

# Conceitos básicos de algoritmos

## ➤ Operadores Aritméticos

➤ Unários: +, -

Exemplos:     +1  
                 -5.9

...

var a: inteiro

...

a <- -a

# Conceitos básicos de algoritmos

## ➤ Operadores Aritméticos

➤ Binários: +, -, \*, /, \, %, ^

### Associação

<i>Símbolo</i>	<i>Operação</i>
+	Soma
-	Subtração
*	Multiplicação
/	Divisão
\	Quociente da divisão inteira
%	Resto da divisão inteira
^	Potenciação

# Conceitos básicos de algoritmos

## ➤ Operadores Aritméticos

### ➤ Binários

## Precedência (Hierarquia nas operações)

Hierarquia	Operação
1	Parênteses
2	Função
3	-, + (unários)
4	^
5	*, /, \, %
6	+, -

# Conceitos básicos de algoritmos

## Expressões aritméticas

Exemplos:

$$3/4+5 = 5.75$$

$$3/(4+5) = 0.333333333$$

$$3\backslash 2*9 = 9$$

$$\underline{11\%3^2} = 2$$

$$\underline{11\%(3^2)} = 2$$

$$(11\%3)^2 = 4$$

$$3\backslash 2+(65-40)^{(1/2)} = 6$$

**Observação: a potenciação gera um valor real!  
Sendo assim, as expressões grifadas são inválidas.**

# Conceitos básicos de algoritmos

## ➤ Operadores Relacionais

Operador	Ação
>	maior que
>=	maior ou igual a
<	menor que
<=	menor ou igual a
=	igual a
<>	diferente de

# Conceitos básicos de algoritmos

## ➔ Operadores Lógicos

Operador
e
ou
nao
xou

# Conceitos básicos de algoritmos

## Expressões lógicas

Exemplos:

$3 > 7$  = FALSO

“A” = “a” = VERDADEIRO

“a” > “B” = FALSO

“Ana” < “Aline” = FALSO

$(3 \geq 13 \setminus 4) \text{ xou } (\text{nao } (5 \% 2 = 0))$  = FALSO

# Método para Construção de Algoritmos

**Os passos necessários para a construção de um algoritmo são:**

- ✦ ler atentamente o enunciado do problema, compreendendo-o e destacando os pontos mais importantes;
- ✦ definir os dados de entrada, ou seja, quais dados serão fornecidos;
- ✦ definir os dados de saída, ou seja, quais dados serão gerados depois do processamento;
- ✦ definir o processamento, ou seja, quais cálculos serão efetuados e quais as restrições para esses cálculos. O processamento é responsável pela obtenção dos dados de saída com base nos dados de entrada;
- ✦ definir as variáveis necessárias para armazenar as entradas e efetuar o processamento;
- ✦ elaborar o algoritmo;
- ✦ testar o algoritmo realizando simulações.



## Pseudocódigo

Com base nos conceitos vistos podemos aprofundar nosso estudo sobre os métodos de representação de algoritmos, iniciando pelo pseudocódigo.

Pseudocódigo é uma técnica textual de representação de um algoritmo. Também é conhecida como Português Estruturado ou Portugol.

Nele os verbos (ações) disponíveis para utilização são limitados e empregados no imperativo, deve-se evitar as expressões excessivamente longas, estas restrições visam eliminar a possibilidade de ambiguidade.

A técnica é baseada em uma PDL (Program Design Language), que é uma linguagem genérica na qual é possível representar um algoritmo de forma semelhante à das linguagens de programação.

## Pseudocódigo – Estrutura

A estrutura de um algoritmo em pseudocódigo pode variar um pouco de acordo com o autor ou com base na linguagem de programação que será utilizada posteriormente, mas essas variações ocorrem apenas na sintaxe, pois a semântica deve ser exatamente a mesma.

A estrutura que empregaremos para a construção de nossos pseudocódigos será a seguinte:

**algoritmo “nome”** //Tem como objetivo identificar o  
//algoritmo, deve-se utilizar um nome o mais significativo  
//possível, para facilitar a identificação

**var**

// Seção de Declarações - Neste ponto são informadas  
//quais variáveis, e seus respectivos tipos, serão  
// utilizadas no algoritmo

**inicio**

// Seção de Comandos - Aqui será escrita a sequência de  
// comandos que deve ser executada para solucionar  
// o problema em questão

**fimalgoritmo** //marca o final do algoritmo

## Pseudocódigo

As palavras reservadas presentes na estrutura do pseudocódigo apresentado foram:

**algoritmo**

**var**

**inicio**

**fimalgoritmo**

## Pseudocódigo – Exemplo

Observaremos agora um pseudocódigo que recebe um valor inteiro, fornecido pelo usuário, e o retorna no monitor.

```
algoritmo "exemplo 1"
```

```
var x: inteiro
```

```
inicio
```

```
    leia (x)
```

```
    escreva (x)
```

```
fimalgoritmo
```

Esta solução não é muito adequada.

Qual o motivo?

## Pseudocódigo – Exemplo

Outro exemplo é o pseudocódigo de um algoritmo que recebe um valor inteiro, acresce duas unidades a este, e exhibe o resultado desta manipulação.

algoritmo " exemplo 2.0 "

var numero, resposta: inteiro

inicio

    escreva ("Digite um número inteiro: ")

    leia (numero)

    resposta <- numero+2

    escreval ("Resultado (número + 2): ", resposta)

fimalgoritmo

## Pseudocódigo – Exemplo

Outro exemplo é o pseudocódigo de um algoritmo que recebe um valor inteiro, acresce duas unidades a este, e exhibe o resultado desta manipulação.

algoritmo " exemplo 2.1 "

var n: inteiro

inicio

    escreva ("Digite um número inteiro: ")

    leia (n)

    n <- n+2

    escreval ("Resultado (número + 2): ", n)

fimalgoritmo

## Pseudocódigo – Exemplo

Outro exemplo é o pseudocódigo de um algoritmo que recebe um valor inteiro, acresce duas unidades a este, e exhibe o resultado desta manipulação.

algoritmo " exemplo 2.2 "

var n: inteiro

inicio

    escreva ("Digite um número inteiro: ")

    leia (n)

    escreval ("Resultado (número + 2): ", n+2)

fimalgoritmo

## Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

algoritmo " exercício 1.1 "

var n1, n2, res: inteiro

inicio

    escreva ("Digite o multiplicando inteiro: ")

    leia (n1)

    escreva ("Digite o multiplicador inteiro: ")

    leia (n2)

    res <- n1 \* n2

    escreva ("Resultado da multiplicação: ", res)

fimalgoritmo



## Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

algoritmo " exercício 1.2 "

var n1, n2: inteiro

inicio

    escreva ("Digite o multiplicando inteiro: ")

    leia (n1)

    escreva ("Digite o multiplicador inteiro: ")

    leia (n2)

    escreva ("Resultado da multiplicação: ", n1\*n2)

fimalgoritmo

## Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

```
algoritmo " exercício 1.3 "
```

```
var n1, n2: inteiro
```

```
inicio
```

```
    escreva ("Digite o multiplicando e em seguida ")
```

```
    escreva (" o multiplicador (ambos inteiros): ")
```

```
    leia (n1, n2)
```

```
    escreva ("Resultado da multiplicação: ", n1*n2)
```

```
fimalgoritmo
```

## Pseudocódigo – Exercício 1

Construa um pseudocódigo para representar um algoritmo que efetue a multiplicação de dois inteiros quaisquer.

```
algoritmo "exercício 1.4"
```

```
var n1, n2: inteiro
```

```
inicio
```

```
    escreva ("Digite o multiplicando e em seguida ")
```

```
    escreva (" o multiplicador (ambos inteiros): ")
```

```
    leia (n1, n2)
```

```
    escreva (n1, "*", n2, "=", n1*n2)
```

```
fimalgoritmo
```

## Pseudocódigo – Exercício 2

Gere um pseudocódigo que aplique um desconto de 30% sobre o valor de um produto, recebido como entrada, e retorne o resultado da manipulação na saída padrão.