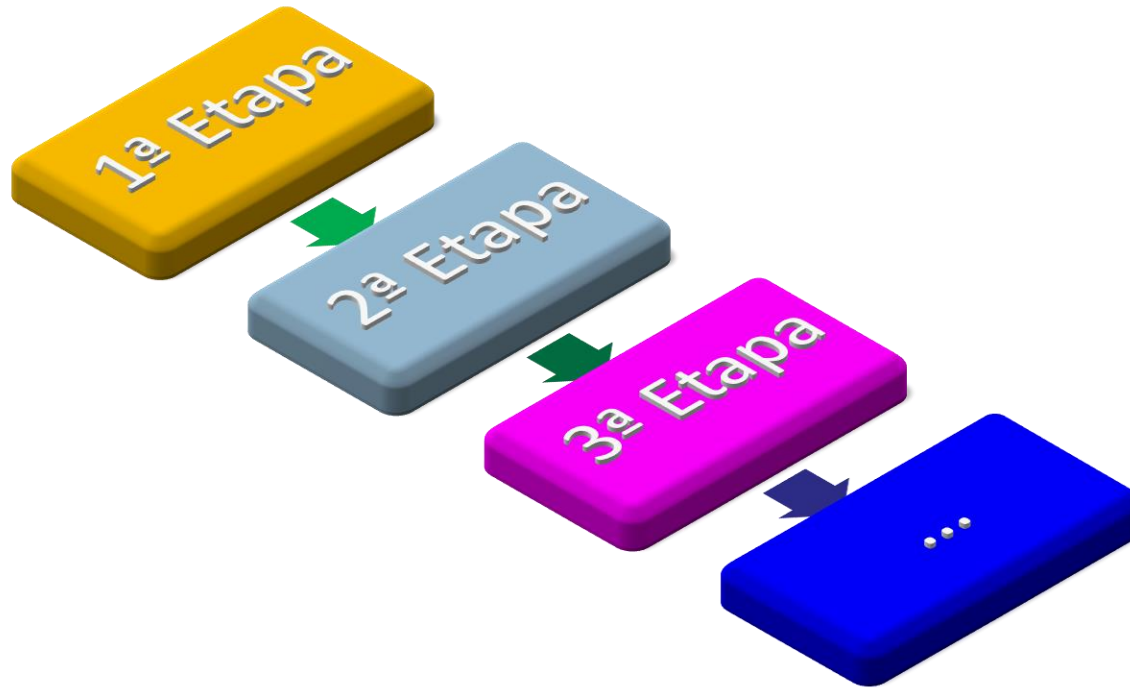


Lógica de Programação

Fabício Curvello Gomes

Rodrigo Dacome Lima



Programação Com Sequência

Dados Primitivos

Os dados representam os elementos do mundo exterior, que são as informações manipuladas pelos seres humanos.

Os dados a serem utilizados no computador devem primeiramente ser abstraídos para então serem processados.

Podem ser categorizados em três tipos primitivos ou básicos:

1. **Numéricos** – Valores numéricos **inteiros** ou **reais**.
2. **Caracteres** – Valores alfabéticos ou alfanuméricos.
3. **Lógicos** – Valores falsos e verdadeiros.



1.1 - Dados Numéricos Inteiros

São inteiros os dados numéricos positivos e negativos pertencentes ao conjunto dos números inteiros, excluindo qualquer valor numérico fracionário.

Serão referenciados em C++ pela palavra reservada ***int***.



1.2 - Dados Numéricos Reais

São reais os dados numéricos positivos e negativos pertencentes ao conjunto dos números reais, incluindo todos os valores fracionários e inteiros.

Serão referenciados em C++ pela palavra reservada ***double***.



2. Dados Caracteres / Cadeias

São delimitados por aspas(" "). Eles são representados por letras (De A a Z), números (De 0 a 9), símbolos (por exemplo, todos os símbolos imprimíveis existentes num teclado) ou palavras contendo estes símbolos.

O dado caractere é conhecido também como **alfanumérico**, **string**, **literal** ou **cadeia**.

Serão referenciados em C++ ao instanciar um objeto da classe **string**.



3. Dados Lógicos

São lógicos os dados com valores binários como sim e não, verdadeiro e falso, 0 e 1, entre outros, e apenas um dos valores pode ser escolhido.

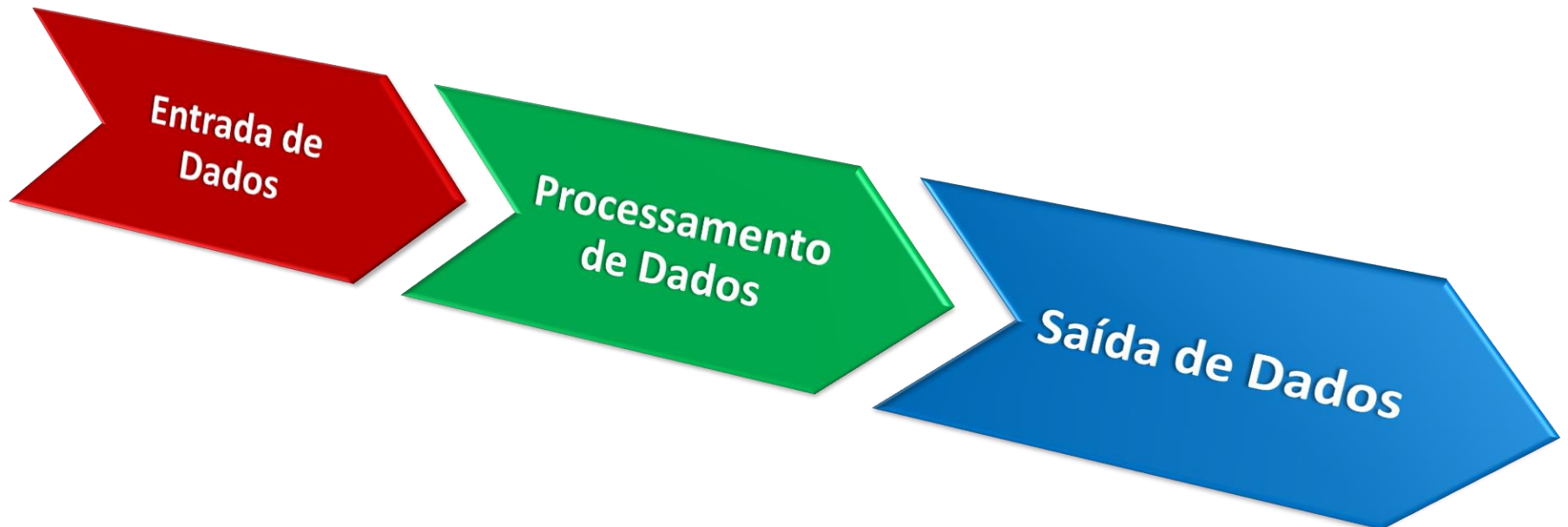
O tipo de dado lógico é também conhecido como **booleano**.

Serão referenciados em C++ pela palavra reservada ***bool***.



Etapas da Ação de um Programa

Todo programa de computador deve executar basicamente três ações:



O Uso de Variáveis

Variável é tudo que está sujeito a variações, que é incerto, instável ou inconstante.

Tratando-se de computador, é preciso ter em mente que o volume de dados a ser tratado é grande e diversificado.

Desta forma, os dados a serem processados são bastante variáveis.

Todo dado a ser armazenado na memória de um computador deve ser previamente identificado segundo seu tipo, ou seja, primeiro é necessário saber o tipo de um dado para depois fazer seu armazenamento adequado.

Armazenado o dado desejado, ele pode ser utilizado e processado a qualquer momento.



O Uso de Variáveis (Cont.)

Imagine a memória de um computador como um grande arquivo com várias gavetas nas quais é possível guardar apenas um valor por vez, e como em um arquivo, essas gavetas devem ser identificadas por uma “etiqueta” com um nome.



Qual Nome Dar a Uma Variável ?

O nome de uma variável é utilizado para sua identificação e representação dentro de um programa de computador. É necessário estabelecer e seguir algumas regras de uso de variáveis:

1. Utilize letras **minúsculas**. Palavras duplas devem virar uma só, onde o letra inicial de cada palavra subsequente fica em maiúscula.
Ex: *casa, a, carroPopular, nomeAluno*
2. O nome de uma variável pode utilizar um ou mais caracteres, exceto letras acentuadas e Ç.
3. O **primeiro** caractere **tem que ser uma letra**.
4. Não pode existir espaço em branco no nome da variável. Siga a regra do item 1.
5. O nome de uma variável não pode ser o mesmo de um comando ou instrução de linguagem de programação.
6. Não pode ser utilizado como nome de uma variável algum rótulo que já tenha sido usado como nome de um programa ou mesmo de outra variável.



Qual Nome Dar a Uma Variável ?(Cont.)

Eis alguns exemplos de nomes de variáveis:

CORRETO:

nome, nomeUsuario, fone1, n, a

ERRADO:

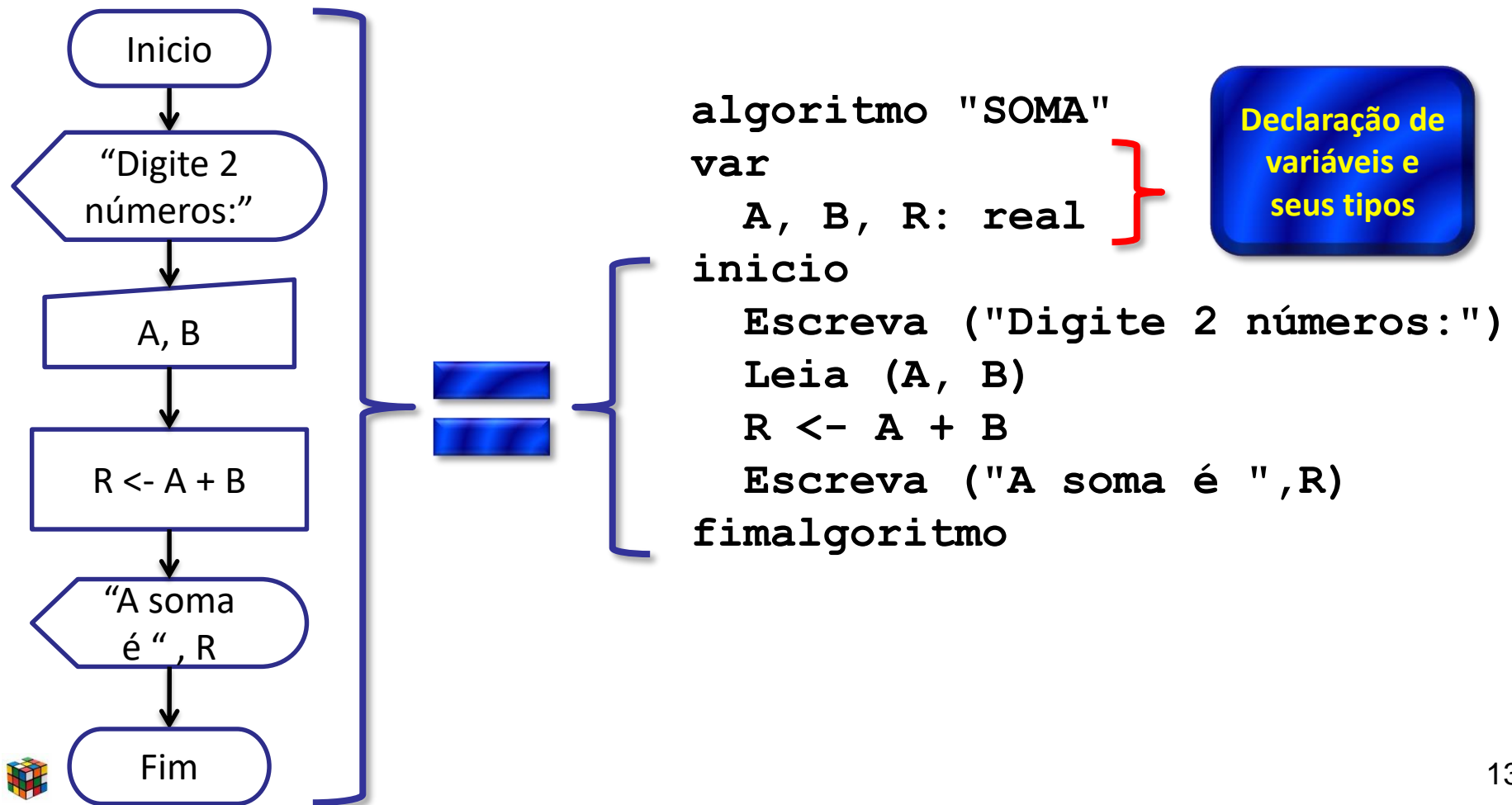
usuário, nome usuario, double, Aluno, nome_aluno, 1, end#1, CIDADE



Declaração de Variáveis

A declaração de variáveis na construção de diagramas de bloco não existe, porém, na estruturação dos pseudocódigos e nos códigos é fundamental.

Observe novamente o exemplo de pseudocódigo presente no capítulo 1:



O Uso de Constantes

Uma **constante** é uma **grandeza numérica fixa**, utilizada normalmente numa expressão aritmética ou matemática, a qual define um valor que será inalterado na expressão, independentemente das variáveis envolvidas na operação realizada.

A declaração de uma constante se dá de forma muito parecida com a declaração de uma variável, mas existe a palavra reservada **const**.

Ex:

```
const int DIAS_SEMANA = 7;
```



Operadores Aritméticos

Em Java são aceitos os seguintes operadores aritméticos:

Operador	Tipo	Descrição	Resultado	Exemplo
+	Unário	Manutenção de sinal	Positivo	+3
-	Unário	Inversão de sinal	Negativo	-8
+	Binário	Soma	Inteiro ou Real	$3 + 5 = 8$
-	Binário	Subtração	Inteiro ou Real	$9 - 4 = 5$
*	Binário	Multiplicação	Inteiro ou Real	$3 * 5 = 15$
/	Binário	Divisão	Inteiro ou Real	$5 / 2 = 2.5$
%	Binário	Resto da divisão inteira	Inteiro	$8 \% 3 = 2$



Operadores Aritméticos (Cont.)

Existem algumas funções já prontas em C++ que podem ser úteis, através da classe **math**:

Ação	Código	Exemplo
Raiz Quadrada	<code>sqrt(valor)</code>	<code>sqrt(16)</code>
Potência	<code>pow(base, expoente)</code>	<code>pow(2, 3)</code>
Valor de Pi	<code>M_PI</code>	<code>M_PI</code>



Expressões Aritméticas

Expressões aritméticas são utilizadas diversas vezes em algoritmos.

Veja um exemplo de uma expressão aritmética tradicional:

$$X = \{ 43 \cdot [55 : (30 + 2)] \}$$

Esta mesma expressão, se trabalhada em um código C++:

$$x = (43 * (55 / (30 + 2)))$$

Vale lembrar que as regras de prioridades da matemática também se aplicam nos algoritmos, ou seja:

- Os conteúdos dos Parênteses são prioridades de resolução.
- Multiplicação e Divisão são prioridades em relação à Adição e Subtração.
- E, em especial nos códigos, parênteses internos (dentro de Parênteses) devem ter seus conteúdos resolvidos antes dos Parênteses externos.



Operadores de Caracteres

Em Java é aceito o seguinte operador de caracteres:

Operador	Descrição	Exemplo
+	Realiza concatenação de strings, quando usado com dois valores (variáveis ou constantes) do tipo “string”.	"Rio " + "de Janeiro" = "Rio de Janeiro".



Operadores Relacionais

Em C++ são aceitos os seguintes operadores relacionais:

Operador	Descrição
==	Igual a *
<	Menor que
>	Maior que
<=	Menor ou igual
>=	Maior ou igual
!=	Diferente de

* OBS: Utiliza-se **==** quando a comparação envolve números, variáveis numéricas (tipos *int*, *double* e *string*) ou variáveis do tipo *boolean*.

Em casos de comparação léxica entre variáveis do tipo *string* ou entre texto utiliza-se **.compare**



Operadores Lógicos

Em C++ são aceitos os seguintes operadores lógicos:

Operador	Operação	Descrição
não (negação)	!	Operador de negação. Não VERDADEIRO = FALSO, e não FALSO = VERDADEIRO.
ou (disjunção)		Operador que resulta VERDADEIRO quando um de seus operandos lógicos for verdadeiro.
e (conjunção)	&&	Operador que resulta em FALSO quando um de seus operandos lógicos for falso.
xou (disjunção exclusiva)	^	Operador que resulta em VERDADEIRO se seus dois operandos lógicos forem diferentes, e FALSO se forem iguais.



Operadores Lógicos (Cont.)

A tabela-verdade abaixo, mostra os resultados das aplicações dos operadores lógicos conforme os valores dos operadores envolvidos

A	B	NÃO A	NÃO B	A E B	A OU B	A XOU B
VERDADEIRO	VERDADEIRO	FALSO	FALSO	VERDADEIRO	VERDADEIRO	FALSO
VERDADEIRO	FALSO	FALSO	VERDADEIRO	FALSO	VERDADEIRO	VERDADEIRO
FALSO	VERDADEIRO	VERDADEIRO	FALSO	FALSO	VERDADEIRO	VERDADEIRO
FALSO	FALSO	VERDADEIRO	VERDADEIRO	FALSO	FALSO	FALSO



Dúvidas?



Bibliografia



Estudo Dirigido de Algoritmos
José Augusto N. G. Manzano e Jayr Figueiredo de Oliveira
Ed. Érica



Fundamentos de Computação e Orientação a Objetos Usando JAVA
Francisco A. C. Pinheiro
Ed. LTC