

# Fundamentos de Programação 1

## Estrutura de Repetição “enquanto faça”

---

Slides 9

Prof. SIMÃO

Jean Marcelo SIMÃO

# Estrutura de Repetição “enquanto faça”

---

**enquanto ( condição for Verdadeira ) faça**

conjunto de comandos

**fimenquanto**

# Exemplo de Algoritmo - 1

**algoritmo** “ Imprimir os números de 1 à 1000 ”

**var**

num : **inteiro**

**inicio**

num <- 1

**enquanto** ( num <= 1000 ) **faca**

**escreval** ( “Número : ”, num )

    num <- num + 1

**fimenquanto**

**fimalgoritmo**

# Exemplo de Algoritmo - 2

---

**algoritmo** “ Somar todos os números de 1 à 1000 ”

**var**

soma, num : **inteiro**

**inicio**

soma <- 0

num <- 1

**enquanto** ( num <= 1000 ) **faca**

soma <- soma + num

num <- num + 1

**fimenquanto**

**escreval** ( “O somatório dos números entre 1 e 1000 é: ”, soma )

**fimalgoritmo**

# Explicando o Exemplo

**algoritmo** “Somar todo os números de 1 a 1000”

**var**

soma, num : **inteiro**

**inicio**

soma <- 0

num <- 1

**enquanto** ( num <= 1000 ) **faca**

    soma ← soma + num

    num ← num + 1

**fimenquanto**

**escreval** ( “O somatório dos números entre 1 e 1000 é: ”, soma )

**fimalgoritmo**

Passo	<i>Num</i>	<i>Soma</i>
1	2	1
2	3	3
3	4	6
4	5	10
5	6	15
6	7	21
7	8	28
8	9	36
...	...	...

# Exercício 1

---

- Algoritmo para somar todos os números de uma seqüência que começa por um e finaliza em um número dado pelo usuário.

# Solução Exercício 1

---

**algoritmo** “ Somar todos os números de 1 a n.”

**var**

soma, num, numdado : **inteiro**

**inicio**

soma <- 0

num <- 1

**escreval** ( “Soma de números de zero até um número dado.” )

**escreval** ( “Informe um número” )

**leia** ( numdado )

**enquanto** ( num <= numdado ) **faca**

soma <- soma + num

num <- num + 1

**fimenquanto**

**escreval** ( “O somatório dos números entre 1 e ”, numdado, “ é ”, soma )

**fimalgoritmo**

# Exercício 2

---

- **2.1 - Algoritmo para decidir se um número é par ou impar.**
- **2.2 - Algoritmo para decidir se cada número entre 1 e 500 é par ou impar.**
- **2.3 - Algoritmo para somar os números pares entre 5 e 500 (inclusive).**



# Solução 2.1

---

```
algoritmo "Par_Impar"  
var  
    num, resto : inteiro  
  
inicio  
  
    escreval ( " Digite um número inteiro: " )  
    leia ( num )  
  
    resto ← num % 2  
  
    se ( resto = 0 ) entao  
        escreval ( "O número é par." )  
    senao  
        escreval ( "O número é impar." )  
    fimse  
  
fimalgoritmo
```

# Solução 2.1 - V.2

---

```
algoritmo "Par_Impar"  
var  
  num, result1, result2 : inteiro  
  
inicio  
  
  escreval ( "Digite um número inteiro:" )  
  leia ( num )  
  
  result1 <- num / 2  
  
  result2 <- result1 * 2  
  
  se ( result2 = num ) entao  
    escreval ( "O número é par." )  
  senao  
    escreval ( "O número é impar." )  
  fimse  
  
fimalgoritmo
```

num	result1	result2
6	3	6
3	1	2

Obs. Isto funcionaria se o Interpretador de algoritmos armazenar a parte inteira da divisão na variável inteira.

Talvez, o interpretador dirá que esta variável não suporta um número real gerando um erro de execução!  
Isto ocorre com o VisuAlg.

# Solução 2.3 - V.1

---

**algoritmo** “Somar os pares entre 5 e 500”

**var**

soma, num, resto : **Inteiro**

**inicio**

soma <- 0

num <- 6

**escreval** ( “Soma de números pares entre 5 e 500.” )

**enquanto** ( num <= 500 ) **faca**

    resto <- num % 2;

**se** ( resto = 0 ) **entao**

        soma <- soma + num

**fimse**

    num <- num + 1

**fimenquanto**

**escreval** ( “O soma dos números pares entre 5 e 500 é”, soma )

**fimalgoritmo**

# Solução 2.3 - V.2

---

**algoritmo** “Somar os pares entre 5 e 500 V2”

**var**

soma, num : **inteiro**

**inicio**

soma <- 0

num <- 6

**escreval** ( “Soma de números pares entre 5 e 500.” )

**enquanto** ( num <= 500 ) **faca**

soma ← soma + num

num ← num + 2

**fimenquanto**

**escreval** ( “O soma dos números pares entre 5 e 500 é”, soma )

**fimalgoritmo**

# Exercício 3

---

- **Elaborar um algoritmo para o cálculo da soma, subtração, multiplicação ou divisão de dois números reais fornecidos pelo usuário, segundo sua opção.**
- **O usuário poderá realizar quantas operações desejar enquanto não optar por sair do programa.**

# Solução exercício.

```
algoritmo    "Operações elementares sobre dois
              números cf. opção do usuário"

var
  prim_num, seg_num, result : real
  opcao                    : inteiro

inicio
  opcao <- 0

enquanto ( opcao <> 5 ) faca
  escreval ( "Operações sobre 2 números reais." )
  escreval ( "Digite 1 para soma," )
  escreval ( "      2 para subtração," )
  escreval ( "      3 para multiplicação ou" )
  escreval ( "      4 para divisão." )
  escreval ( "      5 para sair do programa." )
  leia ( opcao )

se ( ( opcao > 0 ) e ( opcao < 5 ) ) entao

  escreval ( "Digite o primeiro número: " )
  leia ( prim_num )

  escreval ( "Digite o segundo número: " )
  leia ( seg_num )

  escolha ( opcao )
    caso 1
      result <- prim_num + seg_num
      escreval ( "O resultado da soma é: ", result )

    caso 2
      result <- prim_num - seg_num
      escreval ( "O resultado da subtração é: ", result )
```

```
    caso 3
      result <- prim_num * seg_num

      escreval ( "O resultado da multiplic. é: ", result )

    caso 4
      se ( seg_num <> 0 ) entao

        result ← prim_num / seg_num
        escreval ( "O resultado da divisão é: ", result )

      senao

        escreval ( "Divisão por zero impossível" )

      fimse

    fimescolha

  senao

    se ( opcao <> 5 ) entao

      escreval ( "Opção inválida" )

    fimse

  fimse

fimenquanto

fimalgoritmo
```

# Solução exercício.

```
algoritmo "Operações elementares sobre dois
números cf. opção do usuário"
var
  prim_num, seg_num, result : real
  opcao : inteiro

inicio
  opcao <- 0
  enquanto ( opcao <> 5 ) faça

    escreval ( "Operações sobre 2 números reais." )
    escreval ( "Digite 1 para soma," )
    escreval ( " 2 para subtração," )
    escreval ( " 3 para multiplicação ou" )
    escreval ( " 4 para divisão." )
    escreval ( " 5 para sair do programa." )
    leia ( opcao )

  se ( (opcao >0) e (opcao < 6) ) entao

    se ( opcao <> 5 ) entao
      escreval ( "Digite o primeiro número: " )
      leia ( prim_num )

      escreval ( "Digite o segundo número: " )
      leia ( seg_num )

    escolha ( opcao )
      caso 1
        result <- prim_num + seg_num
        escreval ( "O resultado da soma é: ", result )

      caso 2
        result <- prim_num - seg_num
        escreval ( "O resultado da subtração é: ", result )
```

caso 3

```
result <- prim_num * seg_num
```

```
escreval ( "O resultado da multiplic. é: ", result )
```

caso 4

```
se ( seg_num <> 0 ) entao
```

```
result ← prim_num / seg_num
```

```
escreval ( "O resultado da divisão é: ", result )
```

senao

```
escreval ( "Divisão por zero impossível" )
```

fimse

fimescolha

fimse

senao

```
escreval ( "Opção inválida" )
```

fimse

fimenquanto

fimalgoritmo

# Exercício 4

---

**4.1 - Elaborar um algoritmo para receber as notas de 150 alunos e calcular/apresentar a média das notas.**

**4.2 - Elaborar um algoritmo para receber as 4 notas de cada um dos 150 alunos, calculando/apresentando a média de cada um, bem como a média geral da turma.**



# Exercícios - 5

---

- **5.1 - Algoritmo para o cálculo do quadrado e da raiz quadrada de 1 até um número dado pelo usuário.**
- **5.2 Algoritmo para o cálculo do fatorial de um número dado pelo usuário.**
  - **Obs.: Usar a estrutura de repetição *enquanto-faça*.**

# Algoritmo Fatorial 1

```
algoritmo "fatorial"
var
  numero, resultado : inteiro
inicio

  escreval ( "Informe um número decimal inteiro: " )
  leia ( numero )

  se ( numero > 0 ) entao

    resultado <- numero

    enquanto ( numero <> 1 ) faca

      resultado <- resultado * ( numero - 1 )
      numero <- numero - 1

    fimenquanto

    escreval ( "O fatorial é:", resultado )

  senao
    se ( numero = 0 ) entao
      escreval ( "O fatorial é: 1." )
    senao
      escreval ( "Número inválido." )
    fimse
  fimse

fimalgoritmo
```

Numero = 5

	resultado	Numero
0	5*4= 20	4
1	20*3=60	3
2	60*2=120	2
3	120*1=120	1
4		

# Algoritmo Fatorial 2

```
algoritmo "Fatorial"
var
  numero, resultado : inteiro
inicio

  escreval ( "Informe um número decimal inteiro: " )
  leia ( numero )

  se ( numero > 0 ) entao

    resultado <- numero

    enquanto ( numero <> 1 ) faca

      numero <- numero - 1
      resultado <- resultado * numero

    fimenquanto

    escreval ( "O fatorial é:", resultado )

  senao
    se ( numero = 0 ) entao
      escreval ( " O fatorial é: 1." )
    senão
      escreval ( " Número invalido." )
  fimse
fimalgoritmo
```

Numero = 5

	resultado	Numero
0	5*4= 20	4
1	20*3=60	3
2	60*2=120	2
3	120*1=120	1
4		

# Exercícios 6.

---

- **Algoritmo para permitir ao usuário escolher entre o cálculo do fatorial, do quadrado ou da raiz quadrada de um número dado por ele. O usuário também pode escolher como opção 'sair do programa'.**
  - **Obs. 1: Uma primeira versão usando apenas estrutura de decisão *se fim-se* para tratar a opção do usuário.**
  - **Obs. 2: Uma segunda versão usando apenas estrutura de decisão *se senão* para tratar a opção do usuário.**
  - **Obs. 3: Uma terceira versão usando apenas estrutura de decisão *escolha caso* para tratar a opção do usuário.**

# Exercícios 7

---

- **6.1 Algoritmo para permitir o cálculo da área ou do perímetro de uma circunferência.**
  - Obs. 1: Raio fornecido pelo usuário.
  - Obs. 2: Depois de cada cálculo o algoritmo deve permitir ao usuário escolher a mesma ou outra opção.
  - Obs. 3: Utilizar a estrutura escolha-caso.
  - Obs. 4. Utilizar uma variável tipo *character* para tratar a opção do usuário.
  - Obs. 5: O algoritmo só terminará quando o usuário escolher uma opção de término.
  
- **6.2 Algoritmo para cálculo da área de um quadrado, de um triângulo retângulo ou de uma circunferência dependendo da escolha do usuário.**
  - Obs.1: Parâmetros (e.g. lado ou raio) fornecidos pelo usuário.
  - Obs. 2: Utilizar a estrutura escolha-caso.
  - Obs. 3. Utilizar uma variável tipo *character* para tratar a opção do usuário.
  - Obs. 4: Depois de cada cálculo o algoritmo deve permitir ao usuário escolher a mesma ou outra opção.
  - Obs. 5: O algoritmo só terminará quando o usuário escolher uma opção de término.