

# Processamento de Dados aplicado à Geociências

## AULA 7: Comandos de repetição Enquanto-faça

UNIVERSIDADE FEDERAL DE PELOTAS  
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO  
CURSO SUPERIOR DE TECNOLOGIA EM  
GEOPROCESSAMENTO

Professor: Guilherme Tomaschewski Netto  
[guilherme.netto@inf.ufpel.edu.br](mailto:guilherme.netto@inf.ufpel.edu.br)



# Problema

- Calcular o reajuste dos 50 produtos vendidos por um empresa de modo que os produtos com valores inferiores a R\$ 100,00 tenham um reajuste de 10% e os demais tenham um reajuste de 3%. Para cada produto, o usuário deve informar o código e valor atual de cada produto e o algoritmo deve mostrar o valor reajustado, juntamente com o código do produto. Além disso, no final deverá ser mostrado o valor total reajustado.

# Uma solução...

Algoritmo

Declare  $c, v, tot$  Numérico

$tot := 0$

Escreva “Informe o código e o valor”, \n

Leia  $c, v$

Se  $v < 100$

então  $v := v + (v * 0.1)$

senão  $v := v + (v * 0.03)$

fim\_se

Escreva “Código”,  $c$ , “Valor”,  $v$ , \n

$tot := tot + v$

⋮

*repete 48 vezes*

Escreva “Informe o código e o valor”, \n

Leia  $c, v$

Se  $v < 100$

então  $v := v + (v * 0.1)$

senão  $v := v + (v * 0.03)$

fim\_se

Escreva “Código”,  $c$ , “Valor”,  $v$ , \n

$tot := tot + v$

Escreve “Valor total”,  $tot$ , \n

fim\_algoritmo

# Uma solução...

Algoritmo

Declare  $c, v, tot$  Numérico

$tot := 0$

Escreva “Informe o código e o valor”, \n

Leia  $c, v$

Se  $v < 100$

então  $v := v + (v * 0.1)$

senão  $v := v + (v * 0.03)$

fim\_se

Escreva “Código”,  $c$ , “Valor”,  $v$ , \n

$tot := tot + v$

⋮

*repete 48 vezes*

Escreva “Informe o código e o valor”, \n

Leia  $c, v$

Se  $v < 100$

então  $v := v + (v * 0.1)$

senão  $v := v + (v * 0.03)$

fim\_se

Escreva “Código”,  $c$ , “Valor”,  $v$ , \n

$tot := tot + v$

Escreva “Valor total”,  $tot$ , \n

fim\_algoritmo

**NÃO É UMA SOLUÇÃO  
ADEQUADA!**

# Estruturas de Repetição

- Em vários momentos no desenvolvimento de algoritmos, se torna necessário repetir um trecho de um algoritmo um determinado número de vezes.
- Nesse caso, pode ser criado um laço de repetição que efetue um determinado trecho do algoritmo, tantas vezes quantas forem necessárias.

# Estruturas de Repetição

- Permite repetir um conjunto de instruções sem que seja necessário escrevê-las várias vezes.
- Uma sequência de instruções pode ser executada várias vezes, até que uma condição seja satisfeita
- Também são chamadas de Laços ou Loops
- Três tipos:
  - Enquanto-faça
  - Repita-até
  - Para-até-faça

# Características

- Enquanto-faça
  - Repetição com teste no início
- Repita-até
  - Repetição com teste no fim
- Para-até-faça
  - Repetição com variável de controle

# Enquanto-faça

- A estrutura de repetição Enquanto-faça é utilizada quando um conjunto de comandos deve ser executado repetidamente, enquanto uma determinada condição (expressão lógica) permanecer verdadeira.
- Dependendo do resultado do teste da condição, o conjunto de comandos poderá não ser executado nenhuma vez (se for falsa no primeiro teste), ou será executado várias vezes (enquanto for verdadeira).

# Enquanto-faça

- Permite executar uma sequência de instruções enquanto uma condição é verdadeira.
- Sintaxe:

```
Enquanto <condição>  
    faça <lista_comandos>  
fim_enquanto
```

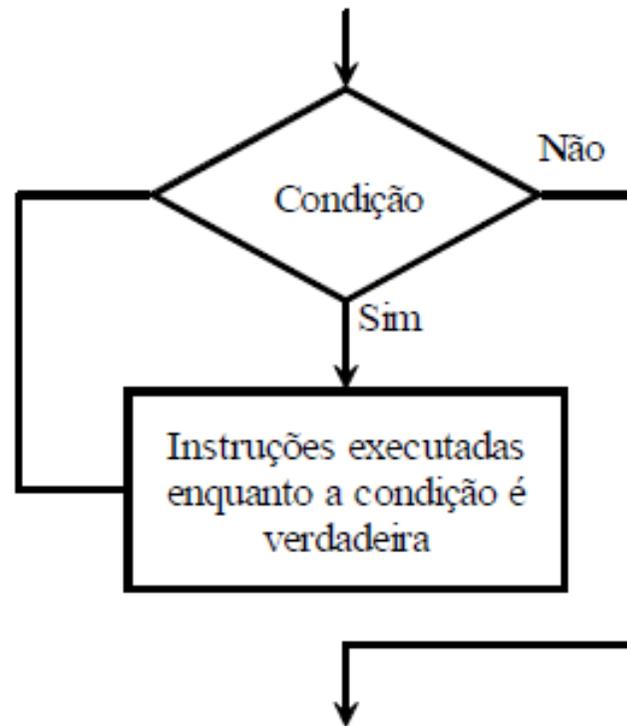
# Semântica

- A execução do comando ocorre da seguinte maneira:
  - Ao início da estrutura Enquanto-faça a condição é testada.
  - Se seu resultado for falso, então a lista de comandos no seu interior não é executada e a execução prossegue normalmente pela instrução seguinte à palavra-reservada fim\_ enquanto que identifica o final da estrutura.
  - Se a condição for verdadeira a lista de comandos é executada e ao seu término retorna-se ao teste da condição. Assim, o processo acima será repetido enquanto a condição testada for verdadeira. Quando esta for falsa, o fluxo de execução prossegue normalmente pela instrução seguinte à palavra-reservada fim\_ enquanto que identifica o final da construção.

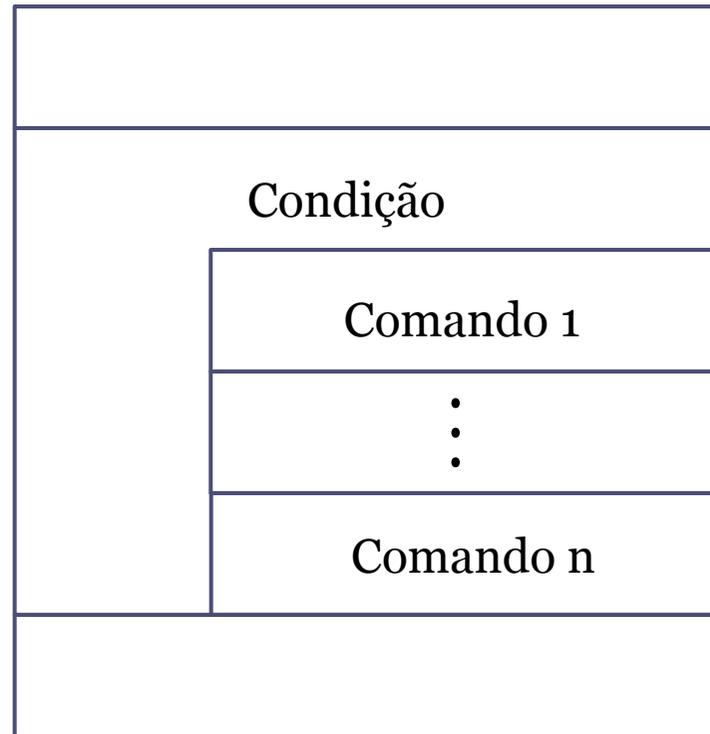
# Semântica

- Uma vez dentro do laço, a execução somente abandonará este laço quando a condição for falsa.
- Deve-se garantir que em algum momento da execução a condição será avaliada como falsa. Caso contrário, o algoritmo permanecerá indefinidamente no interior do laço (laço infinito).

# Fluxograma



# Diagrama de Chapin



# Enquanto-faça

- Solução para o nosso problema usando a estrutura de repetição Enquanto-faça.

# Enquanto-faça

## Algoritmo

```
  Declare c, v, tot, i Numérico
  tot := 0
  i := 1
  Enquanto i <= 50 faça
    Escreva “Informe o código e o valor”, \n
    Leia c, v
    Se v < 100
      então v := v+(v*0.1)
      senão v := v+(v*0.03)
    fim_se
    Escreva “Código”, c, “Valor”, v, \n
    i := i+1
    tot := tot + v
  fim_enquanto
  Escreva “Valor total”, tot, \n
fim_algoritmo
```

# Variável contadora

- A variável  $i$  é quem controla o número de repetições do laço. É chamada variável contadora.
- Uma variável contadora é uma variável que recebe um valor inicial, é incrementada de um valor constante no laço e tem seu valor testado em algum ponto do laço. Ao chegar a um determinado valor o laço é interrompido.
- A inicialização da variável contadora deve ir, necessariamente, fora do laço.

# Variável contadora

- Existem diversas maneiras de implementar o mesmo laço, mas todo laço com variável de controle deve conter:
  - inicialização;
  - incremento (ou decremento);
  - teste de valor final.

# Variável contadora

## Algoritmo

```
Declare c, v, tot, i Numérico
tot := 0
i := 1 /*inicialização*/
Enquanto i <= 50 faça /*teste*/
    Escreva “Informe o código e o valor”, \n
    Leia c, v
    Se v < 100
        então v := v+(v*0.1)
        senão v := v+(v*0.03)
    fim_se
    Escreva “Código”, c, “Valor”, v, \n
    i := i+1 /*incremento*/
    tot := tot + v
fim_enquanto
Escreva “Valor total”, tot, \n
fim_algoritmo
```

# Algoritmo de Fibonacci

- Faça um algoritmo que escreva os 6 primeiros números da série de Fibonacci.

# Algoritmo de Fibonacci Com repetição

Algoritmo

Declare A,B, i Numérico

A:=0

B:=1

i:=1

Enquanto i <= 6 faça

    A:= A+B

    B:= A-B

    Escreva B, \n

    i:= i+1

fim\_enquanto

fim\_algoritmo

# Teste de mesa

Linha	Algoritmo
1	Algoritmo
2	Declare A,B,i Numerico
3	A:=0
4	B:=1
5	i:= 1
6	Enquanto i <= 6 faça
7	A := A+B
8	B := A - B
9	Escreva B, \n
10	i:= i+1
11	fim_enquanto
12	fim_algoritmo

Linha	Teste de mesa			
	A	B	i	Saída
2	?	?	?	
3	0	?	?	
4	0	1	?	
5	0	1	1	
6	0	1	1	
7	1	1	1	
8	1	0	1	
9	1	0	1	0
10	1	0	2	
6	1	0	2	
7	1	0	2	
8	1	1	2	
9	1	1	2	1
10	1	1	3	
6	1	1	3	
7	2	1	3	
8	2	1	3	
9	2	1	3	1
10	2	1	4	
6	2	1	4	

# Teste de mesa

## Continuação

Linha	Algoritmo
1	Algoritmo
2	Declare A,B,i Numerico
3	A:=0
4	B:=1
5	i:= 1
6	Enquanto i <= 6 faça
7	A := A+B
8	B := A - B
9	Escreva B, \n
10	i:= i+1
11	fim_enquanto
12	fim_algoritmo

Linha	Teste de mesa			
	A	B	i	Saída
6	2	1	4	
7	3	1	4	
8	3	2	4	
9	3	2	4	2
10	3	2	5	
6	3	2	5	
7	5	2	5	
8	5	3	5	
9	5	3	5	3
10	5	3	6	
6	5	3	6	
7	8	3	6	
8	8	5	6	
9	8	5	6	5
10	8	5	7	
6	8	5	7	
11	8	5	7	

# Exercício 1

- Escreva um algoritmo que escreva a tabuada do 5 (de 1 à 10).

# Exercício 1 - Solução

## Algoritmo

**Declare** Num, Tabuada **Numérico**

Num:=1

**Enquanto** Num<=10 **faça**

    Tabuada:= Num\*5

**Escreva** 'Tabuada', Num, ' \* 5:', Tabuada, \n

    Num:=Num+1

**fim\_enquanto**

**fim\_algoritmo**

## Exercício 2- Médias

- Faça um algoritmo que, para 10 alunos: leia o nome do aluno e duas notas deste aluno e que escreva o nome do aluno, informando se ele está aprovado caso sua média seja maior ou igual a 7, juntamente com a sua média e reprovado caso contrário.

# Exercício 2- Médias

## Algoritmo

```
Declare Nota1, Nota2, Media, i Numerico
Declare Nome Literal
i:= 1
Enquanto i<=10 faça
    Escreva 'Nome do aluno', \n
    Leia Nome
    Escreva 'Entre com a primeira nota', \n
    Leia Nota1
    Escreva 'Entre com a segunda nota', \n
    Leia Nota2
    Media := (Nota1+Nota2)/2
    Se Media >= 7
        então Escreva 'Aluno', Nome, 'Aprovado com nota', Media,\n
        senão Escreva 'Aluno', Nome, 'Reprovado com nota', Media,\n
    fim_se
    i:=i+1
fim_enquanto
fim_algoritmo
```

## Exercício 3

- Escreva um algoritmo para ler 2 valores e se o segundo valor informado for ZERO, deve ser mostrada uma mensagem de valor inválido e deve ser lido um novo valor (ou seja, para o segundo valor não pode ser aceito o valor zero), e imprimir o resultado da divisão do primeiro valor lido pelo segundo valor lido.

# Exercício 3 - Solução

## Algoritmo

```
Declare Valor1, Valor2, Divisao Numérico
Escreva 'Entre com o primeiro valor', \n
Leia Valor1
Escreva 'Entre com o segundo valor', \n
Leia Valor2
Enquanto Valor2=0 faça
    Escreva 'Valor inválido, entre com um novo valor', \n
    Leia Valor2
fim_enquanto
Divisao:= valor1/valor2
Escreva 'A divisão de', Valor1, 'por', Valor2, 'é', Divisao, \n
fim_algoritmo
```