



Fazendo o "Chinês"

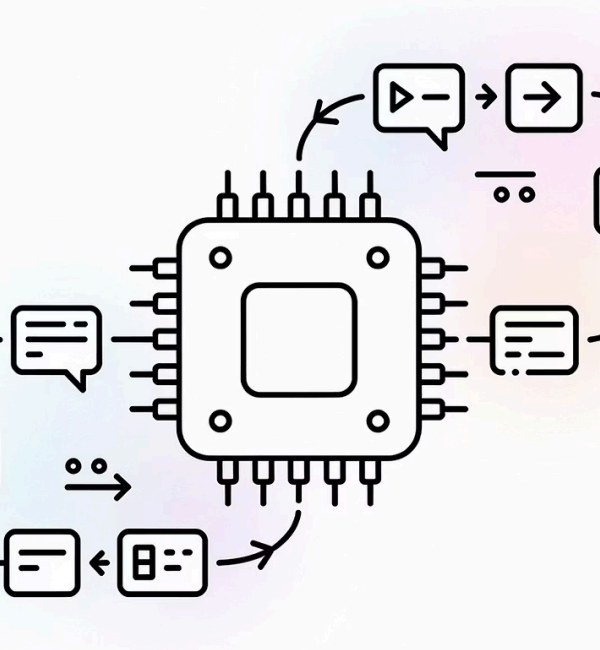
Execução Passo a Passo de Algoritmos

Aprenda a executar algoritmos linha por linha, exatamente como o computador faz.

Eduardo Ogasawara

eduardo.ogasawara@cefet-rj.br

<https://eic.cefet-rj.br/~eogasawara>



 **CONCEITO FUNDAMENTAL**

O Que É Fazer o "Chinês"?

Execução Linear

O computador não pula linhas — ele executa uma de cada vez, na ordem.

Sem Memória do Passado

Ele não lembra valores antigos, apenas o estado atual das variáveis.


Valores Atuais

Trabalha exclusivamente com o valor presente de cada variável.

O Pseudocódigo

Vamos acompanhar a execução deste algoritmo simples:

```
x <- 5  
x <- x + 2  
se x > 6 entao  
  escreva(x)  
senao  
  escreva(0)  
fimse
```

 **Importante:** Não vamos modificar o código. Vamos apenas observar o que acontece com as variáveis durante a execução.

Montando a Tabela de Variáveis

A tabela de execução registra cada mudança nas variáveis:

- Cada linha executada gera uma nova entrada
- Variável só tem um valor por vez
- Começamos com a tabela vazia

Este é o ritual mental essencial para dominar algoritmos.

Linha executada	x
(início)	—

Executando as Primeiras Linhas

Linha 1: $x \leftarrow 5$

O símbolo \leftarrow significa "guardar". A variável x passa a valer 5.

Linha	x
$x \leftarrow 5$	5

1

2

Linha 2: $x \leftarrow x + 2$

O computador lê o valor atual de x (5), calcula $5 + 2$, e guarda o resultado (7) em x . O valor 5 deixa de existir.

Linha	x
$x \leftarrow 5$	5
$x \leftarrow x + 2$	7

📌 **Atenção:** Não existe "x antigo" e "x novo" — apenas o valor atual!

Executando a Decisão

Código destacado:

```
se x > 6 entao
```

Decisões **não mudam valores** — elas apenas escolhem o caminho.

Como $7 > 6$ é **verdadeiro**, seguimos pelo "então".

Linha executada	x
x <- 5	5
x <- x + 2	7
se x > 6	7



Saída Final e Conclusão

7

Resultado

O comando escreva(x) exibe o valor atual de x.

Fazer o chinês é:

- Acompanhar variáveis
- Linha por linha
- Sem pular nada

Conclusão pedagógica: Quem sabe fazer chinês nunca se perde em laços.



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.