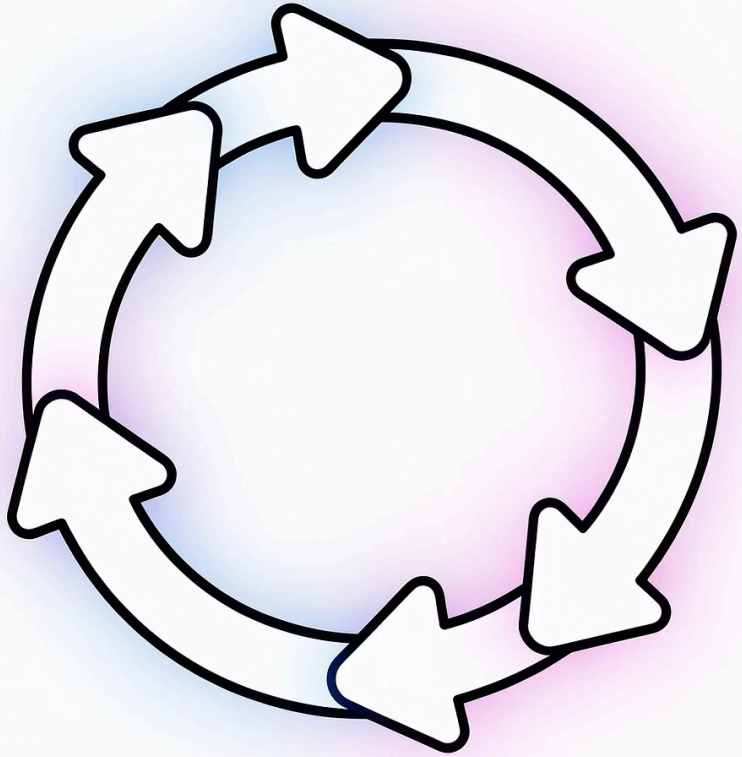


CEFET/RJ

Repetir até uma condição (enquanto)

Aprenda a usar condições para controlar quando o laço para e resolver problemas reais de programação.

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

CONCEITO

Nem sempre sabemos quantas vezes

Às vezes não sabemos quantas repetições serão necessárias. Por exemplo, até o usuário digitar zero ou até acertar um número.

Para isso usamos **enquanto**.

SINTAXE

O comando enquanto

01

Testa a condição

O computador verifica se a condição é verdadeira

02

Executa o bloco

Se verdadeira, executa os comandos dentro

03

Testa novamente

Volta ao início e testa a condição outra vez

04

Para quando falsa

Quando a condição fica falsa, o laço termina

```
enquanto condicao faca
 comandos
fimenquanto
```

Exemplo simples

```
contador <- 1

enquanto contador <= 5 faca
  escreva(contador)
  contador <- contador + 1
fimenquanto
```

Aqui funciona como o **para**, mas a condição controla o laço.

Quando passa de 5, para. O computador verifica sempre.

EXECUÇÃO

Passo a passo

contador = 1

Primeiro contador vale 1. É menor ou igual a 5, então entra.

1

2

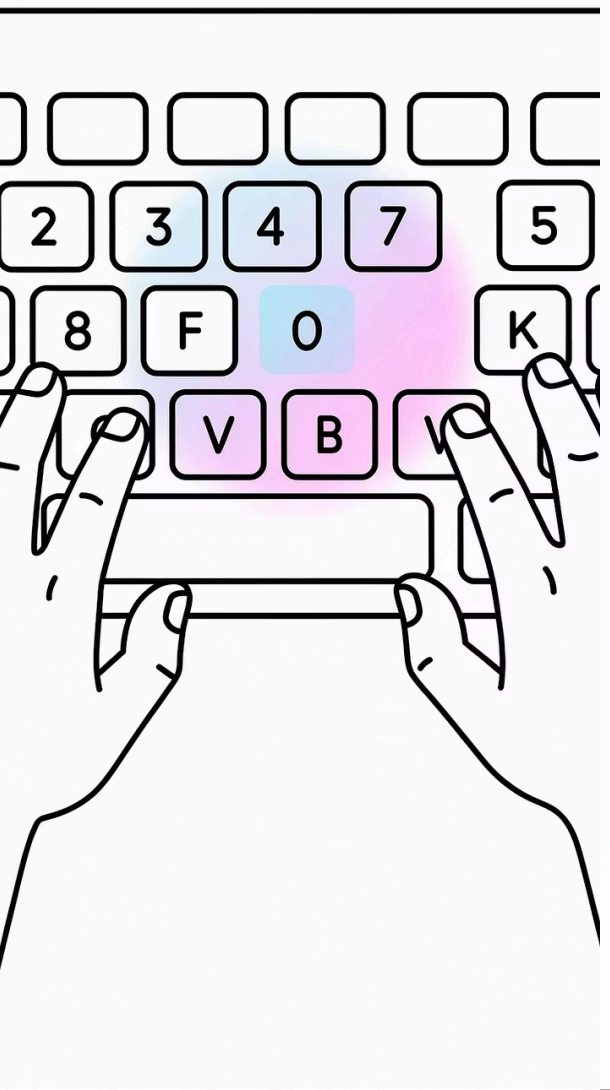
contador = 2, 3, 4, 5

Depois aumenta para 2, 3, 4 e 5. Continua executando.

contador = 6

Quando vira 6, a condição falha e o laço para.

3



Problema real

Desafio

Queremos ler números até o usuário digitar 0

Incerteza

Não sabemos quantos números virão

Solução

Mas sabemos quando parar: isso é perfeito para enquanto

CÓDIGO

Implementação

```
leia(n)

enquanto n <> 0 faca
  escreva(n)
  leia(n)
fimenquanto
```

1

Lê um número

O computador lê o primeiro valor

2

Testa condição

Se não for zero, entra no laço

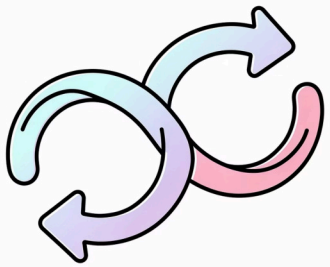
3

Mostra e repete

Exibe o número e lê outro

O que acontece

Se o usuário digitar 5, entra. Depois 3, entra. Depois 0, para. A condição controla tudo.



⚠ ATENÇÃO

Evitar laço infinito

Erro comum: Se esquecermos de mudar a variável, a condição nunca muda.

O laço nunca para. Isso é um erro comum que deve ser evitado!

BOAS PRÁTICAS

Testar sempre

1

Teste a entrada

Veja se o laço entra corretamente

2

Teste a execução

Verifique se os comandos funcionam

3

Teste a saída

Confirme se o laço para quando deve

Isso evita problemas e garante que seu código funcione perfeitamente.

O que aprendemos

Repetição inteligente

Usamos enquanto para repetir com controle

Condição decide

A condição decide quando parar

Resolve problemas

Isso resolve muitos problemas reais

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.