

Conectando Portugol e Scratch

A mesma ideia, duas formas de expressar. Um algoritmo pode ser escrito ou montado com blocos. A decisão é a mesma. A lógica não muda.

De Onde Vem o Valor da Variável?

Do Código

O programador define o valor diretamente no algoritmo via atribuição

idade <- 12

No Scratch, usa-se o comando mude para fazer a atribuição



Do Usuário

O valor é fornecido por meio de entrada durante a execução, pela leitura

leia idade

No Scratch, usa-se o comando pergunte e mude para fazer a leitura



O algoritmo pode rodar várias vezes, com valores diferentes a cada execução.

TIPO 1

Decisão Simples em Portugol

```
leia idade
```

```
se idade > 10 entao
```

```
    escreva("Maior que 10")
```

```
fimse
```

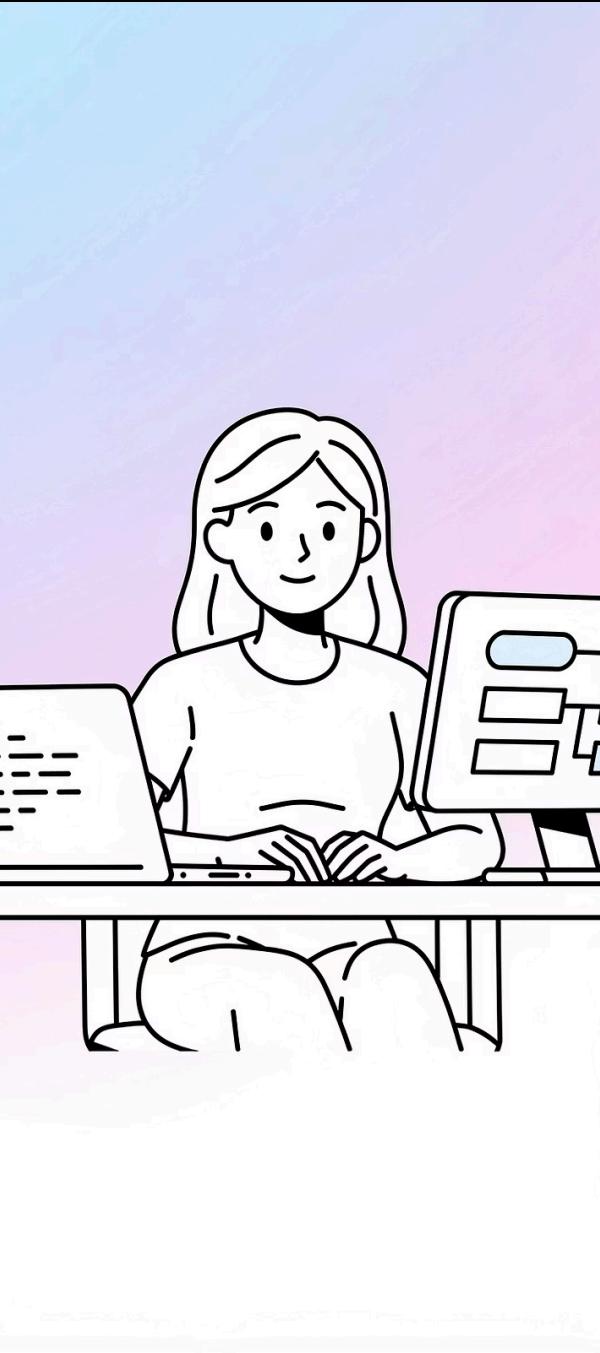
Pergunta importante: O que acontece se a condição for falsa?

A Mesma Decisão no Scratch

Veja como a lógica se traduz em blocos visuais (pergunte, mude, se e diga):

A estrutura visual facilita o entendimento, mas a lógica permanece idêntica





Testando a Decisão Simples



Caso 1: $idade = 12$

Condição verdadeira

✓ Mensagem aparece



Caso 2: $idade = 8$

Condição falsa

✗ Nada acontece



Isso não é erro. É uma decisão simples funcionando corretamente.

TIPO 2

Decisão Completa em Portugol

leia pontos

```
se pontos >= 100 entao
    escreva("Você venceu")
senao
    escreva("Tente de novo")
fimse
```

Agora existem dois caminhos possíveis.

A Mesma Decisão no Scratch

A estrutura SE-SENÃO em blocos visuais a partir dos comandos: pergunte, mude, se, diga, senão e diga.

Observe como o bloco "senão" cria um caminho alternativo para quando a condição é falsa.



Testando os Dois Caminhos

Caso 1: pontos = 120

✓ Condição verdadeira

✓ Executa o "então"

Mensagem: "Você venceu"

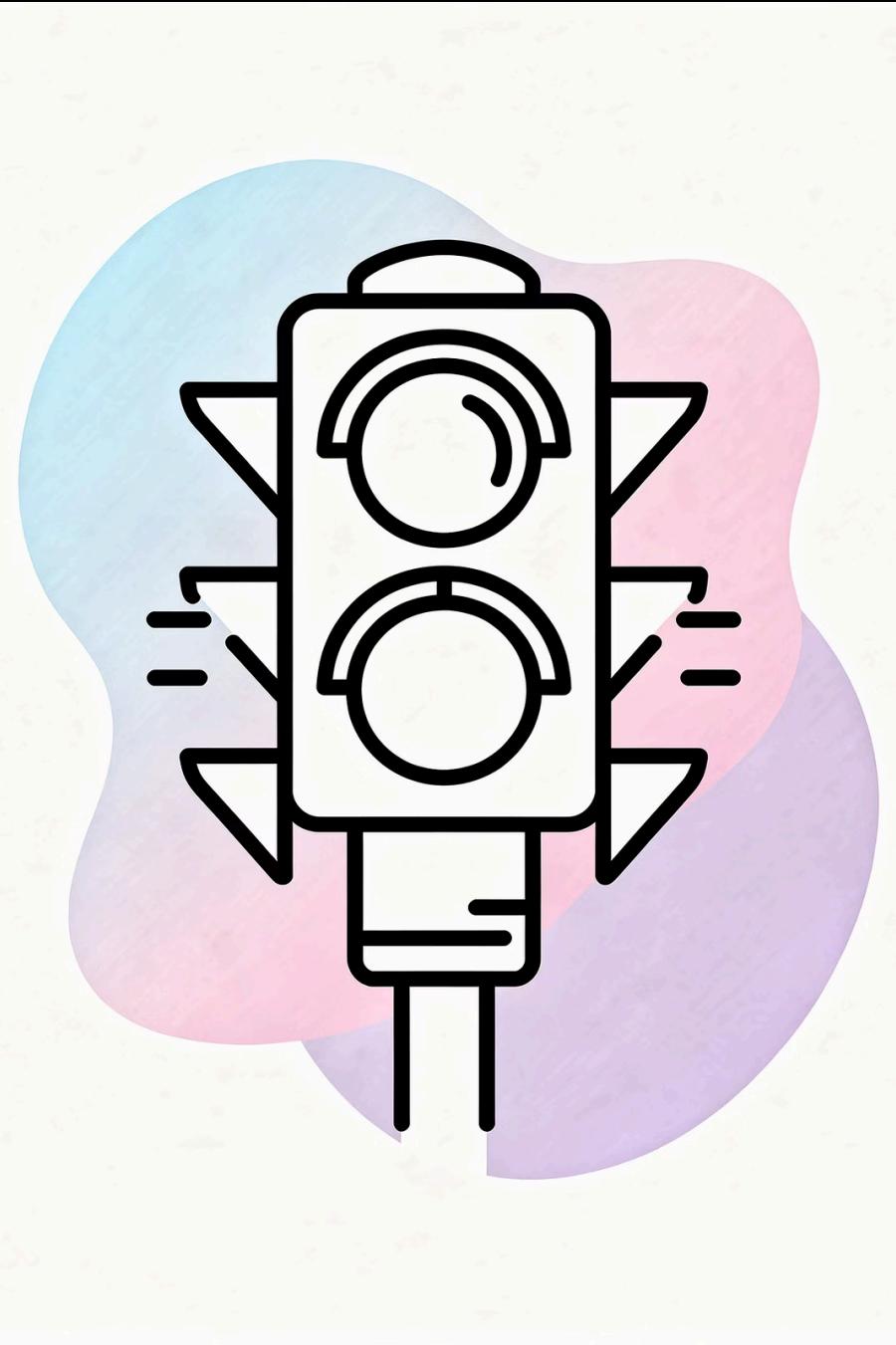
Caso 2: pontos = 80

✗ Condição falsa

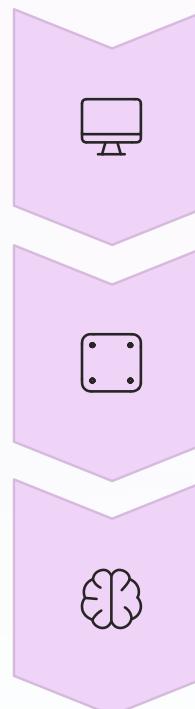
✓ Executa o "senão"

Mensagem: "Tente de novo"

 **Importante:** Somente um caminho executa por vez.



O Caminho Ignorado



Não executa os dois lados

O computador escolhe apenas um caminho

O outro caminho não roda

O código não escolhido é completamente ignorado

Isso torna o programa inteligente

Decisão é controle de fluxo

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.