

ESTRUTURAS DE DECISÃO

Este texto-base aborda os seguintes temas:

Estruturas de Decisão	Estruturas de Decisão if-else	Estrutura de Decisão Aninhada
Como realizar comparações com String no Java	Estrutura de Decisão caso...senão... fim seleccione	Revisando as Estruturas de Decisão



Estruturas de decisão



Até agora, trabalhamos somente com programas que efetuam tarefas simples, como a realização de entrada e saída de dados e pequenos cálculos matemáticos. Você deve ter percebido que os algoritmos criados até aqui não possuem **poder de decisão**.

Em outras palavras, eles sempre executam as **mesmas tarefas**, independentemente dos resultados obtidos. Mas, na vida real precisamos tomar decisões que muitas vezes são difíceis e que podem alterar o rumo de nossas vidas. Quando falamos de programas, ocorre a mesma coisa.

Em programação, chamamos essas decisões de *Estrutura de Decisão*. No Java, chamamos de comando *If*. Estruturas de Decisão *if* (Se... Fim.)

[← Voltar](#)

As Estruturas de Decisão ou Testes Condicionais nos permitem executar um conjunto diferente de comandos dependendo do resultado de um teste utilizando **operadores relacionais**. Este resultado pode ser verdadeiro ou falso conforme indicado na tabela a seguir:

PSEUDOCÓDIGO	FLUXOGRAMA	JAVA
<div>SE (condição) Então {comando(s)} Fim-Se</div>		<div>if (condição){ {comando(s)}; }</div>

Exemplificando...



Imagine que em um determinado trecho de um programa precisamos tomar uma decisão para saber se uma pessoa é maior de idade. O programa codificado é apresentado a seguir:

PSEUDOCÓDIGO	FLUXOGRAMA	JAVA
--------------	------------	------

PSEUDOCÓDIGO	FLUXOGRAMA	JAVA
<div>SE (idade >=18) Então Escreva (“Maior de idade”) Fim-Se</div>		<div>if (idade >=18){ JOptionPane.showMessageDialog(“Maior de idade”); }</div>

O comando condicional **SE** testa a condição **idade >=18**. Ou seja, se a idade for maior ou igual a 18 anos **ENTÃO** a condição é verdadeira (segue-se o caminho “sim”). Desse modo, será executado o comando **Escreva (“Maior de idade”)**. Caso contrário (se a idade for menor que 18 anos), nada será feito (**Fim-Se**). Observando o fluxograma, você compreenderá com mais facilidade o que ocorre.

Você deve ter reparado, com base na sintaxe apresentada na tabela acima, que na programação Java a palavra **SE** é substituída pelo **if** e que a palavra **Então** é substituída por **{**. O comando Escreva é substituído por **JOptionPane.showMessageDialog** e, finalmente, o **Fim-Se** é substituído por **}**.



Agora, veja a codificação em Java de um programa completo:

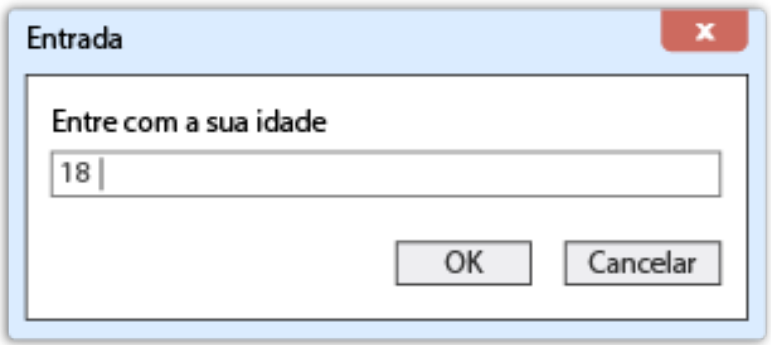
```
1  import javax.swing.JOptionPane;
2
3  public class IfSimples {
4
5      public static void main(String[] args) {
6          //declaração de variáveis
7          int idade; // armazena a idade
8          String aux; //variável auxiliar
9
10         //entrada de dados
11         aux = JOptionPane.showInputDialog('Entre com a sua idade');
12         //conversão de tipos
13         idade = Integer.parseInt(aux);
14
15         //Decisão
16         if (idade >=18) {
17             JOptionPane.showMessageDialog(null, 'Maior de Idade');
18         } //fim do if
19     } //fim do main
20
21 }
```

Copiar

Explicando a Estrutura de Decisão presente na linha 16, o comando de decisão **if (idade >=18)** irá executar o que estiver dentro das chaves até a linha 18, **se e somente se o valor da idade for maior ou igual a 18**. Caso contrário, não executará nenhum comando adicional e o programa será encerrado.

Se, ao executarmos o programa, digitarmos a idade de 18 anos, por exemplo:

← Voltar



Teremos o seguinte retorno:



Note que se a condição for falsa (Não) nenhum comando é executado.

Estruturas de Decisão if-else

(Se... Senão... Fim-se)

Você acabou de conhecer uma Estrutura de Decisão que realiza somente uma ação distinta caso o teste condicional seja verdadeiro. Contudo, em geral também costuma ser necessário que alguma ação seja realizada **caso o teste condicional seja falso**. Para isso, temos o comando if-else:

PSEUDOCÓDIGO

FLUXOGRAMA

JAVA

PSEUDOCÓDIGO

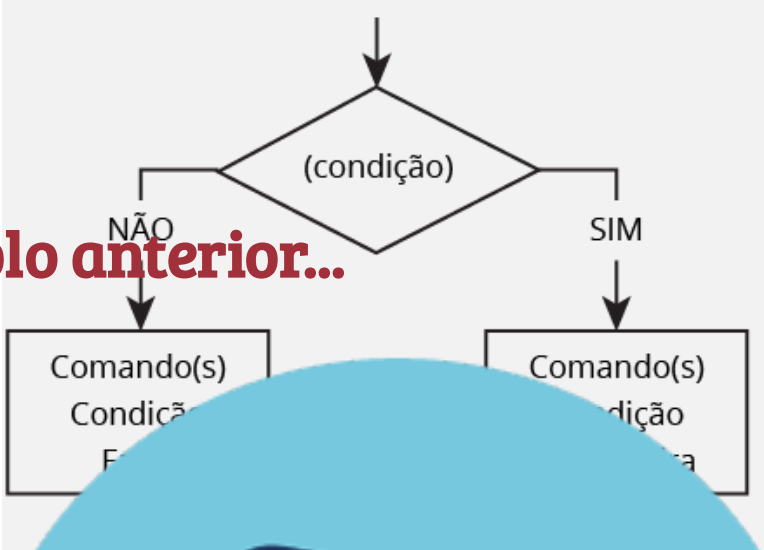
FLUXOGRAMA

JAVA

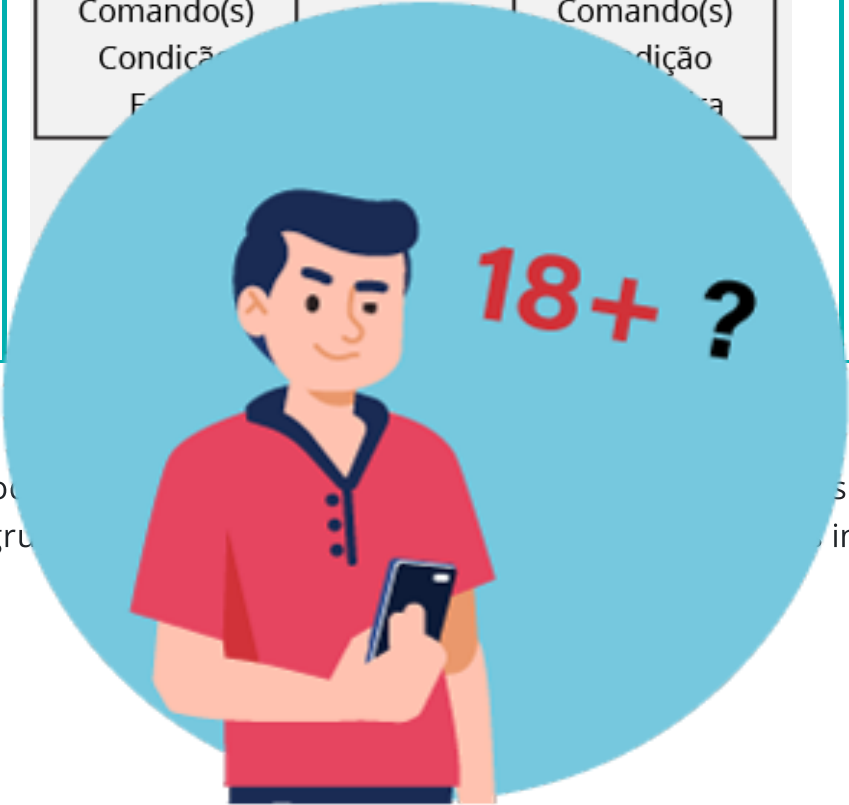
← Voltar

Retomando o exemplo anterior...

```
SE (condição) Então
    {comando(s)
    condição verdadeira}
Senão
    {Comando(s)
    condição falsa}
Fim-Se
```



```
if (condição){
    {comando(s)
    condição verdadeira};
}
else {
    {comando(s)
    condição falsa};
}
```



Observando a tabela acima, você pode perceber que, caso a condição não seja atendida), temos um comando ou grupo de comandos executados após o **Senão**.

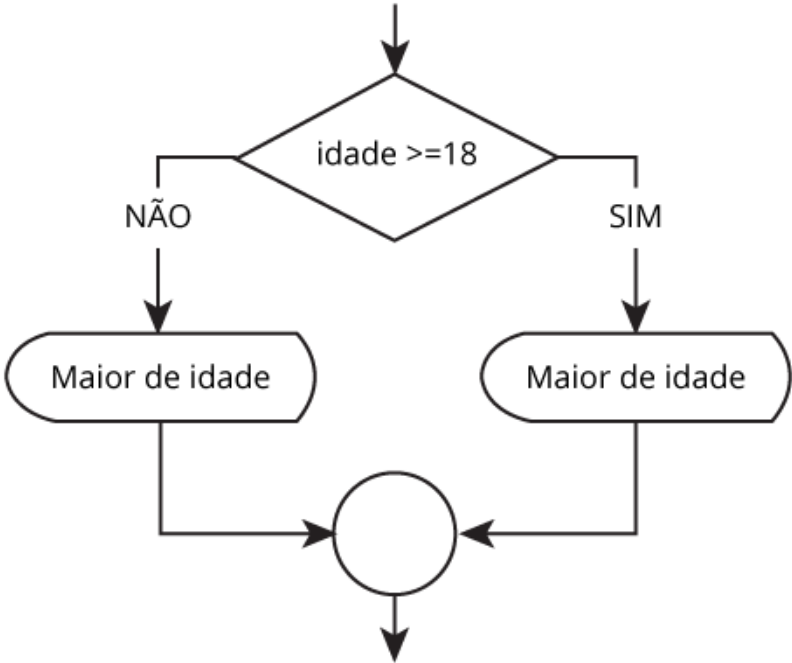
Considere o programa que analisava se uma pessoa era ou não maior de idade. Agora, porém, ele conta com comandos que serão executados se o teste condicional for falso.

PSEUDOCÓDIGO

FLUXOGRAMA

JAVA

```
SE (idade >=18)
Então
    Escreva
    ("Maior de idade")
Senão
    Escreva
    ("Menor de idade")
Fim-Se
```



```
if (idade >=18){
    JOptionPane.showMessageDialog
        (null, "Maior de
    idade");
}
else {
    JOptionPane.showMessageDialog
        (null, "Menor de
    idade");
}
```

Perceba que o exemplo é praticamente idêntico ao caso anterior, porém, se o teste condicional falhar (caso o resultado seja falso, seguindo a resposta “não” do fluxograma), executamos um comando que exibe a mensagem “menor de idade” para o usuário. Isso é feito por meio da utilização da cláusula **Senão**, no Pseudocódigo, e **else**, no Java.

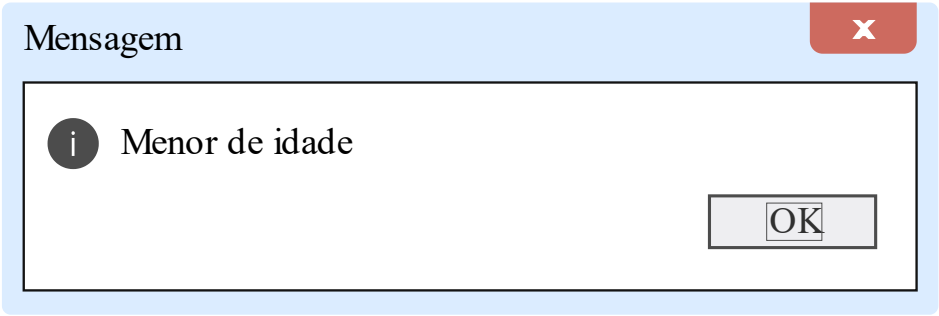
Agora, veja o programa completo codificado em Java:

← Voltar

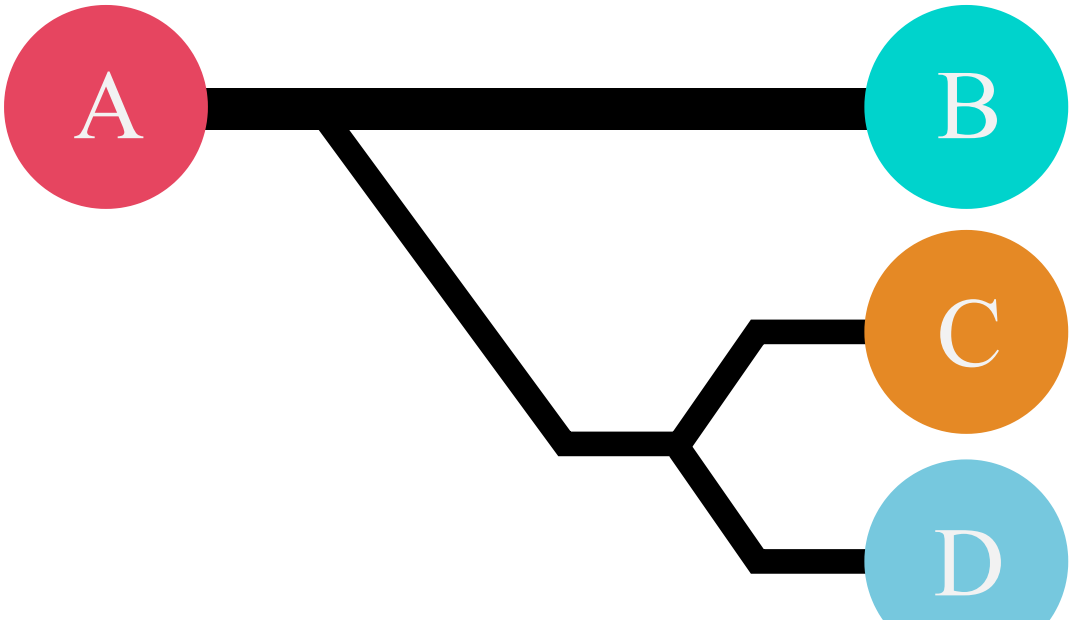
```
1  import javax.swing.JOptionPane;
2
3  public class ifComposto {
4
5      public static void main(String[] args) {
6          //declaração de variáveis
7          int idade; // armazena a idade
8          String aux; //variável auxiliar
9
10         //entrada de dados
11         aux = JOptionPane.showInputDialog("Entre com a sua idade");
12         //conversão de tipos
13         idade = Integer.parseInt(aux);
14
15         //Decisão
16         if (idade >=18) {
17             //comandos se a condição for verdadeira
18             JOptionPane.showMessageDialog(null, "Maior de Idade");
19         } else {
20             //comandos se a condição for falsa
21             JOptionPane.showMessageDialog(null, "Menor de Idade");
22         } // fim do if
23     } //fim do main
24
25 }
```

Copiar

Se digitarmos 17 na caixa de diálogo, o resultado será:



Estrutura de decisão aninhada



O que aconteceria se precisássemos de mais de duas alternativas para resolver um problema?

Talvez você tenha reparado que em uma Estrutura de Decisão podemos ter somente duas saídas: verdadeiro ou falso. Mas o que ocorre quando necessitamos de uma saída com mais de duas alternativas simultaneamente? Essa situação é bastante comum e, para isso, usamos as **Estruturas de Decisão Aninhadas**, que consistem em utilizar um comando **SE** encadeado no interior de outro.

Retomando o exemplo anterior...

Pensando no software que analisa a maioridade de um indivíduo, suponha que você também queira verificar se a idade é igual a 18 anos. Veja como ficaria a codificação do programa:

PSEUDOCÓDIGO

```
SE (idade >=18)
Então
    SE (idade = 18)
Então
    Escreva ("Igual a 18")
Senão
    Escreva ("Maior de 18")
Fim-Se
Senão
    Escreva ("Menor de idade")
Fim-Se
```

FLUXOGRAMA

JAVA

```
if (idade >=18){
    if (idade==18){
        JOptionPane.showMessageDialog
            (null, "igual a 18");
    }
    else {
        JOptionPane.showMessageDialog
            (null, "Maior de 18");
    }
}
else {
    JOptionPane.showMessageDialog
        (null, "Menor de idade");
}
```

Note que ao executar a primeira tomada de decisão **SE** ($idade \geq 18$), em caso verdadeiro, sabe-se somente que a idade é maior ou igual a 18. Para saber se a idade é igual a 18, é necessária a execução de outra Estrutura de Decisão **SE** ($idade = 18$). Em caso afirmativo, sabemos que é igual e em caso negativo, sabemos que é maior de 18. É isso que chamamos de Estrutura de Decisão Aninhada.

Agora, veja o programa completo codificado em Java:

```
1 import javax.swing.JOptionPane;
2
3 public class ifAninhado {
4
5     public static void main(String[] args) {
6         //declaração de variáveis
7         int idade; // armazena a idade
8         String aux; //variável auxiliar
9
10        //entrada de dados
11        aux = JOptionPane.showInputDialog("Entre com a sua idade");
```



```
12 //conversão de tipos
13 idade = Integer.parseInt(aux);
14
15 //Decisão
16 if (idade >=18) { // primeiro if
17     //comandos se a condição for verdadeira
18     if (idade == 18) { // segundo if
19         JOptionPane.showMessageDialog(null, "igual a 18");
20     }else {
21         JOptionPane.showMessageDialog(null, "Maior de 18");
22     }//fim do segundo if
23 } else {
24     //comandos se a condição for falsa
25     JOptionPane.showMessageDialog(null, "Menor de Idade");
26 } // fim do primeiro if
27 }//fim do main
28
29 }//fim da classe
```

[Copiar](#)

Conhecendo um novo exemplo



Agora, veja esse segundo exemplo: suponha que você precise fazer um programa em que o usuário insira um número de 1 a 7 e o programa apresente qual é o dia da semana correspondente. Você sabe que domingo é o início da semana, correspondendo ao número 1 e assim sucessivamente.

Como você poderia resolver esse programa?

1. Primeiramente, você deve pensar nas variáveis necessárias:

Como o usuário deverá inserir um número de 1 a 7, é preciso que exista uma variável que vamos chamar de **entrada**.

Qual seria o **tipo da variável** entrada?

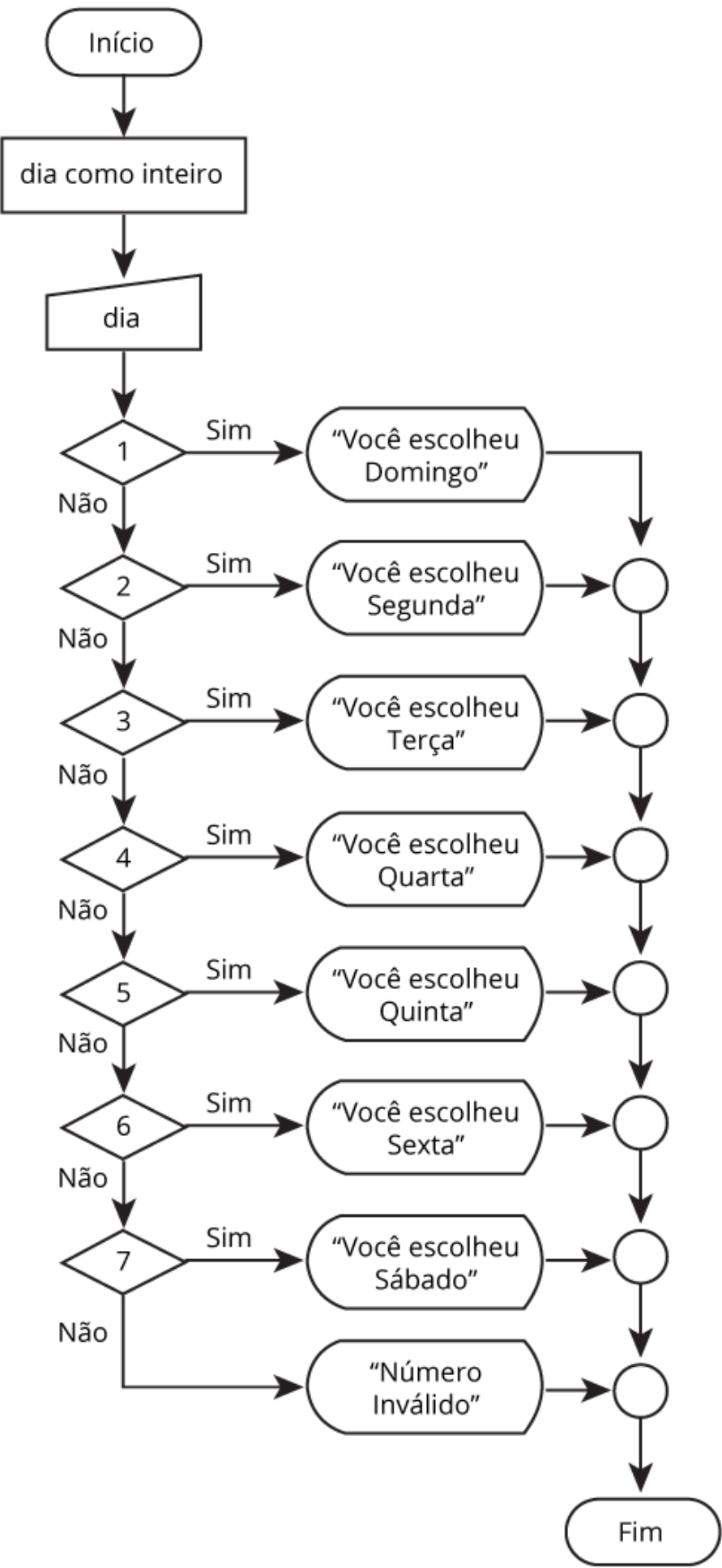
Se você pensou “inteiro”, acertou, pois a variável irá armazenar somente números inteiros.

Será que mais alguma variável é necessária?

A resposta é não, porque vamos fazer a saída diretamente imprimindo na tela.

2. Em seguida, pense no fluxograma, pois é mais fácil de entender e visualizar:

← Voltar



Você pode também optar por escrever o pseudocódigo. Escolha entre o fluxograma ou pseudocódigo, aquele que você preferir para simbolizar a sequência lógica do seu programa.

```
Programa Semana
Declare
    dia como inteiro
Início
    Escreva("Digite um Número de 1 a 7")
    Leia(dia)
    Se (dia = 1) Então
        Escreva ("Você escolheu domingo")
    senão
        Se (dia = 2) Então
            Escreva ("Você escolheu segunda")
```

```
senão
  Se (dia = 3) Então
    Escreva ("Você escolheu terça")
senão
  Se (dia = 4) Então
    Escreva ("Você escolheu quarta")
senão
  Se (dia = 5) Então
    Escreva ("Você escolheu quinta")
senão
  Se (dia = 6) Então
    Escreva ("Você escolheu sexta")
senão
  Se (dia = 7) Então
    Escreva ("Você escolheu sábado")
senão
  Escreva("Número Inválido")
Fim-Se
Fim-Se
Fim-Se
Fim-Se
Fim-Se
Fim-Se
Fim-Se
Fim.
```

3. Agora que você já pensou no problema na sua linguagem, fica mais fácil de traduzir para a linguagem Java. Veja:

```
1  import javax.swing.JOptionPane;
2
3  public class ifAninhadoSemana {
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         //processamento
13
14         if (dia == 1) { //if 1
15             JOptionPane.showMessageDialog(null, "Você escolheu Domingo");
16         } else {
17             if (dia == 2) { //if 2
18                 JOptionPane.showMessageDialog(null, "Você escolheu Segunda");
19             } else {
20                 if (dia == 3) { //if 3
21                     JOptionPane.showMessageDialog(null, "Você escolheu Terça");
22                 } else {
23                     if (dia == 4) { //if 4
24                         JOptionPane.showMessageDialog(null, "Você escolheu Quarta");
25                     } else {
26                         if (dia == 5) { //if 5
27                             JOptionPane.showMessageDialog(null, "Você escolheu Quinta");
28                         } else {
29                             if (dia == 6) { //if 6
30                                 JOptionPane.showMessageDialog(null, "Você escolheu Sexta");
31                             } else {
```

```
32         if (dia == 7) { //if 7
33             JOptionPane.showMessageDialog(null, "Você escolheu Sábado");
34         } else {
35             JOptionPane.showMessageDialog(null, "Número Inválido");
36         } // fim do if 7
37     } // fim do if 6
38 } // fim do if 5
39 } // fim do if 4
40 } // fim do if 3
41 } // fim do if 2
42 } // fim do if 1
43 } // fim do método main
44
45 } // fim da classe
```

← Voltar

Copiar

Após observar os exemplos sobre Estruturas de Decisão Aninhadas, você deve estar se perguntando:

Existe alguma relação entre o **número de alternativas** e o **número necessário de comparações** a serem efetuadas?

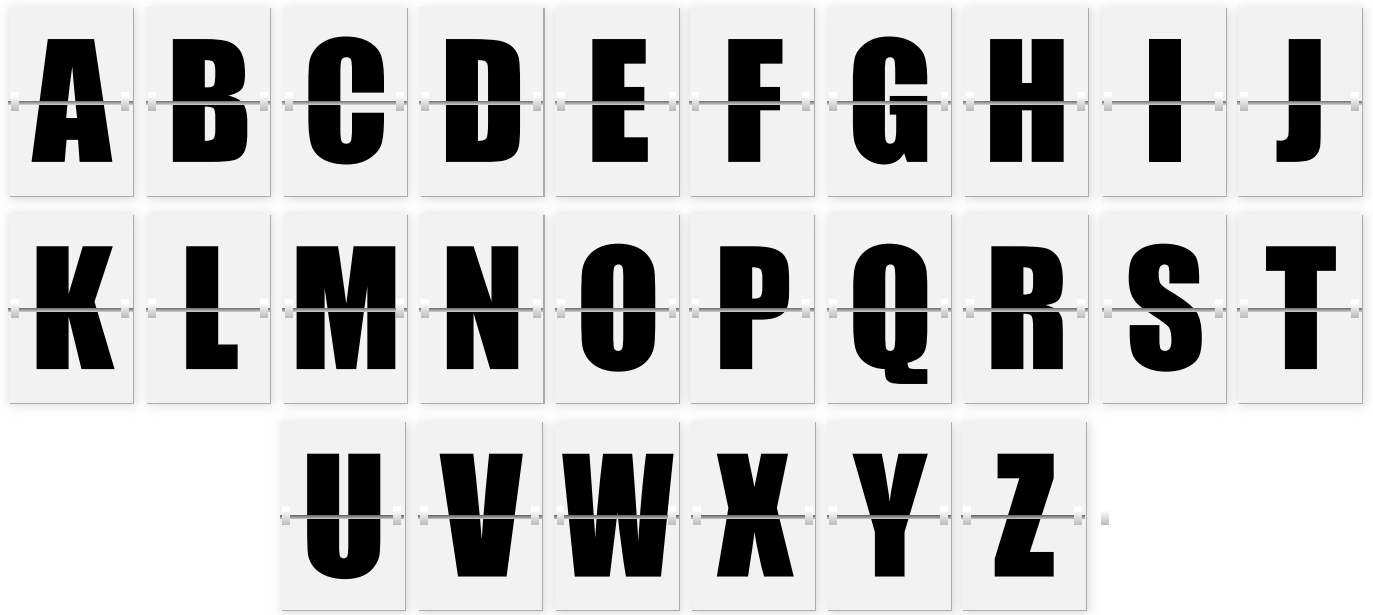
Pare e reflita! Se você acredita que sim, você acertou! Existe sim uma relação. O número de comparações necessárias em um programa é o **número de alternativas menos uma unidade**. Ou seja, se temos 6 alternativas, teremos 5 comparações em nosso programa.



Como realizar comparações com String no Java

Até agora você viu como as Estruturas de Decisão se aplicam a situações que envolvem tipos numéricos, mas já pensou como poderíamos realizar comparações com uma sequência de caracteres em Java? Seria da mesma forma?

Na verdade, não. Para realizarmos uma comparação de um conteúdo de uma variável com uma String – **sequência de caracteres** – no Java, temos que utilizar um método especial: o **.equals()**.



Mas como utilizamos o **.equals()**? Vejamos o exemplo abaixo:

```
1  import javax.swing.JOptionPane;
2
3  public class IfEquals {
4
5      public static void main(String[] args) {
6          //declaração de variáveis
7          String nome;
8
9          //entrada de dados
10         nome = JOptionPane.showInputDialog("Entre com um nome");
11
12         //Processamento e saída
13         if (nome.equals("Jose")) {
14             JOptionPane.showMessageDialog(null, "O Nome Digitado é Jose");
15         } else {
16             JOptionPane.showMessageDialog(null, "O nome digitado foi " + nome);
17         }
18     }
19 }
```

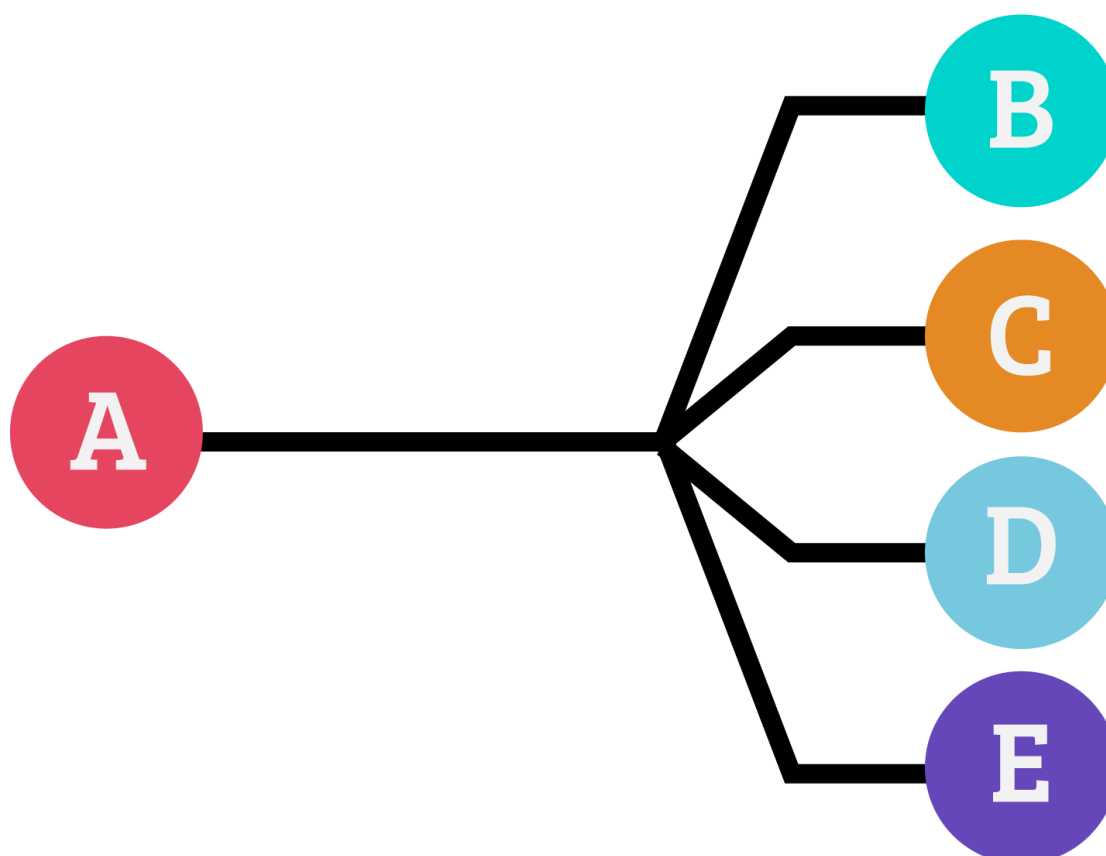
[Copiar](#)

Note que na linha 14 utilizamos a expressão **nome.equals("Jose")**, ou seja, estamos comparando se o valor armazenado na variável **nome** é igual a Jose. Se o usuário digitar qualquer outro nome, ou inclusive jose ou José o resultado da comparação é falso.

Então o **.equals()** é utilizado como:

<nomedavariável> **.equals** (valor a comparar)

Estrutura de Decisão caso... senão... fim_seleccione



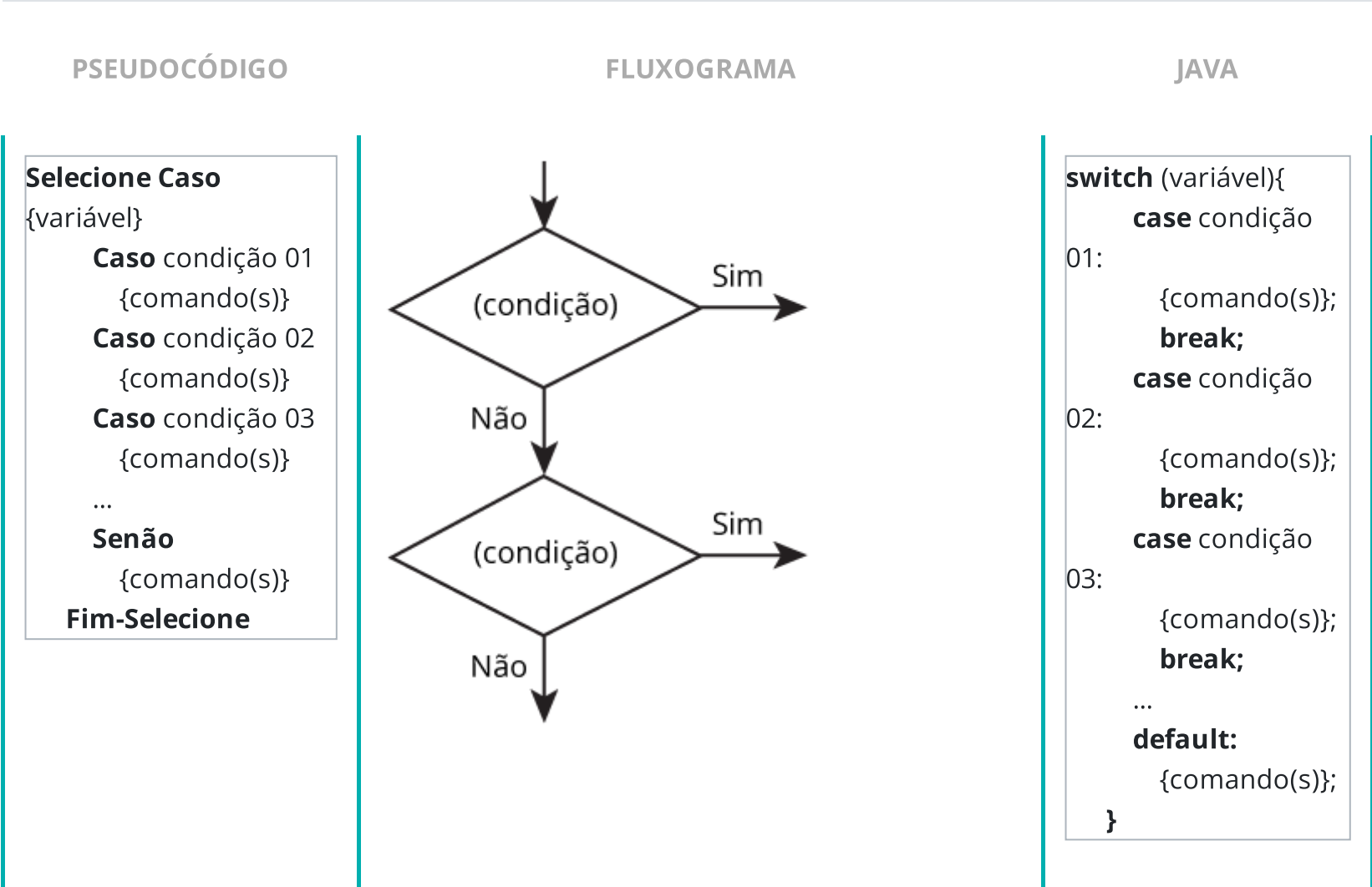
Já sabemos que a lógica de programação possui mecanismos que nos permitem tomar decisões dentro de um algoritmo. Sabemos também que esses mecanismos são denominados “Estruturas de Decisão”. A novidade é que essas estruturas não se restringem a apenas o “**se... senão...fim_se**”.

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 **Selecione...caso...senão...fim_selecione** e sua função principal 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 “selecione...caso...senão...fim_selecione”.

← Voltar

A estrutura “selecione...caso...senão...fim_selecione” 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.



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.

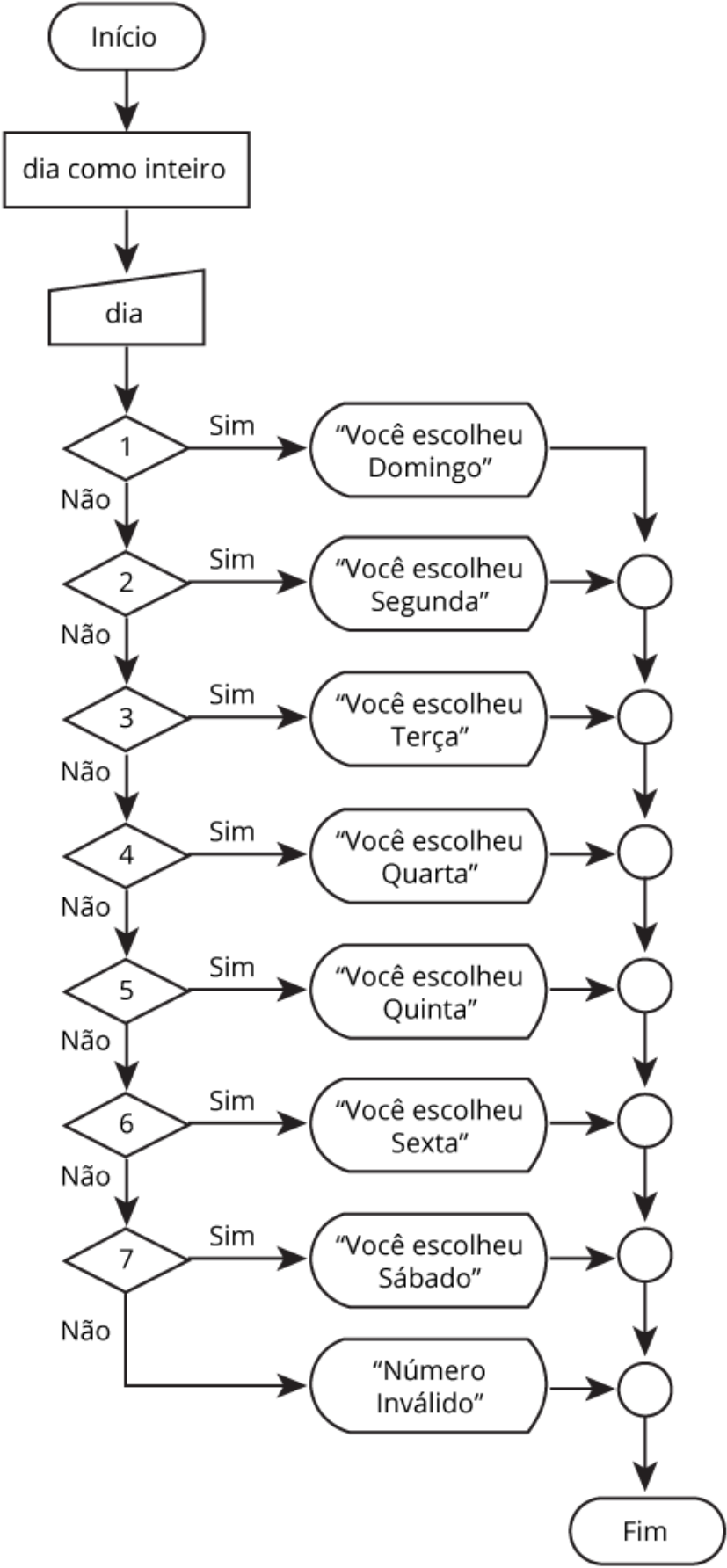
Para essas situações podemos utilizar a estrutura “selecione caso...senão...fim_selecione” do pseudocódigo ou a estrutura switch-case do Java.

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

PSEUDOCÓDIGO

FLUXOGRAMA

Programa semana
Declare
dia como inteiro
Início
Escreva ("Digite um Número de 1 a 7")
Leia(dia)
Selecione Caso (dia)
Caso 1
Escreva ("Você Escolheu Domingo")
Caso 2
Escreva ("Você Escolheu Segunda")
Caso 3
Escreva ("Você Escolheu Terça")
Caso 4
Escreva ("Você Escolheu Quarta")
Caso 5
Escreva ("Você Escolheu Quinta")
Caso 6
Escreva ("Você Escolheu Sexta")
Caso 7
Escreva ("Você Escolheu Sábado")
Senão
Escreva("Número Inválido")
Fim-Selezione
Fim.



Agora, veja o código em Java:

```
1 import javax.swing.JOptionPane;
2
3 public class caseDiaSemana {
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        //processamento
13    }
```

```
14  switch (dia) {
15      case 1:
16          JOptionPane.showMessageDialog(null, "Você escolheu Domingo");
17          break;
18      case 2:
19          JOptionPane.showMessageDialog(null, "Você escolheu Segunda");
20          break;
21      case 3:
22          JOptionPane.showMessageDialog(null, "Você escolheu Terça");
23          break;
24      case 4:
25          JOptionPane.showMessageDialog(null, "Você escolheu Quarta");
26          break;
27      case 5:
28          JOptionPane.showMessageDialog(null, "Você escolheu Quinta");
29          break;
30      case 6:
31          JOptionPane.showMessageDialog(null, "Você escolheu Sexta");
32          break;
33      case 7:
34          JOptionPane.showMessageDialog(null, "Você escolheu Sábado");
35      default:
36          JOptionPane.showMessageDialog(null, "Número inválido");
37          break;
38  } //fim do switch-case
39  } // fim do método main
40  } // fim da classe
```



Copiar

Você percebeu que esse código corresponde ao utilizado no exemplo anterior, utilizando a Estrutura de Decisão **“se”**? Ambos têm o mesmo efeito prático, mas com a estrutura **"caso"**, o código foi escrito de maneira mais elegante e o fluxograma permaneceu o mesmo sem alteração nenhuma.

Revisando as Estruturas de Decisão

Agora que você já estudou o conteúdo da unidade, assista ao vídeo do professor Sandro Valérius, preparado para o curso técnico de Informática, Módulo I - Agenda 13.

Informática - Módulo I - Agenda 13 - Estruturas de Decisão II



Secretaria de
Desenvolvimento Econômico

Secretaria da
Educação

