

TMEC001 - CÁLCULO NUMÉRICO

CAPÍTULO 01 – MODELAGEM COMPUTACIONAL E MATEMÁTICA E ANÁLISE DE ERROS

Prof. Felipe R. Loyola
Disciplina: Cálculo Numérico
1º Semestre de 2020

1.2 – Programação

1.2.1 – Programas Computacionais

- Programas computacionais são, basicamente, um **conjunto de instruções** que direcionam o computador para executar certas **tarefas**. Uma vez que muitos indivíduos escrevem programas para um **amplo conjunto de aplicações**, muitas linguagens de computação de alto nível têm grandes recursos. Pode-se, contudo, reduzir a complexidade de um código para um conjunto limitado de tópicos de programação. Estes são:

1.2.1 – Programas Computacionais

- Representação **simples** da informação (constantes, variáveis e declarações de tipo);
- Representação de informação **avançada** (estrutura de dados, vetores e registros);
- Fórmulas **matemáticas** (atribuições, regras de prioridade e funções intrínsecas);
- Entrada / saída;
- Representação **lógica** (sequência, seleção e repetição);
- Programação **modular** (funções e sub-rotinas)

1.2.2 – Programação Estruturada

- Nos primórdios da programação, os programadores usualmente não prestavam muita atenção ao fato de o programa ser **claro** e **fácil de entender**. Hoje é reconhecido que existem muitos **benefícios** em escrever um código bem organizado e bem estruturado. Além do benefício óbvio de tornar o *software* mais **fácil** de ser **compartilhado**, isso também ajuda a garantir um **desenvolvimento** de programa **mais eficiente**.

1.2.2 – Programação Estruturada

- Essencialmente, **programação estruturada** é um **conjunto de regras** que exige da **disciplina** do programador. Embora a programação estruturada seja **flexível** o suficiente para permitir considerável **criatividade** e expressões pessoais, suas regras impõem **restrições** suficientes para conferir ao código resultante grande **superioridade** comparada à versão não-estruturada em particular, o produto final é **mais elegante e fácil de entender**.

1.2.2 – Programação Estruturada


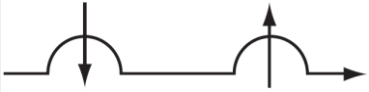


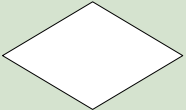

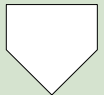

- A ideia por trás da **programação estruturada** é que qualquer algoritmo numérico pode ser composto usando-se as **três estruturas** de controle **fundamentais**:
 - Sequência
 - Seleção
 - Repetição
- Caso se limite a essas estruturas, o código computacional resultante será claro e fácil de seguir.

1.2.2 – Programação Estruturada

Para se manter uma descrição genérica, empregam-se fluxogramas e/ou pseudocódigos

- Fluxograma: **representação gráfica** visual de um algoritmo, emprega uma **série** de **blocos** e **flechas**, cada um dos quais representa uma **operação particular** ou **passos** de um algoritmo
- Pseudocódigo: técnica que utiliza **palavras semelhantes** a um código no lugar dos símbolos gráficos do fluxograma.

1.2.2 – Programação Estruturada

Símbolo	Nome	Função
	Terminal	Representa o início e o fim do programa
	Linhas de Fluxo	Representa o fluxo da lógica. Os morros das flechas horizontais indicam cruzamentos sem que haja conexão com o fluxo vertical
	Processos	Representa cálculos ou manipulação de dados
	Entrada / Saída	Representa entrada ou saída de dados e informação
	Decisão	Representa uma comparação, questão ou decisão que determina caminhos alternativos a serem seguidos
	Junção	Representa a confluência de linhas de fluxo
	Conector de página	Representa uma quebra que é continuada em outra página
	Contadores controladores de laços	Usados para ciclos que se repetem por um número pré-determinado de iterações

1.2.3 – Programação Lógica

- Sequência: A estrutura da sequência expressa a ideia trivial de que, exceto se direcionado de outra forma, o código computacional é implementado uma instrução por vez.

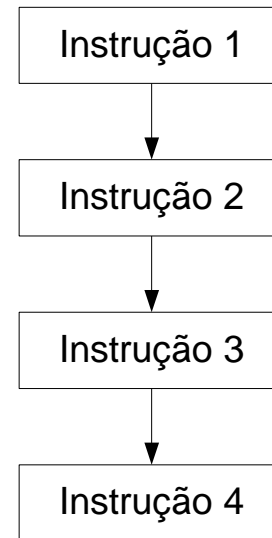
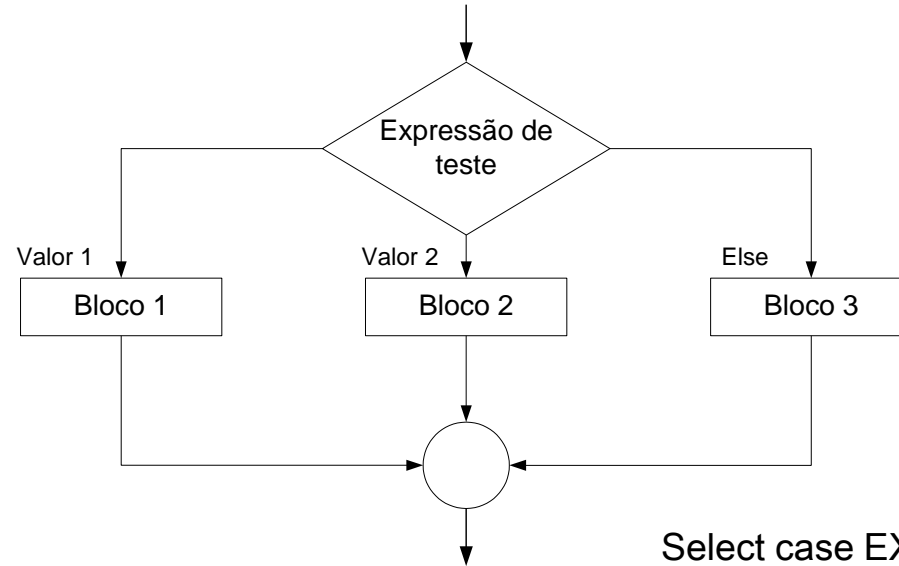
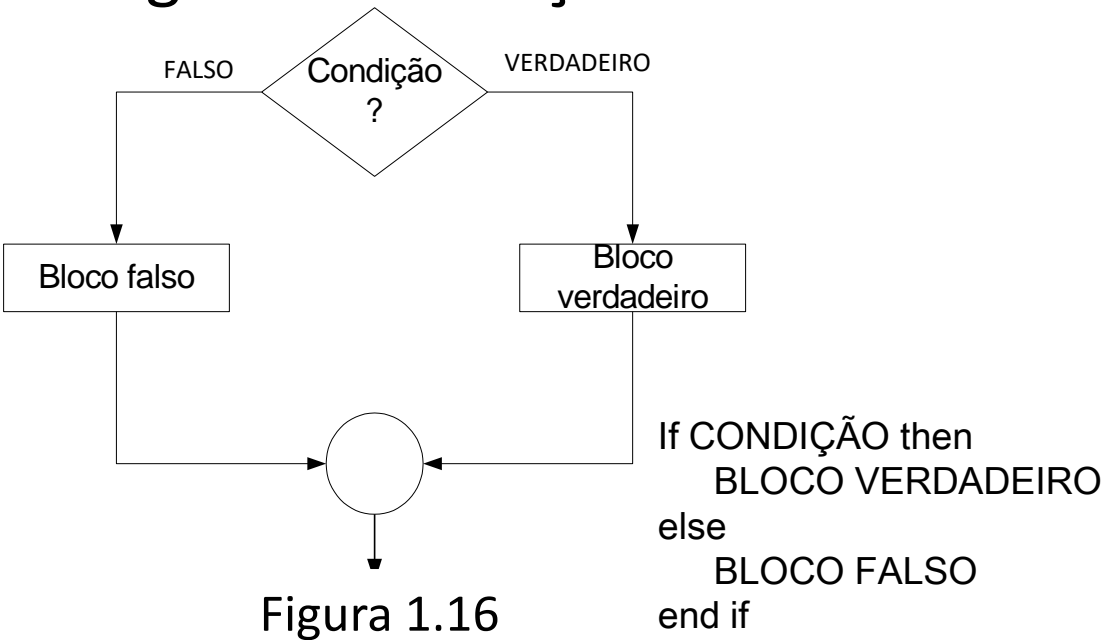


Figura 1.15

1.2.3 – Programação Lógica

- Seleção: Em contraste com a estrutura passo a passo da sequência, a seleção provê um meio de separar o fluxo do programa em ramos com base no valor

lógico da condição



Select case EXPRESSÃO DE TESTE
Case Valor 1
BLOCO 1
Case Valor 2
BLOCO 2
Case else
BLOCO 3

1.2.3 – Programação Lógica

- Seleção: Em contraste com a estrutura passo a passo da sequência, a seleção provê um meio de separar o fluxo do programa em ramos com base no valor lógico da condição

```
If CONDIÇÃO 1 then  
  BLOCO 1  
elseif CONDIÇÃO 2 then  
  BLOCO 2  
elseif CONDIÇÃO 3 then  
  BLOCO 3  
else  
  BLOCO 4  
End if
```

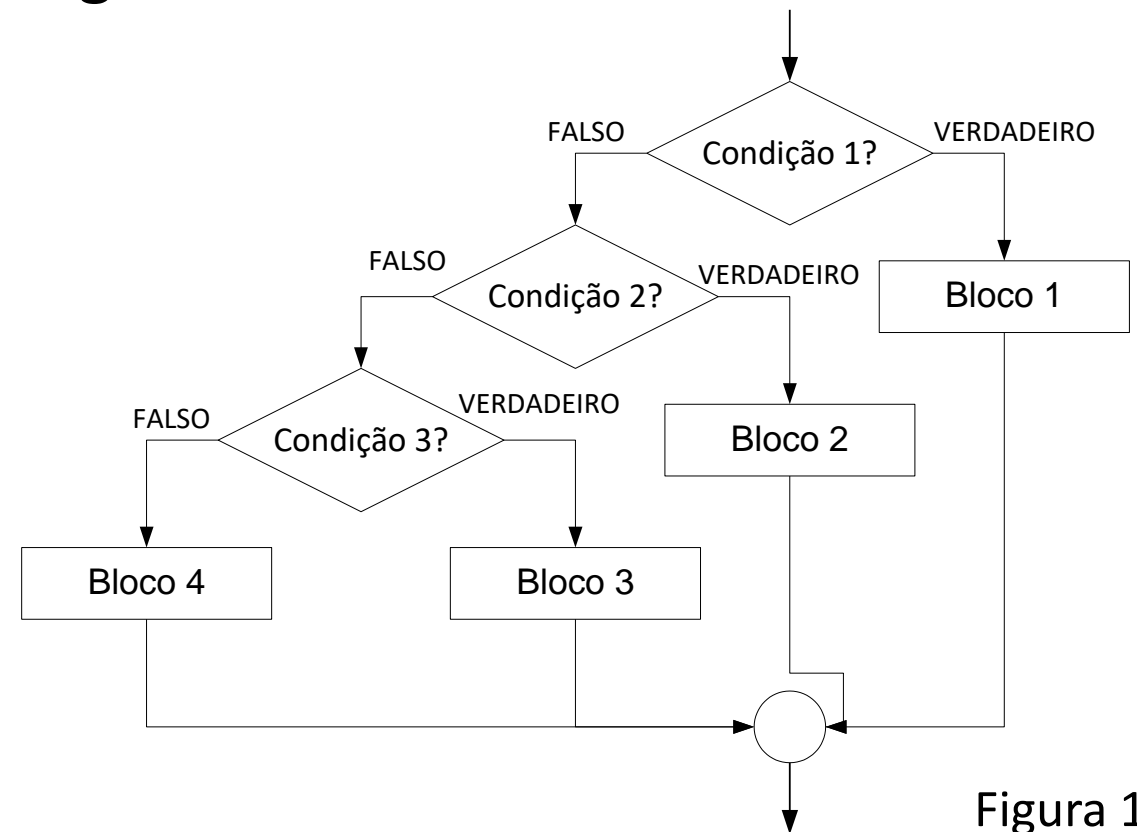
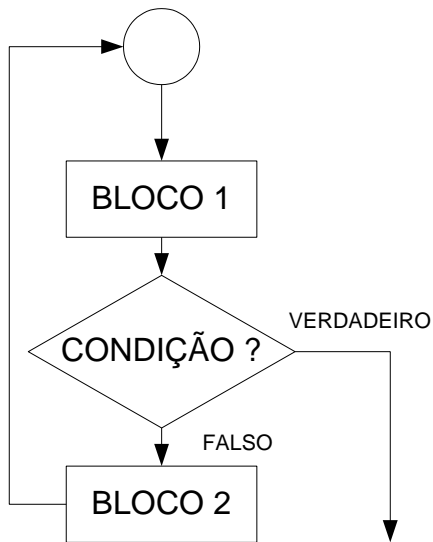


Figura 1.18

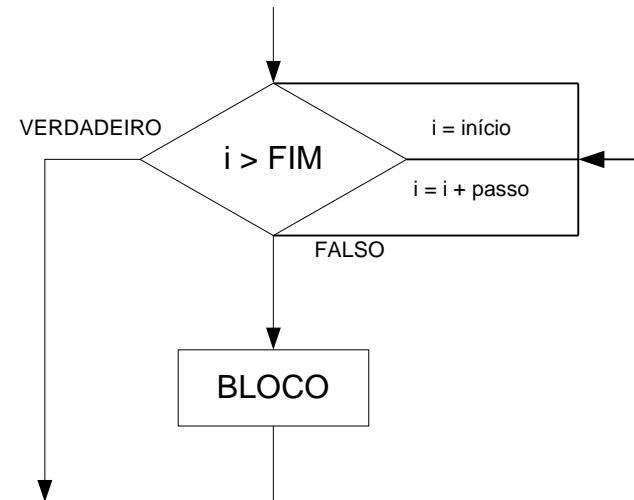
1.2.3 – Programação Lógica

- Repetição: A repetição provê os meios para executar uma instrução repetidamente. A estrutura resultante, chamada laço, pode ter dois “aspectos” distintos de acordo com a forma como termina.



Do
 BLOCO 1
if CONDIÇÃO exit
 BLOCO 2
end do

Figura 1.19



Do i = início, fim, passo
 BLOCO
end do

Figura 1.20

1.2.4 – Programação Modular

- Programas de computador são geralmente divididos em **subprogramas** menores, ou **módulos**, que podem ser desenvolvidos e testados **separadamente**. Essa abordagem é chamada **programação modular**.
- O atributo mais importante dos **módulos** é que sejam tão **independentes** e **autocontidos** quanto possível. Além disso, eles são tipicamente desenvolvidos para desempenhar uma função específicas e bem definida e tem apenas **um ponto de entrada** e **um ponto de saída**. Assim, esses módulos são usualmente **pequenos** (geralmente com 50 a 100 instruções de tamanho) e altamente focados.
- Nas linguagens padrão de alto nível o elemento principal de programação usado para representar cada módulo é o **procedimento**. Um procedimento é **uma série de instruções** computacionais que, juntas, desempenham uma dada tarefa. Dois tipos de procedimentos são empregados comumente: **funções** e **sub-rotinas**. As primeiras usualmente retornam um resultado único, enquanto que as últimas retornam vários resultados.