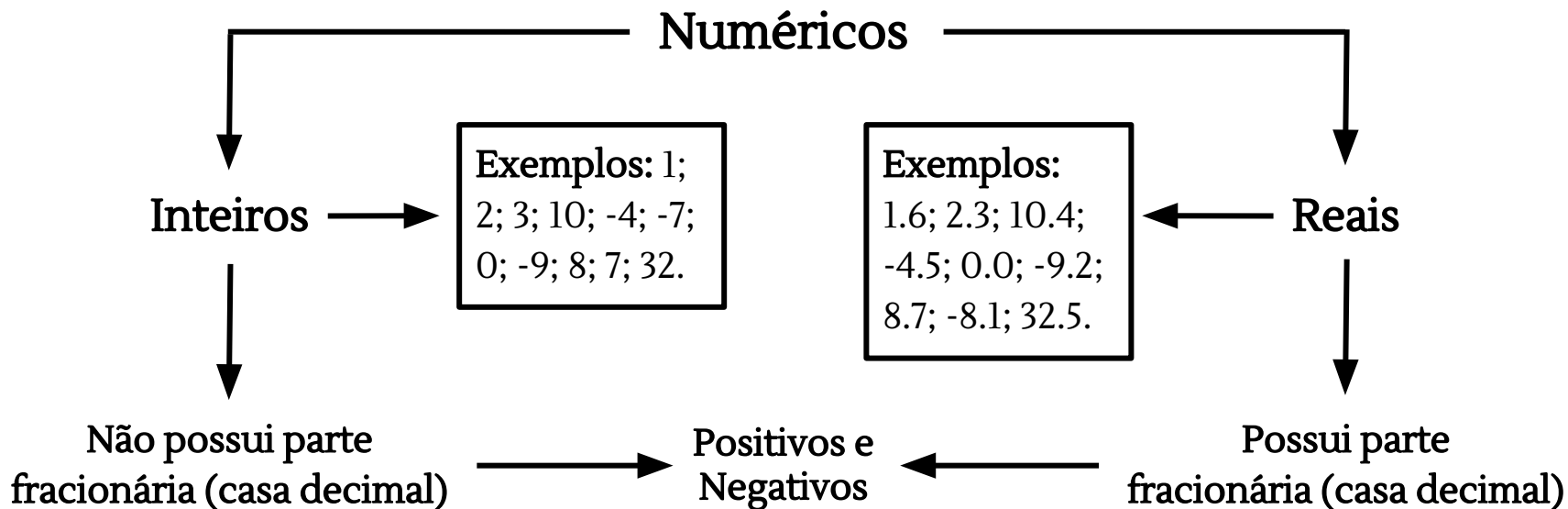
Um algoritmo e, posteriormente, um programa, recebem dados que precisam ser armazenados no computador para serem utilizados no processamento. Esse armazenamento é feito na memória e é representado por **variáveis**.

> Estrutura: **Nome** e **tipo**;

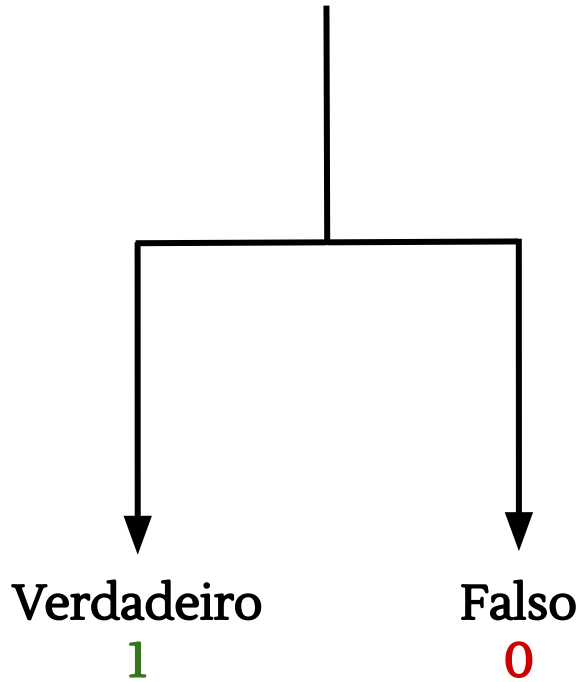
> Seu conteúdo pode variar ao longo do tempo durante a execução de um programa.

Tipos mais utilizados de variáveis

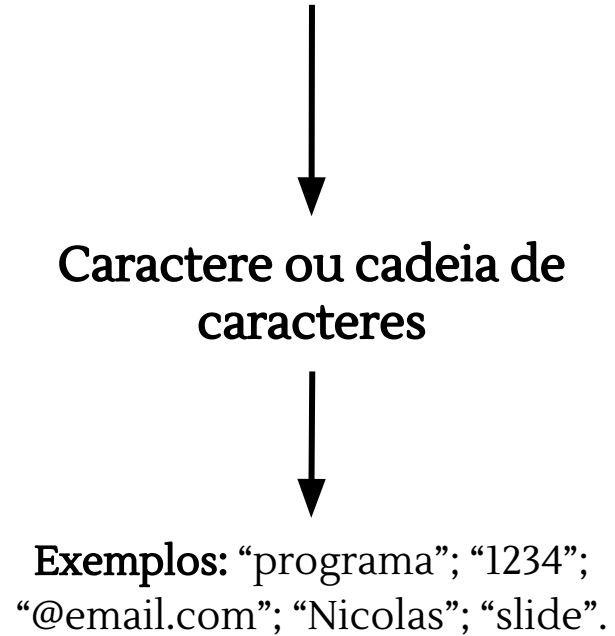
> Numéricos, lógicos e literais.



Lógicos
(Booleanos)



Literais (Strings)



Formação de Identificadores (Nomes)

> O primeiro caractere deve ser sempre uma letra ou o caractere sublinhado;

Exemplos: “Media”; “SOMA”; “_soma”; “X”; “y”; “Nome”; “IDADE”.

> Do segundo caractere em diante pode ser utilizado números;

Exemplos: “X5”; “Y64”; “_media2”; “nota_1”; “Matricula5”; “produto3”.

> Não são permitidos espaços em branco e caracteres especiais (@, \$, +, -, %, !);

Exemplos: “not@”; “\$resultado-0”; “dia%w”; “nome%”; “+Media”.

> Não podemos usar palavras reservadas nos identificadores, ou seja, palavras que pertençam a uma linguagem de programação.

Exemplos: “int”; “main”; “if”; “else”; “while”; “string”; “char”; “for”.

Pseudocódigos

Pseudocódigo é uma forma geral de escrever um algoritmo, utilizando uma linguagem universal (nativa a quem o escreve e entendida por qualquer pessoa) sem a necessidade de conhecer a sintaxe (conjunto de normas) de uma linguagem de programação.

> Exemplo:

Pseudocódigo para escrever uma mensagem

ALGORITMO Escrever

INÍCIO

ESCREVA("Olá Mundo")

FIM.

Estrutura fundamental de um Pseudocódigo

ALGORITMO

-> Nome do algoritmo;

DECLARE

-> Declaração de variáveis;

INÍCIO

-> Início do algoritmo;

-> Desenvolvimento do algoritmo;

FIM.

-> Fim do algoritmo.

Dando continuidade ao exemplo anterior...

> Pseudocódigo para escrever uma mensagem

ALGORITMO Escrever

INÍCIO

ESCREVA("Olá Mundo")

FIM.

Podemos apontar 2 funções básicas frequentemente utilizadas.

> **LEIA**(Variável)

> **ESCREVA**("Mensagem", Variável)

Exemplo de um Pseudocódigo de Somatório

ALGORITMO Somatório

DECLARE **X, Y, Z, SOMA**: INTEIRO

INÍCIO

LEIA(**X**)

LEIA(**Y**)

LEIA(**Z**)

SOMA ← **X** + **Y** + **Z**

ESCREVA (“A soma e:”, **SOMA**)

FIM.

O símbolo “←” tem a função de atribuir um valor a uma variável.

Pseudocódigos x Linguagem de Programação (C++)

ALGORITMO Soma

DECLARE X, Y, SOMA: INTEIRO

INÍCIO

LEIA(X)

LEIA(Y)

SOMA ← X + Y

ESCREVA("A soma é: ", SOMA)

FIM.

```
#include<bits/stdc++.h>
```

```
int main () {
```

```
    int X, Y, SOMA;
```

```
    scanf("%d", &X);
```

```
    scanf("%d", &Y);
```

```
    SOMA = X + Y;
```

```
    printf("A soma é: %d", SOMA);
```

```
    return 0;
```

```
}
```

Pseudocódigos x Linguagem de Programação (C++)

> **Bibliotecas:** São conjuntos de funções utilizadas para realizar determinados tipos de tarefas. São indicadas nas primeiras linhas de um programa. Utilizaremos a biblioteca “`bits/stdc++.h`” nos exemplos deste projeto.

O comando utilizado para indicar uma biblioteca é: `#include<bits/stdc++.h>`

> **Características:**

- C++ é “case sensitive” (Consegue diferenciar maiúsculas e minúsculas);
- Sempre que o código fonte for alterado, ele deve ser novamente compilado;
- Deve-se listar antecipadamente todas as variáveis utilizadas no programa;
- Assim como nos pseudocódigos, não podemos usar palavras reservadas nos identificadores.

Pseudocódigos x Linguagem de Programação (C++)

> Relação quanto a declaração dos tipos de dados:

Pseudocódigos	C++	Código em C++
INTEIRO	<code>int</code>	<code>%d</code>
REAL	<code>double</code> ou <code>float</code>	<code>%lf</code>
LITERAL	<code>char</code>	<code>%c</code> (Apenas 1 caractere) ou <code>%s</code> (Cadeia de caracteres)

Pseudocódigos x Linguagem de Programação (C++)

> Relação quanto às funções e operadores:

Pseudocódigos	C++
<code>LEIA(Variável)</code> <code>LEIA(X)</code>	<code>scanf("Tipo de variável", &Variável);</code> <code>scanf("%d", &X);</code>
<code>ESCREVA("Mensagem", Variável)</code> <code>ESCREVA("Valor final: ", X)</code>	<code>printf("Mensagem e tipo de variável", Variável);</code> <code>printf("Valor final: %d", X);</code>
<code>←</code> (Atribuição de valor a uma variável)	<code>=</code> (Atribuição de valor a uma variável)

Construindo um Programa - Exemplo 1 - Divisão

Pseudocódigo

ALGORITMO Divisão

DECLARE X, Y, D: REAIS

INÍCIO

LEIA(X)

LEIA(Y)

$D \leftarrow X / Y$

ESCREVA("A divisao e: ", D)

FIM.

C++

```
#include <bits/stdc++.h>

int main () {

    double X, Y, D;

    scanf("%lf", &X);

    scanf("%lf", &Y);

    D = X / Y;

    printf("A divisao e: %lf", D);

    return 0;

}
```


Construindo um Programa - Exemplo 2 - Média aritmética entre 2 números

Pseudocódigo

ALGORITMO Média

DECLARE X, Y, MEDIA: REAIS

INÍCIO

LEIA(X)

LEIA(Y)

MEDIA ← (X + Y) / 2

ESCREVA("A media e: ", MEDIA)

FIM.

C++

```
#include <bits/stdc++.h>
```

```
int main () {
```

```
    double X, Y, MEDIA;
```

```
    scanf("%lf", &X);
```

```
    scanf("%lf", &Y);
```

```
    MEDIA = (X + Y) / 2;
```

```
    printf("A media e: %lf", MEDIA);
```

```
    return 0;
```

```
}
```

Referências

RBtech. **Lógica de programação - Aula 01 - Introdução**. Youtube. Disponível em:

<<https://www.youtube.com/watch?v=Ds1n6aHchRU>>. Acesso em: 08/07/2022.

Dev Aprender. **Curso Lógica de Programação #1 - Intro e Algoritmos [FÁCIL]**. Youtube. Disponível em:

<<https://www.youtube.com/watch?v=ngD3sM-zJuE>>. Acesso em: 08/07/2022.

Professor: LeandroS.G.. **Aula 1º Algoritmo em Pseudocódigo**. Youtube. Disponível em:

<<https://www.youtube.com/watch?v=CStPeiAoYR0>>. Acesso em: 08/07/2022.

Marcos - Viver de Programação. **Introdução à Lógica de programação e algoritmos em C/C++**. Youtube. Disponível em:

<<https://www.youtube.com/playlist?list=PLMiLj8NCohLjCkF7-ve9eaXvjEwQZekqX>>. Acesso em: 08/07/2022.

GONÇALVES BOTELHO, Tiago. **Aula 1 - Introdução a lógica de programação; Aula 2 - Programa de Computador; . Aula 3 - Variáveis, tipos de dados, comandos E-S; Introdução a Linguagem C++**. Presencial IFSULDEMINAS (Muzambinho). Acesso em: 08/07/2022.