



Nexaas

TECNOLOGIA DA INFORMAÇÃO
Lógica de Programação

Paulo S R Rios
paulinhorios@gmail.com



Nexaas

Lógica de Programação

Desvios Condicionais: Em Lógica de Programação o desvio condicional é o ponto onde uma determinada proposição é colocada a prova, ou seja, verifica-se se determinada condição é verdadeira ou falsa.

Neste ponto o programa tem alternativas possíveis para decidir qual opção adequada para determinada situação.



Desvios Condicionais Simples: O ponto onde o programa deve verificar se deve ou não executar uma determinada instrução.

Nesse caso o programa se limita a executar ou não uma determinada instrução. No caso verdadeiro a instrução é executada, em caso falso a instrução é ignorada.



Nexaas

Lógica de Programação

```
programa {  
  funcao inicio() {  
    inteiro confirma  
    escreva("Digite 1 para a ler a mensagem secreta ou 0 para ignorar: ")  
    leia(numero)  
    se (confirma == 1)  
    {  
      escreva("A Mensagem secreta é NEXAAS")  
    }  
    escreva("Fim do programa")  
  }  
}
```



Desvios Condicionais Composto: Funciona exatamente como o desvio condicional simples, porém comandos podem ser executados tanto na condição verdadeira quanto na condição falsa.

Neste tipo de desvio condicional podemos direcionar a execução de acordo com a finalidade desejada.

Um exemplo simples é um verificador de idade.



Nexaas

Lógica de Programação

```
programa {  
  funcao inicio() {  
    inteiro idade  
    escreva("Digite a idade: ")  
    leia(idade)  
    se (idade > 17)  
    { escreva("Acesso permitido") }  
    senao  
    { escreva("Acesso Negado") }  
  }  
}
```



Operadores Relacionais: Operadores relacionais são utilizados para comparar valores, o resultado de uma expressão é um valor booleano (VERDADEIRO ou FALSO). Os operadores relacionais são: igual, diferente, maior, menor, maior ou igual, menor ou igual.

* algumas linguagens podem utilizar o igual e diferente com sintaxes diferentes, mas a funcionalidade é a mesma

== igual

<> diferente

> maior

< menor

>= maior ou igual

<= menor ou igual



Nexaas

Lógica de Programação

Exercício 1: Faça um programa em Portugol onde o usuário precisa acessar um determinado conteúdo destinado ao público adulto (18 anos), utilize uma estrutura de desvio condicional composto.



Nexaas

Lógica de Programação

```
programa {  
funcao inicio() {  
    inteiro idade  
    escreva("Digite a idade: ")  
    leia(idade)  
    se (idade >= 18)  
    { escreva("Acesso permitido") }  
    senao  
    { escreva("Acesso Negado") }  
}  
}
```



```
Programa {  
funcao inicio() {  
    inteiro numero  
    escreva("Digite um número inteiro entre 1 e 10: ")  
    leia(numero)  
    se(numero = 10) // Verifica se o número é positivo  
    { escreva("O número é positivo") }  
    senao se(numero < 0) // Verifica se o número é negativo  
    { escreva("O número é negativo") }  
    senao // Se não é positivo nem negativo, só pode ser igual a zero  
    { escreva("O número é igual zero") }  
    escreva("\n") } }
```



Operadores Lógicos: Operadores lógicos verifica sentenças lógicas dentro da condição definida no desvio condicional. Serve muitas vezes para mais de uma variável dentro do desvio condicional.

Um exemplo bastante comum é a validação de Usuário E Senha.

Dentre os operadores lógicos temos E e o OU que são os mais utilizados. Comumente reconhecido como AND e OR em linguagens de programação.



Nexaas

Lógica de Programação

```
programa {  
funcao inicio() {  
    cadeia usuario, senha, v_usuario, v_senha  
    v_usuario = "admin"  
    v_senha = "mudar123"  
    escreva("Digite o usuario: ")  
    leia(usuario)  
    escreva("Digite a senha: ")  
    leia(senha)  
    se (usuario == v_usuario e senha == v_senha)  
    { escreva("Acesso permitido") }  
    senao  
    { escreva("Acesso Negado") }    }    }
```



Exercício 2: Faça um algoritmo em Portugol onde o usuário deve informar a idade e a verificação de idade para votar é feita exibindo a mensagem de acordo com a regra abaixo:

Abaixo de 17 anos → Mensagem = Não pode Votar

17 anos → Mensagem = Voto Facultativo

18 anos até 70 → Voto Obrigatório

Acima de 70 anos → Voto Facultativo



Nexaas

Lógica de Programação

Exercício 3: Faça um programa em Portugol onde o usuário informa o valor do produto e o código da tabela e o sistema deve conceder o desconto conforme a tabela abaixo:

TABELA 1 = 10%

TABELA 2 = 15%

TABELA 3 = 20%

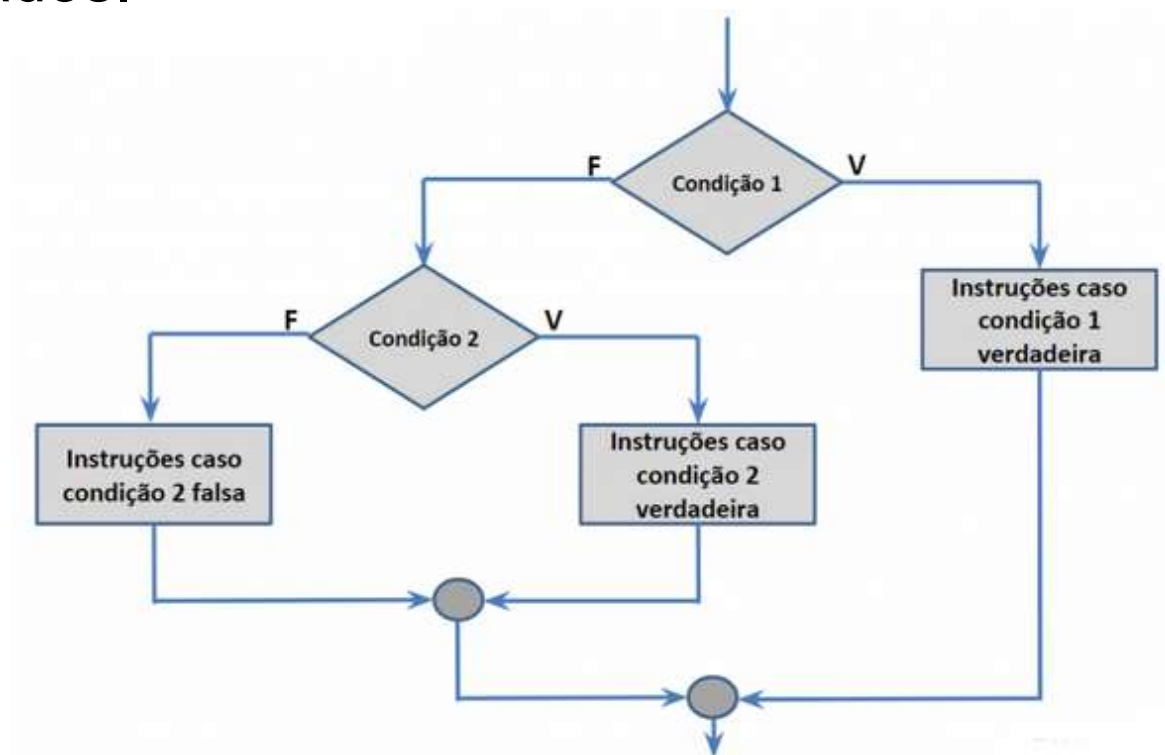
TABELA 4 = 30%

TABELA 5 = 50%

Exemplo: O Usuário informa o valor de R\$500,00 e o código da tabela 1, valor final do produto exibido = R\$450,00



Desvios Condicionais aninhados: Quando precisamos fazer verificações em sequencia podemos ter situações onde precisamos colocar um desvio condicional dentro de outro, nesse caso chamamos de desvios condicionais aninhados.



**Exercício 4:** (upgrade)

O programa deve perguntar inicialmente a senha de gerente (senha = geren01). Se a senha estiver correta deve usar a tabela de gerente, senão deve utilizar a tabela padrão de Funcionários

GERENTE

TABELA 1 = 30%

TABELA 2 = 40%

TABELA 3 = 50%

TABELA 4 = 60%

TABELA 5 = 80%

TABELA 6 = 100%

FUNCIONÁRIO

TABELA 1 = 5%

TABELA 2 = 10%

TABELA 3 = 15%

TABELA 4 = 20%

TABELA 5 = 25%



Exercício 5: O IMC – Índice de Massa Corporal é um critério da Organização Mundial de Saúde para dar uma indicação sobre a condição de peso de uma pessoa adulta.

A fórmula é $IMC = \text{peso} / (\text{altura})^2$

Elabore um algoritmo em Portugol que leia o peso e a altura de um adulto e mostre sua condição de acordo com a tabela abaixo:

IMC em adultos Condição

Abaixo de 18,5 Abaixo do peso

Entre 18,5 e 25 Peso normal

Entre 25 e 30 Acima do peso

Acima de 30 obeso



Estrutura Escolha Caso: Utilizado como uma solução elegante quando se tem várias estruturas de decisão (se-então-senão) aninhadas. Isto é, quando outras verificações são feitas caso a anterior tenha falhado.

Isto permite direcionar ao bloco de código apropriado para a opção correta, dependendo da variável utilizada na verificação da estrutura Escolha-Caso.



Nexaas

Lógica de Programação

```
escreva("Escolha uma opção: ")
leia(opcao)
escolha (opcao)
{
    caso 1:
        escreva ("Instruções da Opção 1")
        pare
    caso 2:
        escreva ("Instruções da Opção 2")
        pare
    caso 3:
        escreva ("Instruções da Opção 3")
        pare
    caso contrario:
        escreva ("Opção Inválida !")
}
```



Exercício 6: Faça um algoritmo em Portugol onde o usuário deve informar o número da opção de acordo com a tabela abaixo e realize a operação entre dois número informado pelo usuário.

Opção:

- 1 – Realizar uma Soma
- 2 – Realizar uma Subtração
- 3 – Realizar uma Divisão
- 4 – Realizar uma Multiplicação



Exercício 7: (upgrade)

Atualize o algoritmo do exercício 4 com estrutura

Escolha – Caso

O programa deve perguntar inicialmente a senha de gerente (senha = geren01). Se a senha esteja correta deve usar a tabela de gerente, senão deve utilizar a tabela padrão de Funcionários

SE operador = GERENTE

caso TABELA 1 = 30%

caso TABELA 2 = 40%

caso TABELA 3 = 50%

caso TABELA 4 = 60%

caso TABELA 5 = 80%

caso TABELA 6 = 100%

SE operador = FUNCIONÁRIO

caso TABELA 1 = 5%

caso TABELA 2 = 10%

caso TABELA 3 = 15%

caso TABELA 4 = 20%

caso TABELA 5 = 25%



Exercício 8: Alguns jogos eletrônicos estilo RPG utilizam estruturas de decisão (se...então ou escolha...caso) para direcionar a história de acordo com as escolhas feitas pelo jogador.

Exemplo: *“No início da jornada você encontra uma carroça tombada na beira da estrada”:*

- 1 – Vasculhar Destroços
- 2 – Procurar Rastros
- 3 – Continuar a Viagem

Caso o Jogador escolha a opção: 1 – Vasculhar Destroços a História Continua dentro dessa estrutura de decisão escolhida;

“Ao começar a revirar os destroços algumas vítimas são encontradas sem vida e barulho de cavalos se aproximam rapidamente”:

- 1 – Esconder-se
- 2 – Sinalizar por Ajuda
- 3 – Fugir do Local



Nexaas

Lógica de Programação

Exercício 8: Faça um Algoritmo em Portugol que simule um jogo de RPG com até 4 níveis (4 escolhas) de decisão que tenha finais bons ou ruins de acordo com as escolhas feita pelo jogador.

Utilize modelos de fluxograma para facilitar a modelagem da sua história de aventura de acordo com as estruturas de decisão a serem utilizadas.



Exercício 9: Faça um algoritmo em Portugol onde o usuário deve informar o valor do serviço e o código do município para que o valor do ISS seja aplicado conforme as tabelas abaixo. (Utiliza a estrutura escolha caso)

1 - PRESTADORES DE SERVIÇO AUTÔNOMOS

- 1 - Angra dos Reis = 2%
- 2 - Volta Redonda = 5%
- 3 - Rio de Janeiro = 4%
- 4- Barra Mansa = 3%

2 - PROFESSORES E OUTRA ATIVIDADES DE ENSINO

- 1 - Angra dos Reis = isento
- 2- Volta Redonda = 2%
- 3- Rio de Janeiro = isento
- 4 - Barra Mansa = 2%