

# Estruturas de Controle

## (1ª Parte)

Na criação de algoritmos, utilizamos conceitos de entrada e saída de dados, variáveis, constantes, atribuições, expressões lógicas, relacionais e aritméticas, bem como comandos que traduzam esses conceitos de forma a representar um conjunto de ações que solucione um determinado problema.

Para que esse conjunto de ações se torne viável, deve existir uma perfeita relação lógica intrínseca ao modo pelo qual essas ações são executadas, ao modo pelo qual é regido o **fluxo de execução** do algoritmo.

Por meio das estruturas básicas de controle do fluxo de execução – **seqüencial, seleção, repetição** – e da combinação delas, poderemos criar algoritmos para solucionar diferentes tipos de problemas.

### Estrutura Seqüencial

A estrutura seqüencial de um algoritmo corresponde à execução do conjunto de ações como uma seqüência linear, de cima para baixo, isto é, os comandos serão executados na ordem em que foram escritos. Essa é a estrutura mais básica de um algoritmo e é a que temos utilizado até agora.

O modelo geral de um algoritmo com estrutura seqüencial é mostrado abaixo. Note que o texto entre chaves é um comentário e, assim, não tem nenhum efeito no algoritmo.

```
programa nome_programa; {identificação do início do algoritmo}
var
    {declaração das variáveis}
início          {identificação do início do corpo do algoritmo}
    ação 1;
    ação 2;
    ação 3;
    ...
    ação n;
fim.            {identificação do fim do algoritmo}
```

### Estruturas de Seleção

Uma estrutura de **seleção** permite a escolha de um grupo de ações (bloco) a ser executado quando determinadas **condições**, representadas por expressões lógicas ou relacionais, são ou não satisfeitas.

Em outras palavras, essas estruturas permitem que façamos algoritmos em que consideramos diferentes condições e então executamos um bloco de ações ou outro, dependendo da situação atual.

Uma estrutura de seleção pode ser: simples, composta, encadeada ou de múltipla escolha. Veremos cada uma delas a seguir.

#### Seleção Simples

Quando precisamos testar uma certa condição antes de executar **uma ação**, usamos uma seleção simples, que segue o seguinte modelo:

```
se condição então
    ação;
```

*condição* é uma expressão lógica ou relacional que, quando avaliada, pode gerar um resultado verdadeiro ou falso.

Se *condição* for verdadeira, a ação sob a cláusula **então** será executada. Caso contrário (*condição* for falsa) encerra-se o comando de seleção sem executar a ação.

Quando precisamos colocar várias ações após a cláusula **então**, temos que delimitar o bloco de ações por **início** e **fim**, como no modelo abaixo.

```
se condição então
início
    ação 1;
    ação 2;
    ação 3;
    ...
    ação n;
fim;
```

Se *condição* for verdadeira, a seqüência de ações (delimitada por **início** e **fim**) sob a cláusula **então** será executada. Caso contrário (*condição* for falsa) encerra-se o comando de seleção sem executar a nenhuma das ações.

## Seleção Composta

Quando tivermos situações em que duas alternativas dependem de uma mesma condição, uma de a condição ser verdadeira e outra de a condição ser falsa, usamos a estrutura de seleção composta.

Supondo que uma ação (ação 1) dependa da avaliação da condição ser verdadeira e outra ação (C2) dependa da avaliação da condição ser falsa, usaremos uma estrutura de seleção composta semelhante ao seguinte modelo:

```
se condição então
    ação 1
senão
    ação 2;
```

Se *condição* for verdadeira, a ação sob a cláusula **então** (ação 1) será executada. Caso contrário (*condição* for falsa), a ação sob a cláusula **senão** (ação 2) será executada.

Note que a ação antes da cláusula **senão** não é seguida por ponto-e-vírgula.

Assim como na seleção simples, também podemos condicionar blocos de ações (seqüências de ações delimitas por **início** e **fim**) às cláusulas **então** e **senão**. Dessa forma, o comando de seleção composta pode aparecer de várias formas. No modelo abaixo, supomos que existe um bloco de ações subordinado ao **então** e outro bloco subordinado ao **senão**.

```
se condição então
início
    ação 1;
    ação 2;
    ...
    ação n;
fim
senão
início
    ação n+1;
    ação n+2;
    ...
    ação m;
fim;
```

Note que o fim antes do **senão** não é seguido por ponto-e-vírgula. Outras formas da seleção composta são:

```
se condição então
início
    ação 1;
    ação 2;
    ...
    ação n;
fim
senão
    ação m;
```

```
se condição então
    ação 1
senão
início
    ação 2;
    ação 3;
    ...
    ação n;
fim;
```

## Seleção encadeada

Quando, devido à necessidade de processamento, agruparmos várias seleções, formaremos uma seleção encadeada. Normalmente, tal formação ocorre quando uma determinada ação ou bloco de ações deve ser executado se um conjunto de possibilidades ou combinações de situações for satisfeito.

As estruturas de seleção podem ser agrupadas de diversas formas visando construir uma sequência lógica de avaliação de condições. Assim, não existe um modelo genérico de encadeamento, de modo que representamos nos modelos abaixo algumas possibilidades de encadeamento.

```
se condição 1 então
    se condição 2 então
        se condição n então
            ação;
```

```
se condição 1 então
    se condição 2 então
        inicio
            ação 1;
            ação 2;
            ...
            ação n;
        fim
    senão
        inicio
            ação n+1;
            ação n+1;
            ...
            ação m;
        fim;
```

```
se condição 1 então
    ação 1
senão
    se condição 2 então
        se condição 3 então
            inicio
                ação 2;
                ação 3;
                ...
                ação n;
            fim;
        fim;
```

```
se condição 1 então
    se condição 2 então
        se condição n então
            inicio
                ação 1;
                ação 2;
                ...
                ação m;
            fim;
```

```
se condição 1 então
    ação 1
senão
    se condição 2 então
        ação 2;
```

```
se condição 1 então
    inicio
        se condição 2 então
            inicio
                ação 1;
                ação 2;
                ...
                ação n;
            fim;
        fim
    senão
        inicio
            ação n+1;
            ação n+1;
            ...
            ação m;
        fim;
```

## Seleção Múltipla Escolha

Quando um conjunto de valores discretos precisa ser testado e ações diferentes são associadas a esses valores, estamos diante de uma seleção de múltipla escolha.

Seja  $X$  uma variável qualquer e  $V_1, V_2, \dots, V_n$ , um conjunto de valores possíveis para  $X$ . O modelo que expressa uma estrutura desse tipo é mostrado abaixo.

```
caso X seja
     $V_1$ : ação 1;
     $V_2$ : ação 2;
    ...
     $V_n$ : ação n;
fim;
```

Dependendo do valor da variável  $X$ , a ação correspondente será executada. Caso  $X = V_1$ , será executada ação 1. Caso  $X = V_2$ , será executada a ação 2. Caso  $X = V_n$ , será executada ação  $n$ .

```

caso  $X$  seja
     $V_1$ : ação 1;
     $V_2$ : ação 2;
    ...
     $V_3$ : ação 3;
    senão
        ação 4;
fim;

```

Outra forma da estrutura de seleção de múltipla escolha inclui uma cláusula **senão**, que será executada caso o valor da variável  $X$  seja diferente de todos os valores descritos nas cláusulas anteriores. Veja o modelo ao lado.

```

caso  $X$  seja
     $V_1$ : inicio
        ação 2;
        ação 3;
        ...
        ação  $n$ ;
        fim;
     $V_2$ : inicio
        ação  $n+1$ ;
        ação  $n+2$ ;
        ...
        ação  $m$ ;
        fim;
    ...
     $V_n$ : ação  $n$ ;
fim;

```

Assim como nas outras estruturas seleção vistas, cada cláusula da estrutura **caso...seja** pode conter uma única ação (como nos exemplos dados até agora) ou blocos de ações. O modelo ao lado mostra uma forma dessa estrutura em que blocos de ações são utilizados.

Embora não seja mostrado no modelo ao lado, poderíamos também incluir uma cláusula **senão** após  $V_n$ .

```

caso  $X$  seja
    0..9 : ação 1;
    10..19: ação 2;
    ...
     $n$ .. $m$  : ação  $n$ ;
fim;

```

A estrutura de múltipla escolha também permite que sejam definidos intervalos de valores para a variável  $x$  em vez de valores únicos. Suponha que a variável  $X$  seja do tipo inteiro. Em vez de se usar um único valor inteiro para cada cláusula do comando **caso**, poderíamos usar intervalos de valores, como mostra o modelo ao lado.

Ressaltamos que qualquer outra estrutura de seleção poderia estar aninhada nessa estrutura de múltipla escolha, assim como uma estrutura de múltipla escolha poderia estar aninhada em qualquer outra estrutura de seleção. **De fato, qualquer tipo de estrutura pode estar aninhada como parte de um bloco de ações.**