

LÓGICA DE PROGRAMAÇÃO

DESVIOS CONDICIONAIS

LISTA DE FIGURAS

Figura 1 - Representação gráfica do SE ENTÃO	7
Figura 2 - Representação gráfica dos blocos	9
Figura 3 - Fluxograma do SE ENTÃO: Módulo matemático	10
Figura 4 - Representação gráfica do SE ENTÃO SENÃO	13
Figura 5 - Fluxograma do SE ENTÃO SENÃO – Maior de idade	15
Figura 6 - Fluxograma da representação do SE ENCADEADO	17
Figura 7 - Fluxograma do SE ENCADEADO – Dia da semana	20
Figura 8 - Fluxograma da estrutura ESCOLHA	23
Figura 9 - Fluxograma do “Dia da Semana” com o ESCOLHA	26
Figura 10 - Fluxograma “Pos/Neg/Nulo” – SE ENCADEADO	35
Figura 11 - Fluxograma “Maior 3 números” – SE ENTÃO	38

LISTA DE QUADROS

Quadro 1 – Execução do SE ENTÃO: Módulo matemático	10
Quadro 2 – Execução do SE ENTÃO SENÃO – Maior de idade.....	14
Quadro 3 – Execução do SE ENCADEADO – dia da semana.....	20
Quadro 4 – Execução do “Dia da semana” com o ESCOLHA.....	25
Quadro 5 – Execução da “Calculadora” com o ESCOLHA	29
Quadro 6 – Execução “Rodízio” com o ESCOLHA	32
Quadro 7 – Execução “Pos/Neg/Nulo” – SE ENCADEADO.....	34
Quadro 8 – Execução “Maior 3 números” – SE ENTÃO	38

LISTA DE CÓDIGOS-FONTE

Código-fonte 1 – Sintaxes do SE ENTÃO	8
Código-fonte 2 – Pseudocódigo do SE ENTÃO: módulo matemático	11
Código-fonte 3 – Python do SE ENTÃO: módulo matemático.....	11
Código-fonte 4 – Java do SE ENTÃO: Módulo matemático	12
Código-fonte 5 – Sintaxes do comando SE ENTÃO SENÃO	14
Código-fonte 6 – Pseudocódigo do SE ENTÃO SENÃO – Maior de idade	15
Código-fonte 7 – Python do SE ENTÃO SENÃO – Maior de idade.....	16
Código-fonte 8 – Java do SE ENTÃO SENÃO – Maior de idade	16
Código-fonte 9 – Sintaxe1 do SE ENCADEADO no Pseudocódigo	17
Código-fonte 10 – Sintaxe2 do SE ENCADEADO no Pseudocódigo	18
Código-fonte 11 – Sintaxe1 do SE ENCADEADO no Python	18
Código-fonte 12 – Sintaxe2 do SE ENCADEADO no Python	18
Código-fonte 13 – Sintaxe1 do SE ENCADEADO no Java	19
Código-fonte 14 – Sintaxe2 do SE ENCADEADO no Java	19
Código-fonte 15 – Pseudocódigo do SE ENCADEADO – Dia da semana.....	21
Código-fonte 16 – Python do SE ENCADEADO – Dia da semana	22
Código-fonte 17 – Python do SE ENCADEADO – Dia da semana	22
Código-fonte 18 – Pseudocódigo da sintaxe do ESCOLHA.....	24
Código-fonte 19 – Java da sintaxe do ESCOLHA (switch).....	24
Código-fonte 20 – Pseudocódigo do “Dia da Semana” com o ESCOLHA	27
Código-fonte 21 – Python do “Dia da Semana” com o SE ENCADEADO (simulando o ESCOLHA)	27
Código-fonte 22 – Java do “Dia da Semana” com o ESCOLHA (switch)	28
Código-fonte 23 – Pseudocódigo da “Calculadora” com o ESCOLHA.....	30
Código-fonte 24 – Python da “Calculadora” com o ESCOLHA (simulação)	30
Código-fonte 25 – Java da “Calculadora” com o ESCOLHA (switch).....	31
Código-fonte 26 – Pseudocódigo – “Rodízio” com o ESCOLHA.....	33
Código-fonte 27 – Python – “Rodízio” com o ESCOLHA	33
Código-fonte 28 – Java – “Rodízio” com o ESCOLHA	34
Código-fonte 29 – Pseudocódigo “Pos/Neg/Nulo” – SE ENCADEADO	35
Código-fonte 30 – Python “Pos/Neg/Nulo” – SE ENCADEADO	36
Código-fonte 31 – Python “Pos/Neg/Nulo” – SE ENCADEADO (elif)	36
Código-fonte 32 – Java “Pos/Neg/Nulo” – SE ENCADEADO.....	37
Código-fonte 33 – Java “Pos/Neg/Nulo” – SE ENCADEADO 2.....	37
Código-fonte 34 – Pseudocódigo “Maior 3 números” – SE ENTÃO	39
Código-fonte 35 – Python “Maior 3 números” – SE ENTÃO	39
Código-fonte 36 – Java “Maior 3 números” – SE ENTÃO	40

SUMÁRIO

<i>Desvios condicionais.....</i>	<i>6</i>
<i>1 Introdução.....</i>	<i>6</i>
<i>2 Desvio simples</i>	<i>7</i>
<i>3 Desvio Composto</i>	<i>12</i>
<i>4 Desvio Encadeado.....</i>	<i>16</i>
<i>5 Escolha</i>	<i>23</i>
<i>6 De-para: Desvio Simples, Composto, Encadeado e Escolha.....</i>	<i>28</i>

DESVIOS CONDICIONAIS

1 INTRODUÇÃO

Até então, aprendemos os comandos simples, os conhecidos Primitivos. Qual a sua característica? Eles são sequenciais e ocupam apenas uma linha, ou seja, eles executam o algoritmo na sequência em que os comandos aparecem.

A partir deste capítulo ensinaremos o nosso algoritmo a “pensar” tomando decisões de acordo com os dados apresentados.

A partir de agora, aprenderemos os comandos estruturados cuja característica é desviar o fluxo do algoritmo fazendo com que as instruções não sejam executadas necessariamente na sequência.

Todos os comandos estruturados têm um início e fim próprio. Por exemplo, o comando SE terá um FIM_SE (em uma outra linha).

Uma estrutura de decisão permite a escolha de um grupo de ações (blocos) a serem executadas quando determinadas condições são ou não satisfeitas, desta forma o fluxo das instruções a serem seguidas são escolhidas de acordo com o resultado da avaliação das condições.

Existem duas classificações desta estrutura: decisão e escolha:

- Decisão:
 - Desvio simples: SE ENTÃO
 - Desvio composto: SE ENTÃO SENÃO
 - Desvio encadeado
- Escolha

Aprenderemos cada uma destas instruções neste capítulo e as aplicaremos em em Pseudocódigo, Python e Java.

2 DESVIO SIMPLES

O comando de decisão SE ENTÃO é a primeira variação dos comandos estruturados de Decisão. Com ele é possível analisar uma condição e executar um determinado <bloco de comandos> caso a condição seja verdadeira, senão nada será executado.

Veja a sua estrutura SE ENTÃO na imagem a seguir:

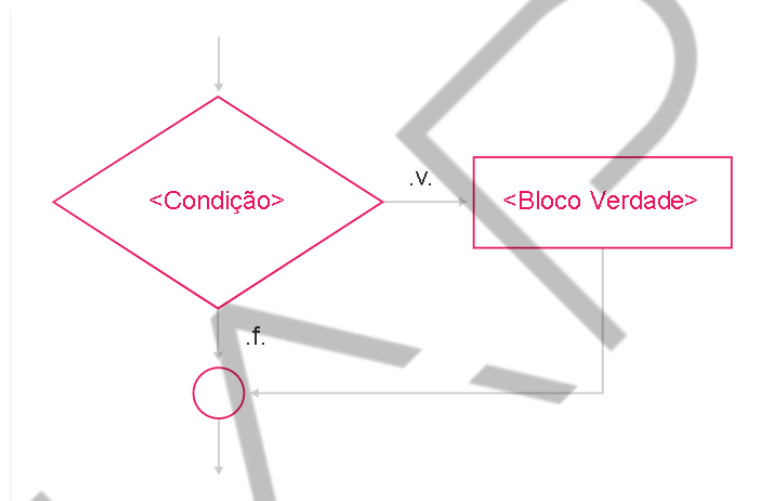


Figura 1 - Representação gráfica do SE ENTÃO
Fonte: Elaborada pelo autor (2022)

Condição é uma expressão qualquer que retorne um valor lógico.

Condição funciona como uma pergunta, onde uma resposta lógica (Verdade ou falso) determinará o andamento do fluxo.

Uma condição é uma expressão composta por três partes, sendo elas:

`variável` operador_relacional `variável_ou_valor`

Ex.: `x >= 3` ou `a = b`

Não se esqueça, sempre retornando uma resposta lógica: Verdade ou Falso

Momento Pseudocódigo

Sintaxe:

```
Se <condição> então  
    <bloco verdade>  
Fim_se
```

Momento Python

Sintaxe:

```
if <condição>:  
    <bloco verdade>
```

Momento Java

Sintaxe:

```
if (condição)  
{  
    <bloco verdade>;  
}
```

Código-fonte 1 – Sintaxes do SE ENTÃO
Fonte: Elaborado pelo autor (2022)

Observação sobre o <bloco verdade>: Este bloco é executado caso a condição resulte verdade.

Conceito de Bloco:

Antes de prosseguirmos vamos conhecer um conceito importante utilizado em todos os comandos estruturados em todas as linguagens de programação: o Bloco.

Bloco é um conjunto de instruções (comandos) que é isolado pelo programador dos demais comandos por atender um objetivo em comum.

Por exemplo, caso não seja possível calcular o valor de delta pelos dados fornecidos não serem numéricos, há a separação de instruções que efetuam o cálculo de delta das instruções que efetuam mensagens de advertência.

Cada linguagem de programação tem uma forma diferente (as vezes igual) de separar e gerenciar os blocos, vamos falar das linguagens que estamos utilizando: Pseudocódigo, Python e Java:

- **Pseudocódigo:** Não há um caractere ou palavra específica para delimitar o bloco. O Pseudocódigo geralmente utiliza o Fim_(nome do comando). Por exemplo, o FIM_SE.
- **Python:** Ele entende o bloco por meio da indentação (neologismo derivado da palavra inglesa *indentation* que significa recuo), em outros termos, as instruções sequenciais na mesma coluna estão no mesmo bloco.

Indentação é uma técnica de edição de código-fonte importante em todas as linguagens de programação por organizar o código-fonte e tornar mais fácil a identificação de um erro de compilação.

Em Python, é a forma de criar blocos, logo, se o código for indentado errado, influencia em sua execução. Nas outras linguagens, se indentar errado, mas a linha de código não tiver com erro de compilação, o programa funciona normalmente.

- **Java:** Ela entende como bloco todas as instruções que estão entre chaves ({bloco}). Lembre-se apenas que a quantidade de chaves que abrem são as mesmas que devem fechar, caso isso não seja respeitado, ocorrerá um erro de compilação.

A figura abaixo exemplifica e compara as construções dos blocos em Pseudocódigo, Python e Java:

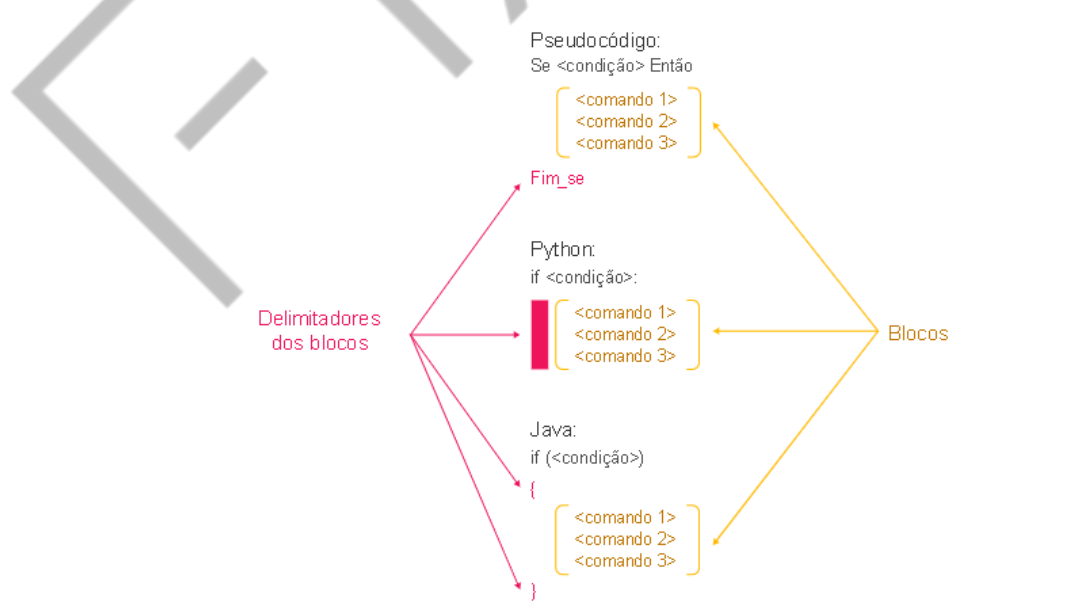


Figura 2 - Representação gráfica dos blocos
Fonte: Elaborada pelo autor (2022)

Vamos agora aplicar o conceito de decisão simples (SE ENTÃO), a partir de um exemplo utilizando: Pseudocódigo, Python e Java.

- Exemplo – SE ENTÃO:

Fazer um algoritmo que solicite um número ao usuário e exiba o seu módulo matemático.

Módulo matemático é tornar um número positivo. Se ele for, continua, senão passa a ser.

Exemplos:

$$|56| = 56$$

$$|-43| = 43$$

A execução deste programa será:

```
Digite um número: -5
Módulo: 5
```

Quadro 1 – Execução do SE ENTÃO: Módulo matemático
Fonte: Elaborado pelo autor (2022)

Segue a ideia do algoritmo sendo representada pelo fluxograma:

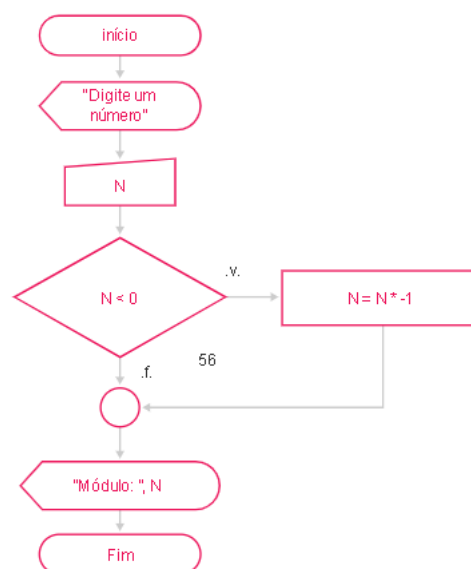


Figura 3 - Fluxograma do SE ENTÃO: Módulo matemático
Fonte: Elaborada pelo autor (2022)

Fluxograma é a representação do algoritmo por intermédio de figuras.

Com ele temos o “desenho” do algoritmo. Este recurso é importante por enxergarmos a resolução do problema graficamente; também por fazer parte da documentação de um sistema.

Representação do exemplo - Pseudocódigo:

```
Programa_exibir_numero_positivo
Var
    n: inteiro
Início
    //exibe "Solicita um número para o usuário"
    Escreva "Digite um número:"
    Leia n
    // Comando de decisão: Verifica se o número é negativo
    Se n < 0 então
        n = n * -1
    Fim_se
    //Exibe o número positivo
    Escreva "Módulo: ", n
Fim
```

Código-fonte 2 – Pseudocódigo do SE ENTÃO: módulo matemático
Fonte: Elaborado pelo autor (2022)

Representação do exemplo - Python:

```
# Solicita um número digitado pelo usuário
n = int(input("Digite um número:"))
#Comando de decisão: Verifica se o número é negativo
if n < 0:
    n = n * (-1)
# Exibe o número positivo
print("Módulo: ",n)
```

Código-fonte 3 – Python do SE ENTÃO: módulo matemático
Fonte: Elaborado pelo autor (2022)

Representação do exemplo - Java:

```
import java.util.Scanner;
public class First {
    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        int n;
```

```
System.out.println("Digite um número:");
n = entrada.nextInt();
// Decisão simples: Verifica se o número é negativo
if (n < 0)
{ // Se verdade executa este bloco
    n = n * -1;
}
//Exibe o número positivo
System.out.println("Módulo: " + n);
}
```

Código-fonte 4 – Java do SE ENTÃO: Módulo matemático
Fonte: Elaborado pelo autor (2022)

Como visto, esta estrutura é utilizada apenas se o lado verdadeiro de uma condição precisar ser operado.

3 DESVIO COMPOSTO

O comando anterior (SE ENTÃO) executava somente o bloco verdade, o lado verdadeiro da condição, ou seja, não existia o bloco falso. O desvio composto tem tanto o <bloco verdadeiro> quanto o <bloco falso>.

O comando SE ENTÃO SENÃO é utilizado quando o desvio for composto por dois blocos, o verdadeiro e o falso.

Este comando é uma bifurcação. Um de dois caminhos é seguido a partir da resposta de uma condição pré-estabelecida (depois da palavra SE), caso a condição resulte verdade, será executado o <bloco verdade> (que é representado pelo ENTÃO), caso contrário será executado o <bloco falso> (que é representado pelo SENÃO).

Independentemente do bloco executado, o algoritmo segue a sequência do fluxo.

Veja o funcionamento da estrutura SE ENTÃO SENÃO na imagem a seguir:

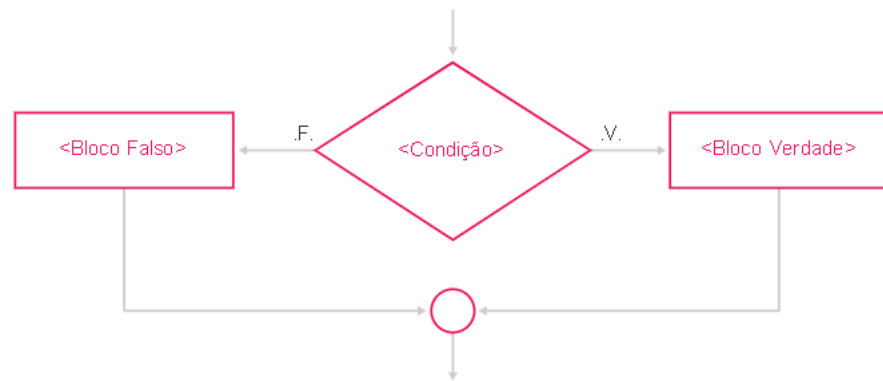


Figura 4 - Representação gráfica do SE ENTÃO SENÃO
Fonte: Elaborada pelo autor (2022)

No código de programação o que modifica em relação ao comando SE ENTÃO, é o acréscimo do SENÃO (Else) no corpo da estrutura. Lembrando que a partir do ENTÃO é executado o bloco verdadeiro e a partir do SENÃO é executado o bloco falso.

Vejamos as sintaxes do comando SE ENTÃO SENÃO no Pseudocódigo, Python e Java:

Momento Pseudocódigo

Sintaxe:

```
Se <condição> então  
    <bloco verdadeiro>  
Senão  
    <bloco falso>  
Fim_se
```

Momento Python

Sintaxe:

```
if <condição>:  
    <bloco Verdadeiro>  
else:  
    <bloco Falso>
```

Momento Java

Sintaxe:

```
if (condição)  
{  
    <bloco verdadeiro>;  
}
```

```
}  
else  
{  
    <bloco falso>;  
}
```

Código-fonte 5 – Sintaxes do comando SE ENTÃO SENÃO
Fonte: Elaborado pelo autor (2019)

Particularidade do Java em relação ao Bloco:

A obrigatoriedade do uso das chaves no bloco se dá caso haja mais de um comando no bloco, se houver apenas um comando, o ponto e vírgula (;) é suficiente para formalizar o bloco.

- Exemplo – SE ENTÃO SENÃO:

Solicitar ao usuário a sua idade e exibir mensagem “Maior de idade” caso a idade seja maior ou igual a 18 anos ou “Menor de idade” caso contrário.

Execução do exemplo:

```
Digite sua idade:4  
Menor de idade
```

Quadro 2 – Execução do SE ENTÃO SENÃO – Maior de idade
Fonte: Elaborado pelo autor (2022)

Segue a ideia do Algoritmo sendo representada pelo fluxograma:

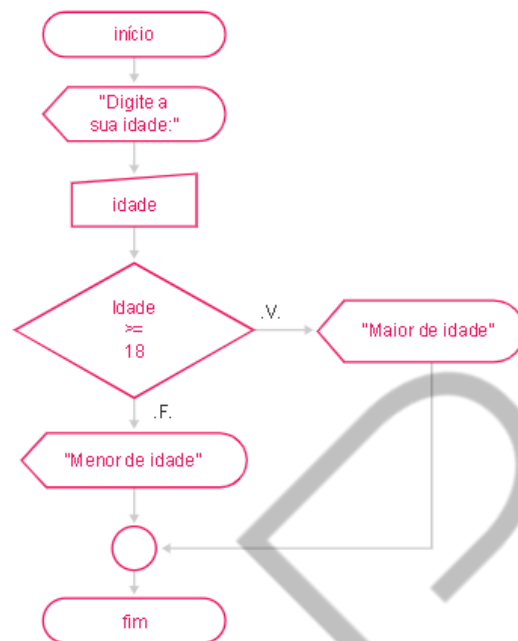


Figura 5 - Fluxograma do SE ENTÃO SENÃO – Maior de idade
 Fonte: Elaborada pelo autor (2022)

Representação do exemplo por meio do Pseudocódigo:

```

Programa desvio_composto
Var
    idade: inteiro
Início
    // Solicita a idade do usuário
    Escreva "Digite sua idade:"
    Leia idade
    // Desvio composto exibe mensagem "Maior de idade" quando a idade
    for maior ou igual a 18 anos e "Menor de idade" caso contrário
    Se idade >= 18 então
        Escreva "Maior de idade"
    Senão
        Escreva "Menor de idade"
    Fim_se
Fim
  
```

Código-fonte 6 – Pseudocódigo do SE ENTÃO SENÃO – Maior de idade
 Fonte: Elaborado pelo autor (2022)

Representação do exemplo em Python:

```

# Pede a digitação da idade do usuário
idade = input("Digite sua idade: ")
idade = int(idade)
# Verifica se a idade atende a condição e exibe mensagem
if idade >= 18:
    print("Maior de idade")
  
```

```
else:  
    print("Menor de idade")
```

Código-fonte 7 – Python do SE ENTÃO SENÃO – Maior de idade
Fonte: Elaborado pelo autor (2022)

Representação do exemplo em Java:

```
import java.util.Scanner;  
public class First {  
    public static void main(String[] args) {  
        Scanner entrada = new Scanner(System.in);  
  
        int idade;  
        System.out.println("Digite sua idade:");  
        idade = entrada.nextInt();  
        // Desvio composto exibe mensagem "Maior de idade" quando a idade for  
        maior ou igual a 18 anos e "Menor de idade" caso contrário  
        if (idade >= 18)  
        {  
            System.out.println("Maior de idade");  
        }  
        else  
        {  
            System.out.println("Menor de idade");  
        }  
    }  
}
```

Código-fonte 8 – Java do SE ENTÃO SENÃO – Maior de idade
Fonte: Elaborado pelo autor (2022)

4 DESVIO ENCADEADO

Com a estrutura de decisão simples, podemos percorrer um caminho (o bloco verdade); na estrutura de decisão composta podemos percorrer um entre dois caminhos (bloco verdade e bloco falso). Daí vêm a pergunta: “E se precisarmos percorrer um entre vários caminhos?”

Neste caso utilizaremos o SE ENCADEADO.

Uma estrutura de decisão encadeada é uma sequência de testes de seleção, os quais serão executados ou não de acordo com o resultado das condições e com o encadeamento dos testes.

Isto é, um teste de seleção pode ter dois conjuntos de instruções, um para o resultado verdadeiro e outro para o falso, porém esses conjuntos podem conter outros testes de seleção, que por sua vez, também podem conter outros e assim por diante.

Vejamos a imagem que representa o se encadeado:

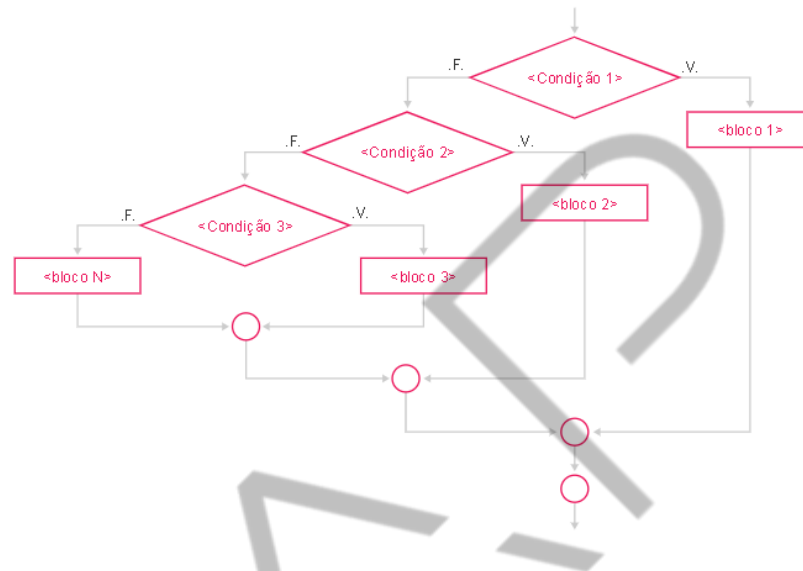


Figura 6 - Fluxograma da representação do SE ENCADEADO

Fonte: Elaborada pelo autor (2022)

Resumidamente, denominamos SE ENCADEADO em situações em que há um comando de decisão – não importa se é o SE ENTÃO ou SE ENTÃO SENÃO – dentro do outro.

Vejamos as sintaxes nas nossas linguagens:

Momento Pseudocódigo

Sintaxe:

```
Se <condição 1> então
    <bloco de instruções 1>
Senão
    Se <condição 2> então
        <bloco de instruções 2>
    Senão
        Se <condicao3> então
            <bloco de instruções 3>
        Senão
            <bloco de instruções N>
    Fim_se
Fim_se
Fim_se
```

Código-fonte 9 – Sintaxe1 do SE ENCADEADO no Pseudocódigo

Fonte: Elaborado pelo autor (2022)

No Pseudocódigo para cada SE aberto deve ter um FIM_SE

Ou podemos escrever:

Sintaxe:

```
Se <condição 1> então
    <bloco de instruções 1>
Senão Se <condição 2> então
    <bloco de instruções 2>
Senão Se <condicao3> então
    <bloco de instruções 3>
Senão
    <bloco de instruções N>
Fim_se
```

Código-fonte 10 – Sintaxe2 do SE ENCADEADO no Pseudocódigo
Fonte: Elaborado pelo autor (2022)

Momento Python

Sintaxe:

```
if <condição 1>:
    <bloco de instruções 1>
else:
    if <condição 2>:
        <bloco de instruções 2>
    else:
        if <condicao3>:
            <bloco de instruções 3>
        else
            <bloco de instruções N>
```

Código-fonte 11 – Sintaxe1 do SE ENCADEADO no Python
Fonte: Elaborado pelo autor (2022)

Ou podemos escrever:

Sintaxe:

```
if <condição 1>:
    <bloco de instruções 1>
elif <condição 2>:
    <bloco de instruções 2>
elif <condicao3>:
    <bloco de instruções 3>
else
    <bloco de instruções N>
```

Código-fonte 12 – Sintaxe2 do SE ENCADEADO no Python
Fonte: Elaborado pelo autor (2022)

No Python o “elif” abrevia o encadeamento dos else if’s, tornando a sua estrutura mais legível.

Momento Java

Sintaxe:

```
if(condição1)
{
    <expressão>;
}
else
{
    if(condicao2)
    {
        <expressão2>;
    }
    else
    {
        if(condicao3)
        {
            <expressão3>;
        }
        else
        {
            <expressão_n>;
        }
    }
}
```

Código-fonte 13 – Sintaxe1 do SE ENCADEADO no Java
Fonte: Elaborado pelo autor (2022)

Ou podemos escrever:

```
if(condição1)
    <expressão>;
else if(condicao2)
    <expressão2>;
else if(condicao3)
    <expressão3>;
else
    <expressãoN>;
```

Código-fonte 14 – Sintaxe2 do SE ENCADEADO no Java
Fonte: Elaborado pelo autor (2022)

Vale lembrar que no Java a utilização das chaves, para delimitar o bloco, será obrigatório caso haja mais de um comando compondo o bloco.

Vistas as sintaxes no Pseudocódigo, Python e Java, vamos utilizar o mesmo exemplo para todas as linguagens:

- Exemplo – SE ENCADEADO:

Solicitar um número ao usuário. A partir do número, exibir o dia da semana correspondente.

Execução do exemplo:

Digite um número: 3
Terça-Feira

Quadro 3 – Execução do SE ENCADEADO – dia da semana
Fonte: Elaborado pelo autor (2022)

Segue a ideia do Algoritmo sendo representada pelo fluxograma:

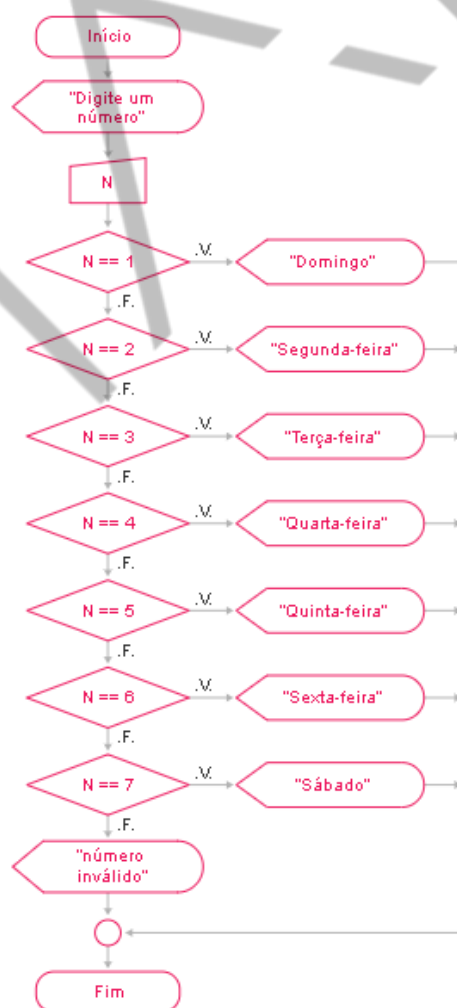


Figura 7 - Fluxograma do SE ENCADEADO – Dia da semana
Fonte: Elaborada pelo autor (2022)

Momento Pseudocódigo

Aplicação do exemplo no Pseudocódigo:

```
Programa exemplo1
Var
    n : inteiro
Início
    // Solicita um número ao usuário
    Escreva "Digite um número:"
    Leia n
    // Compara o n aos valores possíveis
    Se (n == 1) então
        Escreva "Domingo "
    Senão se (n == 2) então
        Escreva "Segunda-feira"
    Senão se (n == 3) então
        Escreva "Terça-feira "
    Senão se (n == 4) então
        Escreva "Quarta-feira "
    Senão se (n == 5) então
        Escreva "Quinta-feira "
    Senão se (n == 6) então
        Escreva "Sexta-feira "
    Senão se (n == 7) então
        Escreva "Sábado "
    Senão
        Escreva "Dia inválido "
    Fim_Se
Fim
```

Código-fonte 15 – Pseudocódigo do SE ENCADEADO – Dia da semana
Fonte: Elaborado pelo autor (2022)

Momento Python

Aplicação do exemplo “dia da semana” no Python:

```
# Pede a digitação do dia da semana
n = int(input("Digite um número: "))

# Verifica se o número atende a condição e exibe mensagem
if n == 1:
    print("Domingo")
elif n == 2:
    print("Segunda-feira")
elif n == 3:
    print("Terça-feira")
elif n == 4:
    print("Quarta-feira")
```

```
elif n == 5:
    print("Quinta-feira")
elif n == 6:
    print("Sexta-feira")
elif n == 7:
    print("Sábado")
else:
    print("Número inválido")
```

Código-fonte 16 – Python do SE ENCADEADO – Dia da semana

Fonte: Elaborado pelo autor (2022)

Momento Java

Aplicação do exemplo “dia da semana” no Java:

```
import java.util.Scanner;

public class ExibirNumeroPositivo {

    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);

        int n;
        System.out.println("Digite um número:");
        n = entrada.nextInt();
        // Desvio encadeado exibe o dia da semana de acordo com o valor numérico
        digitado, caso seja digitado um número inválido exibe a mensagem Dia inválido.
        if (n == 1)
            System.out.println("Domingo");
        else if (n == 2)
            System.out.println("Segunda");
        else if (n == 3)
            System.out.println("Terça");
        else if (n == 4)
            System.out.println("Quarta");
        else if (n == 5)
            System.out.println("Quinta");
        else if (n == 6)
            System.out.println("Sexta");
        else if (n == 7)
            System.out.println("Sábado");
        else
            System.out.println("Dia inválido");
    }
}
```

Código-fonte 17 – Python do SE ENCADEADO – Dia da semana

Fonte: Elaborado pelo autor (2022)

5 ESCOLHA

A estrutura de seleção escolha é similar ao SE ENCADEADO por verificar diversas possibilidades de execução. A desvantagem é de escolha só comparar equivalência (igual a) enquanto o SE ENCADEADO pode utilizar qualquer operador lógico ou relacional (>, <, and...), a vantagem é a de fechar apenas uma vez independente da quantidade de caminhos, agora no SE para cada SE há um FIM_SE.

A estrutura de decisão ESCOLHA testa o valor contido em um identificador (que pode ser uma variável ou um cálculo inteiro) colocada entre parênteses e o compara com valores fornecidos em cada caso. Nesta estrutura podemos ter quantos casos desejarmos e quando um dos valores corresponder ao do identificador de teste, o código é executado.

Vejamos o seu funcionamento no fluxograma:

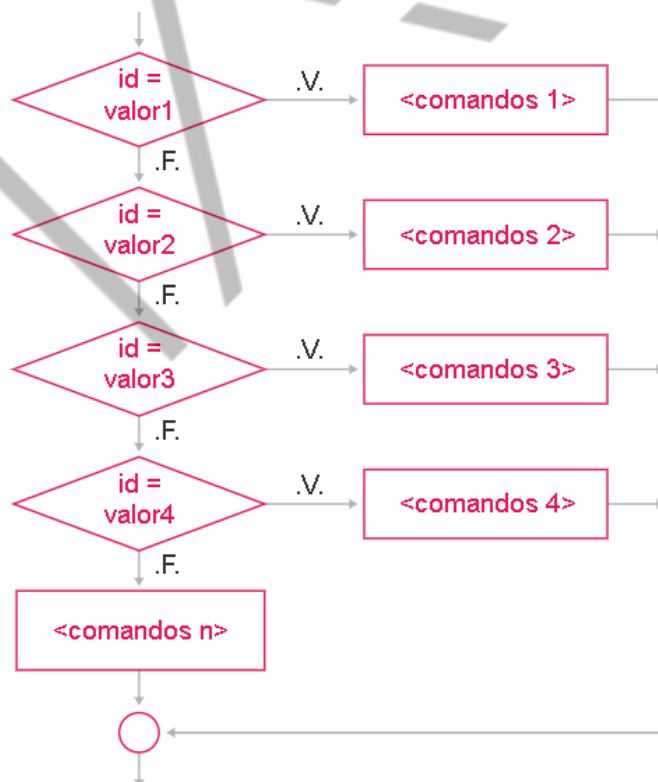


Figura 8 - Fluxograma da estrutura ESCOLHA

Fonte: Elaborada pelo autor (2022)

Vejamos as sintaxes nas nossas linguagens:

Momento Pseudocódigo

Sintaxe:

```
Escolha (<identificador>)
  Caso <valor1>:
    <comandos 1>
  Caso <valor2>:
    <comandos2>
  Caso <valor3>:
    <comandos3>
  Caso <valor4>:
    <comandos4>
  Outrocaso:
    <outros comandos>
Fim_Escolha
```

Código-fonte 18 – Pseudocódigo da sintaxe do ESCOLHA
Fonte: Elaborado pelo autor (2022)

Momento Python

NO PYTHON NÃO EXISTE A ESTRUTURA “ESCOLHA”, MAS ELA PODE SER SIMULADA COM O SE ENCADEADO (IF ELIF) QUE UTILIZE APENAS A EQUIVALENCIA (==) COMO OPERADOR.

Momento JAVA

Sintaxe:

```
switch (<identificador>)
{
  case <valor1>:
    <comandos 1>;
    [break;]
  case <valor1>:
    <comandos 1>;
    [break;]
  case <valor1>:
    <comandos 1>;
    [break;]
  case <valor1>:
    <comandos 1>;
    [break;]
  default:
    <comandos default>
}
```

Código-fonte 19 – Java da sintaxe do ESCOLHA (switch)
Fonte: Elaborado pelo autor (2022)

Em Java o comando `break` força a saída da estrutura (serve para todas as estruturas, inclusive as que ainda não vimos).

Caso o `break`; não seja colocado, assim que o fluxo entrar em um `case`, todas as linhas subsequentes serão executadas até o final da estrutura.

Para exemplificar a estrutura ESCOLHA (`switch`) utilizaremos o mesmo exemplo do SE ENCADEADO (exibir o dia da semana) para que consigamos analisar a diferença entre ambos.

- Exemplo – ESCOLHA:

Solicitar um número ao usuário. A partir do número, exibir o dia da semana correspondente.

Execução do exemplo “dia da semana” com a estrutura escolha:

```
Digite um número: 3  
Terça-Feira
```

Quadro 4 – Execução do “Dia da semana” com o ESCOLHA
Fonte: Elaborado pelo autor (2022)

Segue a ideia do Algoritmo sendo representada pelo fluxograma:

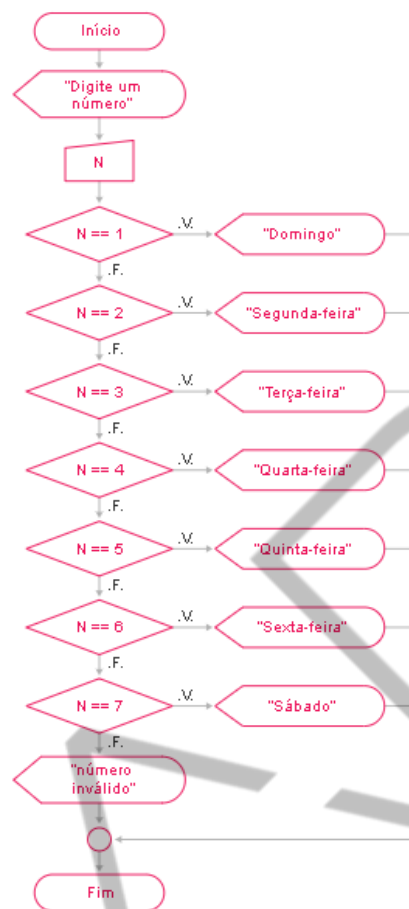


Figura 9 - Fluxograma do “Dia da Semana” com o ESCOLHA
 Fonte: Elaborada pelo autor (2022)

Lembrando que a estrutura ESCOLHA no fluxograma é similar ao SE ENCADEADO.

Momento Pseudocódigo

Aplicação do exemplo no Pseudocódigo:

```

Programa exemplo_escolha
Var
    n : inteiro
Início
    // Solicita um número ao usuário
    Escreva "Digite um número:"
    Leia n
    // Compara o n aos valores possíveis
    Escolha(n)
        Caso 1:
            Escreva "Domíngo"
        Caso 2:
            Escreva "Segunda-feira"
        Caso 3:

```

```
        Escreva "Terça-feira"
Caso 4:
        Escreva "Quarta-feira"
Caso 5:
        Escreva "Quinta-feira"
Caso 6:
        Escreva "Sexta-feira"
Caso 7:
        Escreva "Sábado"
Outro caso:
        Escreva "Dia inválido"
Fim_Escolha
Fim
```

Código-fonte 20 – Pseudocódigo do "Dia da Semana" com o ESCOLHA
Fonte: Elaborado pelo autor (2022)

Momento Python

No Python **não existe** uma estrutura Escolha (switch), como no Java. Sendo assim, vamos improvisar o **if elif** para simulá-lo:

```
dia_semana = int(input("Digite um número: "))
if dia_semana == 1:
    print("Domingo")
elif dia_semana == 2:
    print("Segunda-feira")
elif dia_semana == 3:
    print("Terça-feira")
elif dia_semana == 4:
    print("Quarta-feira")
elif dia_semana == 5:
    print("Quinta-feira")
elif dia_semana == 6:
    print("Sexta-feira")
elif dia_semana == 7:
    print("Sábado")
else:
    print("Número inválido")
```

Código-fonte 21 – Python do "Dia da Semana" com o SE ENCADEADO (simulando o ESCOLHA)
Fonte: Elaborado pelo autor (2022)

Momento Java

Segue a codificação do exemplo com o ESCOLHA (switch) no Java:

```
import java.util.Scanner;
public class First
{
    public static void main(String[] args)
```

```
{
    Scanner entrada = new Scanner(System.in);
    int n;
    System.out.println("Digite um número:");
    n = entrada.nextInt();
    // Verifica qual número foi digitado.
    switch (n)
    {
        case 1:
            System.out.println("Domingo"); break;
        case 2:
            System.out.println("Segunda-feira"); break;
        case 3:
            System.out.println("Terça-feira"); break;
        case 4:
            System.out.println("Quarta-feira"); break;
        case 5:
            System.out.println("Quinta-feira"); break;
        case 6:
            System.out.println("Sexta-feira"); break;
        case 7:
            System.out.println("Sábado"); break;
        default:
            System.out.println("Dia inválido");
    }
}
```

Código-fonte 22 – Java do “Dia da Semana” com o ESCOLHA (switch)
Fonte: Elaborado pelo autor (2022)

O comando break força a saída da estrutura.

Caso ele não seja colocado, serão executados todos os cases a partir do case que tiver o valor verdadeiro

6 DE-PARA: DESVIO SIMPLES, COMPOSTO, ENCADEADO E ESCOLHA

Neste de-para, faremos alguns exercícios para aplicarmos os comandos de Decisão aprendidos neste capítulo:

- Exemplo 1 – calculadora – ESCOLHA:

Dados dois números inteiros e uma operação aritmética desejada calcule a resposta adequada: Utilize os símbolos a seguir para ler qual a operação escolhida:

+ soma - subtração * multiplicação / divisão

Segue a execução do exemplo:

```
Digite o primeiro número: 5
Digite o segundo número: 5
Digite a operação desejada (+ - * /): +
Soma = 10
```

Quadro 5 – Execução da “Calculadora” com o ESCOLHA
Fonte: Elaborado pelo autor (2022)

Momento Pseudocódigo

```
Programa operacao_aritmetica
Var
    numero1, numero2, resultado: real
    operador: caractere
Início
    // Solicitar os dados ao usuário
    Escreva "Digite o primeiro número:"
    Leia numero1
    Escreva "Digite o segundo número:"
    Leia numero2
    Escreva "Digite a operação desejada (+ - * /):"
    Leia operador
    // Verifica a operação desejada
    Escolha (operador)
        caso '+':
            resultado = numero1 + numero2
            Escreva "Soma = ", resultado
        caso '-':
            resultado = numero1 - numero2
            Escreva "Subtração = ", resultado
        caso '*':
            resultado = numero1 * numero2
            Escreva "Multiplicação = ", resultado
        caso '/':
            Se numero2 == 0 então
                Escreva "Não há divisão por zero"
            Senão
                resultado = numero1 / numero2
                Escreva "Divisão = ", resultado
        Fim_se
    caso contrário:
```

```
        Escreva "Operador inválido"
    Fim_Escolha
Fim
```

Código-fonte 23 – Pseudocódigo da "Calculadora" com o ESCOLHA
Fonte: Elaborado pelo autor (2022)

Momento Python

```
numero1= float(input("Digite o primeiro número:"))
numero2= float(input("Digite o primeiro número:"))
operador = str(input("Digite a operação desejada (+ - * /):"))
#Desvio encadeado exibe o dia da semana de acordo com o valor numérico
digitado, caso seja digitado um número inválido exibe a mensagem Dia inválido.
if operador == '+':
    resultado = numero1 + numero2
    print("Soma = ", resultado)
elif operador == '-':
    resultado = numero1 - numero2
    print("Subtração = ", resultado)
elif operador == '*':
    resultado = numero1 * numero2
    print("Multiplicação = ", resultado)
elif operador == '/':
    if numero2 == 0:
        print("Não existe divisão por zero")
    else:
        resultado = numero1 / numero2
        print("Divisão = ", resultado)
else:
    print("Operador inválido")
```

Código-fonte 24 – Python da "Calculadora" com o ESCOLHA (simulação)
Fonte: Elaborado pelo autor (2022)

Lembrando que, no Python, não temos o comando switch por isso utilizamos o desvio encadeado.

Momento Java

```
import java.util.Scanner;
public class First
{
    public static void main(String[] args)
    {
        Scanner entrada = new Scanner(System.in);
```

```
float numero1, numero2, resultado;
String operador;
System.out.println("Digite o primeiro número:");
numero1 = entrada.nextFloat();
System.out.println("Digite o segundo número:");
numero2 = entrada.nextFloat();
System.out.println("Digite a operação desejada (+ - * /):");
operador = entrada.next();
// Analisa qual o sinal digitado para fazer a operação correspondente
switch (operador)
{
    case "+":
        resultado = numero1 + numero2;
        System.out.println("Soma = " + resultado);
        break;
    case "-":
        resultado = numero1 - numero2;
        System.out.println("Subtração = " + resultado);
        break;
    case "*":
        resultado = numero1 * numero2;
        System.out.println("Multiplicação = " + resultado);
        break;
    case "/":
        // Certifica-se que o segundo número não é zero
        if(numero2 == 0)
            System.out.println("Não há divisão por zero");
        else
        {
            resultado = numero1 / numero2;
            System.out.println("Divisão = " + resultado);
        }
        break;
    default:
        System.out.println("Operador inválido");
}
}
```

Código-fonte 25 – Java da “Calculadora” com o ESCOLHA (switch)
Fonte: Elaborado pelo autor (2022)

- Exemplo 3 – Rodízio – ESCOLHA:

Dada a parte numérica da placa de um automóvel, fazer um algoritmo que exiba o dia do rodízio.

Considere a regra de São Paulo. Para veículos com final da placa :

- 1 e 2 o rodízio é segunda-feira,
- 3 e 4 terça-feira,
- 5 e 6 quarta-feira,
- 7 e 8 quinta-feira e
- 9 e 0 sexta-feira.

Segue a execução do exemplo “Rodízio”:

Digite o número da placa do veículo: 9587
Quinta-feira

Quadro 6 – Execução “Rodízio” com o ESCOLHA
Fonte: Elaborado pelo autor (2022)

Momento Pseudocódigo

```
Programa exemplo3
Var
    numeroPlaca, finalPlaca: inteiro
Início
    /* Solicita a placa do veículo */
    Escreva "Digite o número da placa do veículo:"
    Leia numeroPlaca
    /* Efetua o cálculo para saber o último número */
    finalPlaca = numeroPlaca % 10
    /* Verifica o final da placa com o comando Escolha e exibe o dia da
    semana do rodízio */
    Escolha (finalPlaca)
        caso 1:
        caso 2:
            Escreva "Segunda-feira"
        caso 3:
        caso 4:
            Escreva "Terça-feira"
        caso 5:
        caso 6:
            Escreva "Quarta-feira"
        caso 7:
        caso 8:
            Escreva "Quinta-feira"
        caso 9:
```



```
    caso 0:  
        Escreva "Sexta-feira"  
    Fim_Escolha  
Fim
```

Código-fonte 26 – Pseudocódigo – “Rodízio” com o ESCOLHA
Fonte: Elaborado pelo autor (2022)

Momento Python

```
# Solicita o número da placa  
numeroPlaca=int(input("Digite o número da placa do veículo:"))  
finalPlaca = numeroPlaca % 10  
if finalPlaca == 1 or finalPlaca==2:  
    print("Segunda-feira");  
elif finalPlaca == 3 or finalPlaca==3:  
    print("Terça-feira");  
elif finalPlaca == 5 or finalPlaca==6:  
    print("Quarta-feira");  
elif finalPlaca == 7 or finalPlaca==8:  
    print("Quinta-feira");  
else:  
    print("Sexta-feira");
```

Código-fonte 27 – Python – “Rodízio” com o ESCOLHA
Fonte: Elaborado pelo autor (2022)

Em Python, o comando:

```
if finalPlaca == 1 or finalPlaca == 2:
```

Pode ser substituído pelo:

```
if finalPlaca in (1,2):
```

Momento Java

```
import java.util.Scanner;  
public class First  
{  
    public static void main(String[] args)  
    {  
        Scanner entrada = new Scanner(System.in);  
  
        int numeroPlaca, finalPlaca;  
        System.out.println("Digite o número da placa:");  
        numeroPlaca = entrada.nextInt();  
        finalPlaca = numeroPlaca % 10;  
        switch(finalPlaca) {
```

```
case 1:  
case 2: System.out.println("Segunda-feira"); break;  
case 3:  
case 4: System.out.println("Terça-feira"); break;  
case 5:  
case 6: System.out.println("Quarta-feira"); break;  
case 7:  
case 8: System.out.println("Quinta-feira"); break;  
case 9:  
case 0: System.out.println("Sexta-feira"); break;  
}  
}
```

Código-fonte 28 – Java – “Rodízio” com o ESCOLHA
Fonte: Elaborado pelo autor (2022)

- Exemplo 4: Pos/Neg/Nulo - SE ENCADEADO

Dado um número pelo usuário, informar se ele é positivo, negativo ou nulo.

Segue a execução do exemplo:

Digite um número: 7
Positivo

Quadro 7 – Execução “Pos/Neg/Nulo” – SE ENCADEADO
Fonte: Elaborado pelo autor (2022)

Seque o fluxograma:

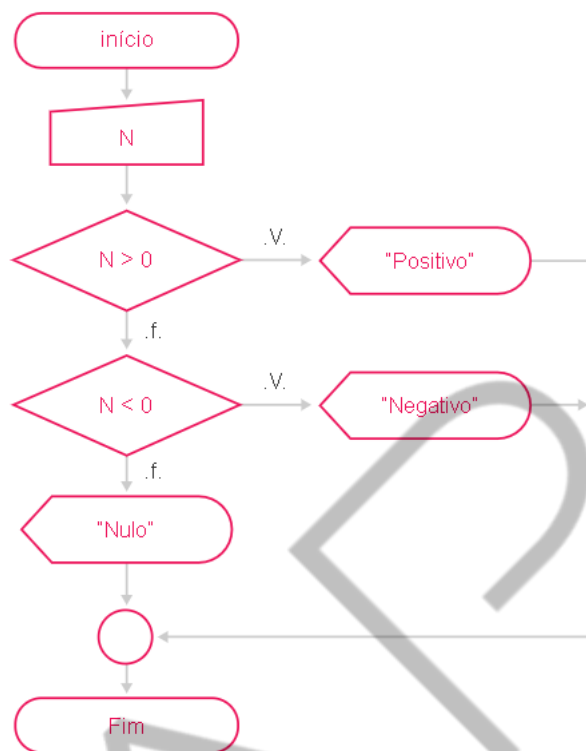


Figura 10 - Fluxograma "Pos/Neg/Nulo" – SE ENCADEADO
 Fonte: Elaborada pelo autor (2022)

Momento Pseudocódigo

```

Programa exemplo4
Var
  n: inteiro
Início
  Escreva "Digite um número: "
  Leia n
  Se n > 0 então
    Escreva "Positivo"
  Senão
    Se n < 0 então
      Escreva "Negativo"
    Senão
      Escreva "Nulo"
  Fim_se
Fim_se
Fim
  
```

Código-fonte 29 – Pseudocódigo "Pos/Neg/Nulo" – SE ENCADEADO
 Fonte: Elaborado pelo autor (2022)

Momento Python

```
n = int(input("Digite um número: "))
if n > 0:
    print("Positivo")
else:
    if n < 0:
        print("Negativo")
    else:
        print("Nulo")
```

Código-fonte 30 – Python “Pos/Neg/Nulo” – SE ENCADEADO
Fonte: Elaborado pelo autor (2022)

Ou poderia ser executado com o elif, como segue:

```
n = int(input("Digite um número: "))
if n > 0:
    print("Positivo")
elif n < 0:
    print("Negativo")
else:
    print("Nulo")
```

Código-fonte 31 – Python “Pos/Neg/Nulo” – SE ENCADEADO (elif)
Fonte: Elaborado pelo autor (2022)

Momento Java

```
import java.util.Scanner;
public class First
{
    public static void main(String[] args)
    {
        Scanner entrada = new Scanner(System.in);

        int n;
        System.out.println("Digite um número:");
        n = entrada.nextInt();
        if (n > 0)
        {
            System.out.println("Positivo");
        }
        else
        {
```

```
        if (n < 0)
        {
            System.out.println("Negativo");
        }
        else
        {
            System.out.println("Nulo");
        }
    }
}
```

Código-fonte 32 – Java “Pos/Neg/Nulo” – SE ENCADEADO

Fonte: Elaborado pelo autor (2022)

Ou poderia ser executado com o else if, como segue:

```
import java.util.Scanner;
public class First
{
    public static void main(String[] args)
    {
        Scanner entrada = new Scanner(System.in);

        int n;
        System.out.println("Digite um número:");
        n = entrada.nextInt();
        if (n > 0)
            System.out.println("Positivo");
        else if (n < 0)
            System.out.println("Negativo");
        else
            System.out.println("Nulo");
    }
}
```

Código-fonte 33 – Java “Pos/Neg/Nulo” – SE ENCADEADO 2

Fonte: Elaborado pelo autor (2022)

- Exemplo 5 – Maior 3 numeros – SE ENTÃO:

Dados três números, exibir o de maior valor

Segue a execução do exemplo:

```
Digite 3 números:  
7  
45  
2  
Maior número: 45
```

Quadro 8 – Execução “Maior 3 números” – SE ENTÃO
Fonte: Elaborado pelo autor (2022)

Seque o fluxograma:

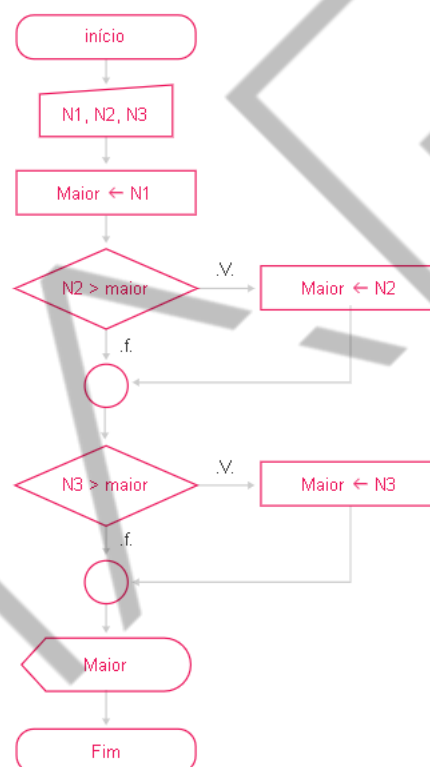


Figura 11 - Fluxograma “Maior 3 números” – SE ENTÃO
Fonte: Elaborada pelo autor (2022)

Momento Pseudocódigo

```
Programa exemplo4  
Var  
    n1, n2, n3, maior: inteiro  
Início  
    Escreva "Digite 3 números: "  
    Leia n1  
    Leia n2  
    Leia n3  
    maior = n1
```

```
Se n2 > maior então
    maior = n2
Fim_se
Se n3 > maior então
    maior = n3
Fim_se
Escreva "Maior número: ", maior
Fim
```

Código-fonte 34 – Pseudocódigo “Maior 3 números” – SE ENTÃO

Fonte: Elaborado pelo autor (2022)

Momento Python

```
print ("Digite 3 números: ")
n1 = int(input())
n2 = int(input())
n3 = int(input())
maior = n1
if n2 > maior:
    maior = n2
if n3 > maior:
    maior = n3
print ("Maior número: ", maior)
```

Código-fonte 35 – Python “Maior 3 números” – SE ENTÃO

Fonte: Elaborado pelo autor (2022)

Momento Java

```
import java.util.Scanner;
public class First
{
    public static void main(String[] args)
    {
        Scanner entrada = new Scanner(System.in);

        int n1, n2, n3, maior;
        System.out.println("Digite 3 números:");
        n1 = entrada.nextInt();
        n2 = entrada.nextInt();
        n3 = entrada.nextInt();

        maior = n1;
```

```
    if (n2 > maior)
        maior = n2;
    if (n3 > maior)
        maior = n3;

    System.out.println("Maior número: maior");
}
```

Código-fonte 36 – Java “Maior 3 números” – SE ENTÃO
Fonte: Elaborado pelo autor (2022)

Neste capítulo vimos os primeiros comandos estruturados, especificamente os de desvio de fluxo. A versatilidade do comando SE (if) nos permite adaptar a resolução do problema a uma das variedades dos comandos de decisão vistos:

- SE ENTÃO (if);
- SE ENTÃO SENÃO (if else) ou
- SE ENCADEADO (if elif else).

Muitos dos problemas podem ser resolvidos com mais de uma variação destes, mas isso é uma questão de gosto.

Com o tempo, praticando, adquirimos habilidades e percebemos que dependendo da situação sempre há uma estrutura melhor para utilizar.

Vimos também a estrutura de seleção ESCOLHA. Ela existe para otimizar o SE ENCADEADO quando a comparação é uma equivalência (igualdade).

Lembrando que esta estrutura não existe no Python, mas os problemas que necessitariam dela podem ser bem resolvidos com IF ELIF ELSE.

Saibam que o aprendizado na área de programação se dá praticando. Pratiquem!

GLOSSÁRIO

Switch	Estrutura Escolha no Java.
Comando estruturado	Aquele comando que ocupa mais de uma linha e possui um fim próprio.
Decisão	Termo utilizado em programação para direcionar o fluxo do algoritmo.
Condição	Expressão que envolve quaisquer tipos de dados, mas que retorna um valor lógico
Bloco	Conjunto de comandos que atendem um objetivo em comum