

Pseudocódigo

Lucas Araújo Campos Szuster

23 de dezembro de 2025

Sumário

1. Fundamentos

2. Identação

3. Regras gerais

3.1 Especificações

4. Exemplos

5. Exercícios

O que é pseudocódigo?

O pseudocódigo é uma forma simplificada de descrever algoritmos e programas, sem a necessidade de seguir rigorosamente a sintaxe de uma linguagem de programação específica.

Principais características do pseudocódigo:

- Leitura sequencial, de cima para baixo;
- Descrição clara e geral, evitando detalhes técnicos excessivos;
- Uso de comandos explícitos para indicar início e fim das estruturas;
- **Indentação** semelhante à utilizada em códigos reais.

Indentação [parte 1]

Indentação é definida da seguinte forma o afastamento de um texto em relação à margem, que é utilizado para facilitar a leitura de um código qualquer.

Não identado

```
#include <iostream>

int main() {
    std::cout << "teste de código";

    try {
        std::cout << 1/0;
    } catch(const std::exception& e) {
        std::cout << "erro";
    }
}
```

Identado

```
#include <iostream>

int main() {
    std::cout << "teste de código";

    try {
        std::cout << 1/0;
    } catch(const std::exception& e) {
        std::cout << "erro";
    }
}
```

Indentação [parte 2]

Não identado

```
#include <iostream>

int main() {
    std::cout << "teste de código";

    try {
        std::cout << 1/0;
    } catch(const std::exception& e) {
        std::cout << "erro";
    }
}
```

Identado

```
#include <iostream>

int main() {
    std::cout << "teste de código";

    try {
        std::cout << 1/0;
    } catch(const std::exception& e) {
        std::cout << "erro";
    }
}
```

Observação

Mesmo sem qualquer conhecimento sobre C++, a linguagem utilizada no código acima, ou sobre a funcionalidade do programa apresentado, o exemplo à direita é significativamente mais claro e legível.

Indentação [parte 3]

Portanto, o pseudocódigo deve ser devidamente identado para garantir uma leitura clara e intuitiva, facilitando a compreensão da lógica do algoritmo. A indentação é usada para indicar a quais blocos de código cada comando pertence dentro das diferentes estruturas.

Por exemplo, com a indentação correta, torna-se imediatamente evidente quais comandos serão executados quando a condição de um bloco condicional for verdadeira.

Observação

Não veremos isto neste instante, mas a linguagem de programação Python nos obriga a identar corretamente todo código escrito.

Dados utilizados

Para descrever corretamente um algoritmo na forma de pseudocódigo, utilizamos dados identificados por nomes simbólicos, de maneira semelhante ao que ocorre na matemática.

Por exemplo:

- Ao descrever um algoritmo que calcula uma função matemática $f(x)$, podemos representar o valor de entrada por x ;
- Em um algoritmo que armazena um dito popular, podemos representar esse dado no pseudocódigo pelo identificador *ditado*;
- De forma geral, todo dado utilizado em um algoritmo recebe um nome, o que facilita a leitura, a compreensão e a organização do pseudocódigo.

Operadores utilizados

Para que o pseudocódigo descreva corretamente o comportamento desejado, utilizamos diferentes operadores responsáveis por realizar operações e manipular os dados de um algoritmo.

Por exemplo:

- Em algoritmos que lidam com valores numéricos, podemos utilizar operadores como $+$, $-$, \times , $/$, $\%$, entre outros, para representar operações aritméticas;
- Para atribuir um valor a uma variável, é comum utilizar uma seta (\leftarrow), indicando que um dado recebe determinado valor;
- Outros operadores podem ser utilizados conforme a necessidade do algoritmo.

Comandos utilizados

Para que o pseudocódigo descreva corretamente o comportamento desejado, utilizamos comandos específicos que representam as principais estruturas algorítmicas, como condicionais, estruturas de repetição e outros tipos de controle de fluxo.

Por exemplo:

- Para representar uma estrutura condicional, utiliza-se o comando **se**;
- Para representar uma estrutura de repetição controlada por uma condição, utiliza-se o comando **enquanto**;
- Outros comandos podem ser utilizados conforme a necessidade do algoritmo.

Leitura de dados [entradas]

Para a maioria dos algoritmos, é desejável que eles sejam capazes de operar sobre um amplo domínio de dados do mesmo tipo. Por exemplo, ao criar um algoritmo que multiplica dois números, não faz sentido especificar valores fixos; em vez disso, o algoritmo deve ler esses dados de uma fonte externa, tornando a operação genérica e aplicável a quaisquer dois números.

Para isto, será utilizado um comando que indica leitura de dados, o seguinte:

Leia x

Escrita de dados [saídas]

Para a maioria dos algoritmos, é desejável que eles sejam capazes de exibir seus resultado, pois só assim podemos verificar se estão corretos (especialmente quando adaptados para código real). Por exemplo, um algoritmo que multiplica dois números deve ser capaz de mostrar o resultado da multiplicação ao final da execução.

Para isto, será utilizado um comando que indica escrita de dados, o seguinte:

Escreva x

Exemplo 1 [parte 1]

Aqui está um simples algoritmo que calcular a soma entre dois diferentes números, α e β :

Leia α

Leia β

resultado $\leftarrow (\alpha + \beta)$

Escreva resultado

Observação importante

Note que nenhuma linha está identada. A indentação é utilizada para indicar escopos, blocos condicionais e estruturas de repetição, e não há nenhuma destas estruturas neste caso

Exemplo 1 [parte 2]

Aqui está uma leve simplificação sobre o algoritmo previamente apresentado, lendo α e β em uma só linha:

Leia α, β

resultado $\leftarrow (\alpha + \beta)$

Escreva resultado

Observação importante

Note que, novamente, nenhuma linha está identada. A indentação é utilizada para indicar escopos, blocos condicionais e estruturas de repetição, e não há nenhuma destas estruturas neste caso

Exemplo 2

Aqui está um simples algoritmo que lê um número α e o incrementa por 1:

Leia α

$\alpha \leftarrow \alpha + 1$

Escreva α

Observação importante

A lógica deste algoritmo é extremamente importante, sendo essencial para diversas operações em programação. Observe que a linha “ $\alpha \leftarrow \alpha + 1$ ” significa que α recebe o valor que já possuía acrescido de um, indicando um incremento.

Exercício 1

Escreva um algoritmo que seja capaz de calcular a equação de Heron para quaisquer valores de a , b e c . Esta é a equação de Heron:

$$A = \sqrt{p(p - a)(p - b)(p - c)}$$

$$p = \frac{a + b + c}{2}$$

Solução [parte 1]

Para solucionar isto, primeiro é necessário ler o valor de a , b e c . Isto é feito da seguinte forma:

Leia a, b, c

Com os valores de a , b e c , é possível calcular o valor de p :

$p \leftarrow (a + b + c)/2$

Solução [parte 2]

Com todos os valores determinados, podemos calcular A :

$$A \leftarrow \sqrt{p(p - a)(p - b)(p - c)}$$

Por fim, basta somente escrever o valor de A :

Escreva A

Solução [parte 3]

Assim, este é o algoritmo completo:

Leia a, b, c

$$p \leftarrow (a + b + c)/2$$

$$A \leftarrow \sqrt{p(p - a)(p - b)(p - c)}$$

Escreva A
