



ESTRUTURA SEQUENCIAL E ESTRUTURA DE SELEÇÃO

AULA 3



Prof. Sandro de Araujo



CONVERSA INICIAL

Esta aula teve como base os livros *Lógica de programação e estruturas de dados* e *Lógica de programação algorítmica*. Em caso de dúvidas ou aprofundamento, consulte-os em nossa Biblioteca Virtual Pearson.

Eis os temas que compõem o conteúdo:

1. Estrutura sequencial e de seleção;
2. Estrutura de seleção simples ou se-então;
3. Estrutura de seleção composta ou se-então-senão;
4. Estrutura de seleção encadeada ou se-então-senão-se-então-senão;
5. Estrutura de seleção de múltipla escolha ou escolha-caso.

O objetivo da aula é conhecer os principais conceitos das estruturas sequencial e de seleção, tanto em pseudocódigo quanto na linguagem de programação C, bem como representar facilmente os algoritmos nas diferentes estruturas para resolver problemas computacionais.

TEMA 1 – ESTRUTURA SEQUENCIAL E DE SELEÇÃO

Construir um algoritmo pressupõe obedecer a uma ordem predeterminada a fim de que sejam dados os passos corretos, à luz de certas regras dentro de uma sequência lógica, para resolver o problema computacional. Tais regras vão desde a sintaxe da linguagem escolhida até a ordenação dos comandos necessários para o algoritmo. A seguir, aprenderemos as estruturas usadas na construção de algoritmos, com alguns exemplos em fluxograma, pseudocódigo e linguagem C.

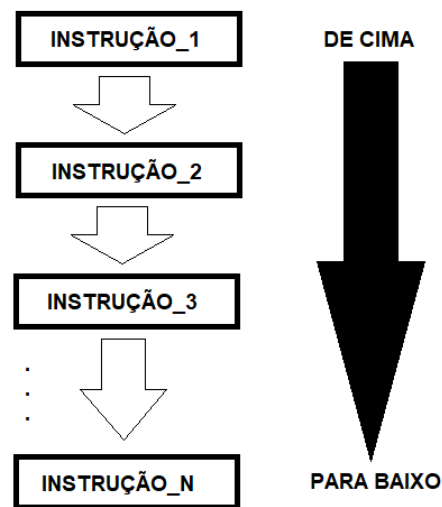
1.1 Estrutura sequencial

É uma estrutura que realiza um conjunto predeterminado de instruções de forma sequencial, de cima para baixo, na ordem em que foram declaradas (Guedes, 2014; Puga e Rissetti, 2016).

Sua estrutura básica é a repetição de comandos sequencialmente, conforme demonstrado pela Figura 1:



Figura 1 – Sintaxe da estrutura sequencial

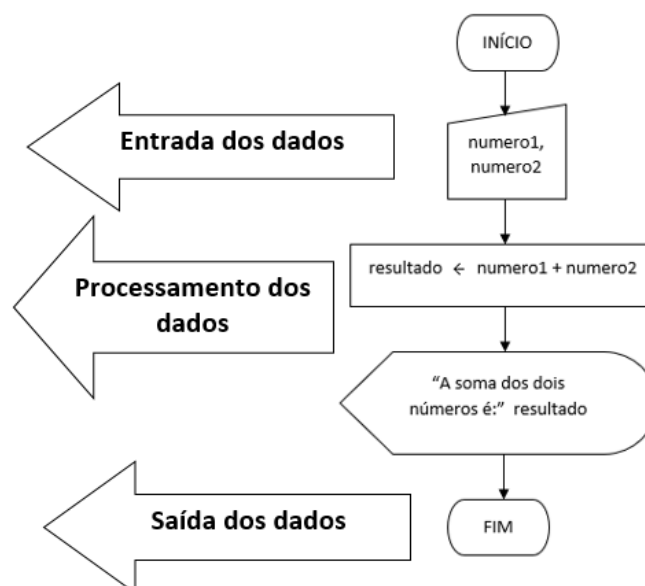


A estrutura sequencial de um algoritmo pode ser facilmente representada por meio de fluxograma, pseudocódigo e linguagem C, conforme os seguintes exemplos.

1.1.1 Exemplo 1

Considere um algoritmo que somará dois números inteiros quaisquer. Para este exemplo, temos, como entrada, dois números inteiros (numero1 e numero2) e, como saída, um número inteiro (resultado), que recebe o resultado de $\text{numero1} + \text{numero2}$.

1.1.1.1 Fluxograma





1.1.1.2 Pseudocódigo

```
algoritmo “exemplo1”  
var numero1, numero2, resultado: inteiro  
início  
    leia (numero1)  
    leia (numero2)  
    resultado ← numero1 + numero2  
    escreva (“A soma dos dois números é: ”, resultado)  
fimalgoritmo
```

1.1.1.3 Linguagem C

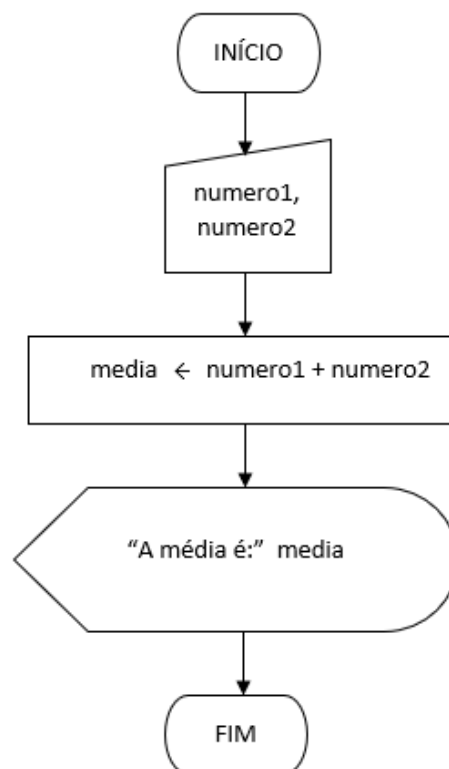
```
#include <stdio.h>  
int main ()  
{  
    int numero1, numero2, resultado;  
    scanf ("%d", &numero1);  
    scanf ("%d", &numero2);  
    resultado = numero1 + numero2;  
    printf (“A soma dos dois números é: %d\n”, resultado);  
    system (“pause”);  
    return (0);  
}
```

1.1.2 Exemplo 2

Considere um algoritmo que somará dois números inteiros e obterá a média entre ambos. Para este exemplo, temos, como entrada, dois números inteiros (numero1, numero2) e, como saída, um número inteiro (media), que recebe o resultado de $(\text{numero1} + \text{numero2}) / 2$.



1.1.2.1 Fluxograma



1.1.2.2 Pseudocódigo

algoritmo “exemplo2”

var numero1, numero2, media: inteiro

início

leia (numero1)

leia (numero2)

media \leftarrow (numero1 + numero2) / 2

escreva (“A média é: ”, media)

fimalgoritmo



1.1.2.3 Linguagem C

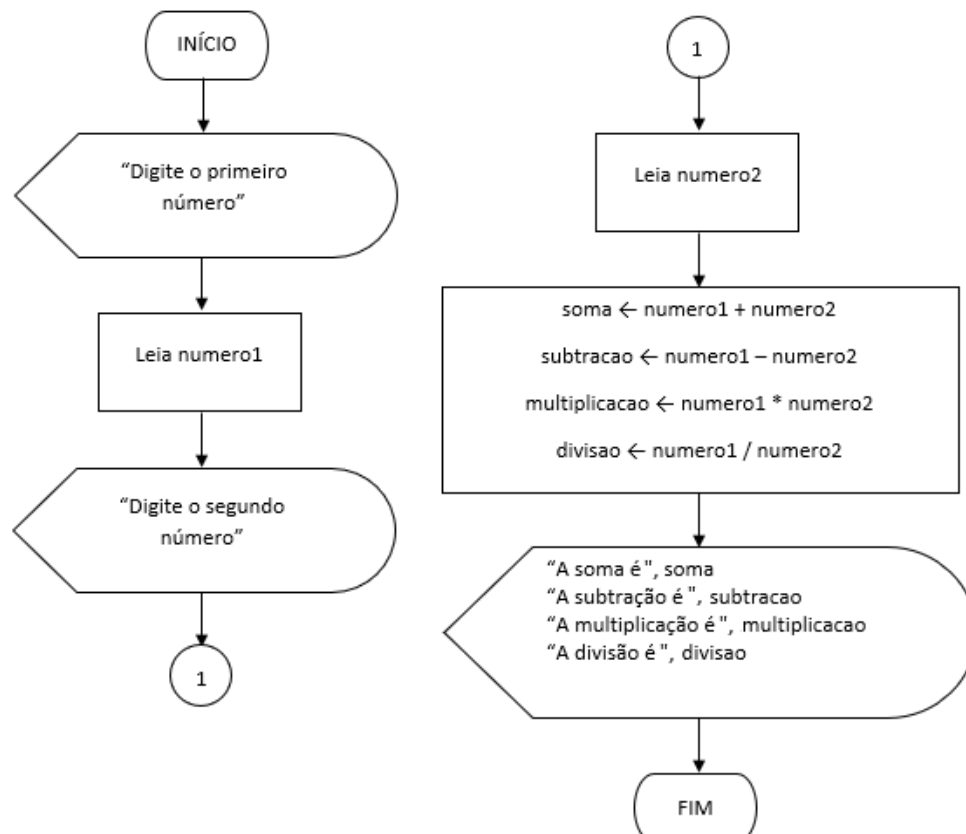
```
#include <stdio.h>

int main()
{
    int numero1, numero2, media;
    scanf ("%d", &numero1);
    scanf ("%d", &numero2);
    media = (numero1 + numero2) / 2;
    printf ("A média é: %d\n", media);
    system ("pause");
    return (0);
}
```

1.1.3 Exemplo 3

Considere um algoritmo que receberá dois números inteiros e, ao final, mostrará a soma, a subtração, a multiplicação e a divisão desses números.

1.1.3.1 Fluxograma





1.1.3.2 Pseudocódigo

algoritmo “exemplo3”

var numero1, numero2: real

início

```
    escreva (“Digite o primeiro número: ”)
    leia (numero1)
    escreva (“Digite o segundo número: ”)
    leia (numero2)
    soma ← numero1 + numero2
    subtracao ← numero1 – numero2
    multiplicacao ← numero1 * numero2
    divisao ← numero1 / numero2
    escreva (“A soma é: ”, soma)
    escreva (“A subtração é: ”, subtracao)
    escreva (“A multiplicação é: ”, multiplicacao)
    escreva (“A divisão é: ”, divisao)
```

fimalgoritmo

1.1.3.3 Linguagem C

```
#include <stdio.h>
int main ()
{
    int numero1, numero2, soma, subtracao, multiplicacao, divisao;
    printf (“Digite o primeiro número: ”);
    scanf ("%d", &numero1);
    printf (“Digite o segundo número: ”);
    scanf ("%d", &numero2);
    soma = numero1 + numero2;
    subtracao = numero1 - numero2;
    multiplicacao = numero1 * numero2;
    divisao = numero1 / numero2;
    printf (“A soma é: %d\n”, soma);
    printf (“A subtração é: %d\n”, subtracao);
    printf (“A multiplicação é: %d\n”, multiplicacao);
    printf (“A divisão é: %d\n”, divisao);

    system (“pause”);
    return (0);
}
```



1.2 Estrutura de seleção

Uma estrutura de seleção viabiliza escolher um grupo de ações (bloco) a ser executado quando determinadas condições, representadas por expressões lógicas ou relacionais, são satisfeitas ou não. Eis os tipos de seleção: simples, composta, encadeada e de múltipla escolha (Guedes, 2014).

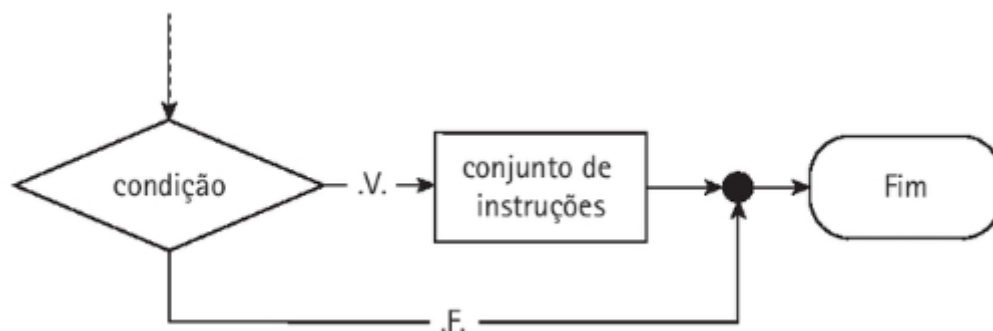
TEMA 2 – ESTRUTURA DE SELEÇÃO SIMPLES OU SE-ENTÃO

As estruturas de seleção simples são utilizadas para testar certa condição antes de executar determinada ação. Se a condição for atendida, um conjunto de instruções deverá ser executado; se não for atendida, o fluxo da execução do algoritmo seguirá após o fim do bloco de decisão (Guedes, 2014; Puga; Rissetti, 2016).

Toda condição pode ser interpretada como uma pergunta, cuja resposta é verdadeira ou falsa. Vejamos, a seguir, a sintaxe da estrutura de seleção simples em fluxograma, pseudocódigo e linguagem C.

A Figura 2 mostra a sintaxe da estrutura de seleção na representação algorítmica em fluxograma, em que a figura geométrica na qual está escrito “condição” corresponde à condição prévia para a execução de determinada ação: se verdadeira (.v.), segue o fluxo de instruções, representado pela segunda figura geométrica; se a condição for falsa (.f.), encerra-se o algoritmo.

Figura 2 – Sintaxe da estrutura de seleção simples



Fonte: Guedes, 2014.

O exemplo a seguir mostra a sintaxe da estrutura de seleção na representação algorítmica em pseudocódigo. Os comandos só serão executados após testada a condição contida em “Se (<condição>) então”: se verdadeira (.v.),



segue o fluxo de instruções representado por “<comandos>”; se a condição for falsa (.f.), encerra-se o algoritmo.

Início

Se (<condição>) **então**
 <comandos>

Fimse

Fimalgoritmo

Do mesmo modo, na sintaxe da estrutura de seleção na linguagem de programação C, a condição testa previamente determinada ação para ver se continua o fluxo, no comando “if (<condição>)”: se verdadeira (.v.), obedecerá às instruções representadas por “<comandos>”; se a condição for falsa (.f.), encerra-se o algoritmo. Confira o exemplo.

Linguagem C

```
if (<condição>)  
{  
    <comandos>  
}
```

Na linguagem de programação C, é obrigatório o uso de parênteses, não há o “então” e, se houver apenas um comando em <comandos>, as chaves não são necessárias.

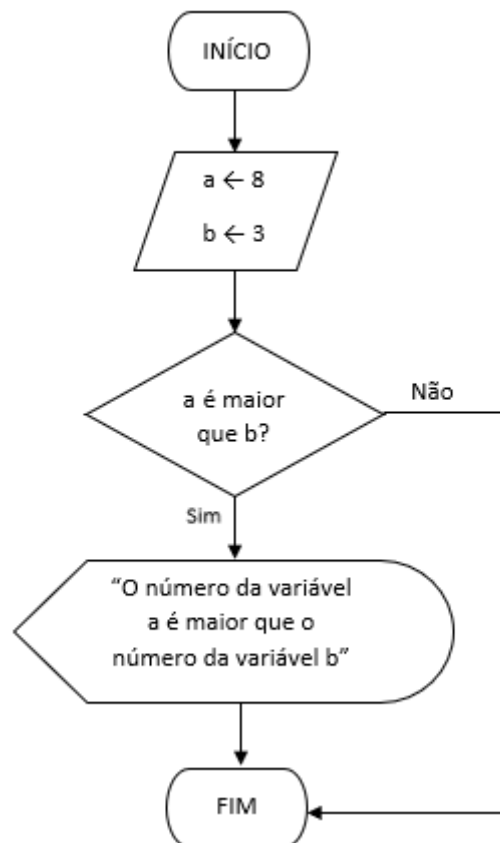
Eis alguns exemplos de estrutura de seleção simples em fluxograma, pseudocódigo e linguagem C.

2.1 Exemplo 1

Considere um algoritmo que pegará dois números inteiros, comparará se o primeiro é maior do que o segundo e, por fim, imprimirá na tela uma mensagem informando o resultado da comparação. Para este exemplo, temos, como entrada, dois números inteiros (a, b) e, como saída, uma mensagem de texto.



2.1.1 Fluxograma



2.1.2 Pseudocódigo

algoritmo “exemplo1”

var **a**, **b**: inteiro

a ← 8

b ← 3

início

Se (**a** > **b**) então

 escreva (“O número da variável **a** é maior que o número da variável **b**”)

Fimse

fimalgoritmo



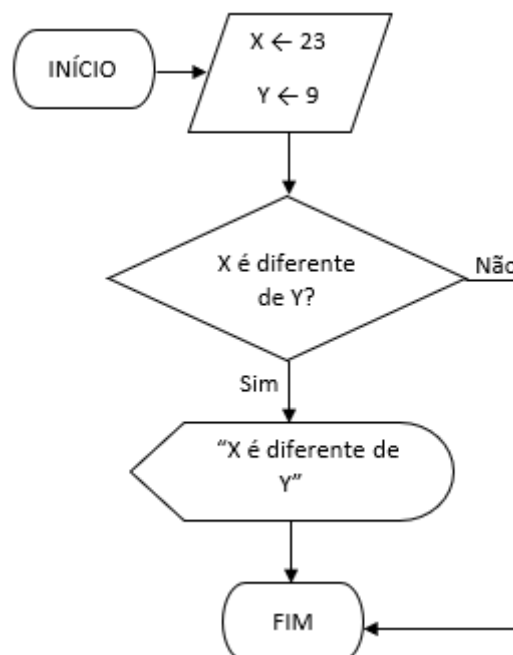
2.1.3 Linguagem C

```
#include <stdio.h>
int main ()
{
    int a = 8, b = 3;
    if (a > b)
    {
        printf ("O número da variável a é maior que o número da variável b \n");
    }
    system ("pause");
    return (0);
}
```

2.2 Exemplo 2

Considere um algoritmo que pegará dois números inteiros, comparará se o primeiro é diferente do segundo e, por fim, imprimirá na tela uma mensagem informando o resultado da comparação. Para este exemplo, temos, como entrada, dois números inteiros (X, Y) e, como saída, uma mensagem de texto.

2.2.1 Fluxograma





2.2.2 Pseudocódigo

```
algoritmo "exemplo2"
var X, Y: inteiro
X ← 23
Y ← 9
início
    Se (X <> Y) então
        escreva ("X é diferente de Y")
    Fimse
finalgoritmo
```

2.2.3 Linguagem C

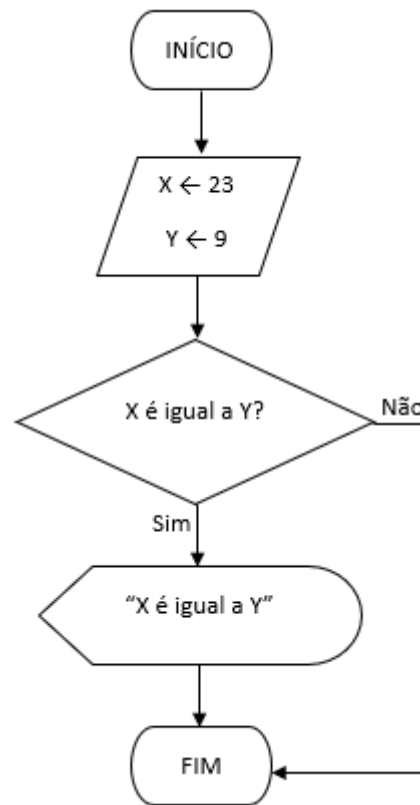
```
#include <stdio.h>
int main ()
{
    int X = 23, Y = 9;
    if (X != Y)
    {
        printf ("X é diferente de Y \n");
    }
    system ("pause");
    return (0);
}
```

2.3 Exemplo 3

Considere um algoritmo que pegará dois números inteiros, comparará se o primeiro é igual ao segundo e, por fim, imprimirá na tela usuário uma mensagem informando o resultado da comparação. Para este exemplo, temos, como entrada, dois números inteiros (X, Y) e, como saída, uma mensagem de texto.



2.3.1 Fluxograma



2.3.2 Pseudocódigo

algoritmo “exemplo2”

var X, Y: inteiro

X ← 23

Y ← 9

início

 Se (X = Y) então

 escreva (“X é igual a Y”)

 Fimse

fimalgoritmo

2.3.3 Linguagem C

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
  int X = 23, Y = 9;
```

```
  if (X = Y)
```

```
  {
```

```

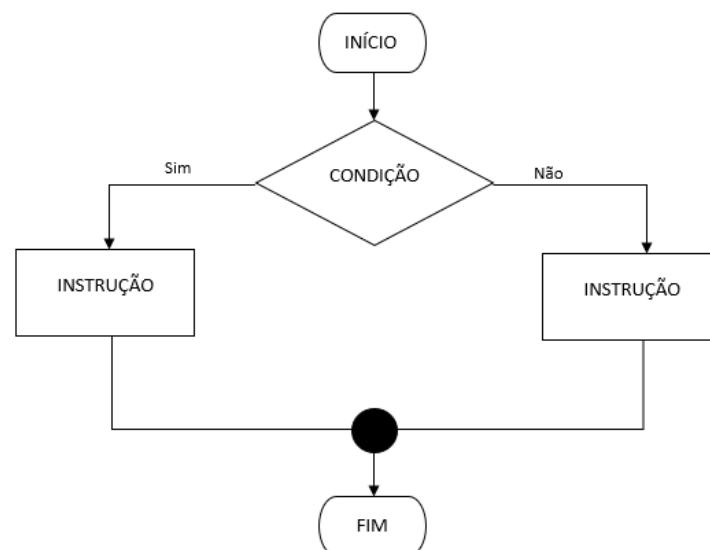
        printf ("X é igual a Y \n");
    }
    system ("pause");
    return (0);
}

```

TEMA 3 – ESTRUTURA DE SELEÇÃO COMPOSTA OU SE-ENTÃO-SENÃO

A estrutura de seleção composta prevê dois conjuntos de instruções, após a avaliação da condição: um conjunto caso a resposta da condição resulte verdadeira; o outro, por sua vez, se resultar falsa (Guedes, 2014; Puga; Rissetti, 2016). A Figura 3 mostra a sintaxe da estrutura de seleção composta representada em um fluxograma.

Figura 3 – Sintaxe da estrutura de seleção composta



Fonte: Adaptado de Guedes, 2014.

O próximo exemplo contempla a sintaxe da estrutura de seleção composta na representação algorítmica em pseudocódigo, cujo funcionamento é simples: com base no resultado da condição (verdadeiro ou falso), o fluxo do algoritmo segue para um dos blocos de instruções.

Início

Se <condição> **então**

<Primeiro conjunto de instruções>

Senão

<Segundo conjunto de instruções>

Fimse

Fimalgoritmo



A linguagem C obedece à mesma lógica do pseudocódigo, conforme o exemplo.

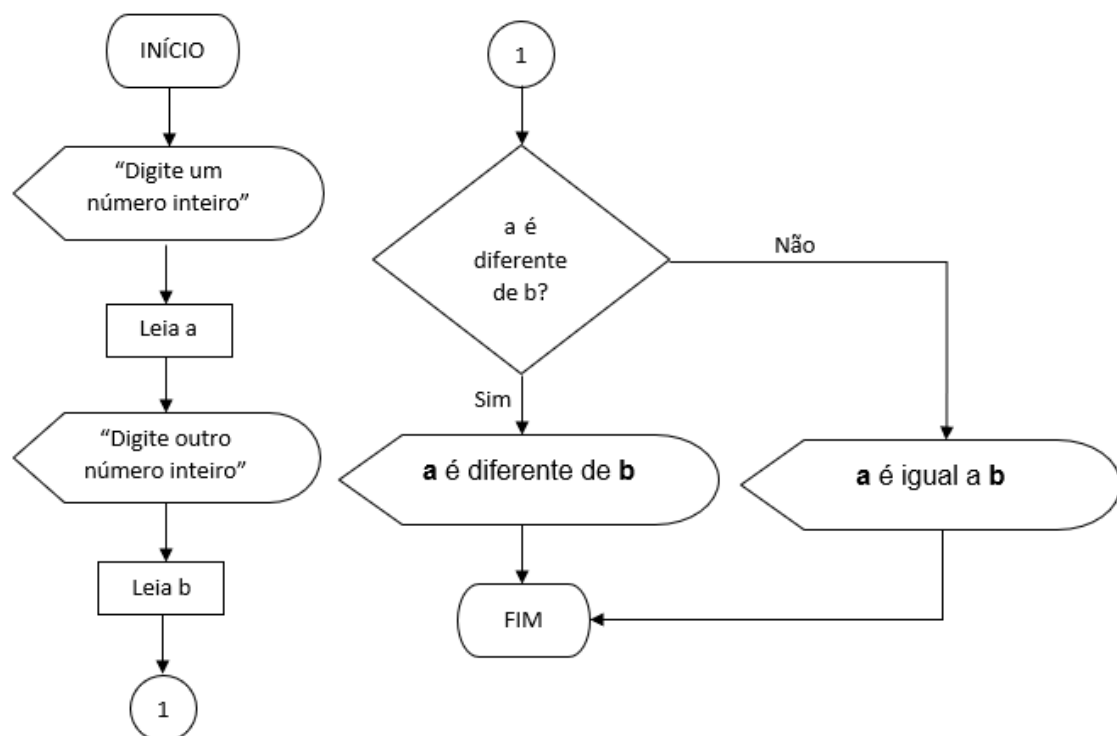
```
if (<condição>)  
{  
    <Primeiro conjunto de instruções>  
}  
else  
{  
    <Segundo conjunto de instruções>  
}
```

Eis um exemplo de estrutura de seleção composta em fluxograma, pseudocódigo e linguagem C.

3.1 Exemplo

Considere um algoritmo que pegará dois números inteiros, comparará se o primeiro é diferente do segundo e, por fim, imprimirá na tela uma mensagem com o resultado dessa comparação.

3.1.1 Fluxograma





3.1.2 Pseudocódigo

algoritmo “exemplo”

var **a, b**: inteiro

início

 escreva (“**Digite um número inteiro**”)

 leia (a)

 escreva (“**Digite outro número inteiro**”)

 leia (b)

 Se (**a <> b**) então

 escreva (“**a é diferente de b**”)

 Senão

 escreva (“**a é igual a b**”)

 Fimse

fimalgoritmo

3.1.3 Linguagem C

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
  int a, b;
```

```
  printf (“Digite um número inteiro \n”);
```

```
  scanf (“%d”, &a);
```

```
  printf (“Digite outro número inteiro \n”);
```

```
  scanf (“%d”, &b);
```

```
  if (a != b)
```

```
  {
```

```
    printf (“a é diferente de b \n”);
```

```
  }
```

```
  else
```

```
  {
```

```
    printf (“a é igual a b \n”);
```

```
  }
```

```
  system (“pause”);
```

```
  return (0);
```

```
}
```



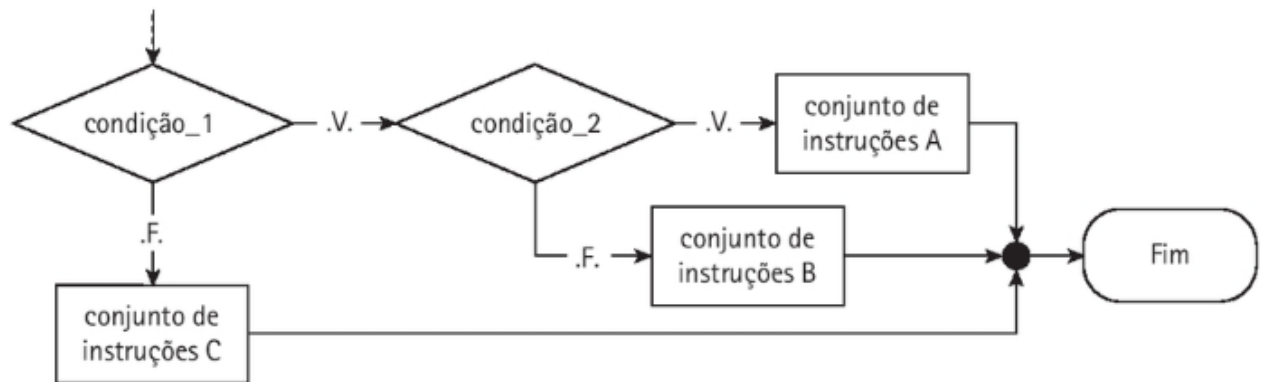

TEMA 4 – ESTRUTURA DE SELEÇÃO ENCADEADA OU SE-ENTÃO-SENÃO-SE-ENTÃO-SENÃO

Uma estrutura de seleção encadeada é formada pela combinação de estruturas de seleção simples e compostas, uma dentro da outra, não havendo limite para o número de combinações e podendo, em alguns casos, gerar um código bastante complexo (Puga, 2016).

Vejamos, a seguir, a sintaxe da estrutura de seleção encadeada em fluxograma, pseudocódigo e linguagem C.

A Figura 4 mostra a sintaxe da estrutura de seleção na representação algorítmica em fluxograma.

Figura 4 – Sintaxe da estrutura de seleção encadeada



Fonte: Puga; Rissetti, 2016.

O próximo exemplo demonstra a sintaxe da estrutura de seleção encadeada na representação algorítmica em pseudocódigo. Se a primeira condição resultar verdadeira, o algoritmo testará a segunda condição, cujo resultado, se verdadeiro, executará o primeiro conjunto de instruções; se for falso, executará o segundo conjunto de instruções. Caso o resultado do teste da primeira condição seja falso, a segunda sequer será executada e, assim, o terceiro conjunto de instruções entrará em ação.



```
Início
  Se (<condição1>) então
    Se (<condição2>) então
      <Primeiro conjunto de instruções>
    Senão
      <Segundo conjunto de instruções>
    Fimse
  Senão
    <Terceiro conjunto de instruções>
  Fimse
Fimalgoritmo
```

A linguagem C segue a mesma lógica do pseudocódigo.

```
if (<condição1>)
{
    if (<condição2>)
    {
        <Primeiro conjunto de instruções>
    }
    else
    {
        <Segundo conjunto de instruções>
    }
}
else
{
    <Terceiro conjunto de instruções>
}
```

Eis alguns exemplos de estrutura de seleção encadeada em fluxograma, pseudocódigo e linguagem C.

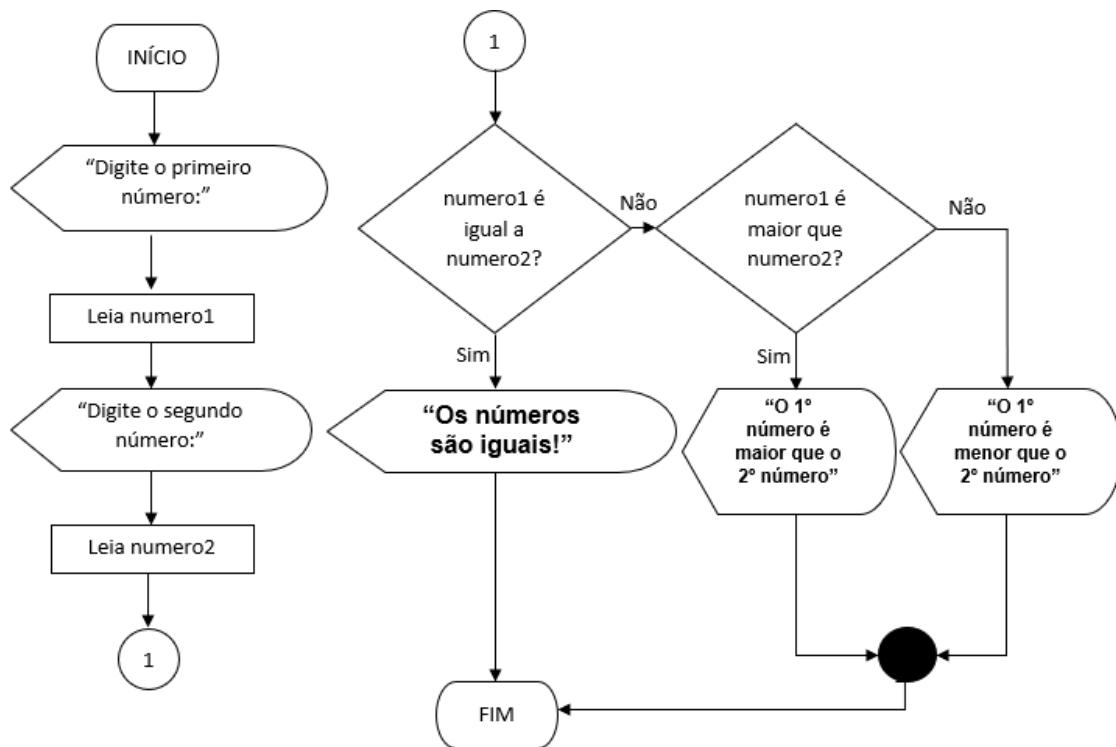
4.1 Exemplo

O seguinte exemplo foi criado com uma estrutura encadeada, cuja lógica difere um pouco da Figura 4.

Considere um algoritmo que pegará dois números inteiros e testará para ver se o primeiro é igual ao segundo. Se a condição for verdadeira, ele imprimirá uma mensagem na tela do usuário; a segunda condição, por seu turno, só entrará em ação caso o resultado da primeira condição seja falso.



4.1.1 Fluxograma



4.1.2 Pseudocódigo

algoritmo “exemplo”

var numero1, numero2: inteiro

inicio

 escreva (“Digite o primeiro número: ”)

 leia (numero1)

 escreva (“Digite o segundo número: ”)

 leia (numero2)

 se (numero1 = numero2) então

 escreva (“Os números são iguais!”)

 senão

 se (numero1 > numero2) então

 escreva (“O maior valor é = ”, numero1)

 senão

 escreva (“O maior valor é = ”, numero1)

 fimse

 fimse

fimalgoritmo



4.1.3 Linguagem C

```
#include <stdio.h>
#include <stdlib.h>
int main ()
{
    int numero1, numero2;
    printf ("Digite o primeiro número: ");
    scanf ("%d", & numero1);
    printf ("Digite o segundo número: ");
    scanf ("%d", & numero2);

    if (numero1== numero2)
        printf ("Os números são iguais!");
    else
        if (numero1> numero2)
            printf ("O maior valor é = %d", numero1);
        else
            printf ("O maior valor é = %d", numero2);
    printf ("\n");
    system ("pause");
    return (0);
}
```

TEMA 5 – ESTRUTURA DE SELEÇÃO DE MÚLTIPLA ESCOLHA OU ESCOLHA-CASO

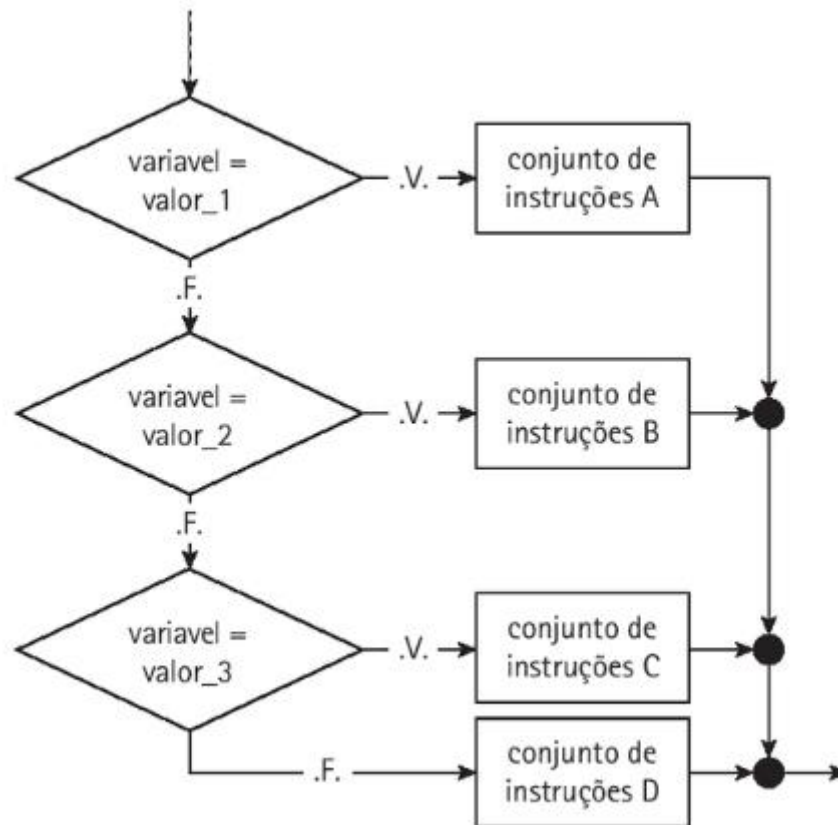
Uma estrutura de seleção de múltipla escolha é uma solução para substituir várias estruturas de decisão (se-então-senão) aninhadas. Também conhecida como estrutura escolha-caso, ela permite ir diretamente ao bloco de instruções desejado, submetendo-se ao valor de uma variável de verificação.

Vejamos, a seguir, a sintaxe da estrutura de seleção de múltipla escolha em fluxograma, pseudocódigo e linguagem C.

A Figura 5 demonstra a sintaxe da estrutura de seleção na representação algorítmica em fluxograma.



Figura 5 – Sintaxe da estrutura de seleção de múltipla escolha



Fonte: Puga; Riseti, 2016.

O próximo exemplo contempla a sintaxe da estrutura de seleção de múltipla escolha na representação algorítmica em pseudocódigo.

```
escolha <expressão>
  caso <valor1>
    <comandos>
  caso <valor2>
    <comandos>
  outrocaso
    <comandos>
fimescolha
```



A linguagem C obedece à mesma lógica do pseudocódigo.

```
...
switch (<expressão>)
{
    case <valor1> :
        <comandos>
        break;
    case <valor2> :
        <comandos>
        break;
    default :
        <comandos>
}
...
```

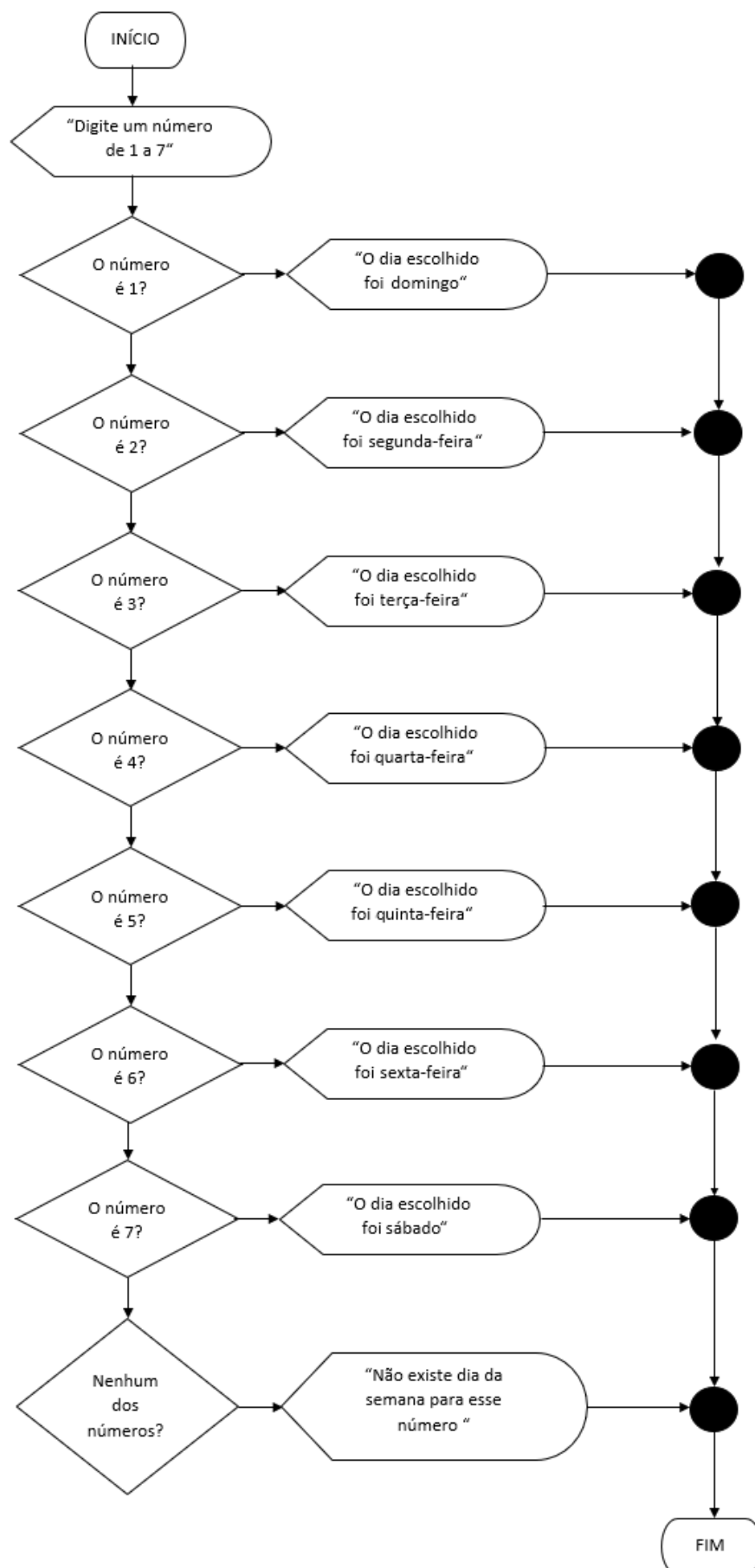
Eis alguns exemplos de estrutura de seleção de múltipla escolha em fluxograma, pseudocódigo e linguagem C.

5.1 Exemplo

Considere um algoritmo que pegará um número e retornará o dia da semana equivalente a ele.



5.1.1 Fluxograma





5.1.2 Pseudocódigo

algoritmo “multiplaescolha”

var

numero1 : inteiro

início

escreva (“Digite um número de 1 a 7”)

leia (numero1)

escolha numero1

caso “1”

escreva (“O dia escolhido foi domingo”)

caso “2”

escreva (“O dia escolhido foi segunda-feira”)

caso “3”

escreva (“O dia escolhido foi terça-feira”)

caso “4”

escreva (“O dia escolhido foi quarta-feira”)

caso “5”

escreva (“O dia escolhido foi quinta-feira”)

caso “6”

escreva (“O dia escolhido foi sexta-feira”)

caso “7”

escreva (“O dia escolhido foi sábado”)

outrocaso

escreva (“Não existe dia da semana para esse número”)

fimescolha

fimalgoritmo



5.1.3 Linguagem C

```
#include <stdio.h>
#include <conio.h>
int main (void)
{
    int opcao;
    printf ("Digite um número de 1 a 7: ");
    scanf ("%d", &opcao);

    switch (opcao)
    {
        case 1:
            printf ("O dia escolhido foi domingo\n");
            break;
        case 2:
            printf ("O dia escolhido foi segunda-feira\n");
            break;
        case 3:
            printf ("O dia escolhido foi terça-feira\n");
            break;
        case 4:
            printf ("O dia escolhido foi quarta-feira\n");
            break;
        case 5:
            printf ("O dia escolhido foi quinta-feira\n");
            break;
        case 6:
            printf ("O dia escolhido foi sexta-feira\n");
            break;
        case 7:
            printf ("O dia escolhido foi sábado\n");
            break;
        default:
            printf ("Não existe dia da semana para esse número\n");
    }
    system ("pause");
    return 0;
}
```



FINALIZANDO

Nesta aula, aprendemos os principais conceitos que envolvem as estruturas sequencial e de seleção (simples, composta, encadeada e de múltipla escolha), além de como utilizá-los nas representações algorítmicas: 1) fluxograma; 2) pseudocódigo; 3) linguagem de programação C. Aproveite a disciplina e bons estudos!



REFERÊNCIAS

GUEDES, S. **Lógica de programação algorítmica**. São Paulo: Pearson, 2014.

PUGA, S.; RISSETTI, G. **Lógica de programação e estruturas de dados**. São Paulo: Pearson, 2016.