



Estruturas de Repetição – Parte I

ENQUANTO-FAÇA

Profs. Do CMCC

Centro de Matemática, Computação e
Cognição

Objetivos

- Reconhecer a necessidade do uso de estruturas de repetição
- Aprender a **sintaxe** da Estrutura de Repetição **“ENQUANTO-FAÇA”**
- Aprender a resolver problemas que requeiram o uso de **ESTRUTURAS DE REPETIÇÃO** a partir de seus enunciados (com a aplicação da técnica de interpretação de enunciados)



Universidade Federal do ABC

Problema

(Estudo de Caso)

Problema

Escreva um algoritmo calcular a média da idade de um grupo de 10 pessoas.

- **SAÍDA:** Quais os resultados que o algoritmo dever fornecer ?
- **ENTRADA:** Quais os insumos necessários para se obter a saída?
- **PROCESSAMENTO:** Como transformar os insumos na saída?

Problema

Escreva um algoritmo **calcular a média** da idade de um grupo de **10** pessoas.

- **SAÍDA:** média da idade das 10 pessoas
- **ENTRADA:** a idade das 10 pessoas
(id1, id2, id3,..., id10)
- **PROCESSAMENTO:**
 - $Media = (id1 + id2 + id3 + \dots + id10) / \underline{10}$

Estruturas de Repetição

```
inteiro id1, id2, id3, id4, id5, id6, id7, id8, id9, id10  
real media
```

```
escreva ("Digite a idade da pessoa 1: ")  
leia(id1)  
  
escreva ("Digite a idade da pessoa 2: ")  
leia(id2)  
■  
■  
■  
escreva ("Digite a idade da pessoa 9: ")  
leia(id9)  
escreva ("Digite a idade da pessoa 10: ")  
leia(id10)
```

**Entrada de
Dados**

Processamento

```
media<- (id1+id2+id3+id4+id5+id6+id7+id8+id9+id10)/10
```

```
escreva("A media das idades é: ", media)
```

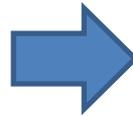
Saída de Dados

Estruturas de Repetição

Processamento

```
media <- (id1+id2+id3+id4+id5+id6+id7+id8+id9+id10) / 10
```

E se fossem 50 pessoas ?
100 pessoas ?
200 pessoas ?



50 variáveis ?
50 “escreva / leia” ?
Média de 50

Padrão de Comportamento

```
inteiro id1, id2, id3, id4, id5, id6, id7, id8, id9, id10  
real media
```

```
escreva ("Digite a idade da pessoa 1: ")  
leia(id1)
```

```
escreva ("Digite a idade da pessoa 2: ")  
leia(id2)
```

```
escreva ("Digite a idade da pessoa 3: ")  
leia(id3)
```

```
escreva ("Digite a idade da pessoa 4: ")  
leia(id4)
```

```
escreva ("Digite a idade da pessoa 5: ")  
leia(id5)
```

```
escreva ("Digite a idade da pessoa 6: ")  
leia(id6)
```

Note que existe um
**“Padrão de
Comportamento”**
nesse algoritmo.

O que muda de
um bloco para
outro?

Apenas o número
da pessoa !!!

Padrão de Comportamento

```
inteiro id1, id2, id3, id4, id5, id6, id7, id8, id9, id10  
real media
```

```
escreva ("Digite a idade da pessoa 1: ")  
leia(id1)
```

```
escreva ("Digite a idade da pessoa 2: ")  
leia(id2)
```

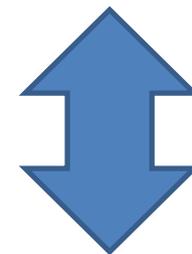
```
escreva ("Digite a idade da pessoa 3: ")  
leia(id3)
```

```
escreva ("Digite a idade da pessoa 4: ")  
leia(id4)
```

```
escreva ("Digite a idade da pessoa 5: ")  
leia(id5)
```

```
escreva ("Digite a idade da pessoa 6: ")  
leia(id6)
```

**Padrão de
Comportamento**



REPETIÇÃO



Universidade Federal do ABC

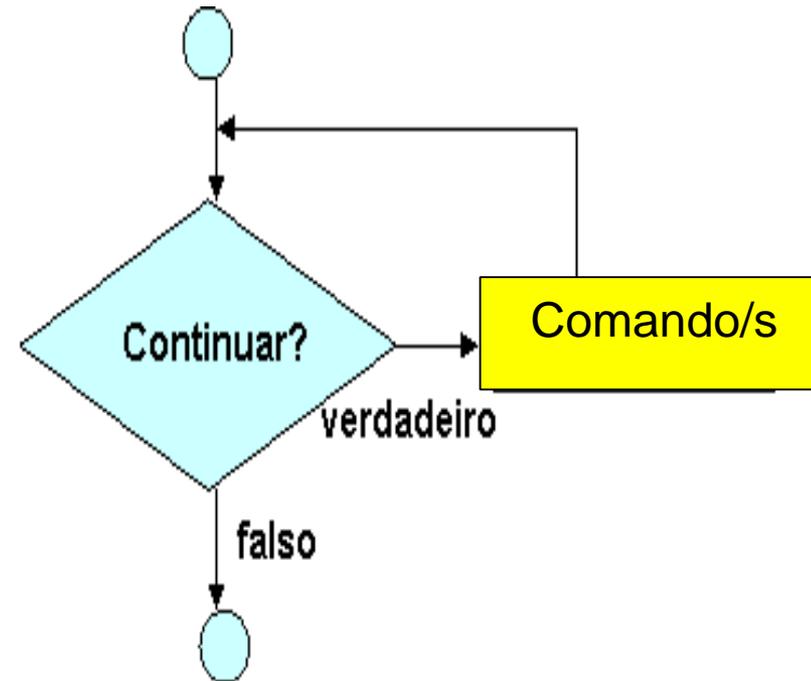
Bacharelado em Ciência e Tecnologia
Processamento da Informação

Estruturas de Repetição

ESTRUTURAS DE REPETIÇÃO

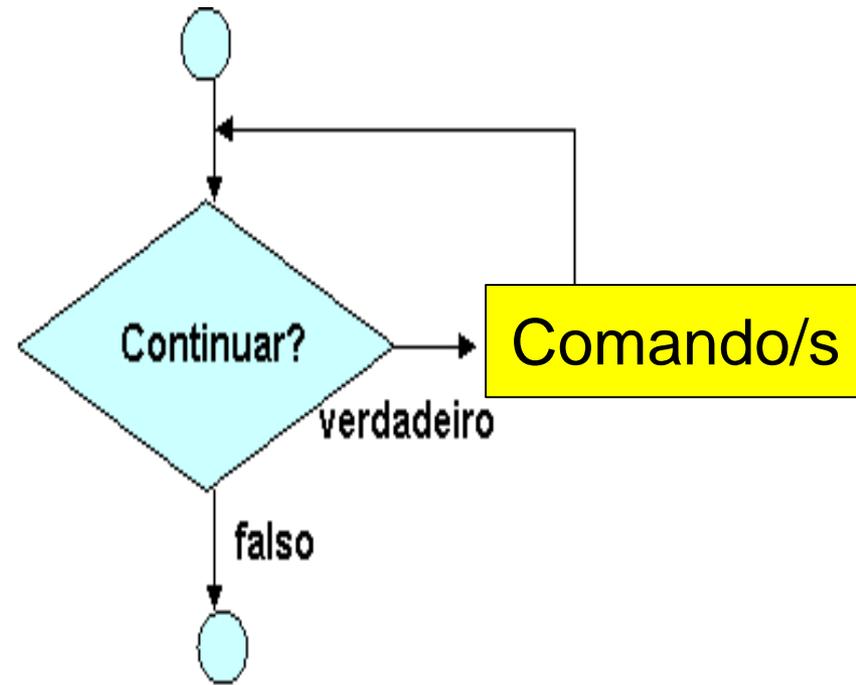
Estruturas de Repetição

- Quando uma seqüência de comandos deve ser executada **repetidas vezes** usamos uma **estrutura de repetição**.
- A estrutura de repetição, assim como a de decisão, envolve sempre a avaliação de uma **condição**.
- Também conhecidas como **laços** de repetição ou ***loops***.



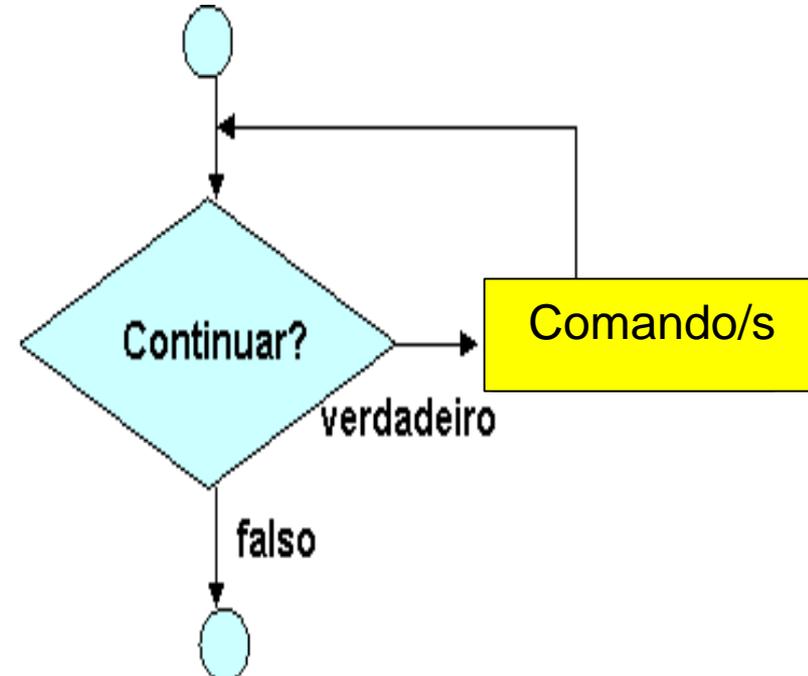
Estrutura de Repetição: ENQUANTO

- A estrutura de repetição **enquanto** permite especificar instruções que devem ser repetidas enquanto determinada condição for verdadeira
- **Exemplo**: enquanto não terminar o arquivo, leia linha e incremente contador.
- O corpo da estrutura **enquanto** pode ser uma instrução ou um bloco (subprograma)



Estrutura de Repetição: ENQUANTO

- Quando a condição se tornar **FALSA**, a ação (ou bloco) da estrutura será pulada.
- O programa continuará com a ação sequencial imediatamente após a estrutura **enquanto**.

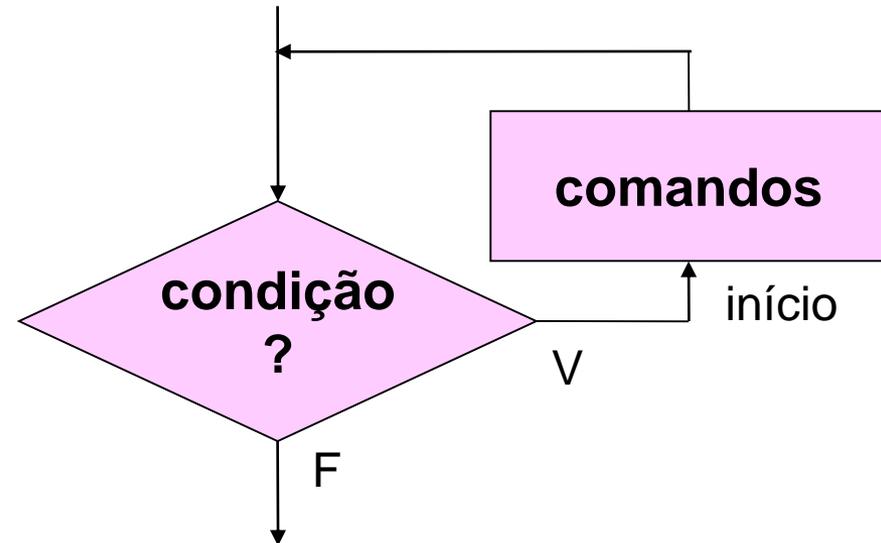


- **IMPORTANTE:** Deve-se sempre prever o comando, ou ação, que tornará falsa a condição do comando **enquanto**, caso contrário, seu programa entrará em **loop infinito**.
- **Condição de SAÍDA (do laço)**

Estrutura de Repetição: ENQUANTO

Pseudolinguagem

```
enquanto <condição> faça  
  <bloco_de_execução>  
fimenquanto
```



PORTUGOL

```
enquanto <condição> {  
  INSTRUÇÃO 1  
  ...  
}
```

JAVA

```
while (<condição>) {  
  INSTRUÇÃO 1  
  ...  
}
```



Universidade Federal do ABC

O problema da **Tabuada**

Problema da Tabuada

Escreva um algoritmo para mostrar na tela a **TABUADA** do número **N**.

5	X	1	=	5
5	X	2	=	10
5	X	3	=	15
5	X	4	=	20
5	X	5	=	25
5	X	6	=	30
5	X	7	=	35
5	X	8	=	40
5	X	9	=	45
5	X	10	=	50

Escreva um algoritmo para mostrar na tela a TABUADA do número N.

- **SAÍDA:** mostrar na tela as 10 linhas com a tabuada do N
- **ENTRADA:** 1 número (n)
- **PROCESSAMENTO:**
 - $n * 1,$
 - $n * 2,$
 - $n * 3,$
 - ...
 - $n * 9,$
 - $n * 10$

```
inteiro n

escreva ("Digite um número : ")
leia(n)

escreva ("\n", n, " X 1 = ", n * 1)
escreva ("\n", n, " X 2 = ", n * 2)
escreva ("\n", n, " X 3 = ", n * 3)
escreva ("\n", n, " X 4 = ", n * 4)
escreva ("\n", n, " X 5 = ", n * 5)
escreva ("\n", n, " X 6 = ", n * 6)
escreva ("\n", n, " X 7 = ", n * 7)
escreva ("\n", n, " X 8 = ", n * 8)
escreva ("\n", n, " X 9 = ", n * 9)
escreva ("\n", n, " X 10 = ", n * 10)
```

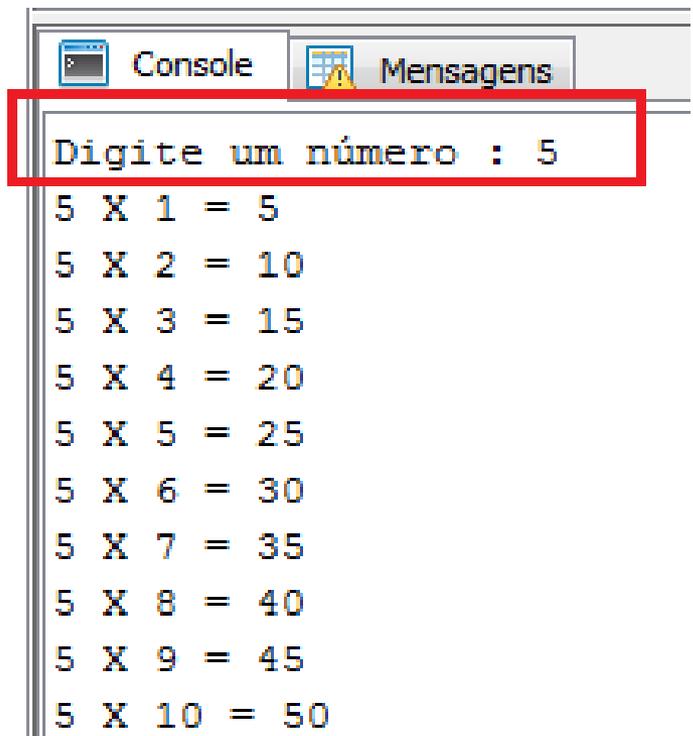
Escreva um algoritmo para mostrar na tela a TABUADA do número N.

```
inteiro n

escreva ("Digite um número : ")
leia(n)

escreva ("\n", n, " X 1 = ", n * 1)
escreva ("\n", n, " X 2 = ", n * 2)
escreva ("\n", n, " X 3 = ", n * 3)
escreva ("\n", n, " X 4 = ", n * 4)
escreva ("\n", n, " X 5 = ", n * 5)
escreva ("\n", n, " X 6 = ", n * 6)
escreva ("\n", n, " X 7 = ", n * 7)
escreva ("\n", n, " X 8 = ", n * 8)
escreva ("\n", n, " X 9 = ", n * 9)
escreva ("\n", n, " X 10 = ", n * 10)
```

RESULTADO



```
Console  Mensagens
Digite um número : 5
5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50
```

Problema da Tabuada – Solução Alternativa

PADRÃO DE COMPORTAMENTO

```
inteiro n, c
```

```
escreva ("Digite um número : ")
```

```
leia(n)
```

```
c = 1
```

```
escreva ("\n", n, " X ", c, " = ", n * c)
c = c + 1
```

```
escreva ("\n", n, " X ", c, " = ", n * c)
c = c + 1
```

```
escreva ("\n", n, " X ", c, " = ", n * c)
c = c + 1
```

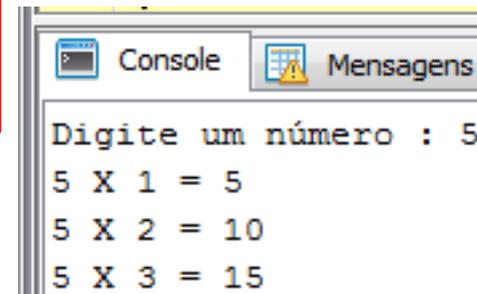
```
// REPETIR MAIS 7 VEZES ...
```

C=1

C=2

C=3

REPETIÇÃO



Problema da Tabuada – Solução com repetição

Escreva um algoritmo para mostrar na tela a TABUADA do número N.

- **SAÍDA:** mostrar na tela as 10 linhas com a tabuada do N
- **ENTRADA:** 1 número (n)
- **PROCESSAMENTO:**
 - Imprimir $(n * y)$ 10 vezes

Problema da Tabuada – Solução com repetição

```
inteiro n, c
```

```
escreva ("Digite um número : ")
```

```
leia(n)
```

```
c = 1
```

C=1

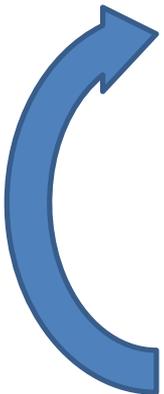
```
enquanto (c <=10) {
```

```
    escreva ("\n", n, " X ", c, " = ", n * c)
```

```
    c = c + 1
```

```
}
```

C=1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11



Java

```
public static void main(String[] args) {  
    int c = 1, n = (int)leia("Qual a taboada?");  
    while (c<=10)  
        System.out.printf("%2d x %2d = %2d\n",  
                            n,      c,      (n*c++) );  
}
```

```
public static float leia(String texto) {  
    java.util.Scanner s = new java.util.Scanner(System.in);  
    System.out.print(texto);  
    return s.nextFloat();  
}
```

Estruturas de Repetição

```
run:
```

```
Qual a taboada?5
```

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

$$5 \times 10 = 50$$

```
BUILD SUCCESSFUL (total time: 2 seconds)
```



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia Processamento da Informação

Estruturas de Repetição

IMPORTANTE !

Padrão de Instrução - CONTADOR

- Um “**padrão de instrução**” bastante utilizado em programas que requerem o uso de REPETIÇÃO é a figura do **CONTADOR**
- Um “contador” geralmente tem a seguinte forma
 - **var = var + 1** ou seja a variável “recebe” o valor que ela já possui somado com mais 1 ... e assim por diante ...

```
inteiro x
x = 1
enquanto (x <= 5) (
    escreva(x)
    x = x + 1
}
```

O que esse trecho de programa faz ?

Padrão de Instrução - CONTADOR

```
inteiro x
x = 1
enquanto (x <= 5) {
    escreva(x)
    x = x + 1
}
```

X
1
2
3
4
5
6

CONTADORES geralmente são inicializados com ZERO ou com 1

Padrão de Instrução - ACUMULADOR

- Um outro “**padrão de instrução**” bastante utilizado em programas que requerem o uso de REPETIÇÃO é a figura do **ACUMULADOR**
- Um “**ACUMULADOR**” geralmente tem a seguinte forma
 - **var = var + X** ou seja, a variável “recebe” o valor dela somado com mais X ... e assim por diante ...

```
inteiro c, s, idade
c = 1
s = 0
enquanto (c <= 5) {
    escreva("Digite a idade ")
    leia (idade)
    s = s + idade // acumulador
    c = c + 1    // contador
}
escreva("s = ", s)
```

O que esse trecho de programa faz ?

Padrão de Instrução - ACUMULADOR

```
inteiro c, s, idade
c = 1
s = 0
enquanto (c <= 5) {
    escreva("Digite a idade ")
    leia (idade)
    s = s + idade // acumulador
    c = c + 1 // contador
}
escreva("s = ", s)
```

C	S	IDADE
1	0	
		10
2	10	
		18
3	28	
		12
4	40	
		15
5	55	
		21
6	76	

ACUMULADORES geralmente são inicializados com o valor **ZERO**



Universidade Federal do ABC

Retomando ...

Média da idade de 50 pessoas

Estruturas de Repetição

Solução [1]

```
inteiro id1, id2, id3, id4, id5, id6, id7, id8, id9, id10  
real media
```

```
escreva ("Digite a idade da pessoa 1: ")
```

```
leia(id1)
```

```
escreva ("Digite a idade da pessoa 2: ")
```

```
leia(id2)
```

```
escreva ("Digite a idade da pessoa 3: ")
```

```
leia(id3)
```

```
escreva ("Digite a idade da pessoa 4: ")
```

```
leia(id4)
```

```
escreva ("Digite a idade da pessoa 5: ")
```

```
leia(id5)
```

```
escreva ("Digite a idade da pessoa 6: ")
```

```
leia(id6)
```

```
escreva ("Digite a idade da pessoa 7: ")
```

```
leia(id7)
```

```
escreva ("Digite a idade da pessoa 8: ")
```

```
leia(id8)
```

```
escreva ("Digite a idade da pessoa 9: ")
```

```
leia(id9)
```

```
escreva ("Digite a idade da pessoa 10: ")
```

```
leia(id10)
```

```
media<- (id1+id2+id3+id4+id5+id6+id7+id8+id9+id10)/10
```

```
escreva("A media das idades é: ", media)
```

E se fossem 50 pessoas ?
100 pessoas ?
200 pessoas ?

Solução com REPETIÇÃO - ENQUANTO

```
inteiro cont = 1, soma = 0, idade  
real media
```

```
enquanto (cont <= 50) {
```

```
    escreva ("\nDigite a idade da pessoa ", cont, " : ")  
    leia(idade)
```

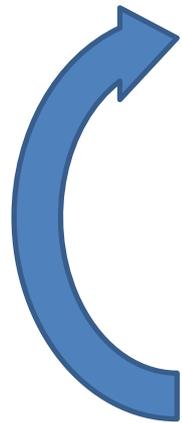
```
    soma = soma + idade
```

```
    cont = cont + 1
```

```
}
```

```
media = soma / 50
```

```
escreva("\nA media das idades é: ", media)
```



← Acumulador

← Contador

Java

```
public static void main(String[] args) {  
    int c = 1;  
    float soma = 0;  
    while (c <= 5) {  
        soma += leia("Idade da pessoa " + c++ + ": ");  
    }  
    System.out.println("A média é " + (soma / c));  
}  
  
public static float leia(String texto) {  
    java.util.Scanner s = new java.util.Scanner(System.in);  
    System.out.print(texto);  
    return s.nextFloat();  
}
```

Estruturas de Repetição

```
run :
```

```
Idade da pessoa 1: 1
```

```
Idade da pessoa 2: 2
```

```
Idade da pessoa 3: 3
```

```
Idade da pessoa 4: 4
```

```
Idade da pessoa 5: 5
```

```
A média é 2.5
```

```
BUILD SUCCESSFUL (total time: 2 seconds)
```



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia Processamento da Informação

Estruturas de Repetição

Outro Exemplo

Problema

Realizou-se uma pesquisa com 20 pessoas que responderam à seguinte pergunta:

Quantos filhos você tem?

Escreva um algoritmo para processar essa pesquisa informando quantas pessoas possuem até 2 filhos e quantas possuem mais de 2 filhos

- **SAÍDA:** Quais os resultados que o algoritmo dever fornecer ?
- **ENTRADA:** Quais os insumos necessários para se obter a saída?
- **PROCESSAMENTO:** Como transformar os insumos na saída?

Problema

Realizou-se uma pesquisa com 20 pessoas que responderam à seguinte pergunta:

Quantos filhos você tem?

Escreva um algoritmo para processar essa pesquisa

 informando quantas pessoas possuem até 2 filhos e quantas possuem mais de 2 filhos

▪ **SAÍDA:** Quais os resultados que o algoritmo deve fornecer ?

 ✓ Quant. Pessoas até 2 filhos e Quant. Pessoas mais 2 filhos
(**quant1, quant2**)

Problema

Realizou-se uma pesquisa com 20 pessoas que responderam à seguinte pergunta:

Quantos filhos você tem?

Escreva um algoritmo para processar essa pesquisa informando quantas pessoas possuem até 2 filhos e quantas possuem mais de 2 filhos

- 
- **ENTRADA:** Quais os insumos necessários para se obter a saída?

Problema

Realizou-se uma pesquisa com 20 pessoas que responderam à seguinte pergunta:



Quantos filhos você tem?

Escreva um algoritmo para processar essa pesquisa informando quantas pessoas possuem até 2 filhos e quantas possuem mais de 2 filhos



- **ENTRADA:** Quais os insumos necessários para se obter a saída?
 - ✓ Quantidade de Filhos de cada entrevistado (**qfilhos**)

Problema

Realizou-se uma pesquisa com 20 pessoas que responderam à seguinte pergunta: **Quantos filhos você tem?**

Escreva um algoritmo para processar essa pesquisa informando quantas pessoas possuem até 2 filhos e quantas possuem mais de 2 filhos

- **SAÍDA:** Quant. Pessoas até 2 filhos e Quant. Pessoas mais 2 filhos
 - **ENTRADA:** Quantidade de filhos de cada entrevistado
 - **PROCESSAMENTO:**
 - Se quant \leq 2 então
 acumular o contador 1
 - Senão**
 acumular o contador 2
- 20 vezes**
- 20 vezes**

Problema: Quantos filhos você tem?

```
inteiro qfilhos, quant1, quant2, cont
```

```
quant1 = 0
```

```
quant2 = 0
```

```
cont = 0
```

```
enquanto (cont < 20) {
```

```
    escreva ("\nQuantos filhos ? ")
```

```
    leia(qfilhos)
```

```
    se (qfilhos <= 2) {
```

```
        quant1 = quant1 + 1
```

```
    }
```

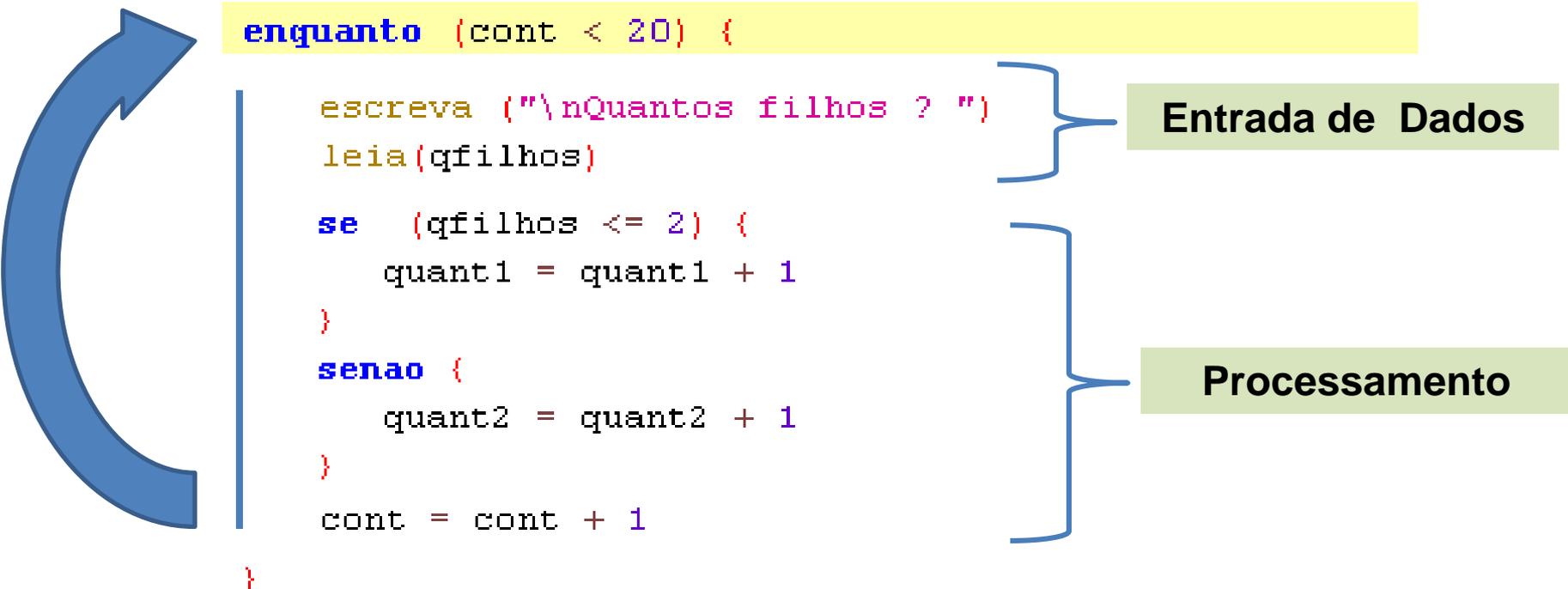
```
    senao {
```

```
        quant2 = quant2 + 1
```

```
    }
```

```
    cont = cont + 1
```

```
}
```



Entrada de Dados

Processamento

Estruturas de Repetição

Problema: Quantos filhos você tem?

```
inteiro qfilhos, quant1, quant2, cont
```

```
quant1 = 0
```

```
quant2 = 0
```

```
cont = 0
```

```
enquanto (cont < 20) {
```

```
    escreva ("\nQuantos filhos ? ")
```

```
    leia(qfilhos)
```

```
    se (qfilhos <= 2) {
```

```
        quant1 = quant1 + 1
```

```
    }
```

```
    senao {
```

```
        quant2 = quant2 + 1
```

```
    }
```

```
    cont = cont + 1
```

```
}
```

```
escreva ("\nPessoas com até 2 filhos: ", quant1)
```

```
escreva ("\nPessoas com MAIS de 2 filhos: ", quant2)
```



Saída de
Dados

Java

```
public static void main(String[] args) {
    int q1 = 0, q2 = 0, c = 1;
    while (c++ <= 20) {
        if (leia("Quantos filhos?") <= 2) q1++;
        else q2++;
    }
    System.out.printf("N° de filhos\n"
        + "%d (%d%%) tem 2 ou menos filhos\n"
        + "%d (%d%%) tem mais de 2 filhos\n",
        q1, (q1 * 4), q2, (q2 * 4));}

public static float leia(String texto) {
    java.util.Scanner s = new java.util.Scanner(System.in);
    System.out.print(texto);
    return s.nextFloat(); }
```

Estruturas de Repetição

```
run :  
Quantos filhos?3  
Quantos filhos?2  
Quantos filhos?1  
Quantos filhos?0  
...  
Quantos filhos?5  
Quantos filhos?5  
N° de filhos  
3 (12%) tem 2 ou menos filhos  
17 (68%) tem mais de 2 filhos  
BUILD SUCCESSFUL (total time: 6 seconds)
```



Universidade Federal do ABC

TESTE

Simulando o computador

Estruturas de Repetição

Problema: Quantos filhos você tem?

quant1	quant2	cont	qfilhos
0	0	0	?

```
inteiro qfilhos, quant1, quant2, cont
```

```
quant1 = 0
```

```
quant2 = 0
```

```
cont = 0
```

```
enquanto (cont < 20) {
```

```
    escreva ("\nQuantos filhos ? ")
```

```
    leia(qfilhos)
```

```
    se (qfilhos <= 2) {
```

```
        quant1 = quant1 + 1
```

```
    }
```

```
    senao {
```

```
        quant2 = quant2 + 1
```

```
    }
```

```
    cont = cont + 1
```

```
}
```

```
escreva ("\nPessoas com até 2 filhos: ", quant1)
```

```
escreva ("\nPessoas com MAIS de 2 filhos: ", quant2)
```

Estruturas de Repetição

Problema: Quantos filhos você tem?

```
inteiro qfilhos, quant1, quant2, cont
```

```
quant1 = 0
```

```
quant2 = 0
```

```
cont = 0
```

```
enquanto (cont < 20) {  
    escreva ("\nQuantos filhos ? ")  
    leia(qfilhos)  
  
    se (qfilhos <= 2) {  
        quant1 = quant1 + 1  
    }  
    senao {  
        quant2 = quant2 + 1  
    }  
  
    cont = cont + 1  
}
```

```
escreva ("\nPessoas com até 2 filhos: ", quant1)
```

```
escreva ("\nPessoas com MAIS de 2 filhos: ", quant2)
```

quant1	quant2	cont	qfilhos
0	0	0	
			(2)

Estruturas de Repetição

Problema: Quantos filhos você tem?

```
inteiro qfilhos, quant1, quant2, cont
```

```
quant1 = 0
```

```
quant2 = 0
```

```
cont = 0
```

```
enquanto (cont < 20) {
    escreva ("\nQuantos filhos ? ")
    leia(qfilhos)
```

```

    se (qfilhos <= 2) {
        quant1 = quant1 + 1
    }
    senao {
        quant2 = quant2 + 1
    }

```

```
cont = cont + 1
```

```
}
```

```
escreva ("\nPessoas com até 2 filhos: ", quant1)
```

```
escreva ("\nPessoas com MAIS de 2 filhos: ", quant2)
```

quant1	quant2	cont	qfilhos
0	0	0	
			(2)
1	0	1	

Estruturas de Repetição

Problema: Quantos filhos você tem?

```
inteiro qfilhos, quant1, quant2, cont
```

```
quant1 = 0
```

```
quant2 = 0
```

```
cont = 0
```

```
enquanto (cont < 20) {  
    escreva ("\nQuantos filhos ? ")  
    leia(qfilhos)
```

```
    se (qfilhos <= 2) {  
        quant1 = quant1 + 1  
    }
```

```
    senao {  
        quant2 = quant2 + 1  
    }
```

```
    cont = cont + 1
```



```
escreva ("\nPessoas com até 2 filhos: ", quant1)
```

```
escreva ("\nPessoas com MAIS de 2 filhos: ", quant2)
```

quant1	quant2	cont	qfilhos
0	0	0	
			(2)
1		1	



Estruturas de Repetição

Problema: Quantos filhos você tem?

```
inteiro qfilhos, quant1, quant2, cont
```

```
quant1 = 0
```

```
quant2 = 0
```

```
cont = 0
```

1 **enquanto** (cont < 20) {

```
    escreva ("\nQuantos filhos ? ")
```

2 **leia**(qfilhos)

```
    se (qfilhos <= 2) {
```

```
        quant1 = quant1 + 1
```

```
    }
```

3 **senao** {

```
    quant2 = quant2 + 1
```

```
    }
```

4 **cont** = cont + 1

```
}
```

```
escreva ("\nPessoas com até 2 filhos: ", quant1)
```

```
escreva ("\nPessoas com MAIS de 2 filhos: ", quant2)
```

quant1	quant2	cont	qfilhos
0	0	0	
			(2)
1		1	
			(1)
2		2	
			(0)
3		3	
			(5)
	1	4	
			(3)
	2	5	
...
{...}	{...}	20	

Coisas para não esquecer:

- **Padrão de Instrução: CONTADOR**

- Exemplo: $\text{cont} = \text{cont} + \mathbf{1}$

- **Padrão de Instrução: ACUMULADOR**

- Exemplo: $\text{soma} = \text{soma} + \mathbf{\text{valor}}$

AGORA É
PRATICAR!!!



Universidade Federal do ABC

EXERCÍCIOS

**Pra VOCÊ
Resolver!**

Exercício 01: Função $f(x)$

Escreva um algoritmo para exibir os valores de $f(x) = 2x^2 + 3x - 5$, para x de -5 a 5

- **SAÍDA:** Quais os resultados que o algoritmo dever fornecer ?
- **ENTRADA:** Quais os insumos necessários para se obter a saída?
- **PROCESSAMENTO:** Como transformar os insumos na saída?

Exercício 01: Função $f(x)$

Escreva um algoritmo para exibir os valores de $f(x) = 2x^2 + 3x - 5$, para x de -5 a 5

- **SAÍDA:** $f(x)$ { para os 10 valores de x }
 - **ENTRADA:** Não há entrada!
{ Apenas a inicialização de $x = -5$ }
 - **PROCESSAMENTO:**
 - $Fx = 2 * (x*x) + 3*x - 5$
 - $x = x + 1$
- } 10 vezes



Universidade Federal do ABC

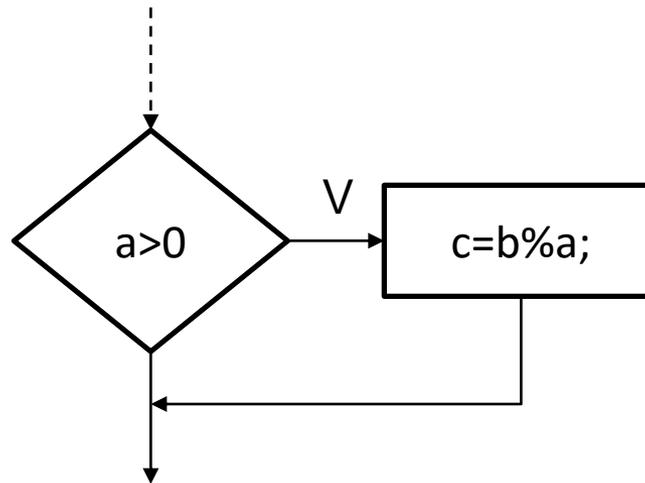
Bacharelado em Ciência e Tecnologia Processamento da Informação

Estruturas de Repetição

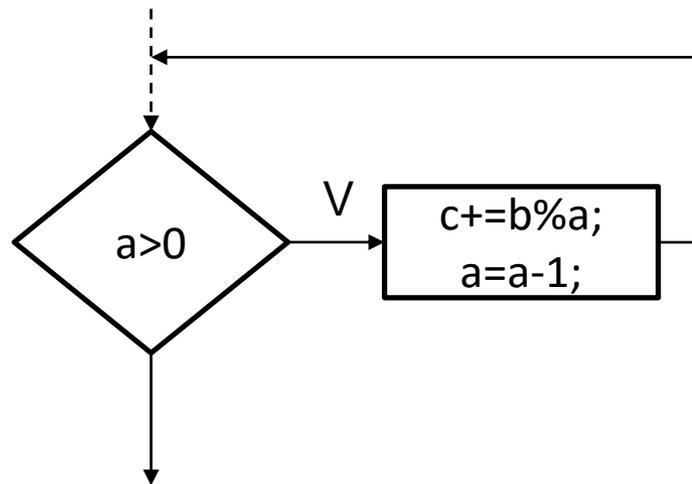
RESUMO

Estruturas de Repetição

Condicional

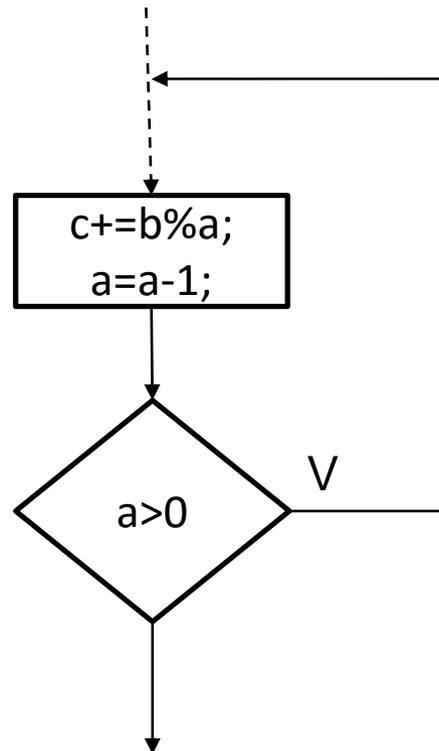


Laços de repetição



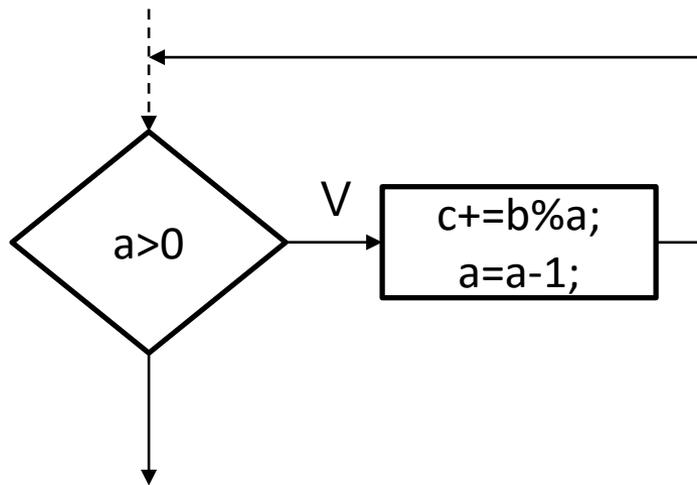
enquanto $(a > 0)$ faça { $c += b \% a;$ $a = a - 1;$ }

Laços de repetição

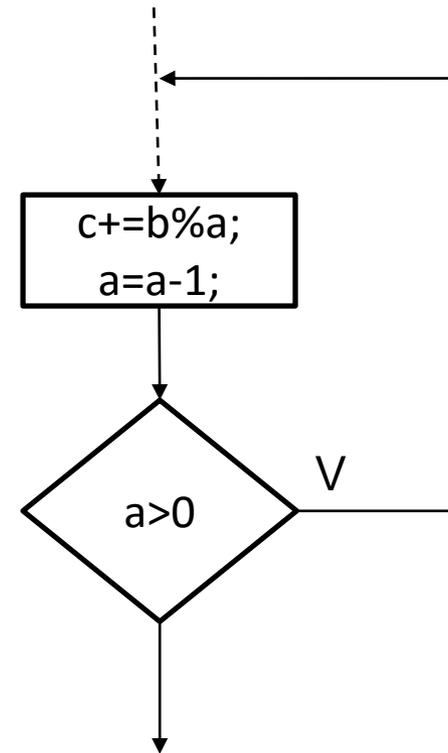


faça { `c += b % a;` `a = a - 1;` } enquanto (`a > 0`)

Laços em Java

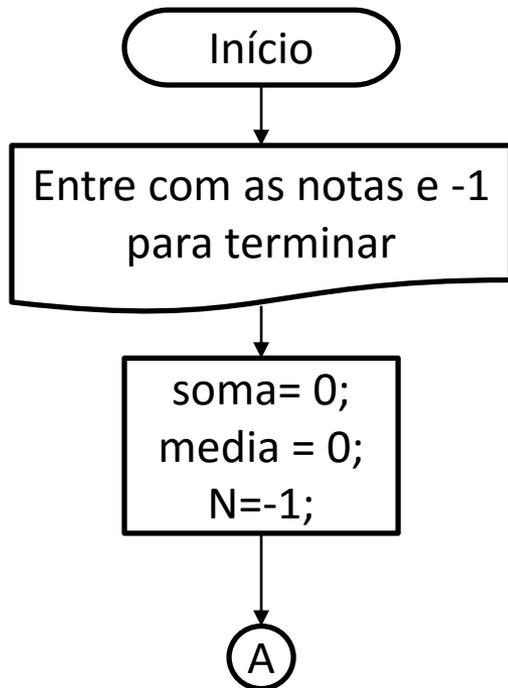


```
while (condição) {  
    comandos ; ; ;  
}
```

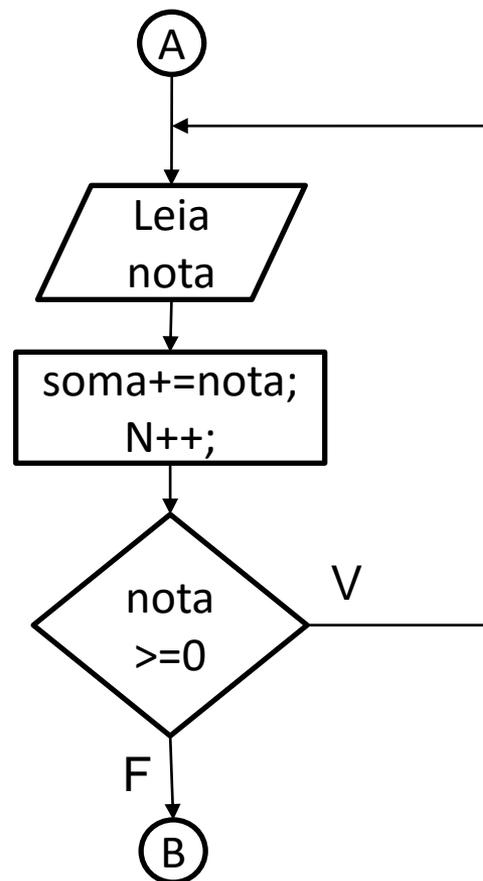


```
do {  
    comandos ; ; ;  
} while (condição);
```

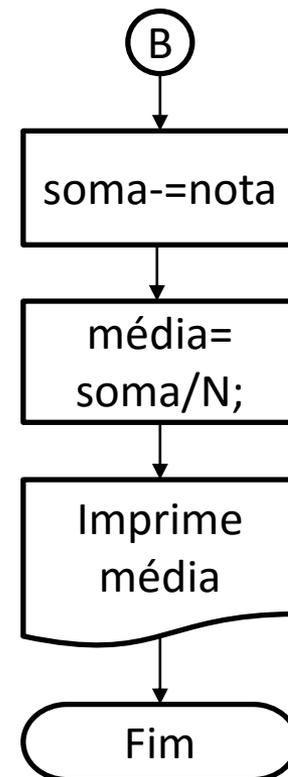
1. Média de N alunos



Sequencial



Repetição



Sequencial



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia Processamento da Informação

Estruturas de Repetição

Estruturas de Repetição – Parte II

PARA-ATÉ

Objetivos

- Aprender a **sintaxe** da Estrutura de Repetição **“PARA-ATÉ”**
- Aprender a resolver problemas que requeiram o uso de **ESTRUTURAS DE REPETIÇÃO** a partir de seus enunciados (com a aplicação da técnica de interpretação de enunciados)



Universidade Federal do ABC

O problema da **Tabuada**

Problema da Tabuada

Escreva um algoritmo para mostrar na tela a **TABUADA** do número **N**.

5	X	1	=	5
5	X	2	=	10
5	X	3	=	15
5	X	4	=	20
5	X	5	=	25
5	X	6	=	30
5	X	7	=	35
5	X	8	=	40
5	X	9	=	45
5	X	10	=	50

Problema da Tabuada – Solução com repetição

Escreva um algoritmo para mostrar na tela a TABUADA do número N.

- **SAÍDA:** mostrar na tela as 10 linhas com a tabuada do N
- **ENTRADA:** 1 número (n)
- **PROCESSAMENTO:**
 - Imprimir $(n * y)$ 10 vezes

Problema da Tabuada – Solução com repetição

```
inteiro n, c
```

```
escreva ("Digite um número : ")
```

```
leia(n)
```

```
c = 1
```

C=1

```
enquanto (c <=10) {
```

```
    escreva ("\n", n, " X ", c, " = ", n * c)
```

```
    c = c + 1
```

```
}
```

C=1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11





Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia Processamento da Informação

Estruturas de Repetição

SOLUÇÃO ALTERNATIVA

Solução com OUTRA estrutura de repetição

```
inteiro n, c
```

```
escreva ("Digite um número : ")  
leia(n)
```

```
para (c =1; c<=10; c= c+1) {
```

```
    escreva ("\n", n, " X ", c, " = ", n * c)
```

```
}
```



Outra “instrução”
de repetição

Estrutura de Repetição PARA-ATÉ

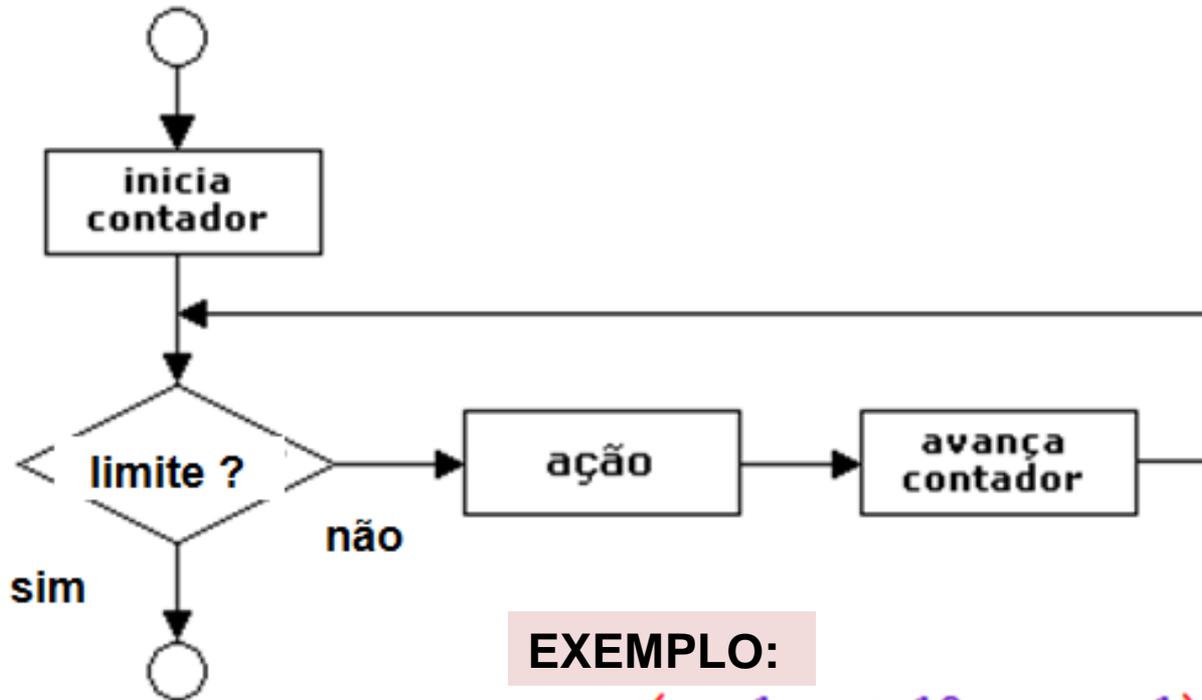
- Executa o bloco de instruções um **DETERMINADO** número de vezes;
- O **PARA** possui uma variável (**contador**) que controla o número de vezes que o laço será executado

```
para (variavel = valor_inicial; variavel = valor_final; variavel incremento) {  
    ...Bloco de Instruções...  
}
```

EXEMPLO: `para (c =1; c<=10; c= c+1) { }`

JAVA: `for (int i = 1; i <= 10; i++) { }`

Estrutura de Repetição PARA



EXEMPLO:

```
para (c =1; c<=10; c= c+1) {
```

```
    escreva ("\n", n, " X ", c, " = ", n * c)
```

```
}
```



Universidade Federal do ABC

ENQUANTO
versus
PARA-ATÉ

Estruturas de Repetição

ENQUANTO X PARA-ATÉ

1 = Inicialização da Variável
2 = Teste de Condição
3 = Incremento

```
inteiro n, c
```

```
escreva ("Digite um número : ")  
leia(n)
```

```
c = 1
```

1

```
enquanto (c <= 10) {  
    escreva ("\n", n, " X ", c, " = ", n * c)
```

2

```
c = c + 1
```

3

```
}
```

```
inteiro n, c
```

```
escreva ("Digite um número : ")  
leia(n)
```

1

2

3

```
para ( c = 1 ; c <= 10 ; c = c + 1 ) {  
    escreva ("\n", n, " X ", c, " = ", n * c)  
}
```

O Comando PARA parece mais “enxuto” pois ele permite em uma única linha, os passos 1, 2 e 3

Estruturas de Repetição

ENQUANTO X PARA-ATÉ

- 1 = Inicialização da Variável
- 2 = Teste de Condição
- 3 = Incremento

```
inteiro n, c
```

```
escreva ("Digite um número : ")  
leia(n)
```

```
c = 1
```

1

```
enquanto (c <= 10) {  
    escreva ("\n", n, " X ", c, " = ", n * c)
```

2

```
c = c + 1
```

3

```
}
```

```
inteiro n, c
```

```
escreva ("Digite um número : ")  
leia(n)
```

1

2

3

```
para ( c = 1 ; c <= 10 ; c = c + 1 ) {  
    escreva ("\n", n, " X ", c, " = ", n * c)  
}
```

QUAL SOLUÇÃO PAECE MAIS “FÁCIL” DE ANALISAR O FUNCIONAMENTO (ENQUANTO ou PARA-ATÉ) ?

Estruturas de Repetição

ENQUANTO X PARA-ATÉ

1 = Inicialização da Variável
2 = Teste de Condição
3 = Instruções
4 = Incremento

```
inteiro n, c
```

```
escreva ("Digite um número : ")  
leia(n)
```

```
c = 1
```

1

```
enquanto (c <= 10) {  
    escreva ("\n", n, " X ", c, " = ", n * c)
```

2

```
c = c + 1
```

3

```
}
```

```
inteiro n, c
```

```
escreva ("Digite um número : ")  
leia(n)
```

1

2

4

```
para ( c = 1 ; c <= 10 ; c = c + 1 ) {
```

3

```
    escreva ("\n", n, " X ", c, " = ", n * c)
```

```
}
```

- O Comando PARA parece mais “**enxuto**” pois ele permite em uma única linha, os passos 1, 2 e 3
- ... Mas o ENQUANTO parece mais “fácil” de analisar seu funcionamento já que cada instrução está em uma linha diferente)

Estruturas de Repetição

ENQUANTO X PARA-ATÉ

0 = Inicialização da Variável
1 = Teste de Condição
2 = Ações
3 = Incremento

```
inteiro n, c
```

```
escreva ("Digite um número : ")  
leia(n)
```

```
c = 1
```

0

```
enquanto (c <= 10) {  
    escreva ("\n", n, " X ", c, " = ", n * c)
```

1

```
c = c + 1
```

3

```
}
```

2

```
inteiro n, c
```

```
escreva ("Digite um número : ")  
leia(n)
```

0

1

3

```
para ( c = 1 ; c <= 10 ; c = c + 1 ) {
```

```
    escreva ("\n", n, " X ", c, " = ", n * c)
```

2

```
}
```

O Comando PARA parece mais “ enxuto ” pois ele permite em uma única linha, os passos 1, 2 e 3



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia Processamento da Informação

Estruturas de Repetição

EXERCÍCIOS

Problema

Um número inteiro x é perfeito se a soma de seus fatores (divisores), exceto ele mesmo, é igual a x . Por exemplo, 6 é perfeito visto que $1 + 2 + 3 = 6$. Escreva um programa para informar se x é um número perfeito

- **SAÍDA:** Quais os resultados que o algoritmo dever fornecer ?
- **ENTRADA:** Quais os insumos necessários para se obter a saída?
- **PROCESSAMENTO:** Como transformar os insumos na saída?

Problema

Um número inteiro x é perfeito se a soma de seus fatores (divisores), exceto ele mesmo, é igual a x . Por exemplo, 6 é perfeito visto que $1 + 2 + 3 = 6$.

Escreva um programa **para informar se x é um número perfeito**

- **SAÍDA:** Mensagem- **É PERFEITO** ou **NÃO É PERFEITO**
- **ENTRADA:** Quais os insumos necessários para se obter a saída?

Problema

Um **número inteiro x** é perfeito se a soma de seus fatores (divisores), exceto ele mesmo, é igual a x . Por exemplo, 6 é perfeito visto que $1 + 2 + 3 = 6$. Escreva um programa para informar se x é um número perfeito

- **SAÍDA:** Mensagem- É PERFEITO ou NÃO É PERFEITO
- **ENTRADA:** UM NÚMERO (x)
- **PROCESSAMENTO:** Como transformar os insumos na saída?

Problema

- **SAÍDA:** x “é um número perfeito” ou
 x “não é um número perfeito”
- **ENTRADA:** o número x
- **PROCESSAMENTO:**
 - Somar (acumular) todos os divisores de X
 - Se (somaDosDivisores = X) então
Informe: X é um numero perfeito
 - Senão ...
Informe: X não é um numero perfeito



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia
Processamento da Informação

Estruturas de Repetição

OPERADOR DE RESTO

OPERADOR DE RESTO (%)

Operação **resto** % (maioria): **div % sor**

Matlab: função **rem(div,sor)**

outras: função **mod(div,sor)**

Exemplo: $9 \% 4 = 1$

$$\begin{array}{r|l} 9 & 4 \\ \hline \mathbf{1} & 2 \end{array}$$



Universidade Federal do ABC

Solução em Portugal e Teste de Mesa [1]

Estruturas de Repetição

SOLUÇÃO

```
inteiro x, y, resto, soma=0, metade
```

```
escreva ("Digite o valor de x: ")
```

```
leia(x)
```

```
metade = x / 2
```

x	y	resto	soma	metade
			0	
(6)				
				3

Estruturas de Repetição

SOLUÇÃO

```
inteiro x, y, resto, soma=0, metade
```

```
escreva ("Digite o valor de x: ")
```

```
leia(x)
```

```
metade = x / 2
```

```
para (y=1; y <= metade; y=y+1) {
    resto = x % y
    se (resto == 0) {
        soma = soma + y
    }
}
```

x	y	resto	soma	metade
			0	
(6)				
				3

	1			
		0	1	

Estruturas de Repetição

SOLUÇÃO

```
inteiro x, y, resto, soma=0, metade
```

```
escreva ("Digite o valor de x: ")
```

```
leia(x)
```

```
metade = x / 2
```

```
para (y=1; y <= metade; y=y+1) {  
    resto = x % y  
    se (resto == 0) {  
        soma = soma + y  
    }  
}
```

x	y	resto	soma	metade
			0	
(6)				
				3

	1			
--	---	--	--	--

		0	1	
--	--	---	---	--

	2	0	3	
--	---	---	---	--

Estruturas de Repetição

SOLUÇÃO

```
inteiro x, y, resto, soma=0, metade
```

```
escreva ("Digite o valor de x: ")
```

```
leia(x)
```

```
metade = x / 2
```

```
para (y=1; y <= metade; y=y+1) {  
    resto = x % y  
    se (resto == 0) {  
        soma = soma + y  
    }  
}
```

x	y	resto	soma	metade
			0	
(6)				
				3

	1			
		0	1	
	2	0	3	
	3	0	6	

Estruturas de Repetição

SOLUÇÃO

```
inteiro x, y, resto, soma=0, metade
```

```
escreva ("Digite o valor de x: ")
```

```
leia(x)
```

```
metade = x / 2
```

```
para (y=1; y <= metade; y=y+1) {  
    resto = x % y  
    se (resto == 0) {  
        soma = soma + y  
    }  
}
```

x	y	resto	soma	metade
			0	
(6)				
				3

	1			
		0	1	
	2	0	3	
	3	0	6	
	4			

SOLUÇÃO

```
inteiro x, y, resto, soma=0, metade
```

```
escreva ("Digite o valor de x: ")  
leia(x)  
metade = x / 2
```

```
para (y=1; y <= metade; y=y+1) {  
    resto = x % y  
    se (resto == 0) {  
        soma = soma + y  
    }  
}
```

```
se (soma == x) {  
    escreva(x, " é um número PERFEITO ")  
}  
senao {  
    escreva(x, " NÃO é um número PERFEITO ")  
}
```

x	y	resto	soma	metade
(6)	4	0	6	3

**6 é um NÚMERO
PERFEITO**



Universidade Federal do ABC

Teste de Mesa

[2]

Estruturas de Repetição

SOLUÇÃO

```
inteiro x, y, resto, soma=0, metade
```

```
escreva ("Digite o valor de x: ")
```

```
leia(x)
```

```
metade = x / 2
```

```
para (y=1; y <= metade; y=y+1) {  
    resto = x % y  
    se (resto == 0) {  
        soma = soma + y  
    }  
}
```

x	y	resto	soma	metade
			0	
(8)				
				4
	1			

	1	0	1	
--	---	---	---	--

	2	0	3	
--	---	---	---	--

	3	2	3	
--	---	---	---	--

	4	0	7	
--	---	---	---	--

	5			
--	---	--	--	--

SOLUÇÃO

```
inteiro x, y, resto, soma=0, metade
```

```
escreva ("Digite o valor de x: ")  
leia(x)  
metade = x / 2
```

```
para (y=1; y <= metade; y=y+1) {  
    resto = x % y  
    se (resto == 0) {  
        soma = soma + y  
    }  
}
```

```
se (soma == x) {  
    escreva(x, " é um número PERFEITO ")  
}  
senao {  
    escreva(x, " NÃO é um número PERFEITO ")  
}
```

x	y	resto	soma	metade
(8)	5	0	7	4

**8 NÃO é um NÚMERO
PERFEITO**



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia Processamento da Informação

Estruturas de Repetição

EXERCÍCIO

Incrementando o Problema

Escreva um programa para informar todos os números perfeitos entre 1 e 10^4

- **SAÍDA:** Quais os resultados que o algoritmo dever fornecer ?
- **ENTRADA:** Quais os insumos necessários para se obter a saída?
- **PROCESSAMENTO:** Como transformar os insumos na saída?

Incrementando o Problema

Escreva um programa para informar todos os números perfeitos entre 1 e 10^4

- **SAÍDA:** todos os números perfeitos entre 1 e 10000
 - **ENTRADA:** Nenhum { $x = 1$ e vai até 10000 }
 - **PROCESSAMENTO:**
 - Somar (acumular) todos os divisores de X
 - Se (somaDosDivisores = X) então
Informe: X é um numero perfeito
fimse
- } Repetir de 1 a 10000

Estruturas de Repetição

```
public static void main(String[] args) {
    escreva("Numeros perfeitos de 1 a 10000\n");
    int soma;
    String divs; // armazena divisores numa String
    for (int n = 1; n <= 10000; n++) {
        soma = 0;
        divs = " = "; // seqüência de divisores
        for (int d = 1; d <= (n / 2); d++)
            if ((n % d) == 0) {
                soma += d;
                divs += (" " + d + " ");
            }
        if (n == soma) escreva("%6d%s\n", n, divs);
    } } }
```

Estruturas de Repetição

```
run:
```

```
Numeros perfeitos entre 1 e 10000
```

```
6 = 1 2 3
28 = 1 2 4 7 14
496 = 1 2 4 8 16 31 62 124 248
8128 = 1 2 4 8 16 32 64 127 254 508
      1016 2032 4064
```

```
BUILD SUCCESSFUL (total time: 0 seconds)
```

RESUMINDO...

Coisas para não esquecer:

- **OPERADOR RESTO**
 - O Operador % retorna o resto da divisão
 - Exemplo: $R = X \% Y$

Coisas para não esquecer:

- **Problemas que envolvam “vários elementos” com quantidade conhecida podem ser tratados com **ESTRUTURAS DE REPETIÇÃO PARA-ATÉ (FOR)****

SINTAXE DO PORTUGOL

```
para (variavel = valor_inicial; variavel = valor_final; variavel incremento) {  
    ...Bloco de Instruções...  
}
```

Java

```
for ([tipo] variável = valor_inicial; condição; incremento) { bloco }
```



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia Processamento da Informação

Estruturas de Repetição

Estruturas de Repetição – Parte III **VALIDAÇÃO DE DADOS** **E REPETIÇÃO FAÇA-ENQUANTO**

Objetivos

- Compreender a necessidade da **VALIDAÇÃO DE DADOS**
- Aprender a utilizar **VALIDAÇÃO DE DADOS**
- Aprender a **sintaxe** da Estrutura de Repetição **“FAÇA-ENQUANTO”** ou **“DO WHILE”** em java
- Resolver problemas que requeiram o uso de **ESTRUTURAS DE REPETIÇÃO** a partir de seus enunciados (com a aplicação da técnica de interpretação de enunciados)

Validação de Dados

- ❑ Validação de Dados é uma técnica que verifica se o dado digitado pelo usuário é um **valor válido** para o problema em questão
- ❑ A validação de dados é feita utilizando-se **CONDIÇÃO**, podendo ser:
 - ❑ **Se (condição) então**
 - ❑ **Enquanto (condição) faça**



Estruturas de Repetição

Exemplo

Problema

Escreva um programa para computar os resultados de uma pesquisa realizada com 50 pessoas em que responderam a pergunta: “**Você gosta de futebol?**”. O programa deve informar quantas pessoas responderam “S” e quantas responderam “N”.

- **SAÍDA:** Quais os resultados que o algoritmo dever fornecer ?
- **ENTRADA:** Quais os insumos necessários para se obter a saída?
- **PROCESSAMENTO:** Como transformar os insumos na saída?

Problema

Escreva um programa para computar os resultados de uma pesquisa realizada com 50 pessoas em que responderam a pergunta: “Você gosta de futebol?”. O programa **deve informar quantas pessoas responderam “S” e quantas responderam “N”**.

- **SAÍDA:** quantidade de respostas S e quantidade de respostas N (**contS, contN**)

Problema

Escreva um programa para computar os resultados de uma pesquisa realizada com 50 pessoas em que responderam a pergunta: **“Você gosta de futebol?”**. O programa deve informar quantas pessoas responderam “S” e quantas responderam “N”.

- **SAÍDA:** quantidade de respostas S e quantidade de respostas N (contS, contN)
- **ENTRADA:** Gosta de Futebol?(resposta)

50 vezes

Problema

Escreva um programa para computar os resultados de uma pesquisa realizada com 50 pessoas em que responderam a pergunta: “**Você gosta de futebol?**”. O programa deve informar quantas pessoas responderam “S” e quantas responderam “N”.

- **SAÍDA:** quantidade de respostas S e quantidade de respostas N (contS, contN)

- **ENTRADA:** Gosta de Futebol?(resposta)

- **PROCESSAMENTO:**

- SE resposta = ‘S’

- ENTÃO contS = contS + 1

- SENÃO contN = contN + 1

50 vezes

Estruturas de Repetição

```
caracter resposta  
inteiro cont=0, contS=0, contN = 0
```

```
para (cont=1; cont <=50 ; cont=cont + 1) {  
  
    escreva("Gosta de Futebol [S/N]? : ")  
    leia (resposta)  
  
    se (resposta == 'S') {  
        contS = contS + 1  
    }  
    senao {  
        contN = contN + 1  
    }  
  
}
```

**COMO GARANTIR QUE
NA ENTRADA DE
DADOS SERÁ ACEITO
APENAS AS LETRAS
“S” ou “N” ?**

```
escreva("\n Total de Pessoas que Gostam de Futebol: ", contS)  
escreva("\n Total de Pessoas que NÃO Gostam de Futebol: ", contN)
```



Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia
Processamento da Informação

Estruturas de Repetição

VALIDAÇÃO DE DADOS

Estruturas de Repetição



```
escreva("Gosta de Futebol [S/N]? : ")
leia (resposta)
enquanto (resposta != 'S' e resposta!= 'N') {
    escreva("Erro! Digite S ou N: Gosta de Futebol [S/N]? : ")
    leia (resposta)
}
```

**T
E
S
T
E**

 Console  Mensagens

```
Gosta de Futebol [S/N]? : X
Erro! Digite S ou N: Gosta de Futebol [S/N]? : X
Erro! Digite S ou N: Gosta de Futebol [S/N]? : X
Erro! Digite S ou N: Gosta de Futebol [S/N]? : X
Erro! Digite S ou N: Gosta de Futebol [S/N]? : S
Gosta de Futebol [S/N]? : |
```

Estruturas de Repetição

```
5 caracter resposta
6 inteiro cont=0, contS=0, contN = 0
7
8 para (cont=1; cont <=50 ; cont=cont + 1) {
9
10     escreva("Gosta de Futebol [S/N]? : ")
11     leia (resposta)
12     enquanto (resposta != 'S' e resposta!= 'N') {
13         escreva("Erro! Digite S ou N: Gosta de Futebol [S/N]? : ")
14         leia (resposta)
15     }
16
17     se (resposta == 'S') {
18         contS = contS + 1
19     }
20     senao {
21         contN = contN + 1
22     }
23
24 }
25 escreva("\n Total de Pessoas que Gostam de Futebol: ", contS)
26 escreva("\n Total de Pessoas que NÃO Gostam de Futebol: ", contN)
```

VALIDAÇÃO

T
E
S
T
E

Console Mensagens

```
Gosta de Futebol [S/N]? : X
Erro! Digite S ou N: Gosta de Futebol [S/N]? : X
Erro! Digite S ou N: Gosta de Futebol [S/N]? : X
Erro! Digite S ou N: Gosta de Futebol [S/N]? : X
Erro! Digite S ou N: Gosta de Futebol [S/N]? : S
Gosta de Futebol [S/N]? : |
```



Universidade Federal do ABC

Uma outra estrutura de Repetição

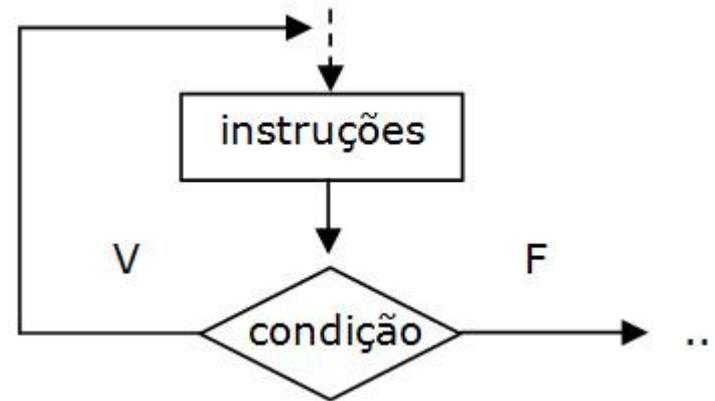
Estrutura de Repetição: FAÇA-ENQUANTO

Pseudolinguagem

Faça

<bloco_de_execução>

enquanto <condição>



SINTAXE DO PORTUGOL

faca {

INSTRUÇÃO 1

...

} enquanto (condição)

SINTAXE DO JAVA

do {

INSTRUÇÃO 1

...

} while (condição)



Universidade Federal do ABC

ENQUANTO
versus
FAÇA-ENQUANTO

Estruturas de Repetição

ENQUANTO

```
escreva("Gosta de Futebol [S/N]? : ")
leia (resposta)

enquanto (resposta != 'S' e resposta!= 'N') {
    escreva("Erro! Digite S ou N: Gosta de Futebol [S/N]? : ")
    leia (resposta)
}
```

FAÇA - ENQUANTO

```
faca {
    escreva("Gosta de Futebol [S/N]? : ")
    leia (resposta)
} enquanto (resposta != 'S' e resposta!= 'N')
```

- No **ENQUANTO** o teste de condição é feito no **INÍCIO**
- No **FAÇA-ENQUANTO** o teste de condição é feito no **FINAL**

Estruturas de Repetição

ENQUANTO

```
escreva("Gosta de Futebol [S/N]? : ")
leia (resposta)

enquanto (resposta != 'S' e resposta != 'N') {
    escreva("Erro! Digite S ou N: Gosta de Futebol [S/N]? : ")
    leia (resposta)
}
```

se (resposta != 'S' e resposta != 'N') {
 escreva ("Erro: Digites S ou N")
}

FAÇA - ENQUANTO

```
faca {
    escreva("Gosta de Futebol [S/N]? : ")
    leia (resposta)
} enquanto (resposta != 'S' e resposta != 'N')
```

- No **ENQUANTO** o teste de condição é feito no **INÍCIO**
- No **ENQUANTO-FAÇA** o teste de condição é feito no **FINAL**



Universidade Federal do ABC

EXERCÍCIOS

**Pra você
resolver!**

Problema

Escreva um programa para exibir a soma de todos os números pares entre um limite inferior e um limite superior (informado pelo usuário)

- **SAÍDA:** Quais os resultados que o algoritmo dever fornecer ?
- **ENTRADA:** Quais os insumos necessários para se obter a saída?
- **PROCESSAMENTO:** Como transformar os insumos na saída?

Problema

Escreva um programa para **exibir a soma** de todos os números pares entre um limite inferior e um limite superior (informado pelo usuário)

- **SAÍDA:** soma dos pares (do intervalo)
- **ENTRADA:** Quais os insumos necessários para se obter a saída?

Problema

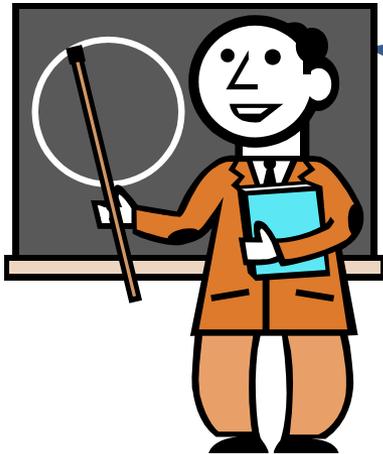
Escreva um programa para exibir a soma de todos os números pares entre um **limite inferior** e um **limite superior** (informado pelo usuário)

- **SAÍDA:** soma dos pares (do intervalo)
- **ENTRADA:** limite inferior e limite superior

Estruturas de Repetição

```
Escreva ("Digite o limite inferior: ")  
leia(inferior)
```

```
Escreva ("Digite o limite superior: ")  
leia(superior)
```



- Espera-se que o usuário digite o segundo número (**superior**) maior que o primeiro (**inferior**)
- O algoritmo deve, portanto, obrigar o usuário a fazer isso, ou seja:
 - O algoritmo **DEVE** exigir que **INFERIOR** seja menor ou igual (\leq) a **SUPERIOR**
- Isso se chama **VALIDAÇÃO DE DADOS**

Problema

Escreva um programa para exibir a soma de todos os números pares entre um **limite inferior** e um **limite superior** (informado pelo usuário)

- **SAÍDA:** soma dos pares (do intervalo)
- **ENTRADA:** limite inferior e limite superior
- **PROCESSAMENTO:** Como transformar os insumos na saída?

Problema

- **SAÍDA:** soma dos pares (do intervalo)
- **ENTRADA:** limite inferior e limite superior
- **PROCESSAMENTO:**
 - **Se $(\text{num} \% 2 == 0)$ então**
 $\text{somapar} = \text{somapar} + \text{num}$
fimse

Para cada
número
dentro do
intervalo

Coisas para não esquecer:

- **Validação de Dados é a forma de garantir que os dados digitados pelo usuário sejam aceitos apenas se estiverem corretos**
- **A Validação de Dados geralmente é feita com ESTRUTURAS DE REPETIÇÃO**
- **A estrutura de repetição mais adequada para validação de dados é o FAÇA-ENQUANTO (teste de condição no final)**

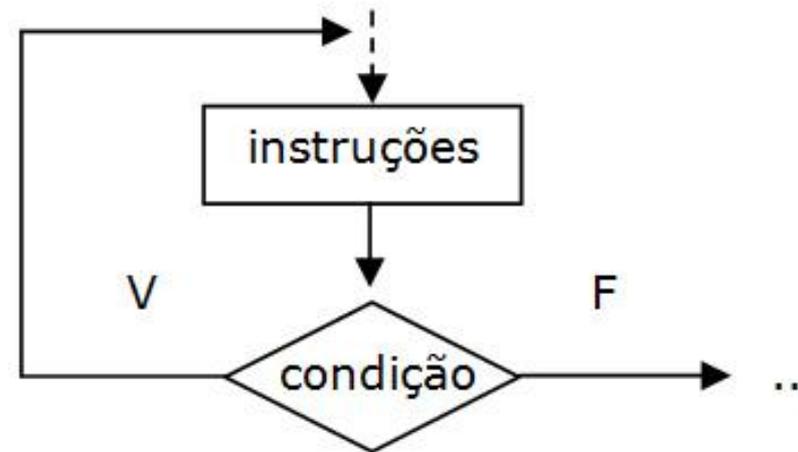
Estrutura de Repetição: FAÇA-ENQUANTO

SINTAXE DO PORTUGOL

```
faça {  
    INSTRUÇÃO 1  
    ...  
} enquanto (condição)
```

SINTAXE DO JAVA

```
do {  
    INSTRUÇÃO 1  
    ...  
} while (condição)
```



RESUMINDO...

Resumo geral: Java

ENQUANTO-FAÇA

WHILE-DO

```
while (condição) {  
    INSTRUÇÕES  
}
```

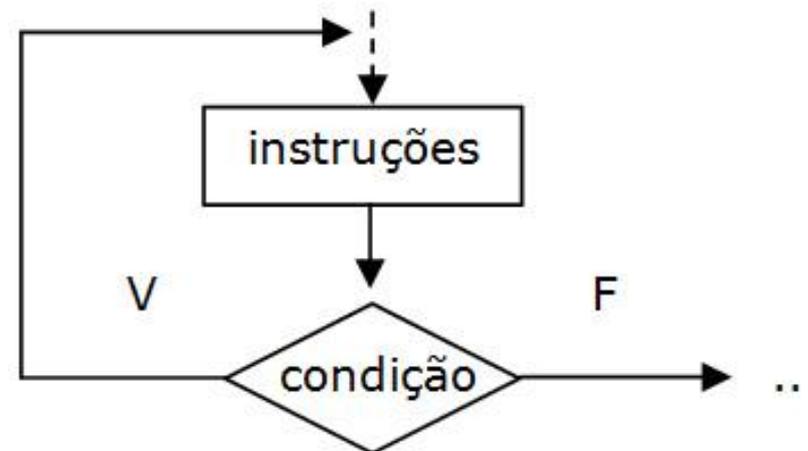
FAÇA-ENQUANTO

DO-WHILE

```
do {  
    INSTRUÇÕES  
} while (condição)
```

FOR (PARA-ATÉ)

```
for (v=1; cond; v++) {  
    INSTRUÇÕES  
} enquanto (condição)
```





Universidade Federal do ABC

Bacharelado em Ciência e Tecnologia Processamento da Informação

Estruturas de Repetição

**AGORA É
PRATICAR!!!**