

# Desvio condicional

APRESENTAR AS ESTRUTURAS CONDICIONAIS SIMPLES, COMPOSTA E ANINHADAS. APRESENTAR SEQUÊNCIAS DE IF-ELSE-IF-ELSE E O OPERADOR TERNÁRIO CONDICIONAL.

AUTOR(A): PROF. APARECIDA DE FATIMA CASTELLO ROSA

## Introdução

Os programas são compostos por comandos ou instruções e são executados sequencialmente. Alguns comandos possibilitam que se criem caminhos diferentes, ou que o código seja desviado da sequência que foi escrito, de acordo com alguma condição que seja necessária ao problema que estamos resolvendo computacionalmente por meio do programa ou algoritmo.

As estruturas condicionais são utilizadas em tomadas de decisão e podem executar diferentes ações dependendo da avaliação de uma determinada condição ou expressão.

Antes de falarmos das estruturas condicionais, vamos relembrar os operadores relacionais e os operadores lógicos.

## Operadores Relacionais

O operador relacional compara dois valores e sempre retorna um valor verdadeiro ou falso. Eles são principalmente utilizados para especificar condições, nas quais envolvem avaliar se uma expressão é verdadeira ou falsa.

Os operadores relacionais são apresentados a seguir.

Operador	Descrição
==	Igual a
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a
!=	Diferente de

## Operadores Lógicos

Os operadores lógicos são utilizados juntamente com os operadores relacionais. Eles avaliam expressões lógicas que resultam em verdadeiro ou falso.

Os operadores lógicos são apresentados na tabela a seguir.

Operador	Significado
&&	Operador E condicional
&	Operador E lógico
	Operador OU condicional
	Operador OU inclusivo lógico
^	XOR Operador OU exclusivo lógico
!	Operado NÃO lógico, também chamado de negação lógica

Vamos relembrar a tabela verdade para os operadores lógicos E, OU.

Assuma que X e Y são booleanos.

X	Y	X && Y	X    Y
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

## Expressão Lógica

Uma condição é uma expressão lógica ou expressão booleana, portanto o resultado dessa expressão é um valor lógico: VERDADEIRO ou FALSO.

Para ter uma expressão lógica como condição, usa-se uma relação lógica entre dois elementos e um operador relacional. Por exemplo: Dada as variáveis A e B, temos

VARIÁVEL versus VARIÁVEL: A = B; A > B; A < B; A >= B; A <= B; A != B;

Desvio condicional

VARIÁVEL versus CONSTANTE: A = 5; A > 5; A < 5; A >= 5; A <= 5; A != 5;

02 / 20

```
1. if (condição)
2. {
3.     /* instruções a serem executadas se a condição for verdadeira */
4. }
5.
6. /* instruções a serem executadas após condição falsa ou após executar instruções:
```

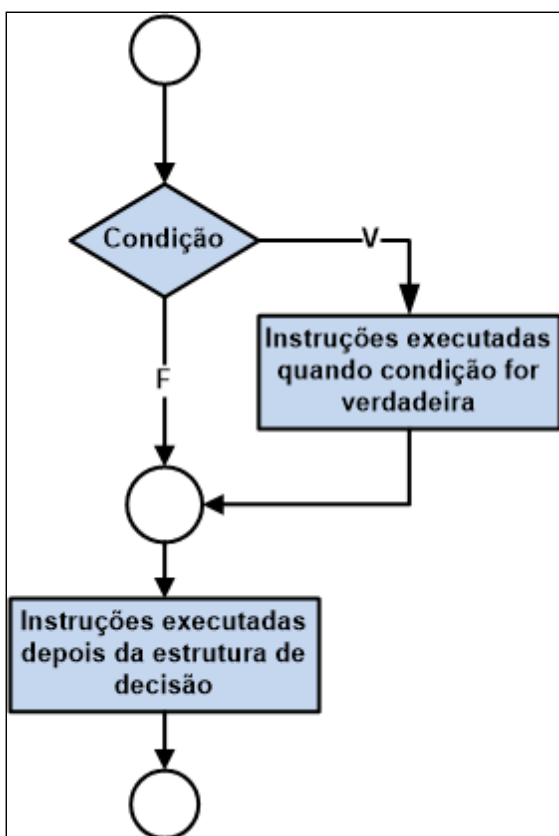
Nessa instrução, primeiramente a condição é avaliada. Se a condição for verdadeira, então o código declarado dentro do corpo ( { } ) da instrução if será executado. Se a condição for falsa, serão executados os códigos que estiverem após o corpo da instrução.

O trecho de código comentado a seguir apresenta um exemplo da utilização da estrutura condicional simples if.

```
1. //declara a variável num do tipo inteiro e atribui o valor 8 a essa variável.
2. int num = 8;
3.
4. //verifica se a variável num tem um valor menor que 10. Se for verdadeiro então
5. //Se a variável num for maior ou igual a 10, o comando if não será executado.
6.
7. if (num < 10)
8. {
9.     MessageBox.Show("O valor da variável num é menor que 10.");
10. } //fim do comando if
11.
```

Fluxograma desvio condicional simples

O fluxograma a seguir representa a estrutura do desvio condicional simples.



## Hora de praticar

Vamos criar um projeto para praticar a estrutura de desvio condicional simples.

Nesse exemplo, o usuário digita um valor que representa a média de notas e se a média for maior ou igual a sete, o programa apresenta a mensagem "Aluno aprovado!". Só verificaremos se a média é maior ou igual a sete (condição verdadeira). Se o teste da condição for falso, não será executada nenhuma ação.

Vamos começar

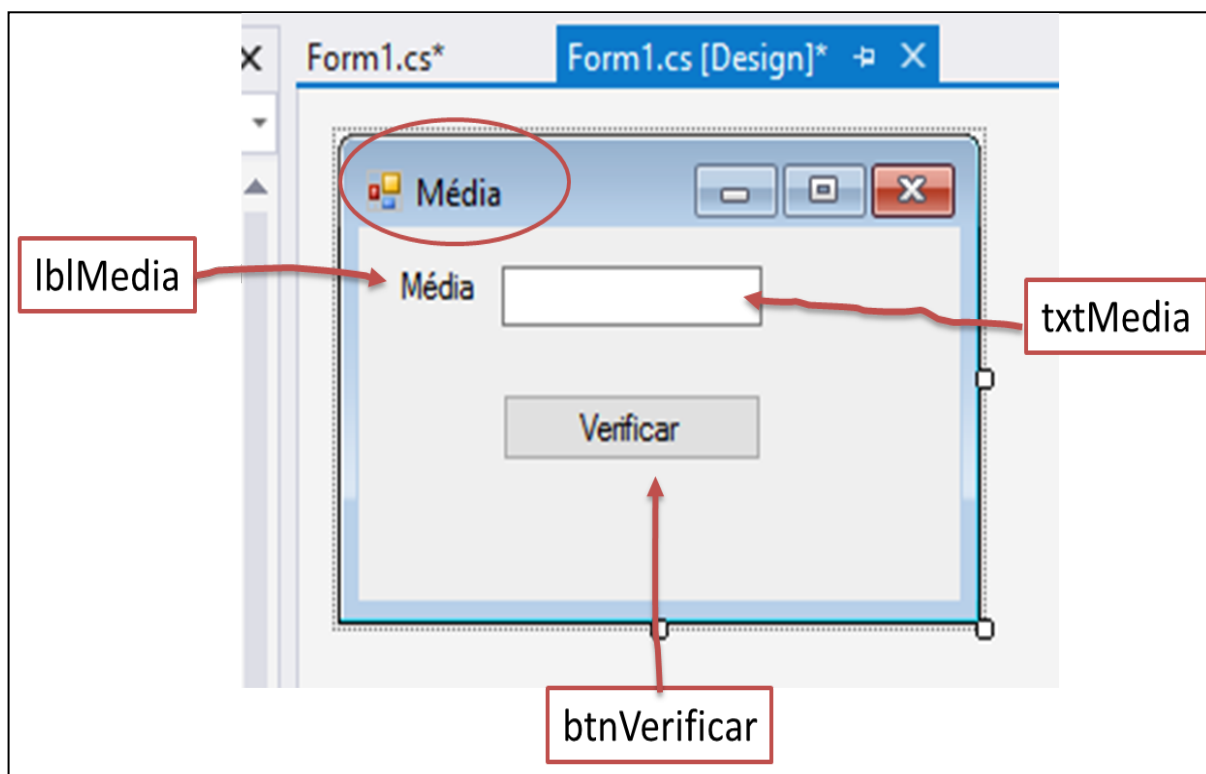
1. Inicie o Visual Studio (se ainda não o fez).
2. Na Start Page escolha New Project, ou caso a Start Page não esteja visível, na barra de menu escolha File, New Project.

A caixa de diálogo é exibida.

3. Selecione no painel à esquerda o template Visual C#, no painel central escolha Windows Forms Application.
4. Na caixa de texto Name: CondicionalSimples.
5. Location: escolha o local onde deseja salvar o seu projeto.
6. Mantenha selecionado: Create directory for solution.
7. Clique em OK

#### Criando o Design do Form

1. Após fixar (Auto Hide) a Toolbox, abra o nó Common Controls, e insira os controles no Form. Lembre-se para inserir os controles, basta selecionar o controle na ToolBox, clicar, segurar e arrastar para a posição que desejar dentro do Form.
2. Faça um layout parecido com a imagem a seguir.



Selecione os controles e altere as propriedades conforme definido na tabela abaixo:

Controle	Propriedade = Valor
Form	Text = Média
Label Média	(name) = lblMedia Text = Média
TextBox	(name) = txtMedia
Button Verificar	(name) = btnVerificar Text = Verificar

### Escrevendo o código do programa

Agora que você já criou a interface e alterou as propriedades dos controles, vamos codificar o botão btnVerificar. Para abrir o editor de código dê duplo clique no botão btnVerificar e digite o código a seguir.

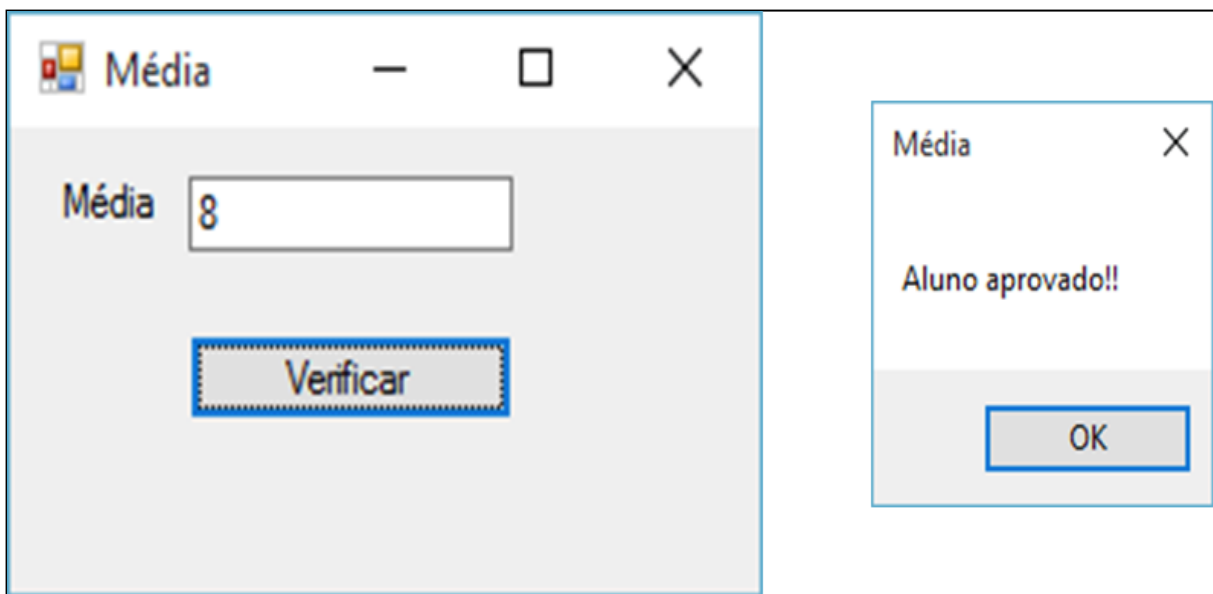
```
1. //declarando a variável media
2. double media;
3.
4. //variavel media recebe o valor que foi digitado no TextBox
5. //Convert.ToDouble, converter o texto do TextBox para o tipo double
6. media = Convert.ToDouble(txtIdade.Text);
7.
8. //testa a condicao: se média é maior ou igual a sete
9. //se condição verdadeira, apresenta a mensagem no MessageBox
10. if (media >= 7.0)
11. {
12. MessageBox.Show("Aluno aprovado!!", "Média");
13. }
14.
15. //OBS.
16. //Se a média for menor que sete, não é realizada nenhuma ação.
```

O código do programa deve ter ficado parecido com a imagem a seguir.

```
13 public partial class Form1 : Form
14 {
15     1 reference
16     public Form1()
17     {
18         InitializeComponent();
19     }
20     1 reference
21     private void btnVerificar_Click(object sender, EventArgs e)
22     {
23         //declarando a variável media
24         double media;
25
26         //variavel media recebe o valor que foi digitado no TextBox
27         //Convert.ToDouble, converter o texto do TextBox para o tipo double
28         media = Convert.ToDouble(txtIdade.Text);
29
30         //testa a condicao: se média é maior ou igual a sete
31         //se condição verdadeira, apresenta a mensagem no MessageBox
32         if (media >= 7.0)
33         {
34             MessageBox.Show("Aluno aprovado!!", "Média");
35         }
36
37         //OBS.
38         //Se a média for menor que sete, não é realizada nenhuma ação.
39     }
40 }
```

Lembrando que o IDE do Visual Studio gera códigos automaticamente e o que você irá digitar é o código da linha 22 até a linha 37.

Veja o resultado final do programa.



## Desvio Condicional Composto – if-else

O desvio condicional composto (if ... else) nos permite tomar decisões para quando a condição é avaliada como verdadeira ou quando a condição é avaliada como falsa.

Sintaxe em C#

```
1. if(condição)
2. {
3.     /* instruções a serem executadas se a condição for verdadeira */
4. }
5. else
6. {
7.     /*instruções a serem executadas se a condição for falsa */
8. }
9.
10. /*instruções a serem executadas após condição ser verdadeira ou após condição se
```



Se a condição for avaliada como verdadeira, serão executadas todas as instruções definidas dentro do bloco if da instrução. Caso seja a condição seja avaliada como falsa, então serão executadas todas as instruções do bloco else. Após a execução de um dos blocos são executadas as eventuais instruções que existirem depois do encerramento do desvio condicional.

O trecho de código comentado a seguir apresenta um exemplo da utilização da estrutura condicional composta if ... else

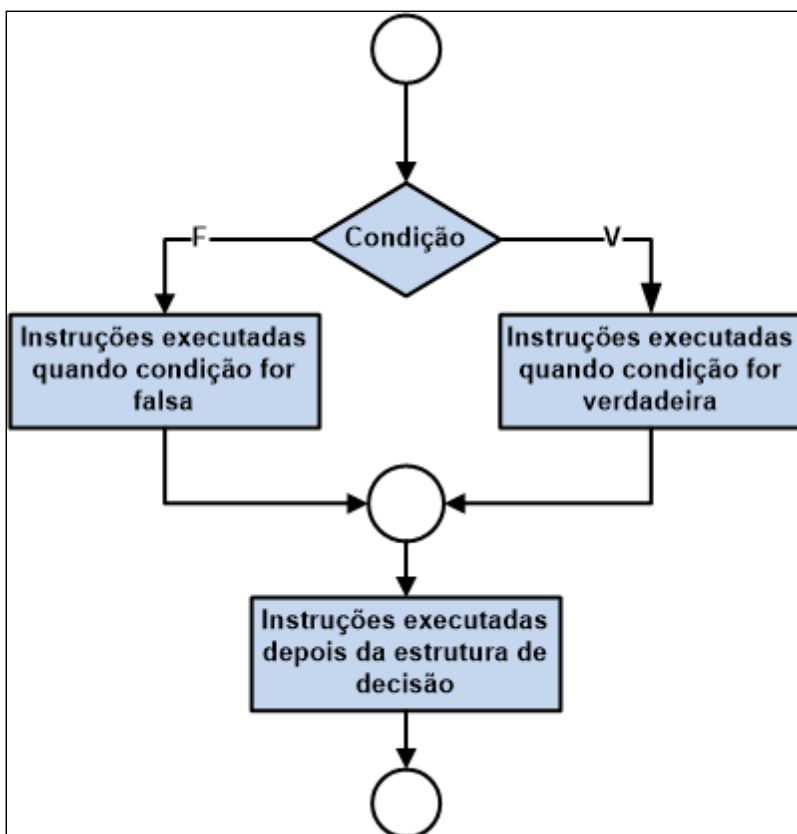
```
1. //declara a variável num do tipo inteiro e atribui o valor 10 a essa variável.  
2.  
3. int num = 10;  
4.  
5. //verifica se a variável num tem valor menor que 10. Se num for menor que 10, c  
6. //Senão o comando else será executado.  
7.  
8. if (num < 10)  
9. {  
10.     MessageBox.Show("O valor da variável num é menor que 10.");  
11. }  
12. else  
13. {  
14.     MessageBox.Show("O valor da variável num é maior ou igual a 10.");  
15. } // fim da instrução if ... else
```

Nesse exemplo, na linha 3, a variável num recebe o valor 10.

Na linha 8, o valor da variável é comparado se ele é menor que 10. Como essa comparação resulta em um valor falso, o código é desviado para a cláusula else, e a mensagem "O valor da variável num é maior ou igual a 10." será apresentada no MessageBox.

#### Fluxograma desvio condicional composto

A estrutura do desvio condicional composto é apresentada no fluxograma a seguir.



## Hora de praticar

Vamos criar um projeto para praticar a estrutura de desvio condicional composto.

Vamos escrever um programa para ler três notas de um aluno, calcular a média aritmética e além de apresentar a média, apresentar a situação do aluno de acordo com a seguinte condição: se a média do aluno for maior ou igual a sete, o aluno está aprovado, senão o aluno está reprovado.

Vamos começar.

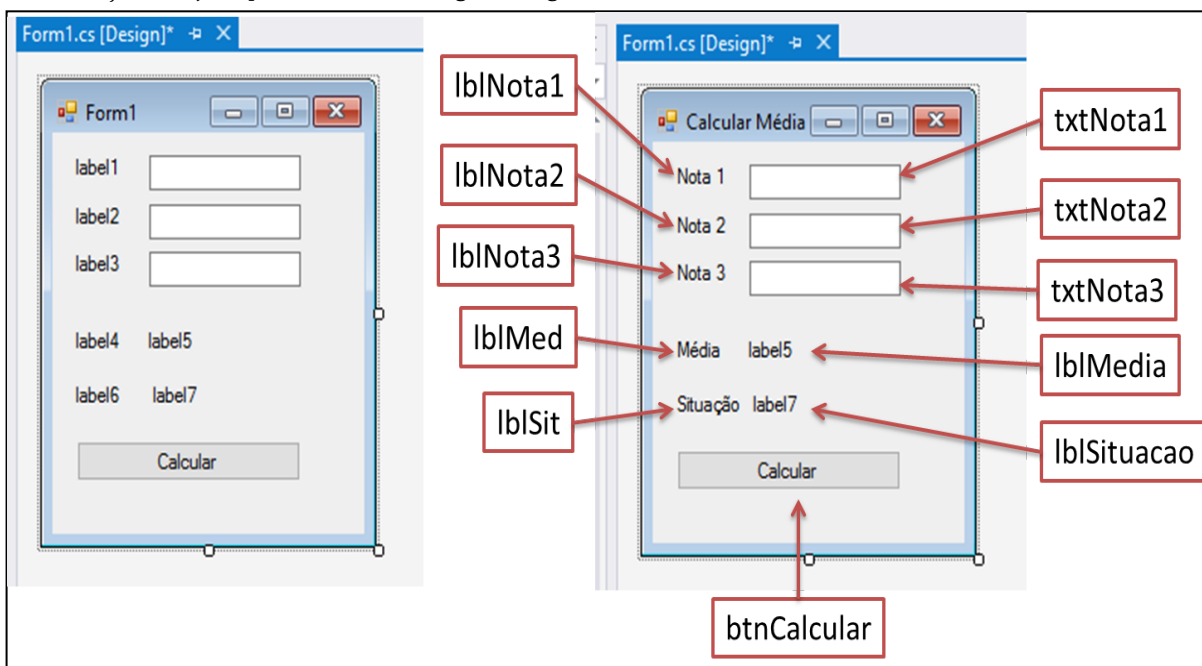
1. Inicie o Visual Studio (se ainda não o fez).
2. Na Start Page escolha New Project, ou caso a Start Page não esteja visível, na barra de menu escolha File, New Project.

A caixa de diálogo é exibida.

3. Selecione no painel à esquerda o template Visual C#, no painel central escolha Windows Forms Application.
4. Na caixa de texto Name: CalcularMedia.
5. Location: escolha o local onde deseja salvar o seu projeto.
6. Mantenha selecionado: Create diretory for solution.
7. Clique em OK

### Criando o Design do Form

1. Após fixar (Auto Hide) a Toolbox, abra o nó Common Controls, e insira os controles no Form. Lembre-se para inserir os controles, basta selecionar o controle na ToolBox, clicar, segurar e arrastar para a posição que desejar dentro do Form.
2. Faça um layout parecido com a imagem a seguir.



Selecione os controles e altere as propriedades conforme definido na tabela abaixo:

Controle	Propriedade = Valor
Form	Text = Calcular Média
Label1	(name) = lblNota1 Text = Nota 1
Label2	(name) = lblNota2 Text = Nota 3
Label3	(name) = lblNota3 Text = Nota 3
Label4	(name) = lblMed Text = Média
Label5	(name) = lblMedia
Label6	(name) = lblSit Text = Situação
Label7	(name) = lblSituacao
TextBox	(name) = txtNota1
TextBox	(name) = txtNota2
TextBox	(name) = txtNota3
Button	(name) = btnCalcular Text = Calcular

Escrevendo o código do programa

Agora que você já criou a interface e alterou as propriedades dos controles, vamos codificar o botão btnCalcular. Para abrir o editor de código dê duplo clique no botão btnCalcular e digite o código a seguir.

```
1. //declaração das variáveis
2. double nota1, nota2, nota3, media;
3.
4. //entrada de dados
5. //as variáveis nota1, nota2, nota3, recebem o valor que foi digitado no TextBox
6. //Convert.ToDouble, faz a conversão do tipo texto para double
7. nota1 = Convert.ToDouble(txtNota1.Text);
8. nota2 = Convert.ToDouble(txtNota2.Text);
9. nota3 = Convert.ToDouble(txtNota3.Text);
10.
11. //processamento, calculando a média:
12. //soma-se as três notas e dividi por 3.
13. media = (nota1+nota2+nota3)/3;
14.
15. //apresenta a média na lblMedia
16. //ToString, converte o valor double que está armazenado na variável
17. //média e converte para texto.
18. lblMedia.Text = media.ToString();
19.
20. //verifica a situação do aluno, analisando o valor da variável média
21.
22. if (media >= 7.0)
23. {
24.     //instruções se a condição (media >= 7.0) for verdadeira
25.     lblSituacao.Text = "Aluno Aprovado!";
26. }
27. else
28. {
29.     //instruções se a condição (media >= 7.0) for falsa
30.     lblSituacao.Text = "Aluno Reprovado!";
31.
32. } //fim da instrução if/else
```

A imagem a seguir apresenta o código do botão calcular.

```
20 private void btnCalcular_Click(object sender, EventArgs e)
21 {
22     //declaração das variáveis
23     double nota1, nota2, nota3, media;
24
25     //entrada de dados
26     //as variáveis nota1, nota2, nota3, recebem o valor que foi digitado no TextBox
27     //Convert.ToDouble, faz a conversão do tipo texto para double
28     nota1 = Convert.ToDouble(txtNota1.Text);
29     nota2 = Convert.ToDouble(txtNota2.Text);
30     nota3 = Convert.ToDouble(txtNota3.Text);
31
32     //processamento, calculando a média:
33     //soma-se as três notas e dividi por 3.
34     media = (nota1+nota2+nota3)/3;
35
36     //apresenta a média na lblMedia
37     //ToString, converte o valor double que está armazenado na variável
38     //média e converte para texto.
39     lblMedia.Text = media.ToString();
40
41     //verifica a situação do aluno, analisando o valor da variável média
42
43     if (media >= 7.0)
44     {
45         //instruções se a condição (media >= 7.0) for verdadeira
46         lblSituacao.Text = "Aluno Aprovado!";
47     }
48     else
49     {
50         //instruções se a condição (media >= 7.0) for falsa
51         lblSituacao.Text = "Aluno Reprovado!";
52     } //fim da instrução if/else
53
54 }
```

## Estruturas Condicionais Aninhadas

Na programação, frequentemente temos que fazer vários testes condicionais, no qual temos que escrever uma instrução condicional dentro da outra. Esse tipo de estrutura é chamada de estrutura condicional aninhada.

Exemplo comentado.

```
1. //declara e atribui valores as variáveis
2. int primeiro = 5;
3. int segundo = 3;
4.
5. //compara se primeiro é igual a segundo
6. //se não forem iguais a cláusula else será executada
7.
8. if (primeiro == segundo)
9. {
10.     MessageBox.Show("Os números são iguais");
11. }
12. else // executado se os números não forem iguais
13. {
14.     //verifica se primeiro é maior que segundo
15.     //se verdadeiro executa as instruções, senão a cláusula else será executada
16.     if (primeiro > segundo)
17.     {
18.         MessageBox.Show("O primeiro número é maior");
19.     }
20.     else //será executado se o segundo if for falso
21.     {
22.         MessageBox.Show("O segundo número é maior");
23.     }
24. }
```

## Sequências de if-else-if-else

Às vezes necessitamos escrever sequências de estruturas if, no qual a cláusula else é uma nova estrutura if. Se usarmos as estruturas aninhadas, o código poderá ficar confuso e extenso. Nesses casos, podemos utilizar um novo if logo após o else.

Veja o exemplo anterior reescrito utilizando sequências de if-else-if-else.

```
1. int primeiro = 5;
2. int segundo = 3;
3.
4. if (primeiro == segundo)
5. {
6.     MessageBox.Show("Os números são iguais");
7. }
8. else if (primeiro > segundo)
9. {
10.     MessageBox.Show("O primeiro número é maior");
11. }
12. else
13. {
14.     MessageBox.Show("O segundo número é maior");
15. }
```

## Estruturas Condicionais e Operadores Lógicos

Até agora vimos as estruturas condicionais utilizando os operadores relacionais.

Podemos formar condições complexas combinando os operadores lógicos. As expressões são avaliadas e tem seu resultado de acordo com a tabela verdade.

Vamos supor que queremos garantir que um número está dentro de um intervalo específico. Por exemplo, queremos verificar se um número está dentro do intervalo de 50 a 100. Nesse caso podemos utilizar o operador && condicional.



```
1. if( num >= 50 && num <= 100 ) {  
2.     MessageBox.Show("O número encontra-se no intervalo de 50 a 100");  
3. }  
4. else  
5. {  
6.     MessageBox.Show("O número não está dentro do intervalo");  
7. }
```

Essa mesma instrução pode ser escrita da seguinte forma:

```
1. if( (num >= 50) && (num <= 100) ) {  
2.     MessageBox.Show("O número encontra-se no intervalo de 50 a 100");  
3. }  
4. else  
5. {  
6.     MessageBox.Show("O número não está dentro do intervalo");  
7. }
```

A expressão `num >= 50 && num <= 100` será avaliada e será verdadeira somente se `num` for maior ou igual a 50 e menor ou igual a 100.

Agora, vamos supor a seguinte situação: um aluno pode ser aprovado se a média final for maior ou igual a 7,0 ou se a nota do exame for maior ou igual 8,0. Nesse caso podemos utilizar o operador `||` condicional.

```
1. if( ( media >= 7.0) || ( exame >= 8.0) )
2. {
3.     MessageBox.Show("Aluno aprovado!");
4. }
5. else
6. {
7.     MessageBox.Show("Aluno Reprovado!");
8. }
```

## Operador ternário ?:

O operador condicional ternário ( ? ) está relacionado com a estrutura condicional if-else e é escrito em uma única linha.

*condição ? expressao1 : expressao2;*

Avalia a *condição*.

Se verdadeira, o resultado é o valor da *expressao1*.

Se falsa, o resultado é o valor da *expressao2*.

Veja o exemplo em C#

```
1. x = num < 10 ? 0 : 1;
```

Se num for menor que 10, a variável x recebe o valor 0, senão a variável x recebe o valor 1.

## ATIVIDADE

Analise o trecho de código a seguir.

```
int num1, num2, num3;  
if (num1 > num2)  
{  
    num3 = num1;  
    num1 = num2;  
    num2 = num3;  
}
```

Considere que o valor da variável num1 é igual a 8 e o valor da variável num2 igual 7.

Após a execução do código, quais os valores das variáveis num1, num2 e num3 respectivamente?

- A. 8; 7; 8
- B. 8; 8; 8
- C. 7; 8; 8
- D. 8; 8; 7

## ATIVIDADE

Em uma instrução condicional composta, o *else* de um comando if-else é executado quando

- A. a avaliação da condição do if for falsa.
- B. a avaliação da condição do if for verdadeira.
- C. a avaliação da condição do else for falsa.
- D. a avaliação da condição do else for verdadeira.

## ATIVIDADE

Analise o trecho de código a seguir.

```
23  int a, b, c;  
24  
25  a = 15;  
26  b = 5;  
27  c = 2;  
28  
29  c += a < b ? a : b;  
30  
31  if ((b > c) && (c < a) )  
32  {  
33      c = a + b;  
34  }  
35  else  
36  {  
37      c = a - b;  
38  }
```

Após a execução do código acima, qual será o valor da variável c?

- A. 15
- B. 5
- C. 2
- D. 10

## REFERÊNCIA

DEITEL, H. M. *et al.* C# Como Programar. São Paulo: Pearson Makron Books, 2003.

OLSSON, Mikael. *C# Quick Syntax Reference*. Berkeley, CA: Apress, 2013.

### SITES

Microsoft <<https://msdn.microsoft.com/pt-br/library/5011f09h.aspx>> (<https://msdn.microsoft.com/pt-br/library/5011f09h.aspx>). Acesso em 25/09/2015

Visual C# Tutorials <<http://visualcsharp tutorials.com/fundamentals/if-else-statement/>> (<http://visualcsharp tutorials.com/fundamentals/if-else-statement/>). Acesso em 25/09/2015

