Noções de algoritmos - Aula 1

Alexandre Diehl

Departamento de Física – UFPel

Definição de algoritmo

→ Sequência ordenada e finita de operações para a realização de uma tarefa.

Tarefa: Experimento de Física I.

- Passo 1: Reunir os equipamentos necessários
- Passo 2: Montar o aparato experimental.
- Passo 3: Realizar o experimento.
- Passo 4: Fazer o relatório do experimento.
- Passo 5: Entregar o relatório para o professor.

Definição de algoritmo

→ Sequência ordenada e finita de instruções ou operações para a solução de um problema computacional.

Tarefa: Calcular a média das idades dos alunos da turma.

- Passo 1: Ler as idades de cada aluno.
- Passo 2: Calcular a média das idades.
- Passo 3: Apresentar o resultado numérico para a média.

Tipos de algoritmos

Um algoritmo é representado de 3 formas:

1. Descrição narrativa: dois exemplos anteriores...
2. Pseudocódigo: forma escrita codificada

```
algoritmo
declare N1, N2, N3, M numerico
leia N1
leia N2
leia N3
M ← (N1+N2+N3)/3
escreva "A idade média é:", M
fim_algoritmo
```

3. Fluxograma: forma gráfica codificada

Tipos de algoritmos

- Fluxograma ou Diagrama de Blocos:

  → forma padronizada e gráfica de representação dos passos do algoritmo.

	Início e final do fluxograma.
	Indica o fluxo de dados e conecta os símbolos existentes.
	Operação de entrada de dados (via teclado ou através de arquivos de entrada).
	Operação de entrada de dados (via teclado ou através de arquivos de entrada).

Tipos de algoritmos

- Fluxograma ou Diagrama de Blocos:

  → forma padronizada e gráfica de representação dos passos do algoritmo.

	Utilizado para indicar cálculos (algoritmos) que serão realizados, atribuições de valores para variáveis.
	Ponto de decisão.
	Saída de dados numa impressora ou arquivo de saída.
	Informações exibidas por dispositivos visuais, vídeos ou monitor.

Notação para o pseudocódigo

Num programa de computador temos 2 estruturas básicas:

1. Variáveis e Constantes: espaços reservados na memória do computador para armazenar elementos de um certo conjunto ou tipo de dados.
   - Variáveis: durante a execução do programa, o conteúdo da variável pode mudar;
   - Constantes: o valor de uma constante não muda durante a execução do programa.
2. Expressões: durante a execução, combinam os valores armazenados nas variáveis e constantes para calcular novos valores.

Notação para o pseudocódigo

Tipos de dados armazenados nas variáveis e constantes

- numerico:
  - → inteiro: $\ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \ldots$
  - → real: $\ldots, -192.291, \ldots, 192.291, \ldots$
- literal:
  - → caractere: definido por um caractere “a”, “b”, etc.
  - → string: junção de caracteres, “bola”, “oi”, etc.
- logico: informação só pode ser verdadeiro ou falso.

Notação para o pseudocódigo

Características dos nomes válidos para variáveis e constantes

- Podem ser usados números, letras minúsculas e maiúsculas e o caractere *underscore* (_).
- Deve começar por uma letra, maiúscula ou minúscula, ou pelo caractere *underscore* (_).
- Não podem ser usados símbolos como $, #, !, ?, &, +, −.
- Não podem ser usados espaços em branco.
- Não podem ser usadas palavras reservadas da linguagem usada: algoritmo, leia, etc.

Regra de ouro: Use nomes que façam algum sentido!

Formato de um pseudocódigo

Portugol Online (https://vinyanalista.github.io/portugol/):
Ambiente de desenvolvimento integrado que suporta a linguagem Portugol, inspirada no livro Fundamentos da Programação de Computadores (Ascencio & Campos)

```
algoritmo
   bloco de declarações
   bloco de comandos
fim_algoritmo
```

- Uma declaração por linha para cada tipo de dado usado.
- Dados de mesmo tipo são separados por vírgula.
- Dados de tipos distintos devem ser declarados em linhas diferentes.

Construção do algoritmo: declaração

Declaração de variávei e constantes:

declare var1, var2, ... tipo de dado

```
algoritmo
   declare i, j, k numerico
           nome literal
           overlap logico
fim_algoritmo
```

- Apenas uma instrução tipo declare deve ser usada.
- A declaração de uma variável ou constante não implica na atribuição de qualquer valor para o dado a ser armazenado.
- A declaração de variáveis e constantes não é representada num fluxograma.

Operadores

Meios pelos quais se realizam operações sobre as variáveis e constantes, tais como atribuição de valores, incremento, decremento, multiplicação, comparação, etc.

1. Operadores de atribuição.
2. Operadores aritméticos.
3. Operadores relacionais.
4. Operadores lógicos.

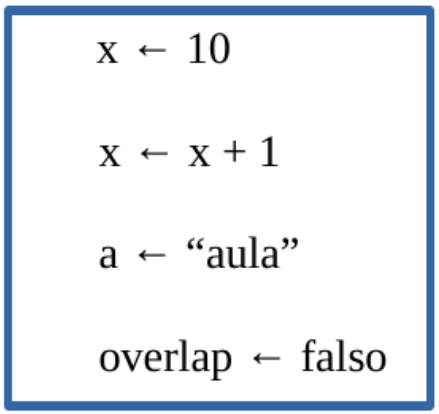
Construção do algoritmo: comandos

Operadores de atribuição: usados para atribuir valores ou operações às variáveis ou constantes.

Pseudocódigo

```
x ← 10
x ← x + 1
a ← "aula"
overlap ← falso
```

Fluxograma



Construção do algoritmo: comandos

Operadores aritméticos: utilizados para operações com valores numéricos entre variáveis e constantes.

Básicos

Operador	Símbolo	Exemplo
Soma	+	$a \leftarrow$ b+c
Subtração	–	$a \leftarrow$ b-c
Produto	*	$a \leftarrow$ b*c
Divisão	/	$a \leftarrow$ b/c

Ordem de prioridade

Menor + - * / Maior

Construção do algoritmo: comandos

Operadores aritméticos: utilizados para operações com valores numéricos entre variáveis e constantes.

Pré-definidos

Operador	Símbolo	Exemplo
inteiro mais próximo do número real x	arredonda(x)	i ← arredonda(3.6)
parte inteira do número real x	parte_inteira(x)	i ← parte_inteira(0.8)
resto da divisão do número x pelo número y	resto(x,y)	r ← resto(8,3)
seno do ângulo x (expresso em radianos)	seno(x)	ang ← seno(3.1415)
cosseno do ângulo x (expresso em radianos)	cosseno(x)	ang ← cosseno(3.1415)
número x elevado ao número y (potência)	potencia(x,y)	p ← potencia(5,2)
raiz quadrada do número x	raiz_quadrada(x)	r2 ← raiz_quadrada(25)
raiz n do número x	raiz_enesima(x)	r3 ← raiz_enesima(3,27)

Construção do algoritmo: comandos

Operadores relacionais: usados na comparação entre valores ou expressões, retornando como resultado um valor lógico.

Operador	Símbolo	Exemplo
Igual a	=	$a = b$
Maior que	>	$a > b$
Menor que	<	$a < b$
Maior ou igual a	>=	$a >= b$
Menor ou igual a	<=	$a <= b$
Diferente de	<>	$a <> b$

Construção do algoritmo: comandos

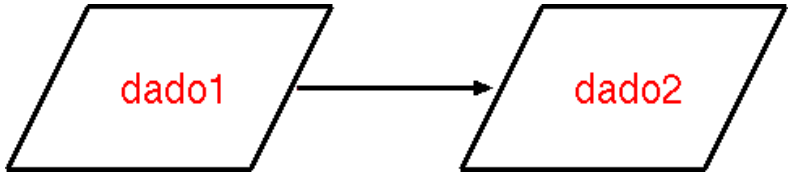
Operadores lógicos: relacionam entre si valores ou expressões lógicas, resultando em valores lógicos.

Operador	Símbolo	Especificação
Conjunção	e	Uma expressão E só é verdadeira se todos os valores das variáveis ou expressões forem verdadeiras.
Disjunção	ou	Uma expressão OU só é falsa se todos os valores das variáveis ou expressões forem falsos.
Negação	nao	O operador NÃO inverte o valor da expressão ou variável, se verdadeira inverte para falsa e vice-versa.

Entrada e Saída de dados num algoritmo

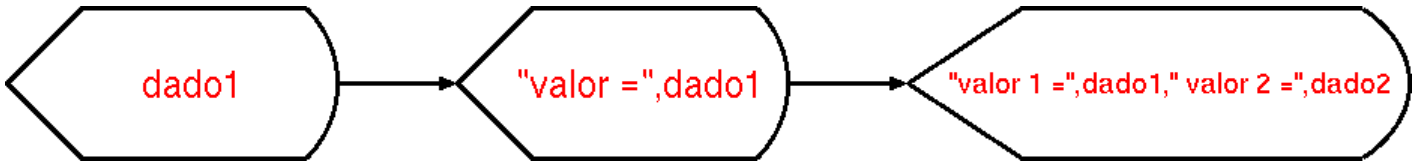
- Entrada: apenas um dado de entrada para cada comando. Não permite comentários.

```
leia dado1
leia dado2
```



- Saída: dados de saída separados por vírgula; texto de comentário (dado literal) entre aspas.

```
escreva dado1
escreva "valor = ",dado1
escreva "valor 1 =",dado1, "valor 2 =",dado2
```



Construção do algoritmo: estruturas

Algumas estruturas possíveis num algoritmo:

1. Estrutura sequencial
2. Estrutura condicional
3. Estrutura de repetição: PARA
4. Estrutura de repetição: ENQUANTO
5. Estrutura de repetição: REPITA

Estrutura sequencial

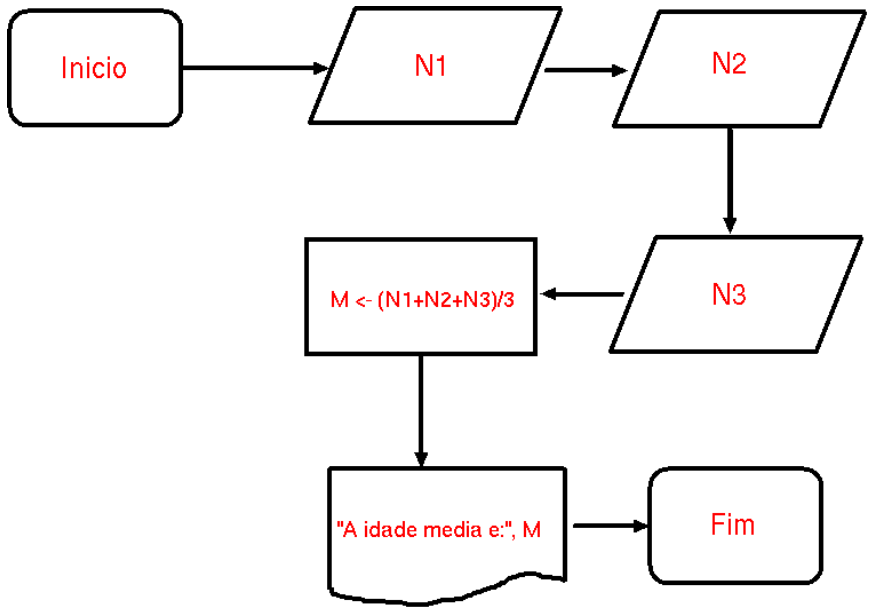
O programa não tem pontos de desvio ou retorno de fluxo.

Pseudocódigo

Cálculo da média simples de 3 idades.

```
algoritmo
declare N1, N2, N3, M numerico
leia N1
leia N2
leia N3
M ← (N1+N2+N3)/3
escreva "A idade média é:", M
fim_algoritmo
```

Fluxograma



Estrutura condicional

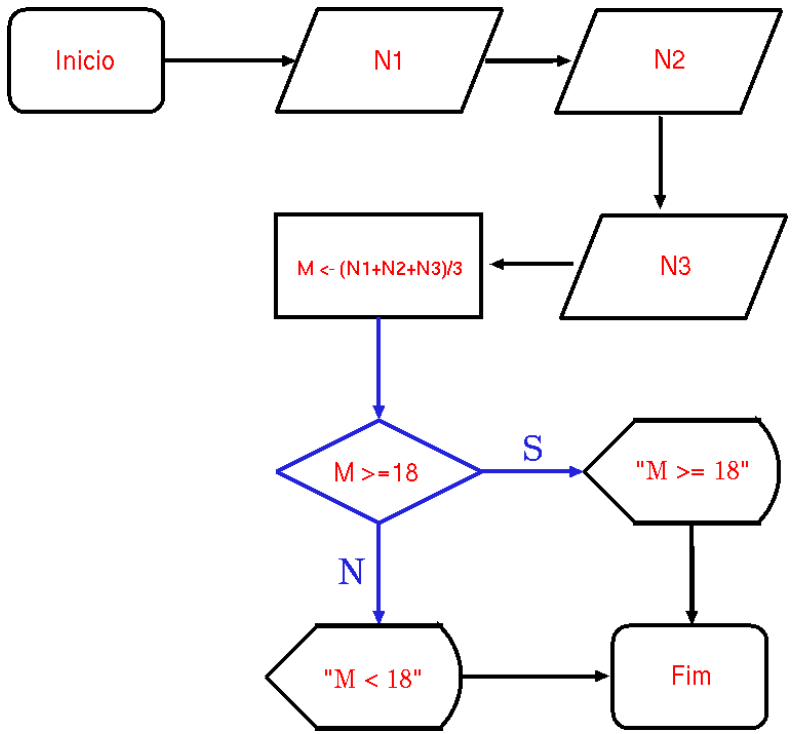
O programa tem pontos de desvio de fluxo mas não de retorno.

Pseudocódigo

Cálculo da média simples de 3 idades, com duas possibilidades de resultado.

```
algoritmo
declare N1, N2, N3 numerico
leia N1
leia N2
leia N3
M ← (N1+N2+N3)/3
se M >= 18
entao escreva "M >= 18"
senao escreva "M < 18"
fim_algoritmo
```

Fluxograma



Estrutura de repetição: PARA

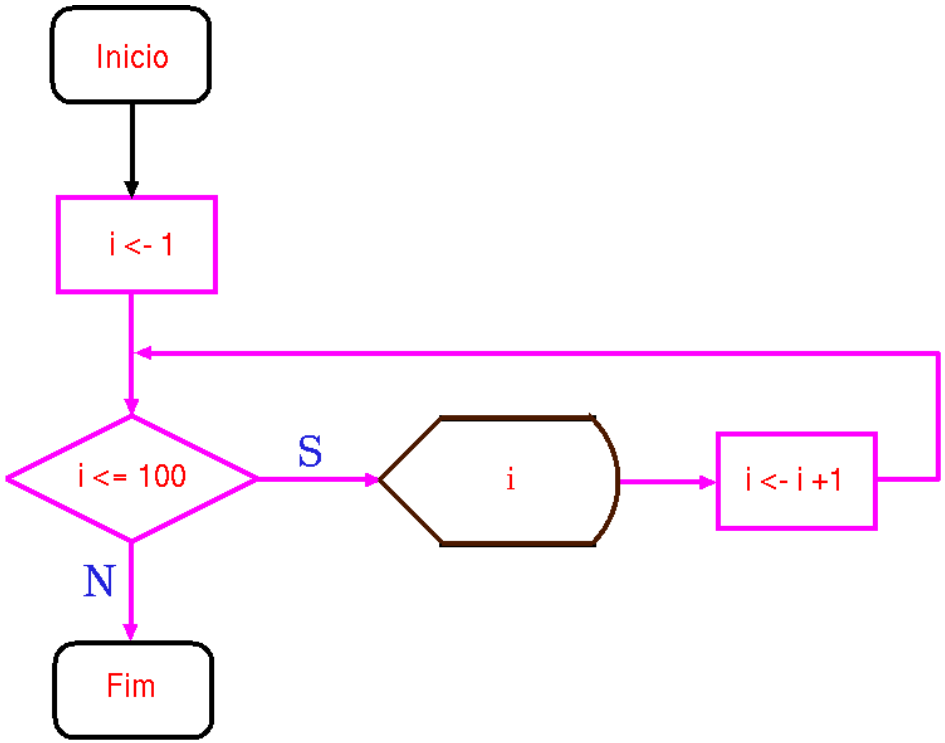
O programa tem pontos de retorno de fluxo, com um número conhecido de repetições.

Pseudocódigo

Sequência de inteiros de 1 até 100.

```
algoritmo
declare i numerico
para i ← 1 ate 100 faca passo 1
     escreva i
fim_algoritmo
```

Fluxograma



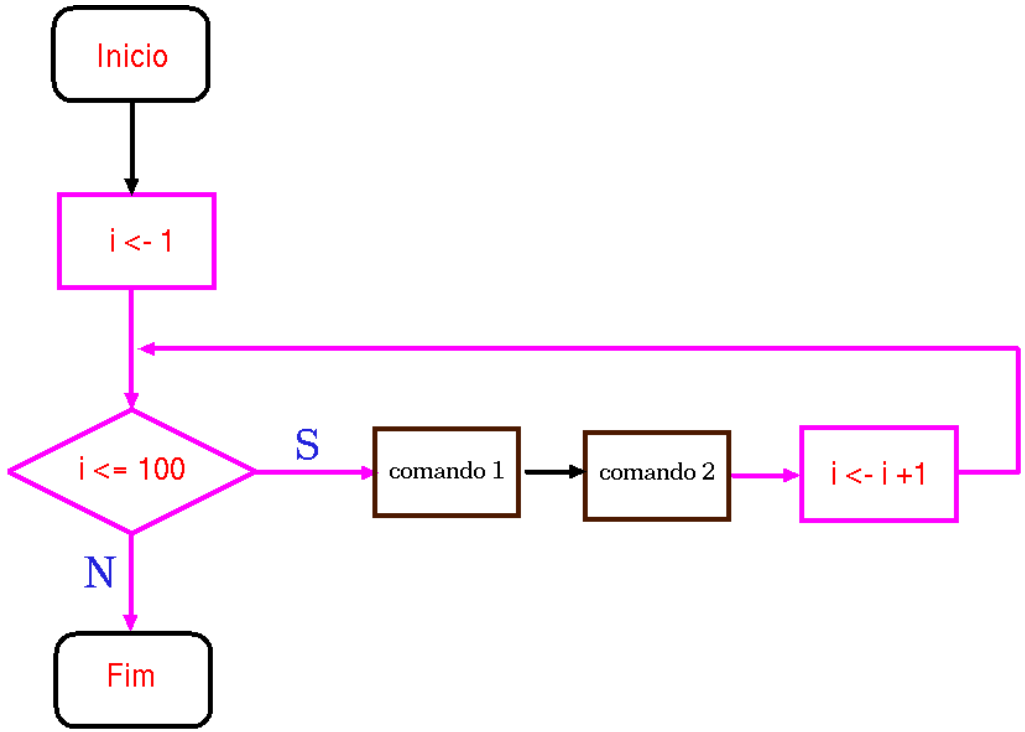
Estrutura de repetição: PARA

O programa tem pontos de retorno de fluxo, com um número conhecido de repetições.

Pseudocódigo

```
algoritmo
declare i numerico
para i ← 1 ate 100 faca passo 1
   inicio
        comando 1
        comando 2
        comando n
   fim
fim_algoritmo
```

Fluxograma



Estrutura de repetição: PARA

O programa tem pontos de retorno de fluxo, com um número conhecido de repetições.

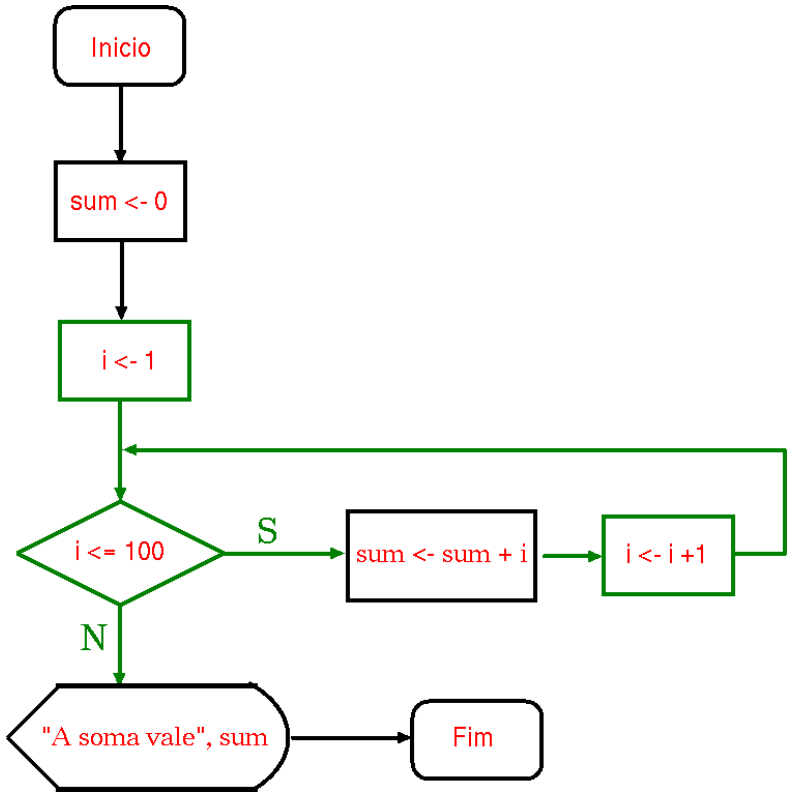
Pseudocódigo

Soma dos primeiros 100 inteiros.

$$\text{sum} = \sum_{i=1}^{100} i$$

```
algoritmo
declare i, sum numerico
sum ← 0
para i ← 1 ate 100 faca passo 1
     sum ← sum + i
escreva ”A soma vale”, sum
fim_algoritmo
```

Fluxograma



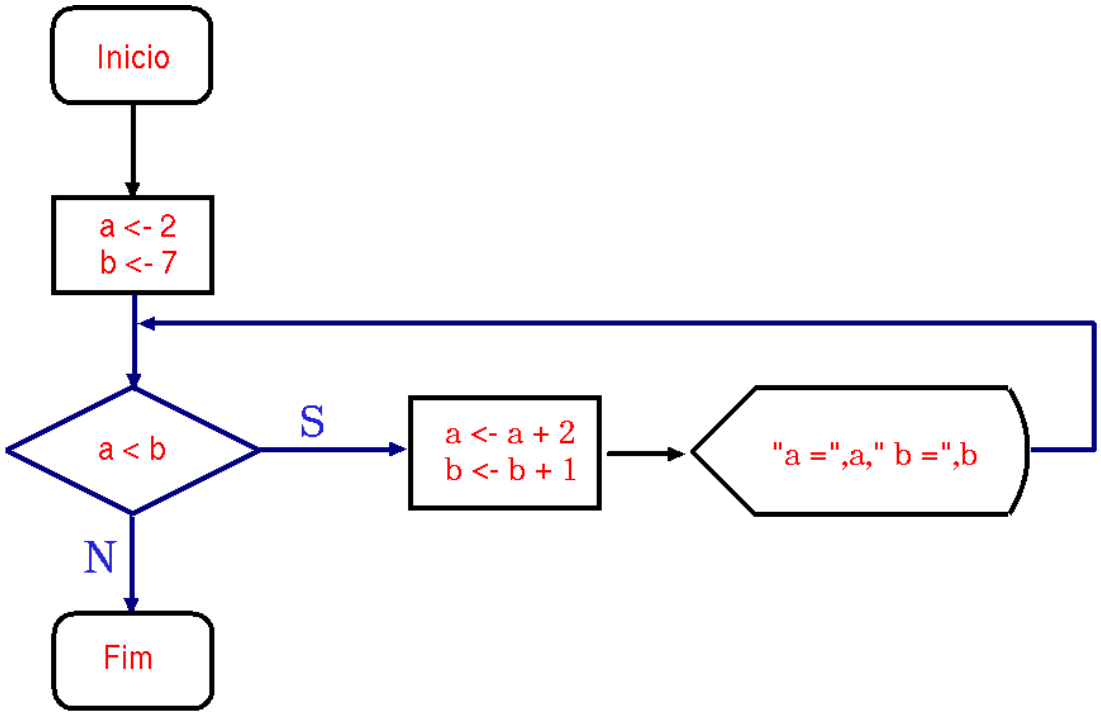
Estrutura de repetição: ENQUANTO

O programa tem pontos de retorno de fluxo, com o número de repetições limitado por uma condição lógica de entrada.

Pseudocódigo

```
algoritmo
declare a, b numerico
a ← 2
b ← 7
enquanto a < b faca
   inicio
      a ← a + 2
      b ← b + 1
      escreva "a =",a,"b =",b
   fim
fim_algoritmo
```

Fluxograma



Estrutura de repetição: REPITA

O programa tem pontos de retorno de fluxo, com o número de repetições limitado por uma condição lógica de saída.

Pseudocódigo

```
algoritmo
declare a, n numerico
a ← 1
b ← 8
repita
      a ← a + 2
      b ← b + 1
      escreva "a =",a,"b =",b
ate a > b
fim_algoritmo
```

Fluxograma