

Etapa 5 – Estruturas de Repetição (Loops)

Trilha: Lógica de Programação

O que são Estruturas de Repetição?

- São usadas para **executar um mesmo conjunto de instruções várias vezes.**
- Evitam **repetição manual** de comandos.
- Permitem que o algoritmo seja mais **eficiente e dinâmico.**
- **Exemplo:**

Exibir uma contagem de 1 até 10 automaticamente.

Tipos de Estruturas de Repetição

- 1. Enquanto (While Loop)** — repete enquanto a condição for verdadeira.
- 2. Repita...até (Do...Until Loop)** — repete até que a condição seja verdadeira.
- 3. Para (For Loop)** — repete um número determinado de vezes.

Cada uma delas é útil em situações diferentes.

Estrutura “Enquanto” (While Loop)

- **Sintaxe em pseudocódigo:**

```
enquanto (condicao) faca
```

```
    // ações repetidas
```

```
fimenquanto
```

- **Exemplo:**

```
inicio
```

```
    contador ← 1
```

```
    enquanto contador <= 5 faca
```

```
        escreva contador
```

```
        contador ← contador + 1
```

```
fimenquanto
```

```
fim
```

- **Saída:**

1

2

3

4

5

Como Funciona o “Enquanto”

- A **condição** é **testada antes** de executar o bloco.
- Se for **verdadeira**, o bloco executa.
- Se for **falsa**, o laço termina.
- Dica: Use quando **não sabe exatamente quantas vezes** o loop vai se repetir.
- **Exemplo prático:**

Ler notas até que o usuário digite “0”.

Estrutura “Repita...até” (Do...Until Loop)

- **Sintaxe:**

```
repita  
  // ações  
  ate (condicao)
```

- **Exemplo:**

```
inicio  
    contador ← 1  
    repita  
        escreva contador  
        contador ← contador + 1  
    ate (contador > 5)  
fim
```

- **Saída:**

```
1  
2  
3  
4  
5
```

Diferença Entre “Enquanto” e “Repita”

Estrutura	Teste da Condição	Execução Mínima
enquanto	Antes do bloco	Pode não executar
repita...ate	Depois do bloco	Executa ao menos uma vez

- **Exemplo de uso:**

repita...ate é ideal para menus que devem aparecer ao menos uma vez.

Estrutura “Para” (For Loop)

- **Sintaxe:**

```
para contador de inicio ate fim faca  
    // ações
```

```
fimpara
```

- **Exemplo:**

```
inicio
```

```
    para i de 1 ate 5 faca  
        escreva "Valor:", i  
    fimpara
```

```
fim
```

- **Saída:**

```
Valor: 1  
Valor: 2  
Valor: 3  
Valor: 4  
Valor: 5
```

Como Funciona o “Para”

- Executa um número **fixo de repetições**.
- O contador é **incrementado automaticamente** a cada iteração.
- Ideal para **listas, contagens e cálculos repetitivos**.
- **Exemplo prático:**

Calcular a soma de 1 até 100.

Controle de Incremento (Step)

- O passo (step) define quanto o contador aumenta ou diminui.

- **Exemplo crescente:**

para i de 1 ate 10 passo 2 faca

 escreva i

fimpara

- **Exemplo decrescente:**

para i de 10 ate 1 passo -1 faca

 escreva i

fimpara

- **Saída:**

10, 9, 8, 7... até 1

Interrupção de Repetição

- Alguns algoritmos permitem **interromper o loop antes do final**.
- Usa-se o comando “saia” (**break**) para encerrar prematuramente.
- Exemplo:

inicio

```
    para i de 1 ate 10 faca
        se i = 5 entao
            saia
        fimse
        escreva i
    fimpara
```

fim

- Saída:

1

2

3

4

Erros Comuns

- Esquecer de atualizar a variável de controle (em enquanto).
- Criar loops infinitos (condição nunca falsa).
- Não usar **parênteses** em condições complexas.
- Dica: Sempre garanta que o loop **tenha uma forma de terminar**.

Exemplo Completo

algoritmo "tabuada"

inicio

 leia numero

 para i de 1 ate 10 faca

 resultado ← numero * i

 escreva numero, "x", i, "=", resultado

 fimpara

fim

- **Saída (para número = 3):**

3 x 1 = 3

3 x 2 = 6

3 x 3 = 9

...

3 x 10 = 30

Dica de Lógica

- Use para → quando souber o número de repetições.
- Use enquanto → quando depender de uma condição.
- Use repita → quando precisar executar **pelo menos uma vez**.

Conclusão

- As estruturas de repetição tornam os algoritmos **automatizados e eficientes**.
- São fundamentais para criar **loops controlados e inteligentes**.
- Preparação para a próxima etapa: **Vetores e Matrizes (Arrays and Matrices)**.