# Unidade 2: Algoritmos e programação

## 2.2. Estruturas de programação

A maioria das linguagens de programação possuem um conjunto de estruturas padrão, o que torna mais fácil implementação de um programa.

Já vimos que os fluxogramas são elaborados com base em três tipos de estruturas, que são:

- Estrutruras seqüenciais;
- Instruções de decisão ou seleção;
- Instruções de repetição ou iteração.

Estas estruturas recebem o nome de estruturas de controle da programação. Podemos dizer que qualquer programa de computador pode ser representado combinando-se estas três estruturas de programação.

### 2.2.1. Estruturas seqüenciais

Este tipo de estrutura representa os comandos que são executados pelo computador de forma seqüencial, ou seja, na ordem em que são apresentados. A leitura de dados, atribuições de valores a variáveis, cálculos, chamadas de funções, saída de resultados são exemplos dessas estruturas. Na Figura 2.5 apresentamos, na forma de fluxograma, algoritmo desenvolvido para o cálculo do peso (P) de uma determinada massa (m) sujeita a ação da força gravitacional (g) dado o valor da massa. Neste tipo de estrutura as ações são executadas de forma sucessiva da primeira a última instrução.

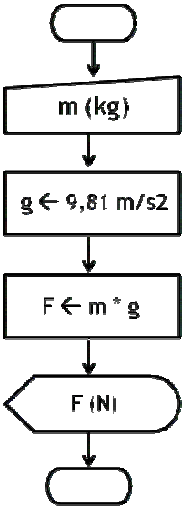


Figura 2.5: Fluxograma que implementa algoritmo para cálculo da força exercida por uma massa sujeita a ação de força gravitacional. Apenas estruturas seqüenciais são utilizadas.

## 2.2.2. Estruturas de decisão ou seleção

Existem três tipos de estruturas de seleção. As estruturas de decisão do tipo SE-ENTÃO e SE-ENTÃO-SENÃO, e as estruturas de derivação ou escolha entre um ou mais caminhos possíveis, estruturas CASO.

O primeiro tipo de estrutura (SE-ENTÃO) avalia uma expressão lógica resultando em uma resposta que pode ser verdadeira ou falsa. Sendo a resposta a resposta verdadeira, uma instrução ou conjunto de instruções é executado. Caso a resposta seja falsa é executado um desvio sem nenhuma instrução. A Figura 2.6 apresenta a implementação desta estrutura na forma de fluxograma.

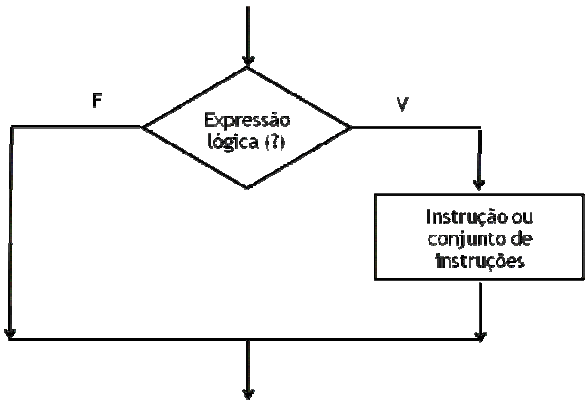


Figura 2.6. Fluxograma ilustrando a estrutura SE-ENTÃO.

Na generalidade das linguagens de programação as estruturas de programação são escritas em inglês. No caso do programa que utilizaremos na disciplina, essa estrutura é representada da forma:

**if** (<condição>) **then**

<instrução 1>

<instrução 2>

...

<instrução n>

**end**

A segunda estrutura de decisão é do tipo SE-ENTÃO-SENÃO. Nesta estrutura se o resultado da expressão lógica for verdadeiro uma instrução ou conjunto de instruções é executado. Caso a expressão lógica assuma valor falso, conjunto diferente de instruções é executado. A Figura 2.7 ilustra este tipo de estrutura.

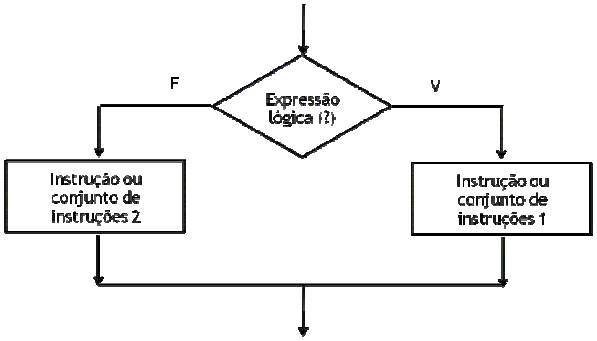


Figura 2.7: Fluxograma ilustrando a estrutura SE-ENTÃO-SENÃO.

Para representar em linguagem de programação a estrutura SE-ENTÃO-SENÃO ilustrada na Figura 2.7 teremos:

**if** (<condição>) **then**

<instrução 1>

<instrução 2>

...

<instrução n>

**else**

<instrução 1>

<instrução 2>

...

<instrução n>

**end**

As estruturas do tipo CASO permitem que se possa realizar a escolha entre dois ou mais caminhos a serem seguidos na programação. Neste tipo de estrutura o caminho do fluxo da informação dependerá do valor atribuído à expressão. Vejamos o exemplo ilustrado na Figura 2.8.

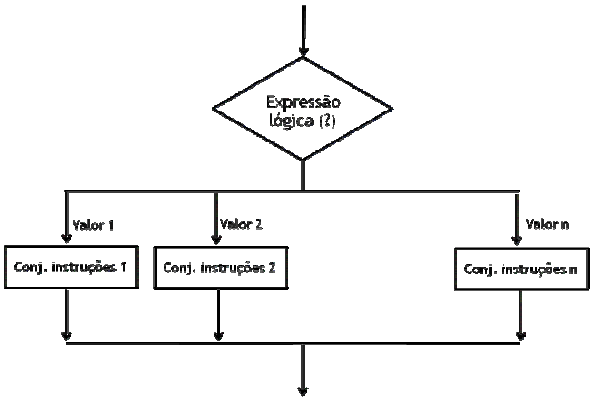


Figura 2.8: Fluxograma ilustrando a estrutura CASO.

No programa Scilab a sintaxe para este comando é da seguinte forma:

**select** <expressão>,
    **case** <valor_1> **then**,
        <conjunto de instruções 1>
    **case** <valor_2> **then**,
        <conjunto de instruções 2>
**...**
    **case** <valor_n> then,
        <conjunto de instruções n>
**end**

## 2.2.2.1 Expressões lógicas

**São expressões que resultam nos valores lógicos verdadeiro ou falso. A Tabela 2.2 apresenta os operadores condicionais empregados nas expressões lógicas.**

| **Tabela 2.2: Operadores condicionais.** | |
|---|---|
| **Operador** | **Símbolo** |

| | |
|---|---|
| **igualdade** | **= =** |
| **diferença** | **< >** |
| **maior do que** | **>** |
| **maior ou igual a** | **> =** |
| **menor do que** | **<** |
| **menor ou igual a** | **<=** |

## 2.2.3. Estruturas de repetição ou iteração

Como o próprio nome diz, este tipo de estrutura permite que se façam repetições controladas de alguma instrução ou conjunto de instruções.

Apresentaremos dois tipos desta estrutura, ambos disponíveis na linguagem de programação do Scilab. A primeira é a estrutura de repetição ENQUANTO-FAÇA. A Figura 2.9 apresenta a representação desta estrutura em forma de fluxograma

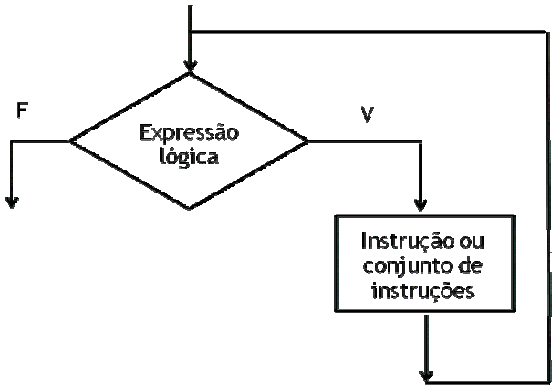


Figura 2.9: Fluxograma representando a estrutura ENQUANTO-FAÇA.

No Scilab esta estrutura é representada pelo comando **while** sendo implementada da seguinte maneira:

**while** <expressão>
    <instrução 1>
    <instrução 2>
    ...
    <instrução 3>
**end**

A segunda estrutura pode ser considerada um caso particular da estrutura ENQUANTO-FAÇA. Esta estrutura gera um ciclo de repetição controlado por uma variável

(do tipo inteira) que assumirá todos os valores entre um valor inicial e um valor final, incrementando-se ou decrementando-se a si própria de um determinado valor que pode ser definido pelo programador. O caso padrão (default) é que esta variável sofra incrementos unitários. Vejamos sua representação na forma de fluxograma, Figura 2.10.

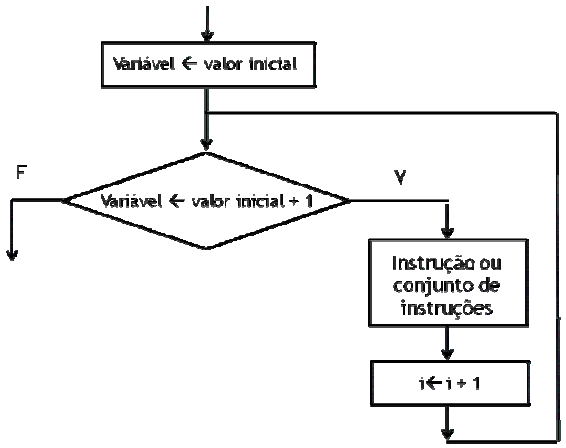


Figura 2.10: Fluxograma representando um caso particular da estrutura ENQUANTO-FAÇA.

No programa Scilab esta estrutura é representada pelo comando **for**, tendo a seguinte sintaxe.

**for** variável = expressão (vetor linha)

<instrução 1>

<instrução 2>

...

<instrução n>

**end**

## 2.4. Referências bibliográficas

Cereda, R. L. D., Maldonado, J. C. Introdução ao FORTRAN 77 para microcomputadores. Ed. McGraw-Hill, São Paulo, 1987.

FERREIRA, Aurelio Buarque de Holanda. Novo dicionário da língua portuguesa. 2 ed. Rio de Janeiro: Nova Fronteira, 1986. 1838 p.

Rodrigues, Diego; Nuno, Fernando; Raggiotti, Naiara Dicionário Larousse ilustrado da língua portuguesa. São Paulo: Larousse do Brasil, 2004. 977 p.

Souza, M. A. F., Gomes, M. M., Soares, M. V., Concilio, R. Algoritmos e Lógica de Programação. Ed. Pioneira Thomson Learning, São Paulo, 2005.