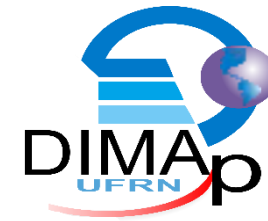




UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA

DIM0320 Algoritmos e Programação de Computadores



#CONTROLE DE FLUXO #CONDICIONAIS #EXPRESSÕES LÓGICAS

ELIEZIO SOARES

ELIEZIOSOARES@DIMAP.UFRN.BR

Expressões Relacionais

- Expressões relacionais são comparações entre dois valores do mesmo tipo.
- Os valores podem ser constantes, variáveis ou expressões aritméticas.
- Possui apenas um dentre dois resultados possíveis:
 - Verdadeiro
 - Falso

Expressões Relacionais

- Igual

- $A == B$
- $2 == 2$
- $A == 2$
- $(5-1) == 4$

- Diferente

- $A != B$
- $5 != 3$
- $A != 2$
- $(5-1) != 4$

- Maior que

- $A > B$
- $3 > 2$

- Menor que

- $B < A$
- $2 < 3$

- Maior ou igual a

- $A >= B$
- $5 >= 5$
- $(7-1) >= 4$

- Menor ou igual a

- $A <= B$
- $3 <= 2$

Expressões Lógicas (booleanas)

- Operadores lógicos:
 - and (Conjunção)
 - or (Disjunção)
 - not (Negação)

Conjunções

- Imagine uma entrevista com 4 participantes e duas perguntas.
- As perguntas podem ser respondidas com sim OU não.
- As respostas não podem ser de outro tipo e nunca sim e não ao mesmo tempo.
- Sim = 1
- Não = 0
 - PERGUNTA 1: Você conhece a linguagem C#?
 - PERGUNTA 2: Você conhece a linguagem Java?
- Somente candidatos que conheçam a linguagem C# **E (and)** a linguagem Java serão selecionados.
 - Ou seja: somente candidatos que conheçam as duas linguagens serão aprovados.

Conjunções

- PERGUNTA 1: Você conhece a linguagem C#?
- PERGUNTA 2: Você conhece a linguagem Java?
- O operador AND só considera a expressão como verdadeira se todas as expressões testadas forem verdadeiras.

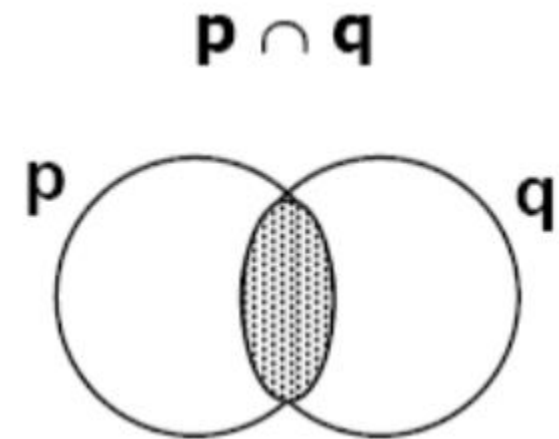
| Candidato | Pergunta 1 | Pergunta 2 | Resultado |
|--------------------|------------|------------|-----------|
| Candidato 1 | 0 | 0 | 0 |
| Candidato 2 | 0 | 1 | 0 |
| Candidato 3 | 1 | 0 | 0 |
| Candidato 4 | 1 | 1 | 1 |

APROVADO

Conjunções – Tabela Verdade

- A expressão lógica tem seu valor lógico verdadeiro somente quando as duas proposições forem verdadeiras;

| p | q | p && q |
|---|---|--------|
| V | V | V |
| V | F | F |
| F | V | F |
| F | F | F |



Disjunções

- Imagine uma entrevista com 4 participantes e duas perguntas.
- As perguntas podem ser respondidas com sim OU não.
- As respostas não podem ser de outro tipo e nunca sim e não ao mesmo tempo.
- Sim = 1
- Não = 0
 - PERGUNTA 1: Você conhece a linguagem C#?
 - PERGUNTA 2: Você conhece a linguagem Java?
- Somente candidatos que conheçam a linguagem C# **OU (or)** a linguagem Java serão selecionados.
 - Ou seja: qualquer candidato que conheça uma das duas linguagens serão aprovados.

Disjunções

- PERGUNTA 1: Você conhece a linguagem C#?
- PERGUNTA 2: Você conhece a linguagem Java?
- O operador OR considera a expressão como verdadeira se pelo menos uma das expressões testadas for verdadeira.

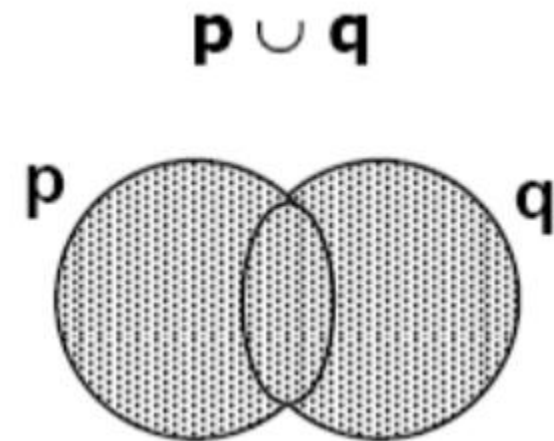
| Candidato | Pergunta 1 | Pergunta 2 | Resultado |
|--------------------|------------|------------|-----------|
| Candidato 1 | 0 | 0 | 0 |
| Candidato 2 | 0 | 1 | 1 |
| Candidato 3 | 1 | 0 | 1 |
| Candidato 4 | 1 | 1 | 1 |

APROVADOS

Disjunções – Tabela Verdade

- A proposição composta p OR q tem seu valor lógico verdadeiro quando p for verdadeira ou quando q for verdadeira;

| p | q | $p \vee q$ |
|-----|-----|------------|
| V | V | V |
| V | F | V |
| F | V | V |
| F | F | F |



Negação - not

- O operador NOT é aplicado a uma proposição e tem o efeito de inverter seu valor lógico.
- Exemplo:
 - (p) Ele trabalha.
 - Ele não trabalha.

 - ! verdadeiro = FALSO
 - ! falso = VERDADEIRO

Negação – Tabela Verdade

- A negação inverte o valor lógico da proposição original.

| p | not p |
|---|-------|
| V | F |
| F | V |

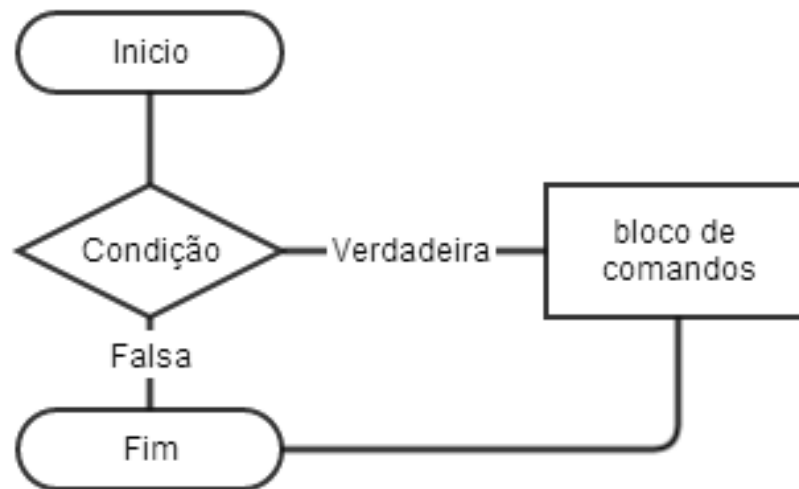
Estruturas Condicionais

- São estruturas de possibilitam a escolha de um grupo de ações a serem executadas quando determinadas condições são ou não são satisfeitas.
- Utilizadas nas tomadas de decisões.
- Podem ser **simples** ou **compostas**.



Estruturas Condicionais Simples

- Executa um bloco de comandos se a condição for verdadeira. Se a condição for falsa, a estrutura é finalizada sem executar o bloco de comandos.
- O comando que define essa estrutura é representado pela palavra **IF**



If (Sintaxe Python)

- If (condicional simples)

```
If expressão-lógica :  
    # bloco de código que será executado  
    # Atenção a indentação  
    # Tudo o que estiver recuado  
    # Será executado
```

- Expressão **deve ser booleana**.

If (Sintaxe C)

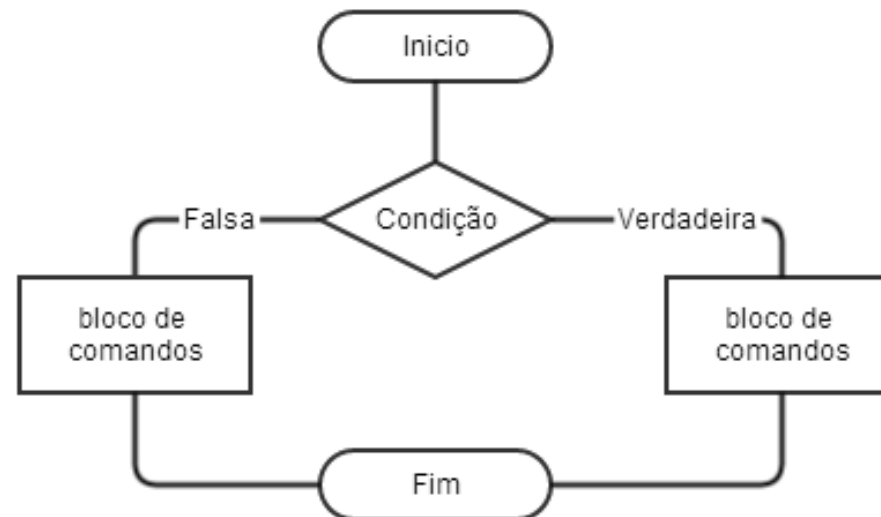
- Abra um novo arquivo python e implemente o seguinte trecho de código:

```
a = int(input("Digite um número: "))
b = int(input("Digite um outro número: "))
if a>b:
    print("O primeiro número é o maior.")
if b>a:
    print("O segundo número é o maior.")
```

- Execute o programa quantas vezes desejar, alterando os valores de entrada.

Estruturas Condicionais Compostas

- Segue o mesmo princípio da estrutura condicional simples, com a diferença de que quando a condição não for satisfeita, será executado outro bloco de comandos.
- O comando que define essa estrutura é representado pelas palavras **IF** e **ELSE**



If – else (Sintaxe Python)

- If – else (condicional composto)

```
if expressão-lógica :  
    # bloco de código que será executado  
    # Atenção a indentação  
    # Tudo o que estiver recuado  
    # Será executado  
else:  
    # bloco de código que será executado  
    # Atenção a indentação  
    # Tudo o que estiver recuado  
    # Será executado
```

- Else é **opcional**.
- Expressão **deve ser booleana**.

If – else (Sintaxe Python)

- Abra um novo arquivo python e implemente o seguinte trecho de código:

```
a = int(input("Digite um número: "))  
if a>0:  
    print("O número digitado é positivo.")  
else:  
    print("O número digitado é negativo.")
```

- Execute o programa quantas vezes desejar, alterando os valores de entrada.
 - Sugestão: Execute a primeira vez digitando 10 e a segunda vez digitando -1.

If – else (Sintaxe Python)

Operador and

- Abra um novo arquivo python e implemente o seguinte trecho de código:

```
a = int(input("Digite um número: "))
if a>10 and a<20:
    print("O número digitado está entre 10 e 20.")
else:
    print("O número digitado NÃO está entre 10 e 20.")
```

- Execute o programa quantas vezes desejar, alterando os valores de entrada.

If – else (Sintaxe Python)

Operador or

- Abra um novo arquivo python e implemente o seguinte trecho de código:

```
a = int(input("Digite um número: "))
if a==0 or a==1:
    print("Você digitou um binário.")
else:
    print("Não é um binário...")
```

- Execute o programa quantas vezes desejar, alterando os valores de entrada.

Exercício

- Construa um programa que leia a categoria de um produto e determine o preço, conforme a tabela abaixo:

| Categoria | Preço |
|-----------|-------|
| 1 | 10,00 |
| 2 | 18,00 |
| 3 | 23,00 |
| 4 | 26,00 |
| 5 | 31,00 |

Vários e INEFICIENTES “ifs”

- O seu programa, certamente, precisou de algo parecido com isso:

```
categoria = int(input("Informe a categoria do produto: "))
preco = 0
if categoria == 1:
    preco = 10
if categoria == 2:
    preco = 18
if categoria == 3:
    preco = 23
if categoria == 4:
    preco = 26
if categoria == 5:
    preco = 31

print("O valor do produto é %3.2f" %preco)
```

Ou...

Estruturas aninhadas

- Essas estruturas aninhadas são mais eficientes, porém deselegantes e desorganizadas.

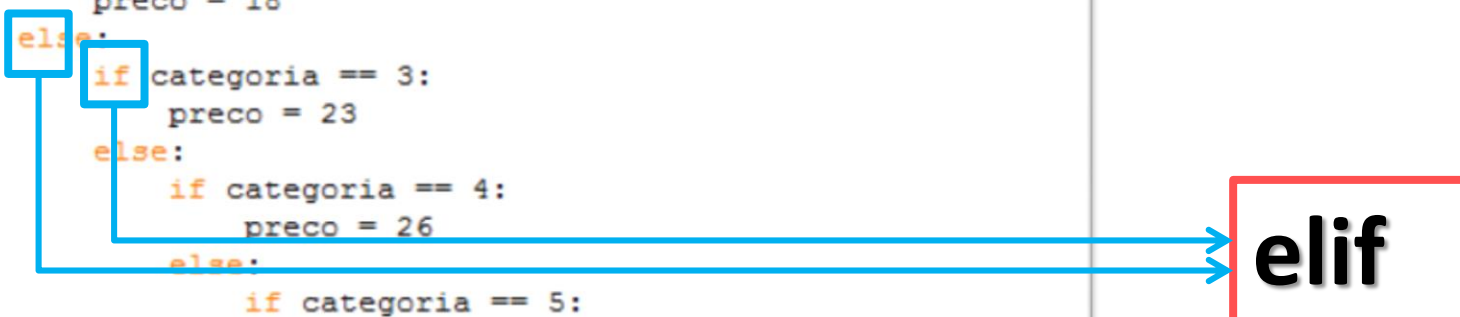
```
categoria = int(input("Informe a categoria do produto: "))
preco = 0
if categoria == 1:
    preco = 10
else:
    if categoria == 2:
        preco = 18
    else:
        if categoria == 3:
            preco = 23
        else:
            if categoria == 4:
                preco = 26
            else:
                if categoria == 5:
                    preco = 31

print("O valor do produto é %3.2f" %preco)
```


Solução: Elif

- Python apresenta uma solução para esse problema com um comando que concatena as funções else e if:

```
categoria = int(input("Informe a categoria do produto: "))
preco = 0
if categoria == 1:
    preco = 10
else:
    if categoria == 2:
        preco = 18
    elif categoria == 3:
        preco = 23
    else:
        if categoria == 4:
            preco = 26
        else:
            if categoria == 5:
                preco = 31
print("O valor do produto é %3.2f" %preco)
```



elif

If – elif

- If – elif

```
if expressão-lógica :  
    # bloco de código que será executado  
    # Atenção a indentação  
    # Tudo o que estiver recuado  
    # Será executado  
elif expressão-lógica:  
    # bloco de código que será executado  
    # Atenção a indentação  
    # Tudo o que estiver recuado  
    # Será executado
```

- If – elif - else

```
if expressão-lógica :  
    # bloco de código que será executado  
    # Atenção a indentação  
elif expressão-lógica:  
    # bloco de código que será executado  
    # Atenção a indentação  
else:  
    # bloco de código que será executado  
    # Atenção a indentação
```

Solução: Elif

- A cláusula “elif” resolve o problema da eficiência e da elegância / organização.

```
categoria = int(input("Informe a categoria do produto: "))
preco = 0
if categoria == 1:
    preco = 10
elif categoria == 2:
    preco = 18
elif categoria == 3:
    preco = 23
elif categoria == 4:
    preco = 26
elif categoria == 5:
    preco = 31

print("O valor do produto é %3.2f" %preco)
```

Dúvidas



Exercícios

1. Construa um programa que leia dois números e pergunte ao usuário qual operação ele deseja executar: soma, subtração, multiplicação ou divisão. Após a operação o programa deve exibir o resultado do processamento.
2. Construa um programa bancário para aprovação de empréstimos. O programa deve receber o valor do empréstimo solicitado, o salário do cliente e a quantidade de meses para se pagar o empréstimo. O valor da prestação mensal não pode ultrapassar 30% do salário. O programa deve considerar o valor da prestação como sendo o valor solicitado dividido pela quantidade de meses.

Exercícios

3. Construa um programa que calcule o preço a pagar pela energia elétrica. O programa deve receber a quantidade de KWh consumidas e o tipo de instalação (R para residências; I para indústrias; C para comércios). O programa deve calcular o valor conforme a tabela abaixo:

| Tipo | Faixa (KWh) | Preço (R\$) |
|-------------|---------------|-------------|
| Residencial | Até 500 | 0,40 |
| | Acima de 500 | 0,65 |
| Comercial | Até 1000 | 0,55 |
| | Acima de 1000 | 0,60 |
| Industrial | Até 5000 | 0,55 |
| | Acima de 5000 | 0,60 |

Exercícios

4. Desenvolva um programa que leia a altura (em metros) e o peso (em quilogramas) e calcule o IMC - Índice de Massa Corporal do usuário e informe sua situação corporal conforme tabela abaixo. O cálculo do IMC é feito dividindo-se o peso pela altura ao quadrado. Sabe-se ainda que a tabela abaixo é válida apenas para pessoas acima dos 15 anos de idade, então o programa deverá invalidar os cálculos que fujam dessa regra.

| RESULTADO | SITUAÇÃO |
|---------------------------|-------------------------|
| Abaixo de 17 | Muito abaixo do peso |
| Entre 17 e 18,49 | Abaixo do peso |
| Entre 18,5 e 24,99 | Peso normal |
| Entre 25 e 29,99 | Acima do peso |
| Entre 30 e 34,99 | Obesidade I |
| Entre 35 e 39,99 | Obesidade II (severa) |
| Acima de 40 | Obesidade III (mórbida) |