



# **1001350 - Construção de Algoritmos e Programação**

## **Aula 2 - 02/04/2025**

**Joice Otsuka**

**Algoritmos - conceitos básicos**

---

# Algoritmos



# Algoritmos

- Sequência de instruções finita e ordenada de forma lógica para a resolução de uma determinada tarefa ou problema
- Caracterização
  - Cada **passo** deve ser uma **instrução** que possa ser realizada
  - Precisa ter **início** e **fim**
  - A **ordem** dos passos deve ser precisamente determinada
  - Não pode haver ambiguidade

# Algoritmos - Clareza e precisão

---



## Clareza e precisão

Com uma caneta (e não lápis, para não poder apagar), execute as seguintes instruções:

1. Desenhe cinco círculos lado a lado
2. Escreva dentro dos círculos as letras de A a E

**Essas orientações estão  
suficientemente claras e precisas??**

**Algoritmos - A ordem das  
instruções é importante!**

---



## Sequência das instruções

Execute as instruções:

O que foi desenhado no centro ??  
A ordem das instruções está adequada?

1. Desenhe um quadrado
2. Desenhe um círculo no centro do quadrado
3. Desenhe um losango no canto superior esquerdo do quadrado
4. Desenhe um círculo no canto inferior direito do quadrado
5. Escreva sua idade no canto inferior esquerdo do quadrado
6. Escreva o ano atual no canto superior direito do quadrado
7. Calcule o valor de  $e = i + 1$ , sendo  $i$  sua idade em anos
8. Se  $e$  for ímpar, ao invés do círculo, desenhe um triângulo no centro



# Sequência das instruções

Considere o algoritmo:

1. Pergunte os nomes de seus colegas
2. Anote quantos começam com vogal

**Será que a sequência descrita é uma solução eficiente???**





## Algoritmos informais

Em grupo de até 3 pessoas, escrevam um algoritmo que ensine uma pessoa a identificar o nome do aluno da turma que mora mais distante de São Carlos.

Não se preocupe com a notação mais formal, foque em escrever um algoritmo que funcione.

---

# Representação de algoritmos computacionais



# Algoritmos computacionais

- Vimos que a linguagem natural não é adequada para elaborar algoritmos para computadores
  - Pouca rigidez sintática e semântica
  - Problemas de ambiguidade
- As linguagens de programação são adequadas, mas pode ser complexo
- Soluções intermediárias
  - Pseudocódigo
  - Fluxogramas



# Representação de algoritmos

- Pseudocódigo: representação usando uma linguagem estruturada, menos rígida que uma linguagem de programação
- Fluxogramas: diagramas com formas bem definidas para a representação do fluxo de execução



# Passos para a elaboração de um algoritmo

1. Entenda o problema
2. Identifique as entradas
3. Identifique as saídas
4. Defina os passos a serem seguidos
  - Identifique as regras e restrições do problema
  - Determine as ações que podem ser realizadas pelo agente alvo do algoritmo para solucionar o problema



# Pseudocódigo

- Utiliza uma linguagem flexível, intermediária entre a linguagem natural e a linguagem de programação
- ‘Pseudocódigo’ ou ‘falso código’; o nome se deve à proximidade que existe entre um algoritmo escrito em pseudocódigo e a maneira pela qual um programa é escrito em uma linguagem de programação de alto nível



# Pseudocódigo

- Especificar entradas (estado inicial) e saídas (estado final)
- Instruções claras e precisas
  - Não usar palavras com duplo sentido
  - Frases simples (e curtas)
  - Colocar comentários sempre que necessário
- A sequência das instruções é importante



# Visualg

O *VisuAlg* é uma ferramenta que permite criar, editar, interpretar e que também executa os algoritmos em pseudocódigo usando Portugol (português estruturado ).

Download: <https://sourceforge.net/projects/visualg30/>

[Manual do VisuAlg \(CEFET - MG\)](#)



---

# Algoritmos - Variáveis, tipo de dados e operações



## Dados e variáveis

- Dados são os valores que serão utilizados na resolução de um problema
  - Esses valores podem ser fornecidos pelo usuário do programa
  - Podem ser originados a partir de processamentos (cálculos)
  - Ou a partir de arquivos, bancos de dados ou outros programas
- Os dados são armazenados temporariamente em variáveis para que sejam processados de acordo com as especificações do algoritmo.



# Variável

- Espaço na memória do computador, reservado para o armazenamento de uma informação
- A informação armazenada pode ser alterada
- Cada variável precisa ter:
  - Um nome (identificador)
  - Tipo de dado



# Identificação de variáveis

- O nome de uma variável deve ser único e deve seguir algumas regras:
  - ser sucinto e utilizar nomes significativos
  - não utilizar espaços entre as letras
    - nome do cliente (errado!)
    - o correto é nome\_do\_cliente, nomeCliente
  - não iniciar o nome da variável com números
    - 2valor (errado)
    - o correto é valor2
  - não utilizar caracteres especiais, como símbolos ( ? / : @ # )
  - não utilizar palavras reservadas



# Tipos de dados básicos

- Inteiros:
  - 10
  - 2200
  - -4
  - 0...
- Reais: possuem parte inteira e parte fracionária
  - 3.5
  - 1.0
  - -5.699



## Tipos de dado básicos

- String (conjunto de caracteres)
  - Representa valores textuais
  - "Brasil", "2.6", "nome", "e-m@il"
- Lógico
  - True, False
  - Verdadeiro,Falso



## Tipos de dados no Visualg

- **inteiro:** define variáveis numéricas do tipo inteiro
- **real:** define variáveis numéricas do tipo real, ou seja, com casas decimais.
- **caractere:** define variáveis do tipo string, ou seja, cadeia de caracteres.
- **logico:** define variáveis do tipo booleano, ou seja, com valor VERDADEIRO ou FALSO.



## Declaração de variáveis no Visualg

```
Algoritmo "Teste"  
// Descrição do algoritmo  
// Autores:  
Var  
    num1, num2: inteiro;  
    nome: caractere;  
Inicio  
...  
FimAlgoritmo
```



---

# Algoritmos - Especificação de entradas e saídas



## Especificação das entradas e saídas

Dados os comprimentos  $a, b, c$  de 3 segmentos retas, verificar se as retas podem formar um triângulo. Forneça as especificações de entrada e saída para o problema. Não precisa escrever o algoritmo.

**Entradas:** os comprimentos  $a, b, c$

**Saída:** verdadeiro, se formarem um triângulo; falso, caso contrário



## Especificação das entradas e saídas

É preciso fazer o cálculo da velocidade média (em km/h) de um percurso feito por um veículo, dada a distância percorrida (em quilômetros) e o tempo gasto (em horas).

Forneça as especificações de entrada e saída para o problema. Não precisa escrever o algoritmo.



# Representação de entrada de dados

- Algumas representações comuns em algoritmos:
  - Obtenha um valor para  $x$  dado pelo usuário
  - No Visualg
    - Leia  $x$  // lê o valor digitado pelo usuário e armazena em  $x$



# Representação saída de dados

- Algumas representações comuns em algoritmos:
  - Apresente o valor de `y`
  - No Visualg
    - Escreva `y` //escreve na tela o valor armazenado em `y`



# Atribuição de valor a uma variável

- Algumas representações comuns:
  - Armazene em **x** o valor -5 ou
  - Atribua à **x** o valor -5 ou
  - No Visualg
    - `x ← -5` //armazena em x o valor inteiro -5



# Operadores

- Aritméticos
- Relacionais
- Lógicos



# Operadores aritméticos

- Utilizados para a realização dos diversos cálculos matemáticos
- Unários:
  - - (negativo)
- Binários:
  - + (adição)
  - - (subtração)
  - \* (multiplicação)
  - / (divisão)
  - \ (divisão inteira) Ex:  $9 \setminus 2 = 4$
  - % ou MOD (resto da divisão inteira) Ex:  $9 \% 2 = 1$
  - ^ (potenciação) Ex:  $2^3 = 8$





# Expressões aritméticas

- Operações são realizadas da esquerda para a direita
- Ordem de precedência:
  1. Expressões entre parênteses
  2. Potenciação
  3. Multiplicação, divisão, resto
  4. Soma e subtração



## Exercício

1. Escreva um algoritmo no Visualg que leia dois números inteiros e imprima a soma dos números lidos
2. Escreva um algoritmo no Visualg que leia o lado de um quadrado e calcule e imprima a sua área



Algoritmo "Soma"

// Descrição : Algoritmo que lê dois números  
// e imprime a soma dos números

Var

num1: inteiro;

num2: inteiro;

Inicio

leia(num1)

leia(num2)

escreva(num1 + num2)

Fimalgoritmo



Algoritmo "AreaQuadrado"

// Descrição : Algoritmo que lê um lado de um  
// quadrado e imprime a área

Var

lado: real;

area: real;

Inicio

leia (lado)

area <- lado \* lado

escreva (area)

Fimalgoritmo



## Exercícios

1. Escreva um algoritmo que receba do usuário a quantidade de dias, horas, minutos e segundos em valores inteiros. Calcule e imprima o total de segundos correspondente.

Exemplo de Entrada	Saída Esperada
1 2 3 4	93784



## Exercícios

- João quer saber o valor do seu novo salário, após o seu aumento anual. Para ajudar João, crie um algoritmo que leia o valor atual do salário de João e a taxa de aumento (%) e calcule e apresente o valor do novo salário.

Exemplo de Entrada	Saída Esperada
1500.00 5	1575.00
1495.00 3	1539.85



## Observação: Formatação de saída no Visualg

Para variáveis reais, pode-se especificar o número de casas decimais que serão exibidas.

Por exemplo, considerando `y` como uma variável real, o comando `escreva(y:6:2)` escreve o valor armazenado na variável `y` em 6 espaços, colocando 2 casas decimais. (Fonte: Manual do VisualAlg 3)



## Exercícios

3. Escreva um algoritmo para calcular e imprimir o número de anos, meses e dias a partir de um número de dias informado pelo usuário. Apenas para facilitar o cálculo, considere todo ano com 365 dias e todo mês com 30 dias. (*Obs: por conta da simplificação, o algoritmo não funcionará para 360 a 364 dias*)

Exemplo de Entrada	Saída Esperada
400	1 ano(s) 1 mes(es) 5 dia(s)
800	2 ano(s) 2 mes(es) 10 dia(s)





## Exercícios

4. Escreva um algoritmo para calcular a redução do tempo de vida de um fumante. O usuário deverá informar a quantidade de cigarros fumados por dia e quantos anos ele já fumou. Considere que um fumante perde 10 minutos de vida a cada cigarro, calcule e imprima a quantidade de dias de vida que o fumante perdeu. Considere todos os anos com 365 dias.

Exemplo de Entrada	Saída Esperada
10 10	253 dias



## Algoritmos - Organização visual

1. Instruções de mesmo nível devem iniciar na mesma coluna
2. Quando houver quebra de linha (linhas longas), deve haver recuo à direita a partir da segunda linha
3. Linhas em branco devem ser utilizadas para separar blocos lógicos de instruções
4. Incluir comentários, para documentar e para facilitar o entendimento do código



Vídeo YouTube

Seu cérebro pode resolver algoritmos - David J. Malan

<https://youtu.be/6hfOvs8pY1k>



Vídeo no youtube: O que é e como funcionam os algoritmos.

Roberto Marcondes Cesar Junior, do IME-USP, e Sérgio Amadeu da Silveira, do CECS-UFABC, explicam como e por que os algoritmos têm impacto crescente no cotidiano.

[https://youtu.be/Xo1V\\_JL1yAg](https://youtu.be/Xo1V_JL1yAg)