

# ALGORITMOS E PROGRAMAÇÃO (PRÁTICA)

Profa. Andreza Leite  
[andrezza.leite@univasf.edu.br](mailto:andrezza.leite@univasf.edu.br)

# Estruturas de Controle de Fluxo

- Os algoritmos desenvolvidos até o momento constituem uma seqüência de ações que sempre são executadas em sua totalidade indiferente do valor da entrada de dados.
- Para a resolução de determinados problemas ou para a execução de determinadas tarefas é necessária a realização de um conjunto distinto de ações e este conjunto é definido com base na análise da entrada de dados.
- Um exemplo simples de uma destas situações é um algoritmo capaz de efetuar o cálculo do imposto de renda devido por um determinado contribuinte. Neste caso dependendo da quantidade de dependentes, do valor de sua renda e outras fatores o cálculo será feito de formas distintas.

# Instrução Condicional

- Estas instruções permitem que o programador especifique a sequência de instruções que será executada.
- Considere um problema que exija uma decisão.
- Sintaxe:

```
se (<expressão-lógica>) então
    <sequência de comandos>
senão
    <sequência de comandos>
fimse
```
- Tomemos como exemplo uma divisão, onde haja a necessidade de que o algoritmo verifique se o divisor é igual ou diferente de zero. Se for igual não é possível dividir. Se for diferente é possível dividir.

# Instrução Condicional

```
algoritmo "divisão - instrução condicional"  
var  
n1, n2: inteiro  
resultado: real  
  
Inicio  
escreva ("Digite o dividendo:")  
leia (n1)  
escreva ("Digite o divisor:")  
leia (n2)  
    se (n2=0) entao  
        escreva ("Impossível dividir por zero")  
    senao  
        resultado <- n1/n2  
        escreva ("O resultado é:", resultado)  
    fimse  
finalgoritmo
```

# Instrução Condicional



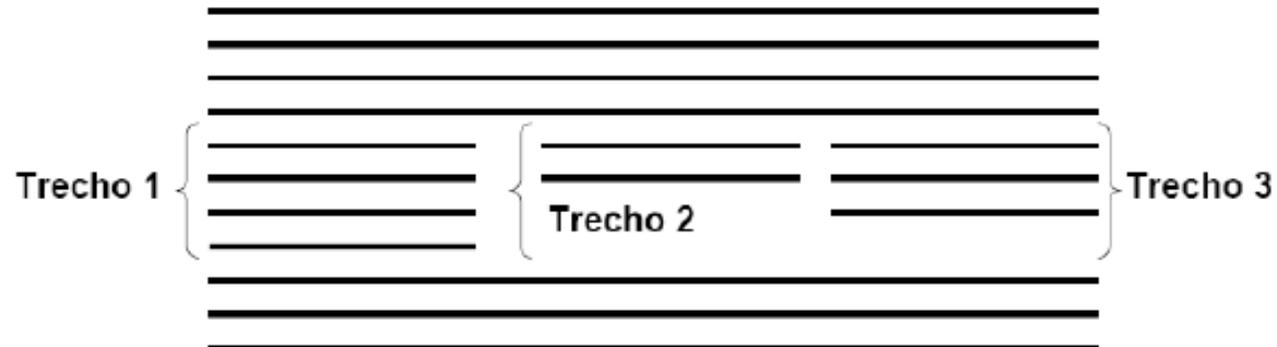
```
Digite o dividendo:6  
Digite o divisor:0  
Impossível dividir por zero  
*** Fim da execução.  
*** Feche esta janela para retornar ao Visualg.
```



```
Digite o dividendo:30  
Digite o divisor:4  
O resultado eh: 7.5  
*** Fim da execução.  
*** Feche esta janela para retornar ao Visualg.
```

# Comando de Seleção Múltipla

- Em algumas situações ao chegarmos a uma determinada instrução de um algoritmo devemos selecionar um dentre alguns trechos a seguir, tendo como base para esta escolha um conjunto de valores.



- Para lidar com casos deste tipo foi criado o comando de seleção múltipla.

# Comando de Seleção Múltipla

- Sintaxe: Comando de seleção múltipla

**escolha** (<variável>)

**caso** <valor11>, <valor12>, ..., <valor1n>  
    <seqüência-de-comandos-1>

**caso** <valor21>, <valor22>, ..., <valor2m>  
    <seqüência-de-comandos-2>

...

**outrocaso**

    <seqüência-de-comandos-extra>

**fimescolha**

# Comando de Seleção Múltipla

```
algoritmo "Times - seleção múltipla"
var
    time: caractere
Inicio
escreval ("Digite um nome de time de futebol:")
leia (time)
escolha (time)
    caso "Sport", "Santa Cruz", "Nautico", "Petrolina"
        escreval ("É um time Pernambucano")
    caso "Vitória", "Bahia", "Camaçari", "Juazeiro"
        escreval ("É um time Baiano")
    outrocaso
        escreval ("É de outro Estado")
fimescolha
fimalgoritmo
```

# Exemplo:

```
Digite um nome de time de futebol:  
Petrolina  
É um time Pernambucano  
  
*** Fim da execução.  
*** Feche esta janela
```

```
Digite um nome de time de futebol:  
Juazeiro  
É um time Baiano  
  
... para retornar ao Visualg.
```

```
Digite um nome de time de futebol:  
Flamengo  
É de outro Estado  
  
*** Fim da execução.  
*** Feche esta janela para retornar ao Visualg.
```

# Exercício



Escreva algoritmos que:

- 1) Receba do usuário um número entre 1 e 7, inclusive 1 e 7. Se ele digitar o número 1 mostre “Hoje é Domingo”, se ele digitar o número 2 mostre “Hoje é Segunda”.....
- 2) Peça uma letra e mostre se ela é vogal ou consoante.
- 3) Peça três números e mostre o maior entre eles.