



PECS | Programa de Pós-Graduação em
Engenharia de Computação e Sistemas

ALGORITMOS E ESTRUTURAS DE DADOS

Disciplina: ALGORITMOS E ESTRUTURAS DE DADOS

Carga Horária: 60h

Professor: **Dr. Reinaldo**

Algoritmo Básico e Programação

Os algoritmos fazem parte do dia-a-dia das pessoas.

Um algoritmo pode ser visto como uma sequência de ações executáveis para a obtenção de uma solução para um determinado tipo de problema.

Conceito de Algoritmo

“Algoritmo é uma sequência de passos que visa atingir um objetivo bem definido.” (Forbellone, 1999)

“Algoritmo é a descrição de uma sequência de passos que deve ser seguida para a realização de uma tarefa.” (Ascencio, 1999).

Algoritmo

Segundo Cormen (2002), um algoritmo é qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como **entrada** e produz algum valor ou conjunto de valores como a **saída**



Algoritmo

Segundo Dijkstra, um algoritmo corresponde a uma descrição de um padrão de comportamento, expresso em termos de um conjunto finito de ações.



Algoritmos

Os algoritmos são utilizados para resolver diversos tipos de problemas, por exemplo:

- Comércio Eletrônico;
- Instruções para o uso de medicamentos;
- Indicações de como montar um aparelho;
- Entre outros.

Estrutura de Dados

Quando os dados obtidos na entrada do algoritmo são dispostos e manipulados de forma homogênea no processo de computação de sua saída, trata-se de *tipo abstrato de dados*.



Estrutura de Dados

Uma estrutura de dados é um meio para armazenar e organizar dados com o objetivo de facilitar o acesso e as modificações (Cormen, 2002).

Estrutura de Dados

Todos os problemas a serem resolvidos por algoritmos **possuem dados**. Estes são armazenados em **estruturas**, escolhidas de acordo com as operações que podem ser realizadas sobre elas e com o custo de cada uma dessas operações.

Estrutura de Dados

Estruturas de dados e algoritmos estão intimamente ligados:

- **não se pode estudar estruturas de dados** sem considerar os algoritmos associados a elas,
- **assim como a escolha dos algoritmos em geral** depende da representação e da estrutura dos dados.

Estrutura de Dados

Para resolver um problema é necessário escolher uma **abstração da realidade**, em geral mediante a definição de um conjunto de dados que representa a situação real.

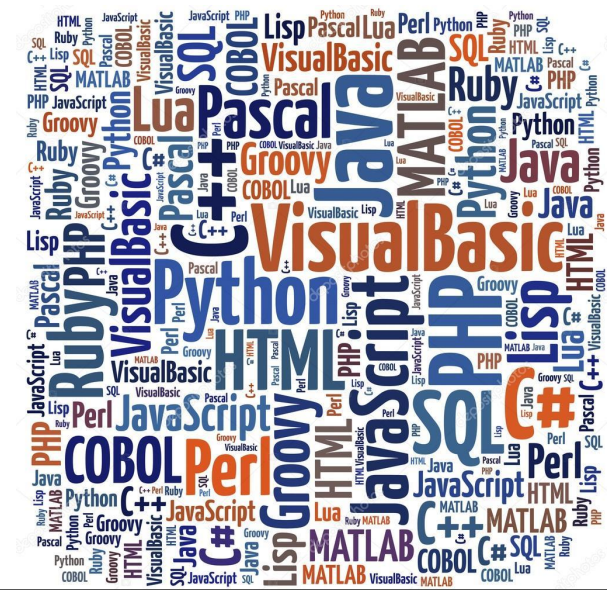
Programas

Programar é basicamente estruturar dados e construir algoritmos.

Programas são formulações concretas de algoritmos abstratos, baseados em representações e estruturas específicas de dados.

Programas representam uma classe especial de algoritmos capazes de serem seguidos por computadores.

- Análise - Enunciado do problema
- Algoritmo – Descrever o problema com suas soluções
- Codificação – transformação em uma linguagem de programação



Descrever algoritmo

- Somar dois números
- Trocar uma lâmpada
- Ir para UEMA
- Troca pneu de um carro
- Saque dinheiro caixa eletrônico



Método para a construção de Algoritmo

Para construção de algoritmo são necessários os seguintes passos:

- Compreender o problema
- Definir os dados de entrada
- Definir Processamento
- Definir os dados de saída
- Construir algoritmo
- Testar o algoritmo realizando simulação

Tipos de algoritmos

- Descrição Narrativa
- Fluxograma
- Pseudocódigo ou Portugol

Exemplo

- Descrição Narrativa

Passo 1: Receber os dois número que serão somado

Passo 2: Soma os números

Passo 3: Mostrar o resultado obtidos na soma

Vantagem: linguagem natural.

Desvantagem: abre espaços para várias interpretação.

Exemplo

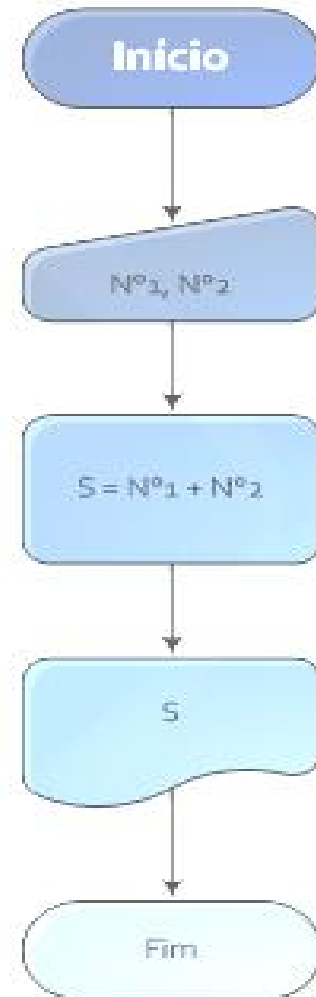
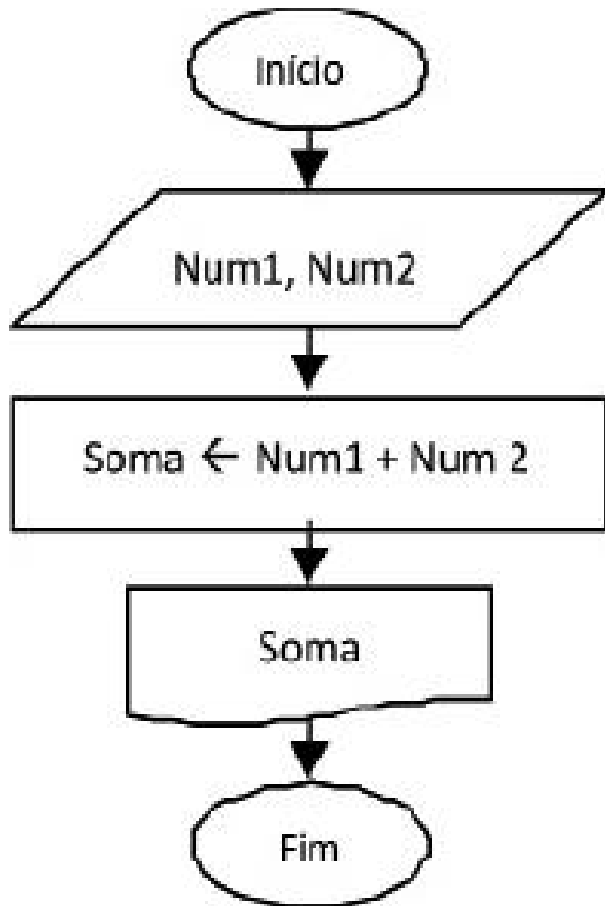
- Fluxograma

Consiste em analisar o enunciado do problema e escrever, utilizando símbolos gráficos pre-definidos.

Vantagem: entendimento do elemento gráficos é mais simples que o entendimento de textos.

Desvantagem: é necessário aprender a simbologia.

Exemplo



Exemplo

- Pseudocódigo ou Portugol – Consiste em analisar e escrever por meio de regras predefinidas.


Vantagens – basta conhecer as palavras reservadas.

Desvantagens – é necessários aprender as regras do pseudocódigo.

Exemplo

VisuAlg 2.0 - Semnome.alg

Arquivo Editar Exibir Algoritmo Código Ferramentas Ajuda



```

algoritmo "somanumero"
// Função :
// Autor :
// Data : 23/08/2012
// Seção de Declarações
var
    n1, n2, soma: inteiro
inicio
// Seção de Comandos
escreva ("Digite 1 Numero..:")
leia (n1)
escreva ("Digite 2 Numero..:")
leia (n2)
soma:=n1+n2
escreva ("soma..:",soma)

finalgoritmo
    
```

Escopo	Nome	Tipo	Valor
GLOBAL	N1	I	89
GLOBAL	N2	I	3
GLOBAL	SOMA	I	92

Conceitos Básicos

Variável

São espaços reservados na memória do computador para guardar informações.

Tipos de Dados

Os tipos de dados influenciam na forma como o algoritmo irá trabalhar, o desempenho do algoritmo e o seu consumo de memória. Os tipos de dados mais utilizados são:

- Numéricos
- Lógicos
- Literais ou Caracteres

Tipos de dados

- Numéricos - os dados numéricos dividem em dois grupos: inteiros e reais.
- Lógicos – são também chamados dados booleanos (por causa da álgebra de *Boole*) e pode assumir os valores verdadeiro ou falso.
- Literais ou Caracteres – são dados formados por um único caractere ou por uma cadeia de caracteres.

Nomes de Variáveis

Para criar nomes de variáveis, é necessário seguir algumas regras básicas de sintaxe:

- Todo nome deve ser composto apenas por letras, números e underline (_)
- Deve começar com uma letra
- Não pode ser igual a nenhuma palavra reservada.

Sendo assim, são nomes válidos de variáveis:

- Valor
- Num1

São nomes inválidos:

- 1Num
- a 10

Atribuição

É utilizado para atribuir valores ou operações a variáveis, sendo representado pelo símbolo \leftarrow .

Exemplo

$$X \leftarrow 5$$
$$Y \leftarrow X + 5$$

Estrutura Sequencial

Estrutura Sequencial em Algoritmo

Exemplo

Algoritmo

Declare

Bloco de comando

Fim_Algoritmo

Estrutura Sequencial

Algoritmo <Nome do Algoritmo>

Declare

X numerico

Nome literal

Tipo lógico

//Atribuição

$X \leftarrow 10$

Nome \leftarrow "Engcomp"

Tipo \leftarrow verdadeiro

Escreva ("Digite o nome")

Leia Nome

Escreva (Nome)

Fim_Algoritmo

Exercício

Desenvolva um programa que receba o valor de um depósito e o valor da taxa de juros, calcule e mostre o valor do rendimento e o valor total depois do rendimento.

Estrutura Condicional

- Simples
- Composta
- Encadeada

Estrutura Condicional Simples

O comando só será executado se a condição for verdadeira.

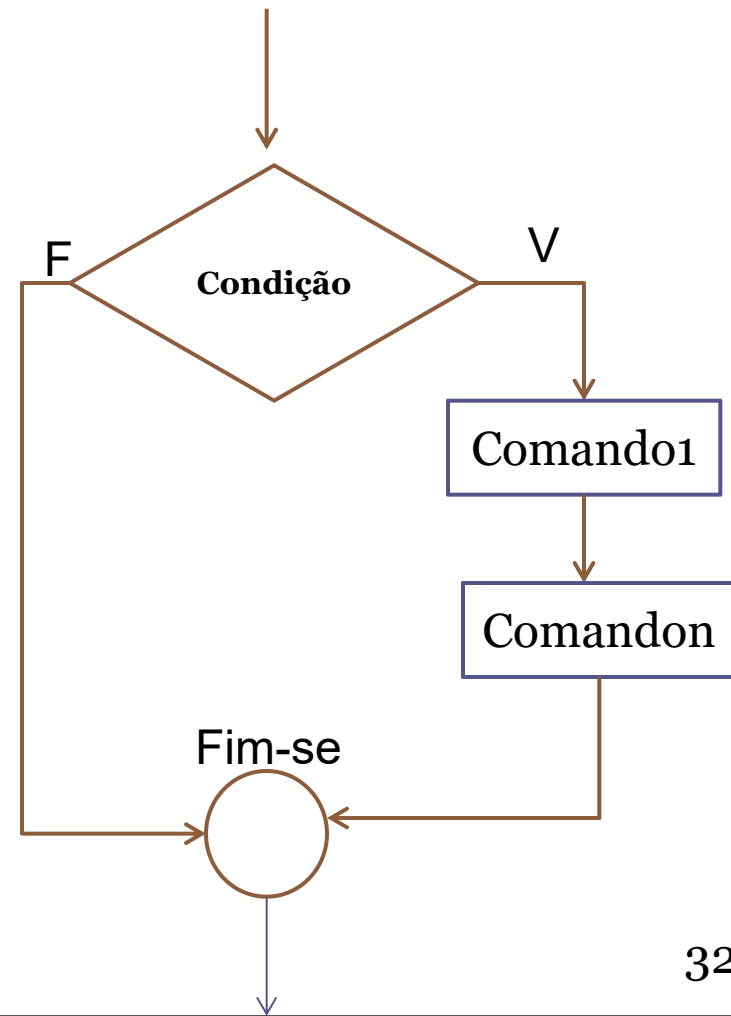
Ex:

Se <condição> então

Comando 1..

Comando n..

Fim-se



Estrutura Condicional Composta

Se a condição for verdadeira, serão executado o comando1 e comando2; caso contrario, serão executado comando3 e comando4.

Ex: Se <condição> então

Comando 1..

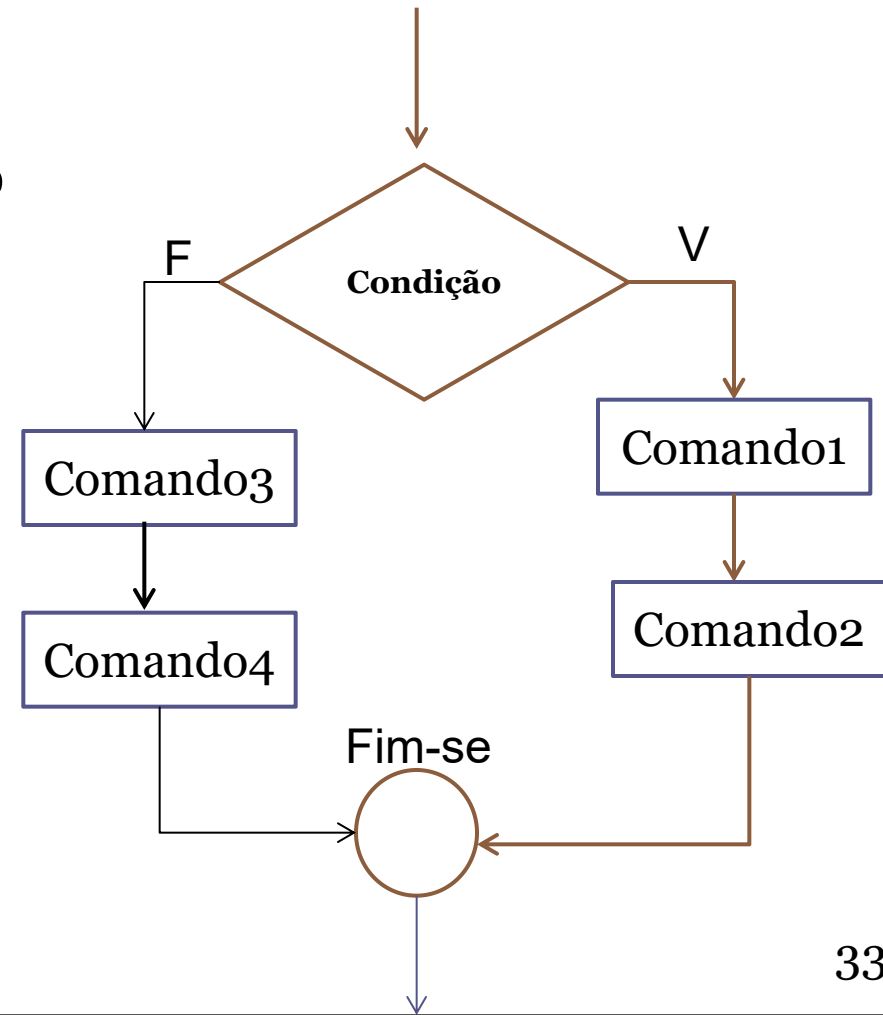
Comando 2..

Senão

Comando 3..

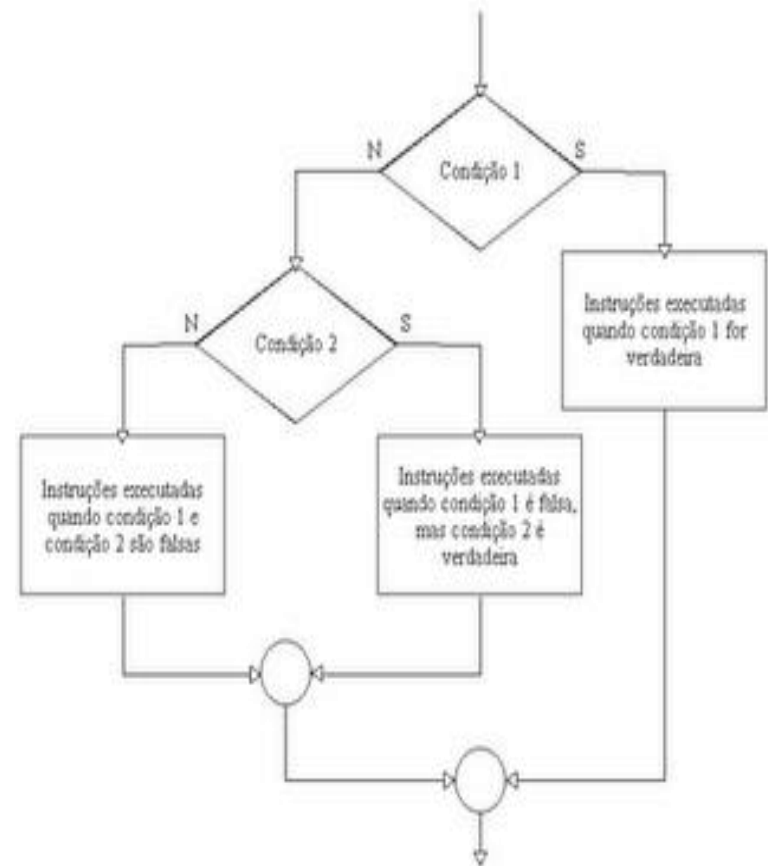
Comando 4..

Fim-se



Estrutura Condicional Encadeada

A instrução condicional pode ser encadeada **se** dentro de outro **se** mais externo.



Estrutura Condicional Múltipla(Seleção)

É equivalente à estrutura condicional composta aninhada, que permite a execução de opções mutuamente exclusivas, isto é, se uma situação for executada, as demais não serão

Exemplo Estrutura Condicional Múltipla (Seleção)

Escolha <seletor>

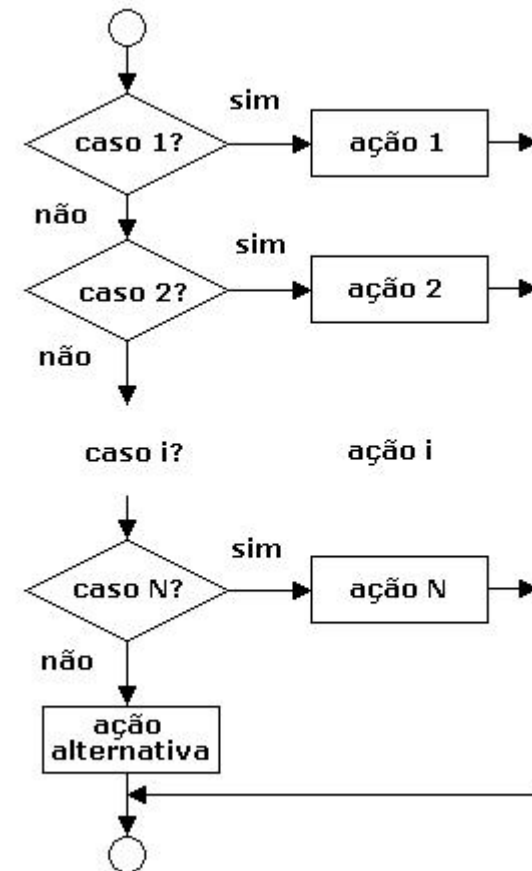
caso <exp1><bloco-comandos-1>

Caso <exp2><bloco-comandos-2>...

Outro caso

<bloco-comandos-extra>

Fimescolha

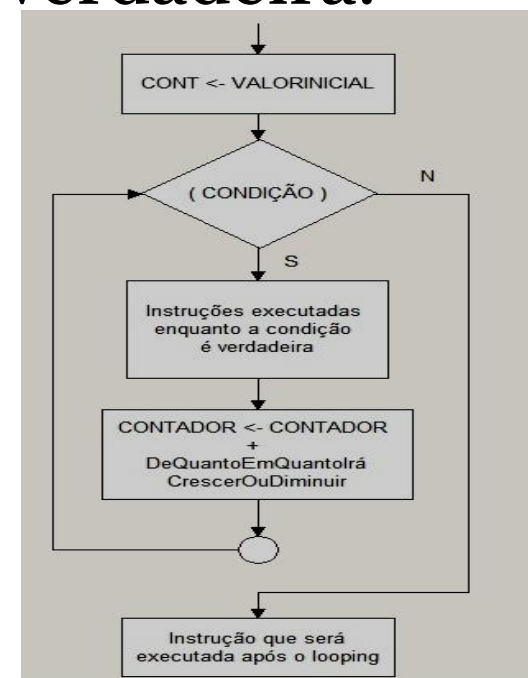


Estrutura Repetição

Enquanto - a estrutura de repetição mais simples.
Ele repete a execução de um bloco de sentenças enquanto uma condição permanecer verdadeira.

Sintaxe

Enquanto <condição> Faça
bloco de código
Fim Enquanto



Estrutura Repetição

Repita - Esta estrutura tem um comportamento muito semelhante ao Enquanto, com uma diferença: a condição é verificada após executar o bloco de instruções correspondente.

Sintaxe

Repita

(bloco de código)

até (condição)



Estrutura Repetição

Para- Esta estrutura algoritmos necessitam executar um bloco de sentenças por um número específico de vezes.

Sintaxe

Para (Variável) De (início) Até (final) Passo (p)

Faça

(bloco de código)

Fim Para

Referência

- FARRER, H. et alii. Algoritmos Estruturados. 2. ed. Rio de Janeiro: LTC, 1989.
- GUIMARÃES, Angelo de Moura. Algoritmos e Estruturas de Dados. Rio de Janeiro: LTC, 1985.
- SALIBA, Walter Luiz Caram. Técnicas de Programação. São Paulo: Makron, McGrawHill, 1992.
- MIZRAHI, Victorine Viviane. Treinamento em Linguagem C – Módulo 1. São Paulo: McGrawHill, 1990.
- MIZRAHI, Victorine Viviane. Treinamento em Linguagem C – Módulo 2. São Paulo: McGrawHill,
- Kernighan, B. & Ritchie, D. C - A linguagem de programação padrão ANSI. Editora Campus, 1990.
- Schildt, H. Turbo C - guia do usuário . Editora McGraw-Hill, 1988.
- Schildt, H. C - completo e total . Editora McGraw-Hill, 1990.
- Carpenter, V. Learn C/C++ today : <http://www.cyberdiem.com/vin/> (uma coleção de referências e tutoriais sobre as linguagens C e C++ disponíveis na Internet)
- Mizrahi, V. V. Treinamento em linguagem C - Módulos 1 e 2. Editora McGraw-Hill, 1990.
- Apostila UFMG,
ftp://ftp.sm.ifes.edu.br/professores/EduardoSilva/Engenharia/1895363640_apostila-c-ufmg.pdf