

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

COMANDOS DE REPETIÇÃO (ENQUANTO, REPITA E PARA)

Disciplina: Algoritmos

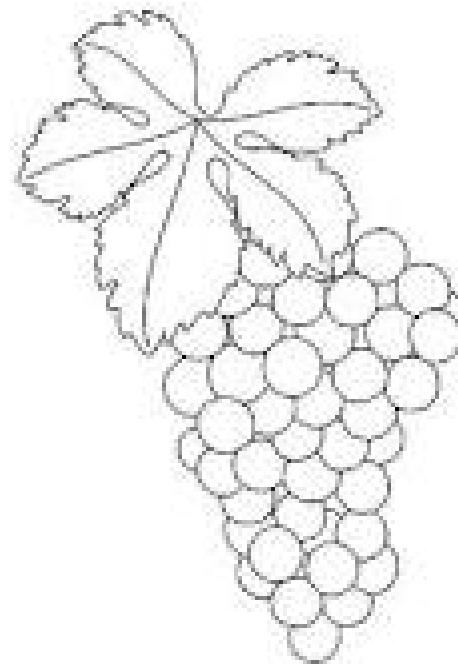
Professora: Alba Lopes

alba.lopes@ifrn.edu.br

<http://docente.ifrn.edu.br/albalopes>

COMANDOS DE REPETIÇÃO

- Usada quando se deseja que um trecho do algoritmo seja repetido várias vezes
- A quantidade de repetições pode ser fixa ou depender de uma condição



COMANDOS DE REPETIÇÃO

- As estruturas de repetição são classificadas em 3 tipos
 - Repetição com **teste no início** do laço
 - Repetição com **teste no final** do laço
 - Repetição com **variável de controle**
- São também chamadas de **laços de repetição!**



REPETIÇÃO COM TESTE NO INÍCIO

○ ENQUANTO...FAÇA

- Essa estrutura repete uma sequencia de comandos enquanto uma determinada condição for verdadeira
- Essa condição é determinada por uma expressão lógica



ENQUANTO...FAÇA

○ Sintaxe:

```
enquanto <expressao-logica> faca  
    <sequencia-de-comandos>  
fimenquanto
```

<i><expressao_logica></i>	Essa expressão é avaliada antes de cada repetição do laço
<i>fimenquanto</i>	Indica o fim da sequencia de comandos . Cada vez que a execução atinge esse ponto, volta-se ao início do laço para que a <expressão-logica> seja avaliada novamente Se o resultado dessa avaliação for VERDADEIRO , a <sequencia-de-comandos> será executada mais uma vez. Caso contrário, a execução prosseguirá a partir do primeiro comando após o fimenquanto

EXEMPLO

- Criar um algoritmo para escrever 10x a mensagem “Algoritmos a melhor disciplina do mundo!”

```
algoritmo "repeticao"
```

```
var
```

```
inicio
```

```
    escreval("Algoritmos é a melhor disciplina do mundo!")
```

```
    escreval("Algoritmos é a melhor disciplina do mundo!")
```

```
    escreval("Algoritmos é a melhor disciplina do mundo!")
```

```
    escreval("Algoritmos é a melhor disciplina do mundo!")
```

```
    escreval("Algoritmos é a melhor disciplina do mundo!")
```

```
    escreval("Algoritmos é a melhor disciplina do mundo!")
```

```
    escreval("Algoritmos é a melhor disciplina do mundo!")
```

```
    escreval("Algoritmos é a melhor disciplina do mundo!")
```

```
    escreval("Algoritmos é a melhor disciplina do mundo!")
```

```
    escreval("Algoritmos é a melhor disciplina do mundo!")
```

```
fimalgoritmo
```



EXEMPLO (NÚMERO FIXO DE VEZES)

- E se tivesse que escrever a mensagem 1000 vezes?
 - Melhor usar um comando de repetição!

```
algoritmo "repeticao"  
var  
    contador: inteiro  
inicio  
    contador <- 1  
    enquanto (contador <= 1000) faca  
        escreval("Algoritmos é a melhor disciplina do mundo!")  
        contador <- contador + 1  
    fimenquanto  
fimalgoritmo
```



EXEMPLO (NÚMERO FIXO DE VEZES)

- E se tivesse que escrever a mensagem 1000 vezes?
 - Melhor usar um comando de repetição!

```
algoritmo "repeticao"  
var  
    contador: inteiro  
inicio  
    contador <- 1  
    enquanto (contador <= 1000) faca  
        escreval("Algoritmos é a melhor disciplina do mundo!")  
        contador <- contador + 1  
    fimenquanto  
fimalgoritmo
```

É preciso declarar uma variável para ser utilizada no teste da condição do laço (variável de controle)



EXEMPLO (NÚMERO FIXO DE VEZES)

- E se tivesse que escrever a mensagem 1000 vezes?
 - Melhor usar um comando de repetição!

```
algoritmo "repeticao"  
var  
    contador: inteiro  
inicio  
    contador <- 1  
    enquanto (contador <= 1000) faca  
        escreval("Algoritmos é a melhor disciplina do mundo!")  
        contador <- contador + 1  
    fimenquanto  
fimalgoritmo
```

É preciso
inicializar a
variável de
controle



EXEMPLO (NÚMERO FIXO DE VEZES)

- E se tivesse que escrever a mensagem 1000 vezes?
 - Melhor usar um comando de repetição!

```
algoritmo "repeticao"  
var  
    contador: inteiro  
inicio  
    contador <- 1  
    enquanto (contador <= 1000) faca  
        escreval("Algoritmos é a melhor disciplina do mundo!")  
        contador <- contador + 1  
    fimenquanto  
fimalgoritmo
```

É preciso
IDENTIFICAR a
condição de
parada do laço



EXEMPLO (NÚMERO FIXO DE VEZES)

- E se tivesse que escrever a mensagem 1000 vezes?
 - Melhor usar um comando de repetição!

```
algoritmo "repeticao"  
var  
    contador: inteiro  
inicio  
    contador <- 1  
    enquanto (contador <= 1000) faca  
        escreval("Algoritmos é a melhor disciplina do mundo!")  
        contador <- contador + 1  
    fimenquanto  
fimalgoritmo
```

É preciso
ATUALIZAR a
variável de
controle do laço



EXEMPLO (NÚMERO INDETERMINADO)



- Algoritmo para comer um cacho de uva
 - Não se sabe ao certo quantas uvas tem no cacho
 - Sempre é preciso identificar se há ou não uvas no cacho para continuar comendo...

```
algoritmo "cachos_de_uva"  
var  
    ha_uvas_no_cacho: logico  
inicio  
    escreval("Há uvas no cacho? Digite verdadeiro ou falso:")  
    leia(ha_uvas_no_cacho)  
    enquanto (ha_uvas_no_cacho) faça  
        escreval("Retire uma uva.")  
        escreval("Ainda há uvas no cacho? Digite verdadeiro ou falso:")  
        leia(ha_uvas_no_cacho)  
    fimenquanto  
    escreval("Não há mais uvas....")  
fimalgoritmo
```



EXERCÍCIOS

- Resolva os exercícios a seguir usando **comandos de repetição**
 1. Calcular a soma dos números de 1 a 20 usando comando de repetição
 2. Escrever os números pares existentes entre 1 e 20
 3. Ler um número e calcular seu fatorial
 - Ex: $5! = 5 * 4 * 3 * 2 * 1$
 4. Ler 10 números do usuário e calcule a média
 5. Ler a idade de várias pessoas e calcular a média das idades. Encerrar a execução quando valor negativo for digitado.



EXERCÍCIOS

6. Escreva um algoritmo para ler e escrever o nome de todos os alunos presentes na aula de hoje de Algoritmos. Encerre a execução quando o nome digitado for igual a “fim”
7. Leia uma sequencia de números do usuário e escreva o dobro de cada número lido. Encerre a execução quando o número digitado for negativo
8. Escreva um algoritmo que calcule a soma dos números pares enquanto o valor da soma for menor do que 100.



COMANDO DE REPETIÇÃO

- As estruturas de repetição são classificadas em 3 tipos
 - Repetição com **teste no início** do laço
 - Repetição com **teste no final** do laço
 - Repetição com **variável de controle**



REPETIÇÃO COM TESTE NO FINAL

- É semelhante à estrutura **enquanto**, porém a condição de teste fica no final do laço
- Porém, os comandos são repetidos até que a condição seja **verdadeira**
- Ao invés de usar a sintaxe “enquanto <condicao> faça...fimenquanto”, será utilizado “repita... ate <condicao>”
- A sequência de comandos dentro do **repita** é realizada no mínimo uma vez, diferente do enquanto, que pode nem ser executada.



REPETIÇÃO COM VARIÁVEL DE CONTROLE

○ Sintaxe:

repita

<comandos>

ate *<condicao>*

<i><comandos></i>	Sequencia de comandos a ser repetida
<i><condicao></i>	Condição a ser avaliada para verificar se o laço deve ser executado mais uma vez. É uma expressão lógica que resulta em verdadeiro ou falso



REPETIÇÃO COM TESTE NO FINAL

- No caso do repita, a sequência de comandos será executada até que a condição testada seja **verdadeira**.
 - Enquanto a condição testada for **falsa** os comandos dentro do repita serão executados.



ESTRUTURA REPITA

○ Exemplo: Subir escada

```
algoritmo "subir_escada"  
var  
    degraus: inteiro  
inicio  
    degraus <- 1  
    repita  
        escreva("Suba um degrau")  
        degraus <- degraus + 1  
    ate (degraus > 10)  
fimalgoritmo
```



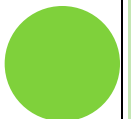
EXEMPLO

- Escreva um algoritmo usando a estrutura **REPITA** que leia uma sequência de dois números, calcule e escreva o valor da soma desses dois números.
Após escrever o resultado de soma, pergunte ao usuário se ele deseja ler novo valor. O programa deve finalizar se a resposta do usuário for “não”.



ESTRUTURA REPITA

```
algoritmo "soma"  
var  
    a, b: inteiro  
    soma: inteiro  
    continuar: caractere  
Inicio  
    repita  
        escreva("Digite o número 1:")  
        leia(a)  
        escreva("Digite o número 2:")  
        leia(b)  
        soma <- a + b  
        escreva ("A soma de ", a, "+", b, "é: ", soma)  
        escreva("Deseja Continuar? ")  
        leia(continuar)  
    ate (continuar = "não")  
fimalgoritmo
```




EXERCÍCIOS

1. Leia 10 números e exiba qual foi o menor número digitado pelo usuário.
2. Leia 20 números e exiba qual foi o maior número digitado pelo usuário.



EXERCÍCIOS

3. Escreva um algoritmo usando a estrutura **REPITA** que leia simule o funcionamento de um banco.
- O algoritmo deve, inicialmente, ler o saldo atual do cliente, e em seguida, entrar em um laço que leia qual operação o usuário deseja realizar:
 - Saque ou Depósito
 - Em seguida, deve ler o valor.
 - Se a operação a ser realizada for saque, o valor deve ser subtraído do saldo.
 - Se a operação for depósito, o valor deve ser adicionado ao saldo.
 - Ao final de cada operação, deve ser lido do usuário a informação se ele deseja realizar uma nova operação. O programa deve continuar executando até que o usuário informe que não possui mais operações para realizar.
- 

COMANDO DE REPETIÇÃO

- As estruturas de repetição são classificadas em 3 tipos
 - Repetição com **teste no início** do laço
 - Repetição com teste no final do laço
 - Repetição com **variável de controle**



REPETIÇÃO COM VARIÁVEL DE CONTROLE

- Quando se conhece o número de vezes que um trecho do algoritmo deve ser repetido, deve ser utilizado uma estrutura com **variável de controle**
- É utilizada a estrutura **para**
 - Essa estrutura descreve a repetição de um **número definido** de vezes, fixando limites **iniciais** e **finais** para a variável de controle



REPETIÇÃO COM VARIÁVEL DE CONTROLE

○ Sintaxe:

```
para <variavel> de <inicio> ate <limite> [passo <incremento>]faca  
    <comandos>  
fimpara
```

<variavel>	É a variável contadora que controla o numero de repetições do laço. Deve ser do tipo inteiro
<inicio>	É uma expressão que especifica o valor de inicialização da variável contadora antes da primeira repetição do laço
<limite>	Expressão que especifica o valor máximo que a variável contadora pode alcançar
<incremento>	É opcional. Quando for necessária, é precedida pela palavra passo . É uma expressão que especifica o incremento que será acrescentado à variável contadora em cada repetição do laço. Quando estiver omitida, o valor do incremento é sempre 1.

REPETIÇÃO COM VARIÁVEL DE CONTROLE

○ Sintaxe:

```
para <variavel> de <inicio> ate <limite> [passo <incremento>]faca  
    <comandos>  
fimpara
```

fimpara

Indica o fim da sequência de comandos a serem repetidos. Cada vez que o programa chega neste ponto, é acrescentado à variável contadora o valor de <incremento>, e comparado a <limite >. Se for menor ou igual, a sequência de comandos será executada mais uma vez; caso contrário, a execução prosseguirá a partir do primeiro comando que esteja após o fimpara.



ESTRUTURA PARA

○ Exemplo: Subir escada

```
algoritmo "subir_escada"  
var  
    degraus: inteiro  
inicio  
    para degraus de 1 ate 10 faca  
        escreval("Suba um degrau")  
    fimpara  
finalgoritmo
```



ESTRUTURA PARA

- **Exemplo utilizando o “passo”**
 - **Escrever número pares entre 0 e 20**

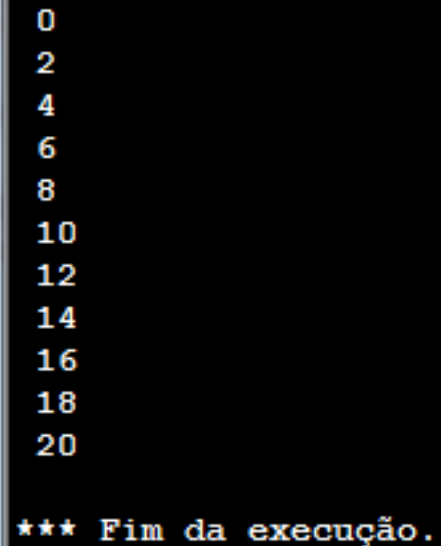
```
algoritmo "numeros_pares"  
var  
    numeros_pares: inteiro  
inicio  
    para numeros_pares de 0 ate 20 passo 2 faca  
        escreval(numeros_pares)  
    fimpara  
fimalgoritmo
```



ESTRUTURA PARA

- Exemplo utilizando o “passo”
 - Escrever número pares entre 0 e 20

```
algoritmo "numeros_pares"  
var  
    numeros_pares: inteiro  
inicio  
    para numeros_pares de 0 ate 20 passo 2 faca  
        escreval(numeros_pares)  
    fimpara  
fimalgoritmo
```



```
0  
2  
4  
6  
8  
10  
12  
14  
16  
18  
20  
*** Fim da execução.
```

ESTRUTURA PARA

○ Exemplo: Calcular a média de 3 números

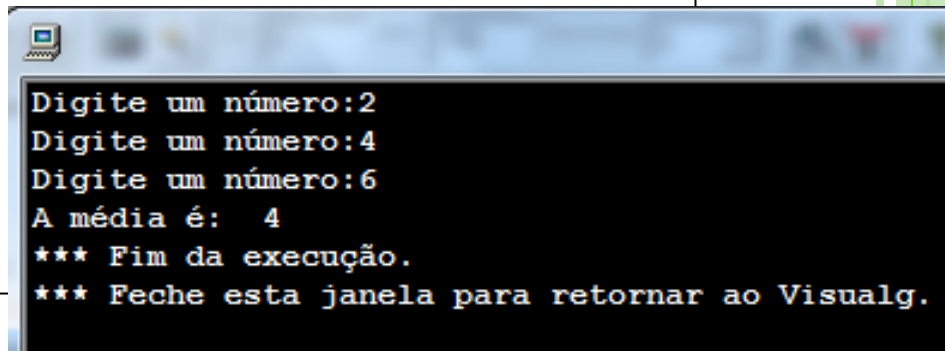
```
algoritmo "exemplo_media"  
var  
    contador: inteiro  
    numero, soma, media: real  
inicio  
    soma <- 0  
    para contador de 1 ate 3 faca  
        escreva("Digite um número:")  
        leia (numero)  
        soma <- soma + numero  
    fimpara  
    media <- soma / 3  
    escreva("A média é: ")  
    escreva(media)  
fimalgoritmo
```



ESTRUTURA PARA

○ Exemplo: Calcular a média de 3 números

```
algoritmo "exemplo_media"  
var  
    contador: inteiro  
    numero, soma, media: real  
inicio  
    soma <- 0  
    para contador de 1 ate 3 faca  
        escreva("Digite um número:")  
        leia (numero)  
        soma <- soma + numero  
    fimpara  
    media <- soma / 3  
    escreva("A média é: ")  
    escreva(media)  
fimalgoritmo
```



```
Digite um número:2  
Digite um número:4  
Digite um número:6  
A média é: 4  
*** Fim da execução.  
*** Feche esta janela para retornar ao Visualg.
```


EXERCÍCIOS

1. Escrever 20 vezes a mensagem “Eu vou tirar 6.0 na próxima prova de Algoritmos!”
2. Escrever os números ímpares de 1 a 20
3. Um cliente de supermercado fez uma compra de 20 itens. Leia o preço de cada item e calcule o total que a pessoa vai pagar.
4. Leia a quantidade de alunos existentes em uma sala de aula e para cada aluno, leia a sua média final. Ao final, escreva a quantidade de alunos que foi aprovada por média (o aluno é aprovado por média se sua média final for maior ou igual a **6,0**)



REFERÊNCIAS

- CARVALHO, Flávia Pereira de. **Apostila de Lógica de Programação – Algoritmos.** Faculdade de Informática de Taquara, 2007.
- SILVEIRA, Angélica. Algoritmos. Curso Técnico de Informática.

