



Alex Sander Resende de Deus

A 25 anos ensinando programação a jovens e adultos.

Apaixonado por tecnologia é atualmente coordenador de cursos na ETEC Albert Einstein. Na FIAP atua como professor na FIAP School, lecionando C#, SQLServer e Desenvolvimento Mobile

Aula 03



Mais desvios lógicos

```
match http_code:  
  case "200":  
    do_something_good()  
  case "418":  
    make_coffee()  
  case _:  
    pass
```

Switch Case



Operadores Lógicos

AULA 3

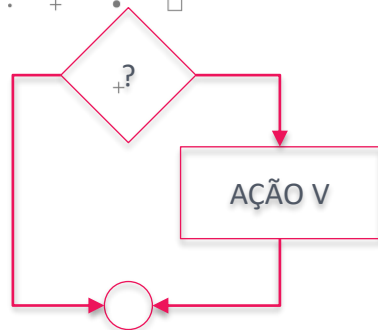
MAIS DESVIOS CONDICIONAIS: OPERADORES LÓGICOS E SWITCH CASE

RELEMBRANDO A

AULA 2

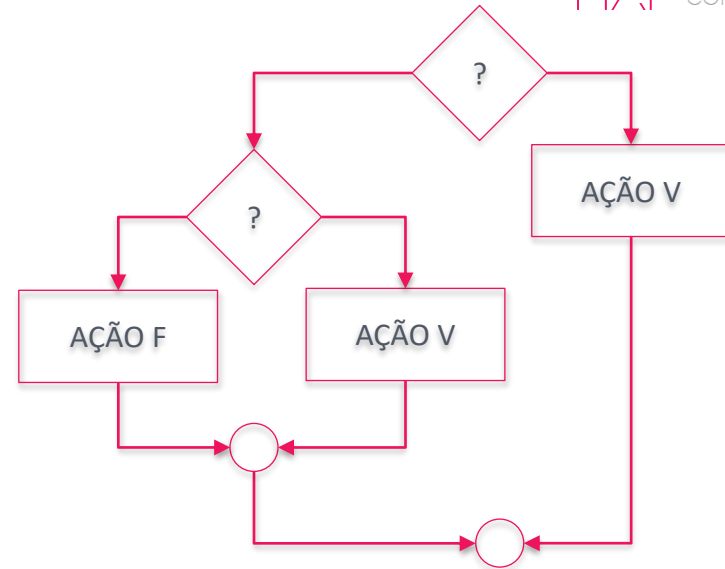
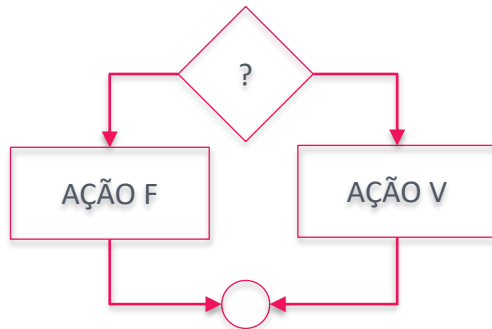
Os **desvios condicionais** são estruturas de programação que permitem executar algumas instruções com base no resultado – verdadeiro ou falso – de um teste lógico.

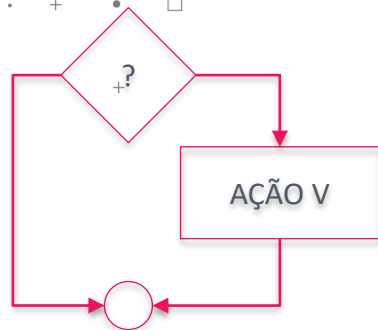
Na última aula, foram estudados 3 *ifs*: o **if simples**, o **if composto** e o **if encadeado**.



O if simples apresenta um teste lógico e executa um conjunto de instruções caso o teste seja verdadeiro.

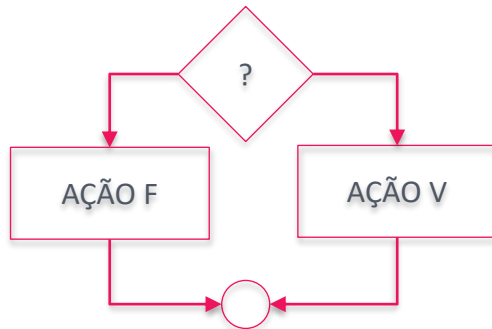
if simples





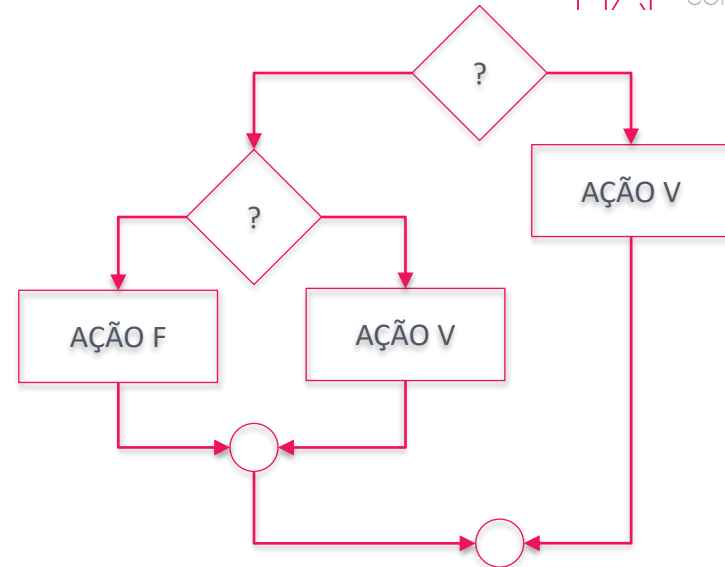
O if simples apresenta um teste lógico e executa um conjunto de instruções caso o teste seja verdadeiro.

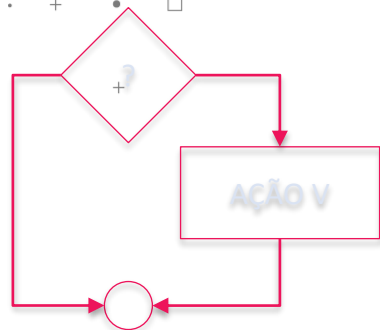
if simples



O if composto apresenta um teste lógico e executa um conjunto de instruções caso o teste seja verdadeiro e outro caso seja falso.

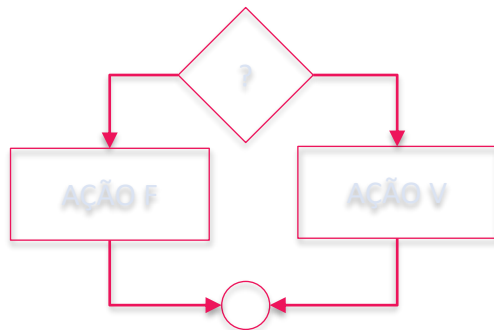
if composto





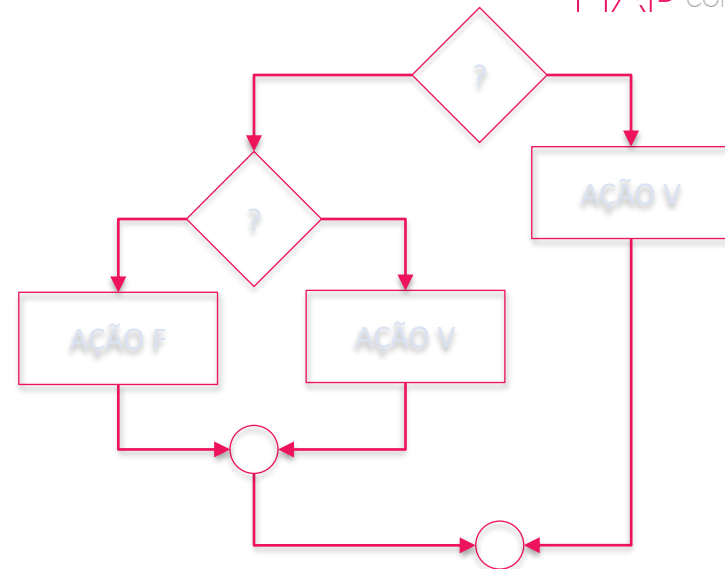
O if simples apresenta um teste lógico e executa um conjunto de instruções caso o teste seja verdadeiro.

if simples



O if composto apresenta um teste lógico e executa um conjunto de instruções caso o teste seja verdadeiro e outro caso seja falso.

if composto



O if encadeado apresenta um if dentro de outro. Geralmente usamos quando a validação de uma condição está condicionada à anterior.

if encadeado

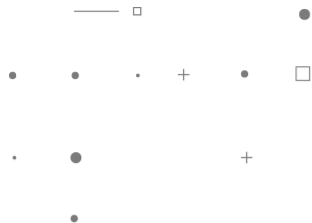
PROGRAMANDO EM C#

CONHECENDO OS
OPERADORES LÓGICOS

Em diversas ocasiões, uma única condição é insuficiente para validar algo dentro dos nossos programas.

Nesses casos, precisaremos fazer uso de *operadores lógicos* para realizarmos a *conexão* entre as diferentes condições.

Esses operadores lógicos servem para determinar *quais são os resultados possíveis* quando as condições forem avaliadas em conjunto.



Existem diversos operadores lógicos, sendo que nem todas as linguagens de programação implementam todos eles.

No caso do Java, existem os operadores **não, e, ou, ou exclusivo, e condicional** e **ou condicional**.

Vamos entender como cada um deles se comporta?



OPERADOR	NOME	SIGNIFICADO
!	NÃO (NOT)	Inverte o estado lógico da condição que acompanhar: verdadeiro vira falso e falso vira verdadeiro.
&&	E CONDICIONAL (CONDITIONAL AND)	Retorna verdadeiro se a primeira condição e a segunda forem verdadeiras. Caso a primeira condição seja falsa, a segunda nem chega a ser avaliada.
	OU CONDICIONAL (CONDITIONAL OR)	Retorna verdadeiro se pelo menos uma das condições forem verdadeiras. Caso a primeira condição seja verdadeira, a segunda nem chega a ser avaliada.



Se nossa condição for: “a lâmpada está acesa?”, nosso resultado é **verdadeiro**.

Se adicionarmos o operador lógico não (“! A lâmpada está acesa?”), nosso resultado é **falso**, pois ele inverte o estado lógico da condição original.





Se nossa condição for: “a lâmpada está acesa?”, nosso resultado é **verdadeiro**.

Se a nossa condição for: “a porta está fechada?”, nosso resultado é **falso**.

Se adicionarmos o operador lógico **e** (“A lâmpada está acesa? & A porta está fechada?”),
nosso resultado é **falso**, pois a primeira condição é verdadeira e a segunda condição é
falsa.



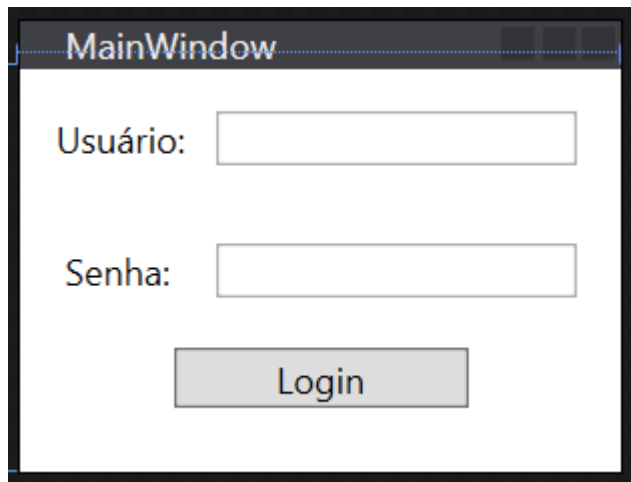


Se nossa condição for: “a lâmpada está acesa?”, nosso resultado é **verdadeiro**.

Se a nossa condição for: “a porta está fechada?”, nosso resultado é **falso**.

Se adicionarmos o operador lógico *ou* (“A lâmpada está acesa? | A porta está fechada?”),
nosso resultado é **verdadeiro**, pois pelo menos uma das condições avaliadas é verdadeira.

Faça um programa que verifique se o usuário digitado é igual a 'administrador' e se a senha digitada é igual a '123'.



The image shows a screenshot of a Windows application window titled "MainWindow". Inside the window, there are two text input fields. The first field is labeled "Usuário:" and the second field is labeled "Senha:". Below these fields is a button labeled "Login".

```
private void btnLogin_Click(object sender, RoutedEventArgs e)
{
    string usuario, senha;
    usuario = txtUsuario.Text;
    senha = txtSenha.Text;
    if(usuario=="administrador" && senha == "123")
    {
        MessageBox.Show("Acesso autorizado!", "Login",
            MessageBoxButton.OK, MessageBoxImage.Information);
    }
    else
    {
        MessageBox.Show("Usuário e/ou senha inválidos", "Login",
            MessageBoxButton.OK, MessageBoxImage.Error);
    }
}
```



Com o uso de um dos operadores E, conseguimos exigir que as duas condições sejam verdadeiras para que a mensagem de acesso seja exibida.

Se usarmos o operador &&, a segunda condição só é avaliada se a primeira for verdadeira.



Com o uso de um dos operadores OU, conseguimos exigir que pelo menos uma das condições seja atendida para que o desconto seja aplicado. Se usarmos o operador `||`, a segunda condição só será avaliada se a primeira for falsa.

PROGRAMANDO EM C#

CONHECENDO O
SWITCH CASE

Depois de fazermos diversos testes, podemos ver quão poderoso é o if!

Para alguns casos, porém, ele não é a ferramenta mais indicada.

Imagine que você precisa verificar se o usuário digitou 1, 2, 3, 4 ou 5 e disparar algumas funcionalidades dependendo do que ele digitou.

Um if ficaria um pouco comprido, certo?

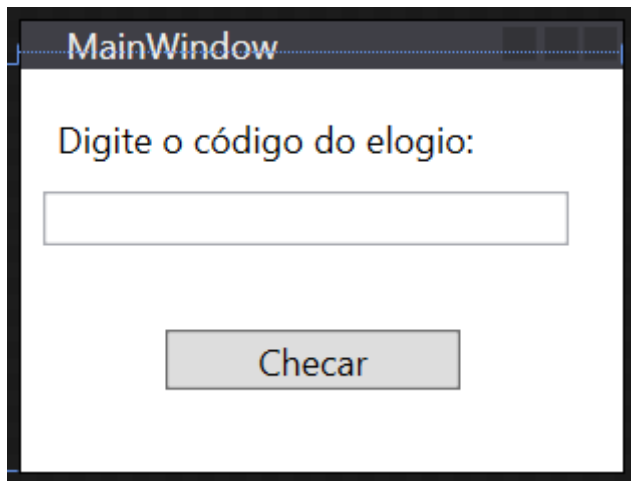
Para resolver esse problema existe a estrutura de seleção chamada **switch**.

Com o switch, podemos **checar em qual caso o valor de uma variável se encaixa**.

O switch é indicado apenas para situações em que conhecemos de antemão todos os casos possíveis de valor para uma variável, sem considerarmos intervalos.

Crie um programa que permita ao usuário escolher o elogio que gostaria de receber, de acordo com a seguinte regra: 1 – Elogio profissional, 2 – Elogio físico, 3 – Elogio pessoal.

Caso digite qualquer outro número, deve ser exibida a mensagem “opção inválida”.



```
private void btnLogin_Click_1(object sender, RoutedEventArgs e)
{
    int codigo;
    string mensagem;

    codigo = Convert.ToInt32(txtCodigo.Text);

    switch (codigo)
    {
        case 1:
            mensagem = "Você escolheu a opção ELOGIO PROFISSIONAL";
            break;
        case 2:
            mensagem = "Você escolheu a opção ELOGIO FÍSICO";
            break;
        case 3:
            mensagem = "Você escolheu a opção ELOGIO PESSOAL";
            break;
        default:
            mensagem = "Opção Inválida!";
            break;
    }

    MessageBox.Show(mensagem, "Elogios", MessageBoxButton.OK, MessageBoxImage.Exclamation);
}
```



Nosso switch monitora a variável “opcao” e exibe diferentes mensagens caso o valor dessa variável seja 1, 2 ou 3.

Caso tenha qualquer outro valor, será exibida a mensagem do bloco default.



Momento Hands On



- Dados três valores X, Y e Z, verificar se eles podem ser os comprimentos dos lados de um triângulo, e se forem, verificar se é um triângulo equilátero, isósceles ou escaleno. Se eles não formarem um triângulo, escrever uma mensagem.

Antes da elaboração do programa, torna-se necessário a revisão de algumas propriedades e definições.

Propriedade – o comprimento de cada lado de um triângulo é menor do que a soma dos comprimentos dos outros dois lados.

Definição 1 - chama-se de triângulo equilátero o que tem os comprimentos dos três lados iguais;

Definição 2 - chama-se de triângulo isósceles o triângulo que tem os comprimentos de dois lados iguais;

Definição 3 - chama-se triângulo escaleno o triângulo que tem os comprimentos dos três lados diferentes.

Utilizando a estrutura switch case, elabore um programa que leia um número referente a um mês do ano e exiba o nome do mês. Se for digitado um valor que não corresponda a um mês válido, exiba uma mensagem indicando tal situação.

Para exercitar mais



Utilizando a estrutura switch case, elabore um programa que leia um valor referente a uma transação bancária e o código de uma transação:

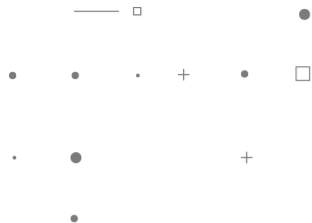
1 – Depósito

2 – Saque

3 – Ver saldo

Lembre-se: o saldo começa com 0 (zero). O programa deve controlar o valor do saldo a cada depósito ou saque.

A conta pode ficar negativa



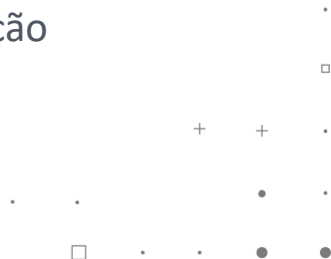
Elabore um programa que leia 2 notas escolares. Calcule a média

Leia também a quantidade de aulas dadas e a quantidade de aulas assistidas pelo aluno e calcule sua frequência :

$$(aulasAssistidas/aulasDadas)*100$$

O aluno somente estará aprovado com média maior ou igual a 7 e frequência mínima de 75%

Exiba a média, a frequência e a mensagem informando sua situação



Lembre-se de versionar seu projeto



OBRIGADO



profalex.deus@fiap.com.br



linkedin.com/in/alexanderresende

FIAP MBA⁺

Copyright © 2019 | Professor (a) Nome do Professor

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.

FIAP