



Lógica de Programação e Algoritmos

Prof. William C. Augustonelli (Billy)
05/02/2025

M2.

Estrutura Básica de Algoritmos

- ✓ Tipos de Dados, variáveis e constantes
- ✓ Operadores aritméticos, relacionais e lógicos
- ✓ Entrada e saída de dados
- ✓ Estruturas de decisão: se, se - senão, caso - se
- Estruturas de repetição: faça, faça enquanto, faça - enquanto



- **Objetivo**

- Compreender o funcionamento das estruturas de repetição (faça, faça enquanto, faça – enquanto)
- Aplicar essas estruturas em problemas computacionais
- Identificar diferenças e usos adequados para cada uma



A5.

Laços de Repetição

- Introdução às Estruturas de Repetição
- Estrutura de repetição com faça/para {for}
- Estrutura de repetição com faça enquanto {while}
- Estrutura de repetição com faça – enquanto {do – while}
- Diferenças entre as estruturas e aplicações práticas



Introdução às Estruturas de Repetição



- O que são estruturas de repetição?
 - São comandos que permitem executar um bloco de código várias vezes, de forma controlada, economizando tempo e evitando repetição manual de código

- Analogia

- Imagine que um professor vai chamar a lista com os nomes de 30 alunos

Sem repetição

```
escreva("João")  
escreva("Maria")  
escreva("Lucas")  
...
```

Com Repetição

```
Para cada aluno da lista:  
    escreva(aluno)
```

Introdução às Estruturas de Repetição

- Por que usamos repetição nos algoritmos?

Situação	Sem repetição	Com repetição
Somar 100 números	100 linhas de código	1 (faça/ para) com soma acumulada
Ler até o usuário digita 0	Condições infinitas	Faça enquanto ou Faça – enquanto
Mostrar menu de opções	Múltiplas repetições manuais	Repetição com faça enquanto

Introdução às Estruturas de Repetição

- Tipos de estruturas de repetições

Pseudo-código	Comando	Descrição	Quando usar
faça/ para	for	Executa com contador fixo	Número conhecido de repetições
faça enquanto	while	Executa com base em condição	Quando não sabemos quantas vezes
faça - enquanto	do – while	Executa pelo menos uma vez	Quando precisamos executar ao menos uma vez

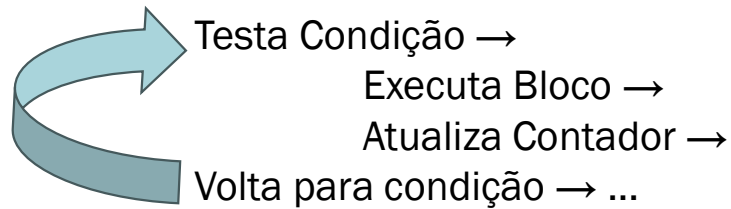
Introdução às Estruturas de Repetição

- Exemplo prático

Pseudo-código	Portugol	Python
<pre>Para i de 1 até 10 faça escreva(i) fimPara</pre>	<pre>algoritmo "Exemplo For" var i: inteiro inicio para i de 1 ate 10 faca escreval(i) fimpara fimalgoritmo</pre>	<pre>for i in range(1, 11): print(i)</pre>

Introdução às Estruturas de Repetição

- Dica visual para uma linha do tempo de repetição



Estrutura de repetição faça/ para {for}

- *O que é o for (faça ou para)?*
 - É uma estrutura de repetição controlada por contador
 - Ele é usado quando
 - Sabemos **quantas vezes** o código deve ser repetido
 - Há uma **sequência definida** (ex.: 1 até 10)



Estrutura de repetição faça/ para {for}

- Sintaxe

Pseudo-código

```
Para i de 1 até 10 faça  
    escreva(i)  
fimPara
```

Portugol

```
para i de 1 ate 10 faca  
    escreval(i)  
fimpara
```

Python

```
for i in range(1, 11):  
    print(i)
```



Estrutura de repetição faça/ para {for}

- Exemplos clássicos

Pseudo-código

```
Para i de 1 até 5 faça  
    escreva("Contando: ", i)  
fimPara
```

```
soma ← 0  
Para i de 1 até 10 faça  
    soma ← soma + i  
fimPara  
escreva("Soma = ", soma)
```

```
for i in range(1, 6):  
    print(f"Contando: {i}")
```

```
soma = 0  
for i in range(1, 11):  
    soma += i  
print("Soma =", soma)
```

Python

Estrutura de repetição faça/ para {for}

- Atividade em sala

Enunciado: Crie um algoritmo que:

Receba um número do usuário

Mostre a tabuada de 1 a 10 desse número

Pseudo-código

```
leia(n)
Para i de 1 até 10 faça
    escreva(n, " x ", i, " = ", n*i)
fimPara
```

Python

```
n = int(input("Digite um número: "))
for i in range(1, 11):
    print(f"{n} x {i} = {n*i}")
```



Estrutura de repetição faça/ para {for}

- for com salto (incremento diferente)

Você pode controlar de quanto em quanto o contador é incrementado

Exemplo: Números pares entre 2 e 20

Pseudo-código

```
Para i de 2 até 20 passo 2 faça  
    escreva(i)  
fimPara
```

Python

```
for i in range(2, 21, 2):  
    print(i)
```



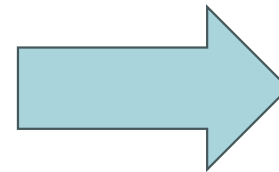
Estrutura de repetição faça/ para {for}

- for aninhado (laço dentro de laço)
 - for dentro de for

Exemplo: Imprimir matriz 3x3

Python

```
for i in range(1, 4):  
    for j in range(1, 4):  
        print(f"{i},{j}", end=" ")  
    print()
```



```
1,1 1,2 1,3  
2,1 2,2 2,3  
3,1 3,2 3,3
```



Estrutura de repetição faça enquanto {while}

- *O que é o comando while (faça enquanto)?*
 - O *while (faça enquanto)* é uma estrutura de repetição que executa um bloco de código enquanto uma condição for verdadeira
 - É usado quando
 - Não sabemos quantas vezes o laço vai se repetir
 - A repetição depende de uma condição lógica

Pseudo-código

```
enquanto <condição> faça  
    <comandos>  
fimenquanto
```

Portugol

```
enquanto <condição> faca  
    // comandos  
fimenquanto
```

Python

```
while <condição>:  
    # comandos
```



Estrutura de repetição faça enquanto {while}

- Exemplo básico: contagem com *while*
 - Exemplo 1: Contar de 1 a 5

Pseudo-código

```
i ← 1
enquanto i ≤ 5 faça
    escreva(i)
    i ← i + 1
fimenquanto
```

Python

```
i = 1
while i <= 5:
    print(i)
    i += 1
```



Estrutura de repetição faça enquanto {while}

- Cuidado com Laço Infinito!!!!

- Exemplo errado (vai travar)

Python

```
i = 1
while i <= 5:
    print(i)
# i não aumenta!
```



Estrutura de repetição faça enquanto {while}

- Atividade em sala

Enunciado: O usuário digita números até digita 0

O algoritmo deve somar todos os valores e mostrar a soma final

Pseudo - código

```
soma ← 0
leia(valor)
enquanto valor ≠ 0 faça
    soma ← soma + valor
    leia(valor)
fimenquanto
escreva("Soma final = ", soma)
```

Python

```
soma = 0
valor = int(input("Digite um número (0 para sair): "))
while valor != 0:
    soma += valor
    valor = int(input("Digite outro número: "))
print("Soma final =", soma)
```



Estrutura de repetição faça enquanto {while}

- Uso do *while* para validação

Enunciado: Solicitar um número entre 1 e 10
Se for inválido, pedir novamente

Pseudo - código

```
leia(n)
enquanto n < 1 ou n > 10 faça
    escreva("Número inválido! Digite entre 1 e 10.")
    leia(n)
fimenquanto
```

Python

```
n = int(input("Digite um número entre 1 e 10: "))
while n < 1 or n > 10:
    print("Número inválido!")
    n = int(input("Digite entre 1 e 10: "))
```



Estrutura de repetição faça enquanto {while}

- *while* com múltiplos testes

Enunciado:

Criar um menu com 3 opções, até o usuário digita 0

Pseudo - código

```
opcao ← -1
enquanto opcao ≠ 0 faça
    escreva("1 - Opção A")
    escreva("2 - Opção B")
    escreva("0 - Sair")
    leia(opcao)
fimenquanto
escreva("Fim do programa.")
```

Python

```
opcao = -1
while opcao != 0:
    print("1 - Opção A")
    print("2 - Opção B")
    print("0 - Sair")
    opcao = int(input("Escolha: "))
print("Fim do programa.")
```



Estrutura de repetição faça - enquanto {do while}

- *O que é o do – while (faça – enquanto)?*
 - do-while é uma estrutura de repetição pós-condicional
 - Executa primeiro o **bloco de comandos**
 - Depois **verifica a condição**
 - **Repete** se a condição ainda for verdadeira
- **Ideal para**
 - Menus de opções
 - Validação de entrada com execução mínima
 - Situações onde a **primeira execução é obrigatória**



Estrutura de repetição faça - enquanto {do while}

- Sintaxe

Pseudo-código

```
repita  
    <comandos>  
até <condição>
```

Portugol

```
repita  
    // comandos  
ate <condição>
```

Python

```
while True:  
    # comandos  
    if condicao:  
        break
```

OBS: O Python não possui *do-while*, mas podemos simular com *while true* e *break*

Estrutura de repetição faça - enquanto {do while}

- Exemplo clássico
 - Enunciado: Ler até digitar 0

Pseudo-código

```
repita  
    leia(valor)  
    escreva("Você digitou: ", valor)  
até valor != 0
```

Python

```
while True:  
    valor = int(input("Digite um número (0 para sair): "))  
    print(f"Você digitou: {valor}")  
    if valor == 0:  
        break
```



Estrutura de repetição faça - enquanto {do while}

- Atividade em sala

- Enunciado:

Criar um menu com as opções:

1. Mostrar data atual
2. Mostrar mensagem
3. Sair

Pseudo-código

```
repita
    escreva("1 - Mostrar data")
    escreva("2 - Mostrar mensagem")
    escreva("0 - Sair")
    leia(opcao)
    se opcao = 1 então
        escreva("Hoje é 23/04/2025")
    senão se opcao = 2 então
        escreva("Bem-vindo ao sistema!")
    fimse
até opcao = 0
```

Python

```
while True:
    print("1 - Mostrar data")
    print("2 - Mostrar mensagem")
    print("0 - Sair")
    opcao = int(input("Escolha: "))

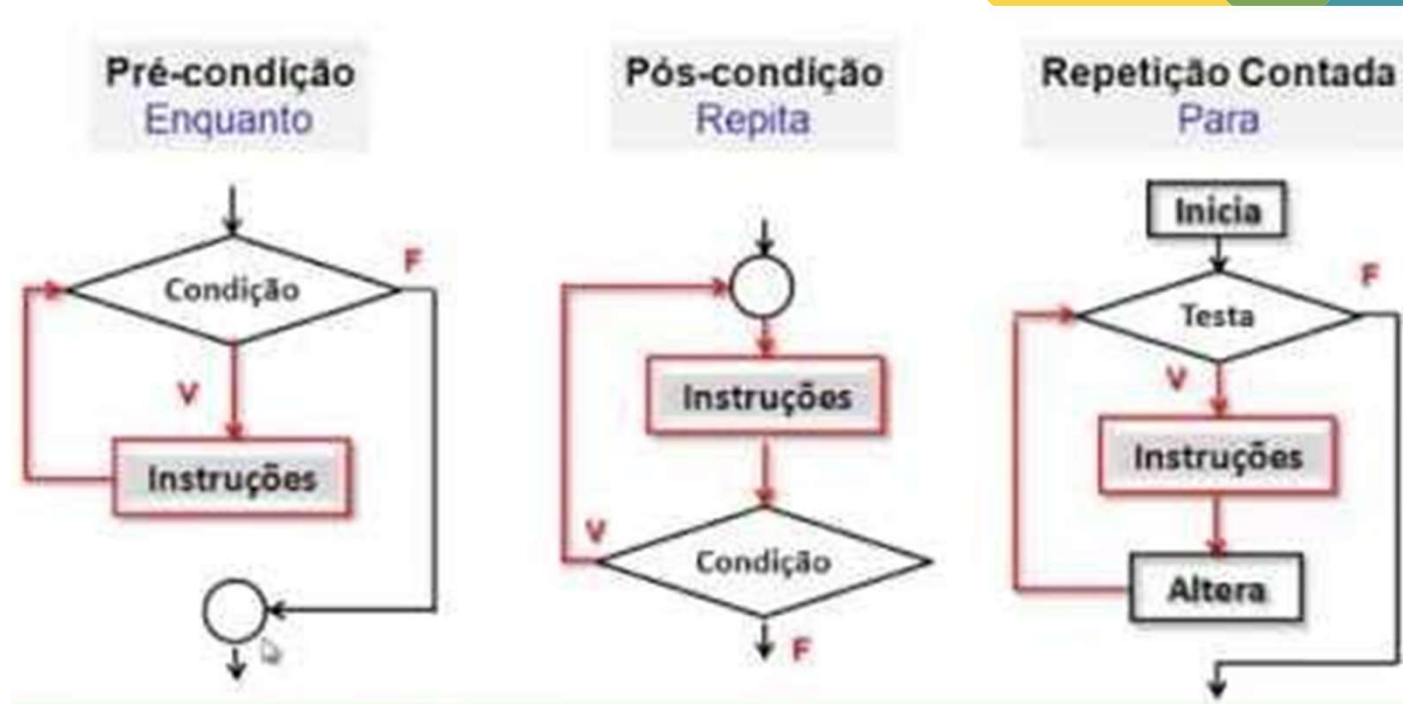
    if opcao == 1:
        print("Hoje é 23/04/2025")
    elif opcao == 2:
        print("Bem-vindo ao sistema!")
    elif opcao == 0:
        break
```

Resumo: Diferenças entre as estruturas e aplicações práticas



Critério	for	while	do-while
Controle por contador	Sim	Opcional	Opcional
Testa a condição no início	Sim	Sim	Não – executa o bloco pelo menos uma vez a condição é testada no final do bloco
Uso comum	Repetições com contador conhecido	Repetições com condição lógica	Execução obrigatório de 1 iteração
Exemplo clássico	Tabuada, soma de N números	Validação de entrada	Menus e senhas com execução obrigatória

Estrutura de repetição Fluxograma



Atividades



1. Qual dos três você achou mais legível e mais produtivo?
2. Qual estrutura te dá mais segurança para evitar laços infinitos?
3. Em qual situação você usaria do-while, sem ser para menu ou validação de senha?
4. Criar um algoritmo que
 - Mostre um menu com 3 opções:
 - Para a opção 1, peça um número e diga se é par ou ímpar
 - Para a opção 2, peça dois números e calcule a média
 - Repita até o usuário escolher a opção 0
 - Implemente utilizando while e do-while