

Curso Técnico em Desenvolvimento de Sistemas Online

LÓGICA E TÉCNICAS DE PROGRAMAÇÃO EM JAVA

GEEaD - Grupo de Estudo de Educação a Distância

Centro de Educação Tecnológica Paula Souza

Expediente

GEEaD – CETEC
GOVERNO DO ESTADO DE SÃO PAULO
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

Autores:

*Eliana Cristina Nogueira Barion
Marcelo Fernando Iguchi
Paulo Henrique Mendes Carvalho
Rute Akie Utida*

Revisão Técnica:

Kelly Dall Pozzo

Revisão Gramatical:

Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação:

Flávio Biazim

AGENDA 7

ESTRUTURAS DE DECISÃO II





MERGULHANDO NO TEMA...

Já sabemos que a lógica de programação possui mecanismos que nos permitem tomar decisões dentro de um algoritmo. Sabemos também que esses mecanismos são denominados “Estruturas de Decisão”. A novidade é que essas estruturas não se restringem a apenas o “**se...senão...fim_se**”.

E se tivéssemos uma decisão a ser tomada entre **dez opções**? Será que o “**se...senão...fim_se**” seria a solução mais apropriada para essa situação? Será que existe alguma outra estrutura mais adequada para esse tipo de ocorrência?

Existe sim! Essa estrutura é chamada de **Seleciona...caso...senão...fim_seleciona**.



A função principal dessa estrutura é **facilitar a escrita do algoritmo** quando se tem muitos caminhos a serem seguidos a partir de uma decisão. Assim como a estrutura “**se...senão...fim_se**”, é necessário saber quando e como utilizar o “**seleciona...caso...senão...fim_seleciona**”.

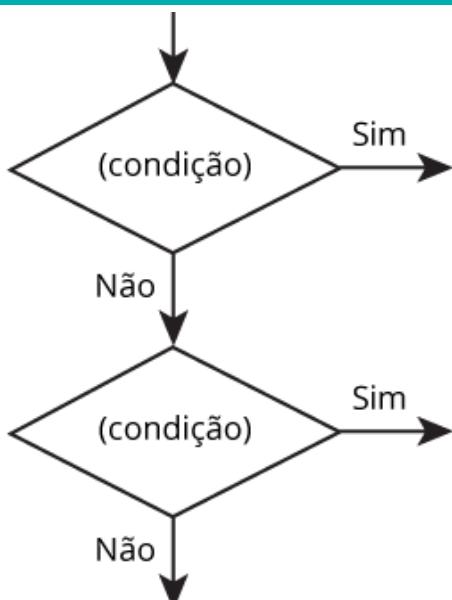
A estrutura “Escolha - Caso” do pseudocódigo corresponde à estrutura **switch- case** do Java. Elas permitem que a seleção correta seja feita a partir da comparação entre o valor do conteúdo da variável e uma lista definida durante a programação. Ao encontrar a correspondência correta, os respectivos comandos são executados e as demais opções ignoradas.

PSEUDOCÓDIGO

```

Escolha (variável){
    Caso condição 01:
        comando(s)
    pare
    Caso condição 02:
        comando(s)
    pare
    Caso condição 03:
        comando(s)
    pare
    ...
    Caso contrario:
        {comando(s)}
}
  
```

FLUXOGRAMA



JAVA

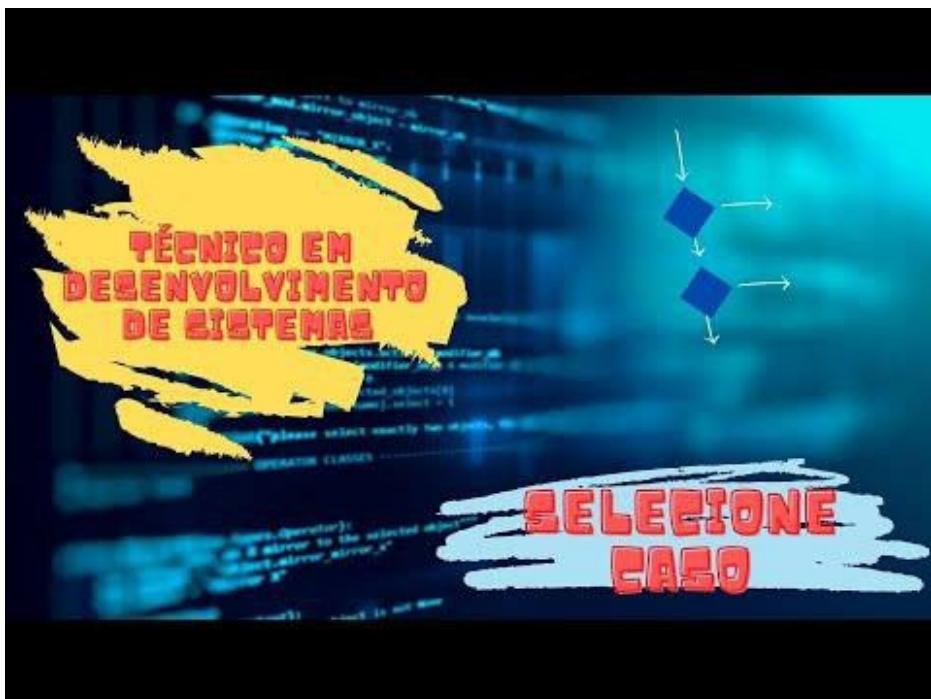
```

switch (variável) {
    case condição 01:
        comando(s);
        break;
    case condição 02:
        comando(s);
        break;
    case condição 03:
        comando(s);
        break;
    ...
    default:
        comando(s);
}
  
```

Se recapitularmos o último exemplo de Estrutura de Decisão Aninhada do tópico anterior, no qual o programa exibia o dia da semana de acordo com o número digitado pelo usuário, notamos que o código do programa fica relativamente **confuso** devido à grande quantidade de comandos de seleção (se). A tendência dessa confusão é aumentar conforme o número de comandos de decisão aninhados for crescendo. Isso, é claro, levando em conta que estamos realizando a comparação sempre com a mesma variável.

Para essas situações podemos utilizar a estrutura “selecione caso...senão...fim_seleccione” do pseudocódigo ou a estrutura switch-case do Java.

Antes de continuar a leitura, assista a aula do prof. Sandro Valérius:



Tanto as sintaxes em pseudocódigo quanto em Java são semelhantes como temos a seguir:

PORUTGOL STUDIO

```

programa
{
    funcao inicio()
    {
        inteiro dia

        escreva("Digite um Número de 1 a
7: ")
        leia(dia)

        limpa()

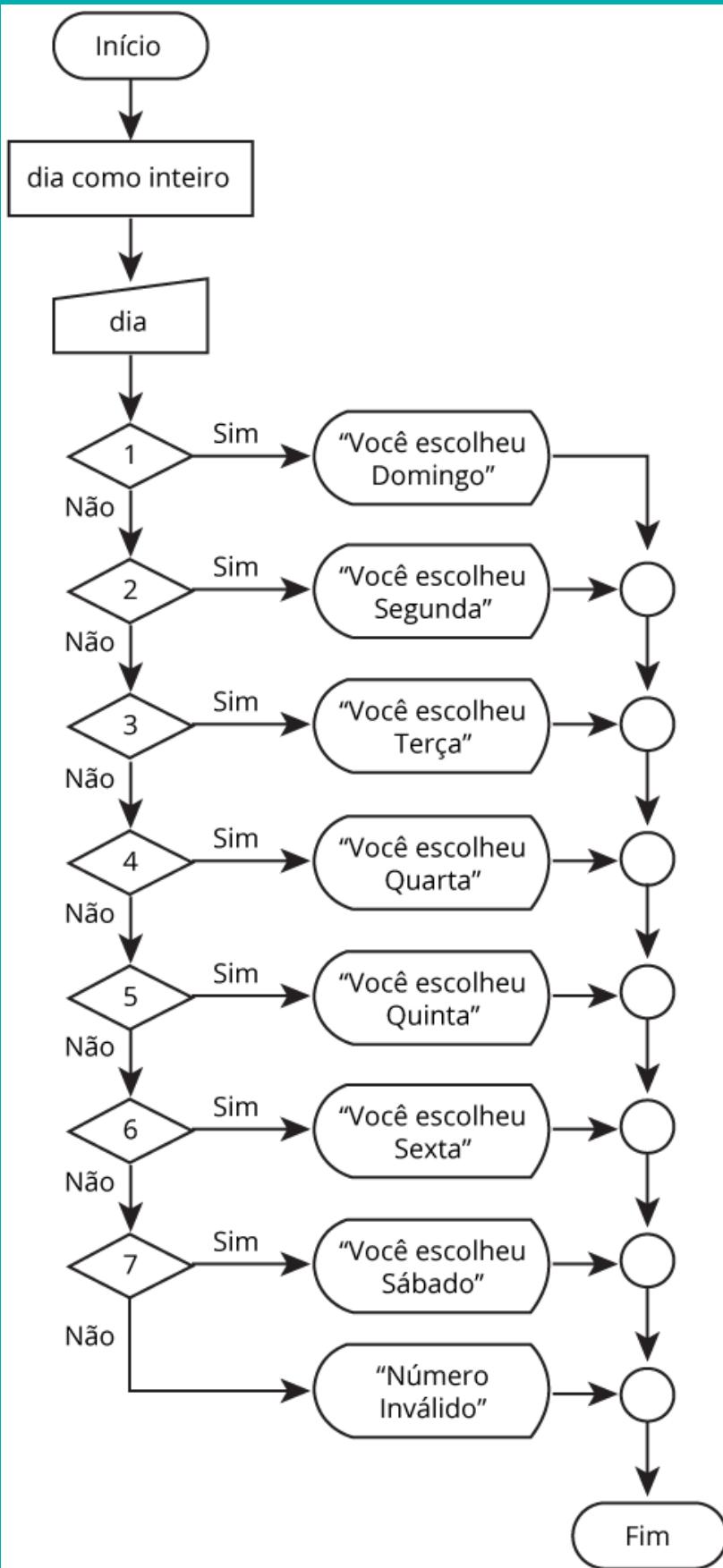
        escolha (dia)
        {
            caso 1:
                escreva ("Você Escolheu
Domingo")
                pare
            caso 2:
                escreva ("Você Escolheu
Segunda")
                pare
            caso 3:
                escreva ("Você Escolheu
Terça")
                pare
            caso 4:
                escreva ("Você Escolheu
Quarta")
                pare
            caso 5:
                escreva ("Você Escolheu
Quinta")
                pare
            caso 6:
                escreva ("Você Escolheu
Sexta")
                pare
            caso 7:
                escreva ("Você Escolheu
Sábado")
                pare

            caso contrario:
                escreva ("Número Inválido")
        }

        escreva("\n")
    }
}

```

FLUXOGRAMA

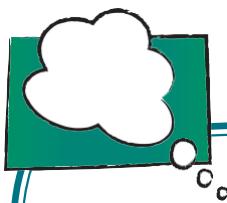


Vamos agora ao código em Java:

```

switchCase.java ✘
1 import javax.swing.JOptionPane;
2
3 public class switchCase {
4
5     public static void main(String[] args) {
6         //declaração de variáveis
7         int dia; // variável para armazenamento da semana
8
9         //entrada de dados com conversão de tipos juntas
10        dia = Integer.parseInt(JOptionPane.showInputDialog("Entre com um número de 1 a 7"));
11
12        switch (dia) {
13
14            case 1:
15                JOptionPane.showMessageDialog(null, "Você escolheu Domingo");
16                break;
17            case 2:
18                JOptionPane.showMessageDialog(null, "Você escolheu Segunda");
19                break;
20            case 3:
21                JOptionPane.showMessageDialog(null, "Você escolheu Terça");
22                break;
23            case 4:
24                JOptionPane.showMessageDialog(null, "Você escolheu Quarta");
25                break;
26            case 5:
27                JOptionPane.showMessageDialog(null, "Você escolheu Quinta");
28                break;
29            case 6:
30                JOptionPane.showMessageDialog(null, "Você escolheu Sexta");
31                break;
32            case 7:
33                JOptionPane.showMessageDialog(null, "Você escolheu Sábado");
34                break;
35
36            default:
37                JOptionPane.showMessageDialog(null, "Número Inválido");
38                break;
39        }
40    } // fim do método main
41
42 } // fim da classe
43

```



Você acha que já viu o programa em algum lugar? Não é impressão, não! É exatamente o mesmo programa de exemplo de Estrutura de decisão aninhada da agenda anterior. Com essa nova estrutura que foi apresentada, o código foi escrito de maneira muito mais elegante. O fluxograma permaneceu o mesmo sem alteração nenhuma.

Analogamente ao pseudocódigo, em Java, após fazermos a entrada do dado pelo usuário, o comando switch (dia) irá comparar o valor armazenado na variável dia com um valor de 1 a 7 em cada comando case.

Exemplo:

Se o usuário entrar com o valor 3, teremos dia = 3 e ao executarmos o comando case 3: (dia=3), como o resultado da comparação será verdadeiro, ele executará o comando JOptionPane.

```
showMessageDialog(null, "Você escolheu Terça");
```

e o comando break; e por fim finalizará o programa.

O comando break tem a finalidade de parar a execução do comando switch, uma vez que já foi executada a ação necessária (comparação verdadeira) e não há a necessidade de se continuar com o comando switch.

Caso nenhuma comparação resulte em uma resposta verdadeira, o comando default: é executado gerando uma mensagem de “número inválido” para o usuário.



VOCÊ NO COMANDO

Exercícios sobre estrutura de seleção:

Os exercícios de 1 a 4 devem ser desenvolvidos elaborando o pseudocódigo, o fluxograma e a linguagem Java:

1. Vanessa precisa desenvolver um programa em que ao se digitar o código de um produto cadastrado da papelaria Lápis Colorido este retorne o nome do produto. Se o código do produto não estiver cadastrado, o programa deve exibir a mensagem produto não cadastrado. A tabela a seguir descreve os códigos e os produtos cadastrados:

Como Vanessa resolveria essa questão?

2. Eliberto deseja fazer um menu de seleção de um programa utilizando o comando de seleção múltipla. O menu consiste nas opções de:
 1. Cadastrar usuário
 2. Alterar dados
 3. Excluir Usuário

Como Eliberto faria esse programa? Elabore uma mensagem diferente para ser exibida para o usuário ao selecionar cada um dos itens dos menus.

3. Além do menu apresentado na questão anterior, Eliberto precisa fazer alguns submenus com a seguinte estrutura:

1. Cadastrar usuário
2. Alterar dados
 1. Alterar nome
 2. Alterar Endereço
 3. Alterar telefone
 4. Alterar RG.
3. Excluir usuário

Altere o programa do exercício anterior para incluir este submenu e exiba uma mensagem correspondente de acordo com o menu escolhido pelo usuário.

4. Crie um programa para uma calculadora com cinco operações básicas (soma, subtração, multiplicação, divisão e resto da divisão) em que dois números reais devem ser inseridos e, posteriormente, em um menu, o usuário escolha qual operação deve ser realizada. Após a execução da operação selecionada o resultado deve ser apresentado na tela do computador.

Exercícios para serem resolvidos somente em linguagem Java (5 e 6):

5. Elabore um programa em que o usuário deva inserir um número do tipo double e a seguir o programa exiba este número.

6. Denis decidiu, por curiosidade, elaborar um programa que realiza a conversão de milhas para quilômetros. Sabe-se que uma milha tem 1,609km. Como Denis resolveria este problema?

Agora confira se você acertou as respostas:

Comando de Seleção de Múltipla

1.

Pseudocódigo	Fluxograma
<p>Programa ex1</p> <p>Declare</p> <p> opcao como inteiro</p> <p>Início</p> <p> Escreva (“Entre com o código do produto”)</p> <p> Leia (opcao)</p> <p> Selecione Caso (opcao)</p> <p> Caso condição 100</p> <p> Escreva (“Lápis preto nº2”)</p> <p> Caso condição 150</p> <p> Escreva (“borracha branca”)</p> <p> Caso condição 200</p> <p> Escreva (“caneta azul”)</p> <p> Caso condição 230</p> <p> Escreva (“caneta vermelha”)</p> <p> Caso condição 256</p> <p> Escreva (“giz de cera 12 unid.”)</p> <p> Caso condição 300</p> <p> Escreva (“cartolina branca”)</p> <p> Caso condição 310</p> <p> Escreva (“resma de sulfite A4”)</p> <p> Caso condição 400</p> <p> Escreva (“estojo escolar verde”)</p> <p> Caso condição 470</p> <p> Escreva (“caderno universitário 100fls.”)</p> <p> Caso condição 500</p> <p> Escreva (“caderno brochura 50fls.”)</p> <p> Senão</p> <p> Escreva (“Produto não cadastrado”)</p> <p> Fim-selezione</p> <p>Fim.</p>	<pre> graph TD Inicio([Início]) --> Inteiro[opcao como Inteiro] Inteiro --> Entrada{“Entre com o código do Produto”} Entrada --> opcao[/opcao/] opcao --> Cond100{100} Cond100 -- Sim --> Lapis["Lapis preto nº2"] Cond100 -- Não --> Cond150{150} Cond150 -- Sim --> Borracha["borracha branca"] Cond150 -- Não --> Cond200{200} Cond200 -- Sim --> CanetaAzul["caneta azul"] Cond200 -- Não --> Cond230{230} Cond230 -- Sim --> CanetaVermelha["caneta vermelha"] Cond230 -- Não --> Cond256{256} Cond256 -- Sim --> Giz["giz de cera 12und."] Cond256 -- Não --> Cond300{300} Cond300 -- Sim --> Cartolina["cartolina branca"] Cond300 -- Não --> Cond310{310} Cond310 -- Sim --> Resma["resma de sulfite A4"] Cond310 -- Não --> Cond400{400} Cond400 -- Sim --> Estojo["estojo escolar verde"] Cond400 -- Não --> Cond470{470} Cond470 -- Sim --> Caderno100["caderno universitário 100fls."] Cond470 -- Não --> Cond500{500} Cond500 -- Sim --> Caderno50["caderno brochura 50fls."] Cond500 -- Não --> ProdutoNaoCadastrado["Produto não cadastrado"] ProdutoNaoCadastrado --> Fim([Fim]) </pre>

Em Java [KDP2]

```

1 import javax.swing.JOptionPane;
2
3 public class Switch_ex1 {
4
5     public static void main(String[] args) {
6         //declaração de variáveis
7         int opcao;
8
9         //Entrada e conversão de dados
10        opcao = Integer.parseInt(JOptionPane.showInputDialog("Entre com o código do produto: "));
11
12        //Processamento dos dados
13        switch (opcao) {
14            case 100:
15                JOptionPane.showMessageDialog(null, "O produto selecionado é Lápis preto n.2");
16                break;
17
18            case 150:
19                JOptionPane.showMessageDialog(null, "O produto selecionado é Borracha branca");
20                break;
21
22            case 200:
23                JOptionPane.showMessageDialog(null, "O produto selecionado é Caneta azul");
24                break;
25
26            case 230:
27                JOptionPane.showMessageDialog(null, "O produto selecionado é Caneta vermelha");
28                break;
29
30            case 256:
31                JOptionPane.showMessageDialog(null, "O produto selecionado é Giz de cera 12 unid.");
32                break;
33
34            case 300:
35                JOptionPane.showMessageDialog(null, "O produto selecionado é Cartolina branca");
36                break;
37
38            case 310:
39                JOptionPane.showMessageDialog(null, "O produto selecionado é Resma de sulfite A4");
40                break;
41
42            case 400:
43                JOptionPane.showMessageDialog(null, "O produto selecionado é estojo escolar verde");
44                break;
45
46            case 470:
47                JOptionPane.showMessageDialog(null, "O produto selecionado é caderno univrsitário 100 fls.");
48                break;
49
50            case 500:
51                JOptionPane.showMessageDialog(null, "O produto selecionado é caderno brochura 50fls.");
52                break;
53
54            default:
55                JOptionPane.showMessageDialog(null, "Produto não cadastrado!");
56                break;
57        }
58    }
59
60}
61

```

2.

Pseudocódigo	Fluxograma
<pre> Programa ex2 Declare opcao como inteiro Início Escreva ("Entre com a opção desejada:") Escreva ("1 – Cadastrar usuário") Escreva ("2 – Alterar dados") Escreva ("3 – Excluir usuário") Leia (opcao) Selecione Caso (opcao) Caso condição 1 Escreva ("Cadastrando Usuário") Caso condição 2 Escreva ("Alterando dados") Caso condição 3 Escreva ("Exclusão de usuários") Senão Escreva ("Opção Inválida") Fim-selecione Fim. </pre>	<pre> graph TD Inicio([Início]) --> Leitura[/opcao como Inteiro/] Leitura --> Entrada("Entre com a opção desejada") Entrada --> Opcao[/opcao/] Opcao --> Cond1{1} Cond1 -- Não --> Cond2{2} Cond2 -- Não --> Cond3{3} Cond3 -- Não --> Invalida("opção Inválida") Cond1 -- Sim --> Cadastro("Cadastrando Usuário") Cadastro --> Fim([Fim]) Cond2 -- Sim --> Alteracao("Alterando dados") Alteracao --> Fim Cond3 -- Sim --> Exclusao("Exclusão de Usuários") Exclusao --> Fim </pre>

Em Java

```
J *Switch_ex2.java ✘
1 import javax.swing.JOptionPane;
2
3 public class Switch_ex2 {
4
5     public static void main(String[] args) {
6         // Exercício 2
7
8         //declaração de variáveis
9         int opcao;
10
11         //entrada de dados
12         opcao = Integer.parseInt(JOptionPane.showInputDialog("Entre com a opção desejada:" +
13             "\n1 - Cadastrar usuário" +
14             "\n2 - Alterar dados" +
15             "\n3 - Excluir usuário"));
16
17
18         //processamento e saída
19         switch (opcao) {
20             case 1:
21                 JOptionPane.showMessageDialog(null, "Cadastrando usuário");
22                 break;
23
24             case 2:
25                 JOptionPane.showMessageDialog(null, "Alterando dados");
26                 break;
27
28             case 3:
29                 JOptionPane.showMessageDialog(null, "Exclusão de usuários");
30                 break;
31
32             default:
33                 JOptionPane.showMessageDialog(null, "Opção Inválida");
34                 break;
35         }//fim do switch-case
36
37     } // fim do método main
38
39 } // fim da classe
40 }
```

3.

Pseudocódigo	Fluxograma
<p>Programa ex3</p> <p>Declare</p> <p> opcao como inteiro</p> <p>Início</p> <p> Escreva ("Entre com a opção desejada:")</p> <p> Escreva ("1 – Cadastrar usuário")</p> <p> Escreva ("2 – Alterar dados")</p> <p> Escreva ("3 – Excluir usuário")</p> <p> Leia (opcao)</p> <p> Seleciona Caso (opcao)</p> <p> Caso condição 1</p> <p> Escreva ("Cadastrando Usuário")</p> <p> Caso condição 2</p> <p> Escreva ("Menu de alteração de dados")</p> <p> Escreva ("Seleciona a opção desejada:")</p> <p> Escreva ("1 – Alterar nome")</p> <p> Escreva ("2 – Alterar Endereço")</p> <p> Escreva ("3 – Alterar Telefone")</p> <p> Escreva ("4 – Alterar RG")</p> <p> Leia (submenu)</p> <p> Seleciona Caso (submenu)</p> <p> Caso condição 1</p> <p> Escreva ("Alterar nome selecionado")</p> <p> Caso condição 2</p> <p> Escreva ("Alterar Endereço selecionado")</p> <p> Caso condição 3</p> <p> Escreva ("Alterar telefone selecionado")</p> <p> Caso condição 4</p> <p> Escreva ("Alterar RG selecionado")</p> <p> Senão 1</p> <p> Escreva ("Opção Inválida")</p> <p> Fim-seleciona</p> <p> Caso condição 3</p> <p> Escreva ("Exclusão de usuários")</p> <p> Senão</p> <p> Escreva ("Opção Inválida")</p> <p> Fim-seleciona</p> <p>Fim.</p>	<pre> graph TD Inicio([Início]) --> Submenu[Submenu] Submenu --> OpcaoInteiro[opcao, submenu como Inteiro] OpcaoInteiro --> Entrada{Entre com a opção desejada} Entrada --> Opcao[opcao] Opcao --> Cond1{1} Cond1 -- Sim --> Cadastrar[Cadastrando Usuário] Cond1 -- Não --> Cond2{2} Cond2 -- Sim --> AlterarDados[Seleciona a opção desejada] Cond2 -- Não --> Fim([Fim]) AlterarDados --> Submenu Submenu --> Cond3{1} Cond3 -- Sim --> AlterarNome[Alterar nome selecionado] Cond3 -- Não --> Cond4{2} Cond4 -- Sim --> AlterarEndereco[Alterar endereço selecionado] Cond4 -- Não --> Cond5{3} Cond5 -- Sim --> AlterarTelefone[Alterar telefone selecionado] Cond5 -- Não --> Cond6{4} Cond6 -- Sim --> AlterarRG[Alterar RG selecionado] Cond6 -- Não --> Invalida[Opção Inválida] AlterarNome --> Fim AlterarEndereco --> Fim AlterarTelefone --> Fim AlterarRG --> Fim Invalida --> Fim Cadastrar --> Fim Exclusao[Exclusão de Usuários] --> Fim </pre>

Em Java

```

J Switch_ex3.java ✘
1 import javax.swing.JOptionPane;
2
3 public class Switch_ex3 {
4
5     public static void main(String[] args) {
6         // Exercício 3
7
8         // declaração de variáveis
9         int opcao, submenu;
10
11        // entrada de dados
12        opcao = Integer.parseInt(JOptionPane.showInputDialog("Entre com a opção desejada:" +
13                                "\n1 - Cadastrar usuário" +
14                                "\n2 - Alterar dados" +
15                                "\n3 - Excluir usuário"));
16
17        // processamento e saída
18        switch (opcao) {
19            case 1:
20                JOptionPane.showMessageDialog(null, "Cadastrando usuário");
21                break;
22
23            case 2:
24                submenu = Integer.parseInt(JOptionPane.showInputDialog(
25                                "Menu de alteração de dados" +
26                                "\nSelecione a opção desejada:" +
27                                "\n1 - Alterar nome" +
28                                "\n2 - Alterar endereço" +
29                                "\n3 - Alterar Telefone" +
30                                "\n4 - Alterar RG"));
31                // início do 2ºswitch
32                switch (submenu) {
33                    case 1:
34                        JOptionPane.showMessageDialog(null, "Alterar nome selecionado");
35                        break;
36                    case 2:
37                        JOptionPane.showMessageDialog(null, "Alterar endereço selecionado");
38                        break;
39                    case 3:
40                        JOptionPane.showMessageDialog(null, "Alterar telefone selecionado");
41                        break;
42                    case 4:
43                        JOptionPane.showMessageDialog(null, "Alterar RG selecionado");
44                        break;
45                    default:
46                        JOptionPane.showMessageDialog(null, "Opção inválida");
47                        break;
48                } // fim do 2º switch
49                break;
50

```

4.

Pseudocódigo	Fluxograma
<p>Programa ex4</p> <p>Declare</p> <p> num1, num2 e resultado como real</p> <p> operacao como inteiro</p> <p>Início</p> <p> Escreva ("Entre com o primeiro número")</p> <p> Leia (num1)</p> <p> Escreva ("Entre com o segundo número")</p> <p> Leia (num2)</p> <p> Escreva ("Digite a operação desejada: ")</p> <p> Escreva ("1 –soma")</p> <p> Escreva ("2 – subtração")</p> <p> Escreva ("3 – multiplicação")</p> <p> Escreva ("4 – divisão")</p> <p> Escreva ("5 – resto da divisão")</p> <p> Leia (operacao)</p> <p> Seleciona Caso (operacao)</p> <p> Caso condição 1</p> <p> Resultado <- num1 + num2</p> <p> Caso condição 2</p> <p> Resultado <- num1 - num2</p> <p> Caso condição 3</p> <p> Resultado <- num1 * num2</p> <p> Caso condição 4</p> <p> Resultado <- num1 / num2</p> <p> Caso condição 5</p> <p> Resultado <- num1 mod num2</p> <p> Senão</p> <p> Resultado <- 0</p> <p> Fim-seleciona</p> <p> Escreva ("O resultado da operação é ", resultado)</p> <p> Fim.</p>	<pre> graph TD Inicio([Início]) --> Declara[/num1, num2, resposta como real
Operacao como inteiro/] Declara --> Entrada1("Entre com o
primeiro
número") Entrada1 --> num1[/num1/] Entrada2("Entre com o
segundo
número") --> num2[/num2/] Entrada3("Entre com a
operação
desejada") --> operacao[/operacao/] operacao --> Cond1{1} Cond1 -- Sim --> Soma[/Resultado <- num1 + num2/] Soma --> Cond1 Cond1 -- Não --> Cond2{2} Cond2 -- Sim --> Subtracao[/Resultado <- num1 - num2/] Subtracao --> Cond2 Cond2 -- Não --> Cond3{3} Cond3 -- Sim --> Multiplicacao[/Resultado <- num1 * num2/] Multiplicacao --> Cond3 Cond3 -- Não --> Cond4{4} Cond4 -- Sim --> Divisao[/Resultado <- num1 / num2/] Divisao --> Cond4 Cond4 -- Não --> Cond5{5} Cond5 -- Sim --> Modulo[/Resultado <- num1 mod num2/] Modulo --> Cond5 Cond5 -- Não --> ResultadoZero[/Resultado <- 0/] ResultadoZero --> Cond5 Soma --> Fim([Fim]) Subtracao --> Fim Multiplicacao --> Fim Divisao --> Fim Modulo --> Fim ResultadoZero --> Fim </pre>

Em Java

```

1 *Switch_ex4.java ✘
2
3 import javax.swing.JOptionPane;
4
5 public class Switch_ex4 {
6     public static void main(String[] args) {
7         // Exercício 4
8
9         //declaração de variáveis
10        double num1, num2, resultado;
11        int operacao;
12
13        //entrada de dados
14        num1 = Double.parseDouble(JOptionPane.showInputDialog("Entre com o primeiro número"));
15        num2 = Double.parseDouble(JOptionPane.showInputDialog("Entre com o segundo número"));
16
17        operacao = Integer.parseInt(JOptionPane.showInputDialog(
18            "Digite a operação desejada: " +
19            "\n1 - Soma" +
20            "\n2 - Subtração" +
21            "\n3 - Multiplicação" +
22            "\n4 - Divisão" +
23            "\n5 - Resto da divisão"));
24
25        //processamento
26        switch (operacao) {
27            case 1: //soma
28                resultado = num1 + num2;
29                break;
30
31            case 2: //subtração
32                resultado = num1 - num2;
33                break;
34
35            case 3: //multiplicação
36                resultado = num1 * num2;
37                break;
38
39            case 4: //divisão
40                resultado = num1 / num2;
41                break;
42
43            case 5: //resto da divisão
44                resultado = num1 % num2;
45                break;
46
47            default:
48                resultado = 0;
49                break;
50        }//fim do switch-case
51
52        //saída de dados
53        JOptionPane.showMessageDialog(null, "O resultado da operação é " + resultado);
54    } // fim do método main
55
56 } // fim da classe

```

Importante deixar claro que a estrutura switch -case a variável de verificação pode ser somente dos tipos int ou char.

Observação: resolução dos exercícios somente em Java.

5.

```
J TryCatch_Ex1.java ✘
1 import javax.swing.JOptionPane;
2
3 public class TryCatch_Ex1 {
4
5     public static void main(String[] args) {
6         // Exercício 1
7
8         // declaração de variáveis
9         double num;
10
11        // entrada de dados
12        num = Double.parseDouble(JOptionPane.showInputDialog("Entre com um número"));
13        // saída de dados
14        JOptionPane.showMessageDialog(null, "O número digitado é " + num);
15
16    }
17
18 }
```

6.

```
J *TryCatch_Ex3.java ✘
1 import javax.swing.JOptionPane;
2
3 public class TryCatch_Ex3 {
4
5     public static void main(String[] args) {
6         // Exercício 3
7
8         // declaração de variáveis
9         double km, milha;
10
11        // entrada de dados
12        milha = Double.parseDouble(JOptionPane.showInputDialog("Entre com um número" +
13            " em milhas"));
14
15        // processamento
16        km = milha * 1.609;
17
18        // saída de dados
19        JOptionPane.showMessageDialog(null,
20            "O valor de " + milha + " milhas convertido para quilômetros é " + km +
21            " quilômetros");
22
23    }
24
25 }
```