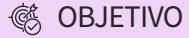


Resolvendo Problemas com Repetição

Aprenda a usar laços de repetição para resolver problemas do mundo real com eficiência e elegância.

Eduardo Ogasawara
eduardo.ogasawara@cefet-rj.br
<https://eic.cefet-rj.br/~eogasawara>



OBJETIVO

Repetição Resolve Problemas Grandes

Somar Valores

Processar vários números de forma automática e eficiente.

Contar Pontos

Acumular resultados ao longo de múltiplas iterações.

Calcular Médias

Processar conjuntos de dados e obter estatísticas.

Muitos problemas do mundo real envolvem processar grandes quantidades de valores. Laços de repetição são a solução.

Problema: Somar Números

Desafio

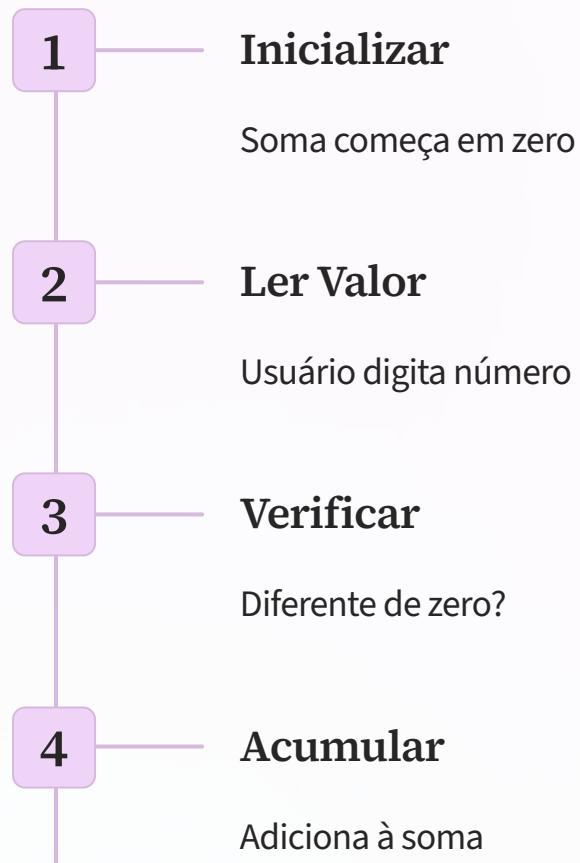
Queremos somar vários números digitados pelo usuário. Não sabemos quantos serão. O usuário digita até zero para encerrar.

Solução

Vamos usar **enquanto** para repetir até que o usuário digite zero.

Código: Soma com Enquanto

```
soma <- 0  
  
leia(n)  
  
enquanto n <> 0 faça  
    soma <- soma + n  
    leia(n)  
fim enquanto  
  
escreva(soma)
```



Passo a Passo da Execução

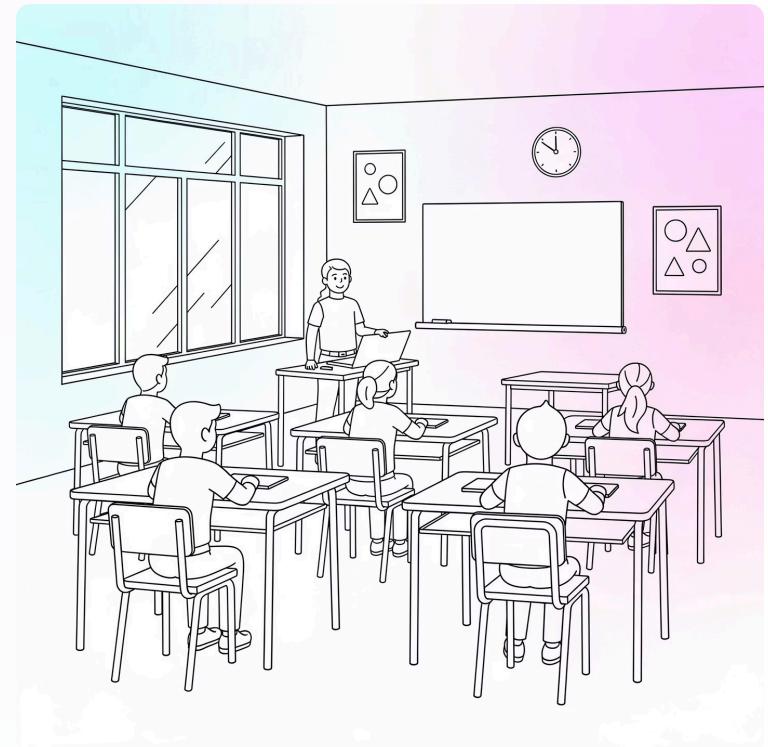
- 1 Usuário digita 3
 soma vira 3
- 2 Usuário digita 5
 soma vira 8
- 3 Usuário digita 2
 soma vira 10
- 4 Usuário digita 0
 Laço para, mostra 10

Problema: Média da Turma

Novo Desafio

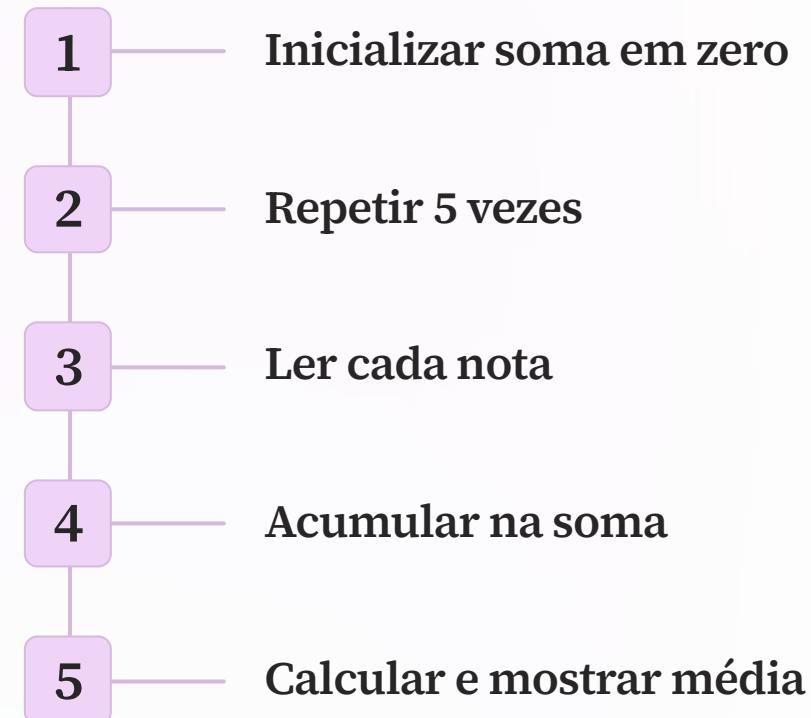
Queremos calcular a média de 5 alunos. Sabemos que são exatamente 5 notas.

Usamos **para** porque conhecemos a quantidade de repetições. Vamos somar todas as notas e dividir por 5.



Código: Média com Para

```
soma <- 0  
  
para i de 1 ate 5 faca  
    leia(nota)  
    soma <- soma + nota  
fimpara  
  
media <- soma / 5  
escreva(media)
```





💡 CONCEITO-CHAVE

Usando Acumulador

O que é?

A variável **soma** guarda o total acumulado ao longo das iterações.

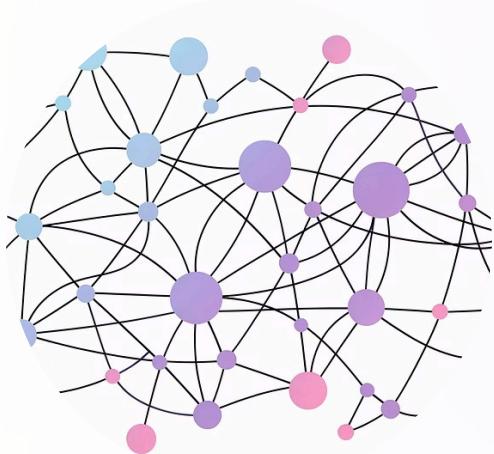
Como funciona?

Ela começa em zero e vai aumentando a cada repetição do laço.

Por que usar?

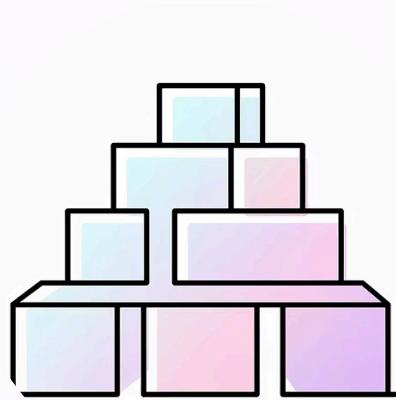
Acumuladores são essenciais para guardar resultados parciais durante a repetição.

Por Que Isso Importa



Processar Muitos Dados

Sem repetição, não poderíamos lidar com grandes volumes de informação.



Programas Robustos

Laços tornam o programa forte e capaz de resolver problemas complexos.



Escalabilidade

Eles resolvem problemas grandes com código simples e elegante.

O Que Aprendemos



Somar com Laços

Usamos laços para acumular valores e calcular somas.



Calcular Médias

Processamos múltiplos valores e calculamos estatísticas.



Usar Acumuladores

Variáveis que guardam resultados parciais durante a repetição.



Resolver Problemas

Agora sabemos repetir para resolver problemas do mundo real.

EXERCÍCIO 1

Contar de 1 a 10

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

O que faz?

Esse programa mostra os números de 1 a 10, um por vez, usando um laço **para** simples.

EXERCÍCIO 2

Números Ímpares

```
para i de 1 ate 9 passo 2 faca  
    escreva(i)  
fimpara
```

1

Saída

1 3 5 7 9

Passo 2



O laço pula de 2 em 2, mostrando apenas ímpares.

 EXERCÍCIO 3

Tabuada

```
leia(n)
```

```
para i de 1 ate 10 faca
```

```
    escreva(n * i)
```

```
fimpara
```

Funcionalidade

Mostra a tabuada do número digitado pelo usuário,
multiplicando de 1 até 10.

Fatorial

```
leia(n)
fat <- 1

para i de 1 ate n faça
    fat <- fat * i
fimpara

escreva(fat)
```

1

2

3

4

Ler número

Usuário informa o valor

Iniciar factorial

fat começa em 1

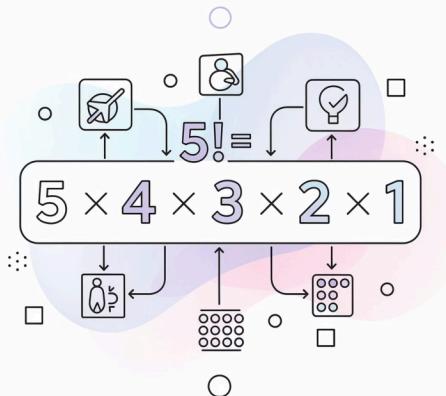
Multiplicar sequência

Multiplica de 1 até n

Mostrar resultado

Exibe o fatorial calculado

Testando o Fatorial



5

Entrada

Número digitado pelo usuário

120

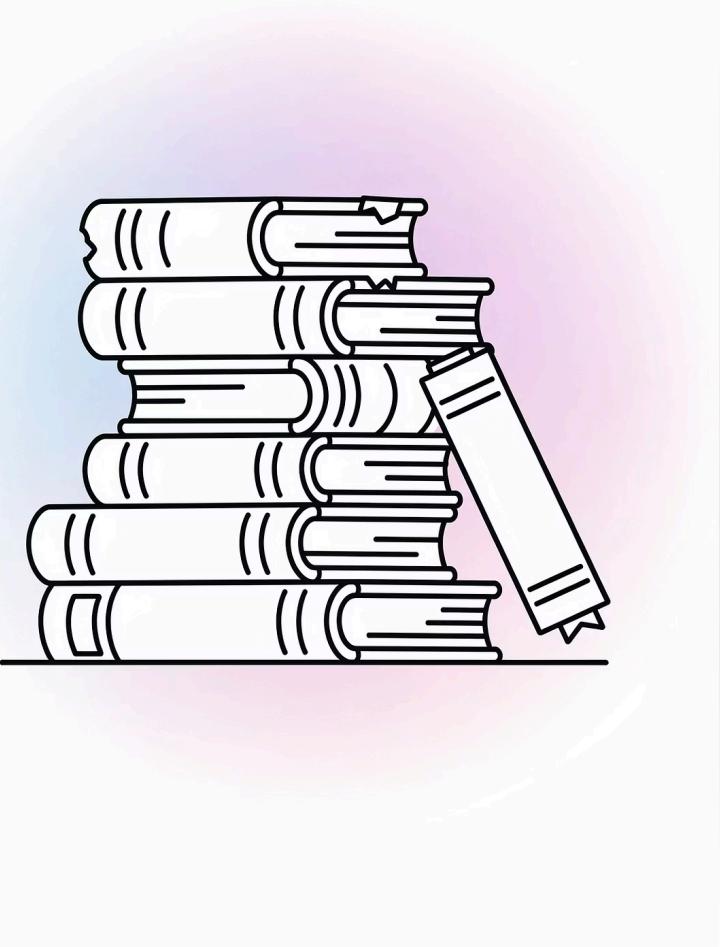
Saída

Resultado do cálculo

Cálculo Passo a Passo

$$1 \times 2 \times 3 \times 4 \times 5 = 120$$

O programa multiplica todos os números de 1 até 5 e mostra o resultado final: **120**



Referências

- 1 WING, Jeannette M. Computational thinking. Communications of the ACM, New York, v. 49, n. 3, p. 33–35, 2006.
- 2 PAPERT, Seymour. Mindstorms: children, computers, and powerful ideas. New York: Basic Books, 1980.
- 3 PÓLYA, George. How to solve it: a new aspect of mathematical method. 2. ed. Princeton: Princeton University Press, 1957.
- 4 CAMPOS, A. F. G. A.; CAMPOS, E. A. V. Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ (padrão ANSI) e Java. 3. ed. São Paulo: Pearson, 2012.