



# COMPUTAÇÃO BÁSICA

## Algoritmos Computacionais e Estrutura Sequencial

Prof. Bruno Macchiavello ([bruno@cic.unb.br](mailto:bruno@cic.unb.br))

Universidade de Brasília – UnB  
Instituto de Ciências Exatas – IE  
Departamento de Ciência da Computação – CIC



# Algoritmos Computacionais

- Um algoritmo computacional é uma seqüência de **instruções** que sejam capaz de ser compreendidas por um computador para manipular certos **dados**.
  - **Instruções**: comandos que determinam a forma pela qual os dados devem ser tratados. Comandos que indicam ou computador realizar uma determinada tarefa básica.
  - **Dados**: elementos recolhidos/fornecidos por diversos meios e que serão processadas pelo computador através das instruções.



# Dados em Algoritmos Computacionais

- Os dados de um algoritmo computacionais podem ser de diferentes tipos, a classificação deles é a seguinte:
  - Dados Numéricos
  - Dados Literais
  - Dados Lógicos



# Tipos de Dados – Dados Numéricos

- Numéricos inteiros
  - Os números inteiros são aqueles que não possuem componentes decimais ou fracionários, podendo ser positivos ou negativos

Exemplos de números inteiros:

36	número inteiro positivo;
0	número inteiro;
-18	número inteiro negativo.



# Tipos de Dados – Dados Numéricos

- Numéricos reais
  - Os dados de tipo real são aqueles que podem possuir componentes decimais ou fracionários, e podem também ser positivos ou negativos.

Exemplos de números reais:

36.01	número real positivo com duas casas decimais;
166.	número real positivo com zero (nenhuma) casa decimal;
-18.8	número real negativo com uma casa decimal;
0.0	número real com uma casa decimal;
0.	Número real com zero (nenhuma) casa decimal.



# Tipos de Dados – Dados Numéricos

- Numéricos Binários
  - Normalmente as linguagens de programação de alto nível não proporcionam a possibilidade de criar um dado numérico do tipo binário de mais de um BIT.
  - Dado binário de um BIT é conhecido como dado booleano.

Verificar em casa:

Como trabalhar com números binários de mais de um bit?

Dica: Verificar a função de Deslocamento (shift)



# Tipos de Dados – Dados Literais

- Dados Literais:
  - Uma seqüência de caracteres contendo letras, dígitos e/ou símbolos especiais.
  - Também chamados de **alfanumérico**, **cadeia de caracteres** ou, ainda, **String**.
  - Usualmente são representados nos algoritmos pela coleção de caracteres delimitada em seu início e fim com aspas (“ ”).

Exemplos de dados literais:

“QUEM?”	Literal de comprimento 5;
“ ”	Literal de comprimento 1;
“cOmO!?#”	Literal de comprimento 7;
“AbcDEFghi”	Literal de comprimento 9;
“4+5-1=”	Literal de comprimento 6;
“1”	Literal de comprimento 1;
“ANA MARIA”	Literal de comprimento 9;

É comum em algumas linguagens a diferenciação entre um caracter (‘A’) e um conjunto de caracteres (“Olá Mundo”).



# Tipos de Dados – Dados Lógicos

- Dados Lógicos
  - São caracterizados como tipos lógicos, os dados com valores **verdadeiro** e **falso**, sendo que este tipo de dados poderá representar apenas um dos valores.
  - É chamado por alguns de tipo **booleano** (matemático George Boole).
  - Se definimos que Verdadeiro = 1, e Falso = 0, então um dado booleano = 1 BIT.





# Conceitos Básicos de Algoritmos Computacionais

- Os conceitos básicos de um algoritmo computacional são os seguintes:
  - Constante
  - Variáveis
  - Palavras Reservadas
  - Expressões Aritméticas
  - Expressões Lógicas
  - Comando de Atribuição e Entrada/Saída



# Constantes

- Entendemos que um dado é uma constante quando não sofre nenhuma variação no decorrer do tempo.
- Em relação à informática, é um espaço em memória identificado por um nome conhecido como **identificador** que pode conter dados de algum dos **tipos** anteriormente mencionados.
- Os dados armazenados são o **conteúdo**, o conteúdo não varia durante a execução de um programa, ou seja, o valor é o mesmo do início ao fim da execução do algoritmo.



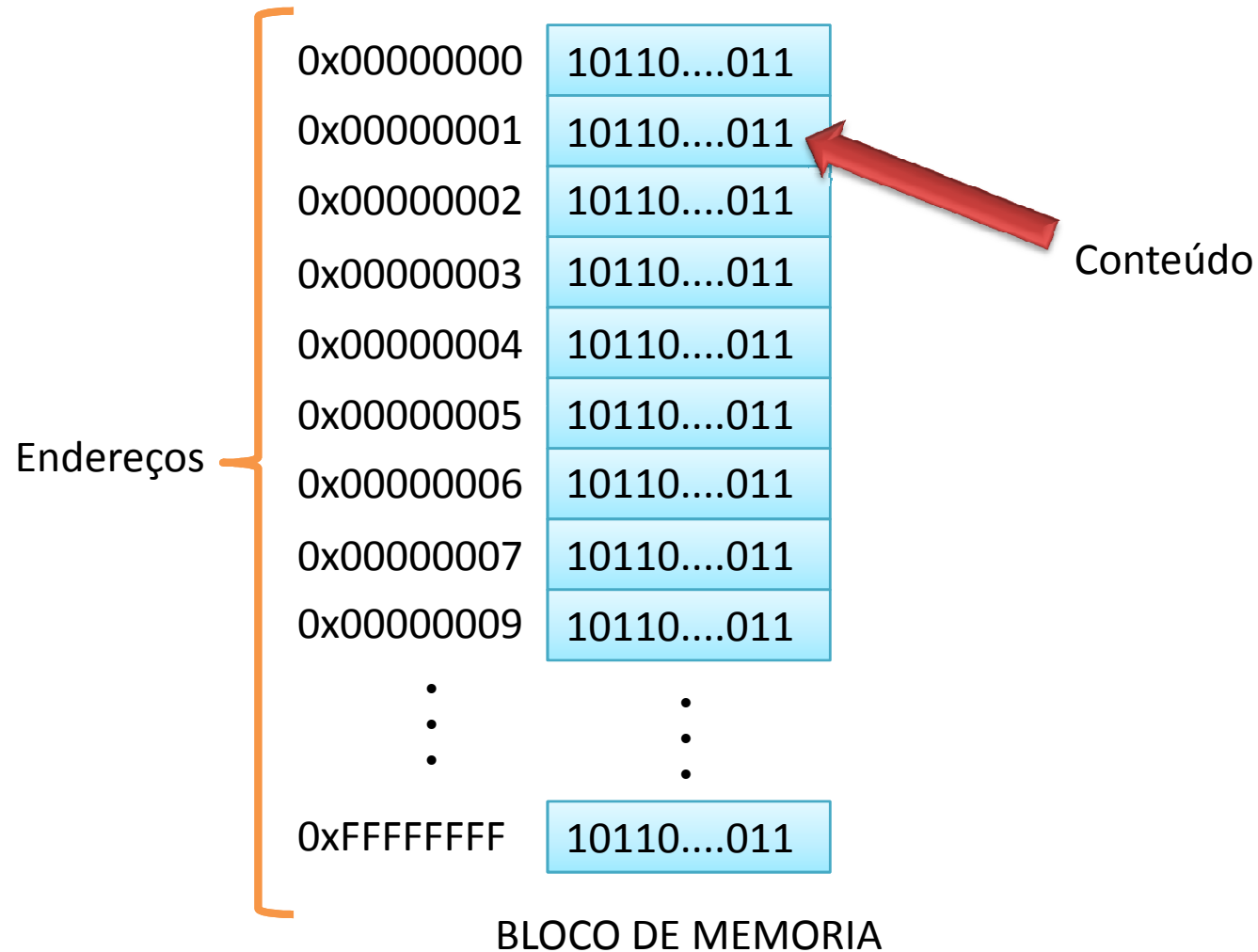
# Constantes

0x00000000	10110...011
0x00000001	10110...011
0x00000002	10110...011
0x00000003	10110...011
0x00000004	10110...011
0x00000005	10110...011
0x00000006	10110...011
0x00000007	10110...011
0x00000009	10110...011
⋮	⋮
0xFFFFFFFF	10110...011

BLOCO DE MEMORIA



# Constantes

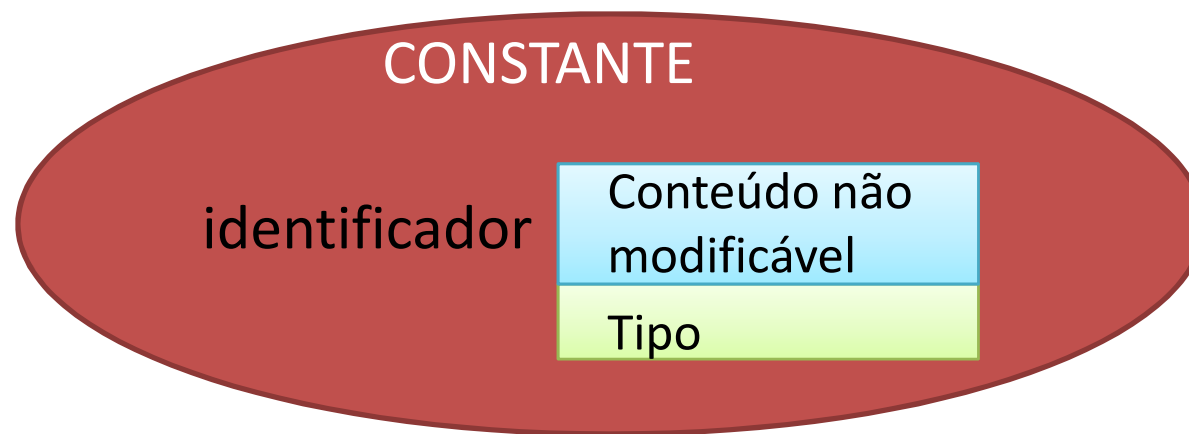


# Constantes

**Identificador:** Nome da constante, é uma palavra que ao compilar o programa é vinculado a um determinado endereço de memória. O identificador **não** pode mudar durante a execução de uma programa.

**Conteúdo:** Bits que representam um dado de um determinado tipo. Esse conjunto de bits **não** muda durante a execução do programa.

**Tipo:** É o tipo de dados que o conteúdo da variável representa. O tipo de uma constante **não** muda durante a execução de um programa.





# Variável

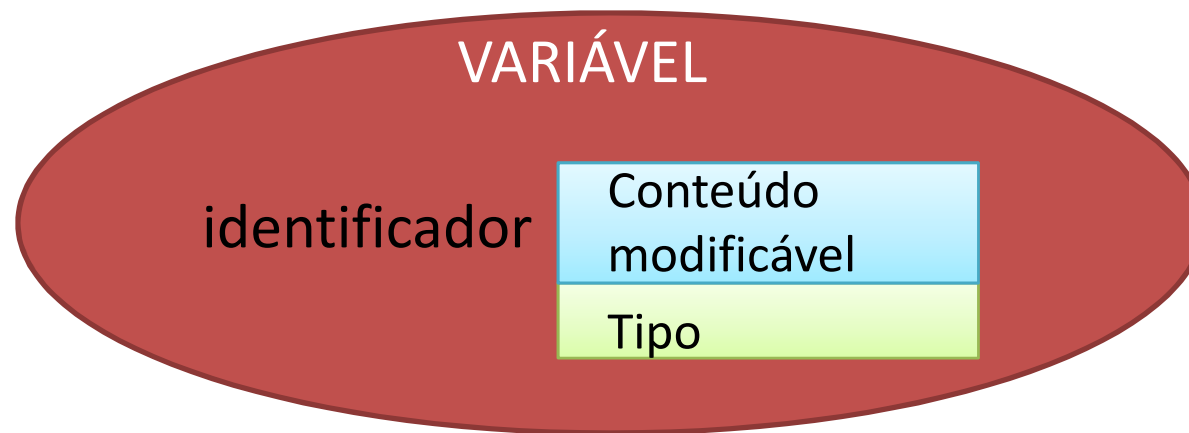
- Um dado é classificado como variável quando tem a possibilidade de ser alterado em algum instante no decorrer do tempo.
- Em relação à informática, variável é um espaço na memória, identificado por um nome conhecido como **identificador**, que pode conter dados de algum determinado **tipo**.
- Os dados armazenados em uma variável são o seu **conteúdo**. O conteúdo pode variar durante a execução do programa à qual ele pertença.

# Variável

**Identificador:** Nome da variável, é uma palavra que ao compilar o programa é vinculado a um determinado endereço de memória. O identificador **não** pode mudar durante a execução de uma programa.

**Conteúdo:** Bits que representam um dado de um determinado tipo. Esse conjunto de bits **pode mudar** em qualquer instante de tempo durante a execução do programa.

**Tipo:** É o tipo de dados que o conteúdo da variável representa. O tipo de uma variável **não** muda durante a execução de um programa.





# Variável

- ESCOPO
  - O escopo de uma variável de programa é a faixa de instruções na qual a variável é visível
  - Uma variável é visível em uma instrução se puder ser referenciada nesta instrução
  - Ex: O escopo de uma variável pode ser uma determinada função dentro de um programa (variável local), ou pode ser vista em todo o programa (variável global).



# Identificador

- Identificadores são conjuntos de caracteres, regido por determinadas leis, que representam constantes, variáveis, tipos, procedimentos, unidades, programas e campos de um registro (estes itens serão analisados mais adiante no curso).
- Regras:
  - O **primeiro caractere** deve ser uma letra;
  - No conteúdo do identificador não pode conter caracteres especiais (!@#\$%&\* /-+<>?), ou seja no identificador só podemos ter, após o primeiro caráter, **letras**, **números** e **sublinhada** ( \_ );
  - Nomes de variáveis escritas com letras **maiúsculas** serão diferentes de letras **minúsculas** (case sensitive);
  - Nenhuma **palavra reservada** à ferramenta (linguagem de programação) poderá ser usada como nome de uma variável;
  - Procure dar nomes significativos para a variável (**mnemônicos**), mas não muito longos.
  - Não podem ter identificadores com o mesmo nome para dois objetos diferentes.



# Identificador

- Existe um tamanho máximo para um identificador?
  - Depende da Linguagem de Programação:
    - FORTRAN I (Máx. 6)
    - COBOL (Máx. 30)
    - C/C++ (sem limite, depende do compilador)
    - Java (sem limite)



# Palavras Reservadas

- Palavras reservadas, são identificadores que já foram usados pelo programa, ou seja se caso utilizarmos uma destas palavras iremos infringir a regra de que não se pode haver identificadores repetidos.
- Em ANSI C algumas das palavras reservadas são:
  - auto, double, int, struct, break, else, long, switch, case, enum, register, typedef, char, extern, return, union, const, float, short, unsigned, continue, for, signed, void, default, goto, sizeof, volatile, do, if, static, while.



# Exemplos de Nomes de Identificadores

- Identificadores validos:
  - Alpha, X, BJ153, K7, Notas, Média, ABC, INPS, FGTS, i, j, nome\_1.
- Identificadores inválidos:
  - 5X, E(13), A:B, X-Y, Nota/2, AWQ\*, P&AA, nome 2.
- É sempre melhor usar nomes significativos para a variável (**mnemônicos**), mas não muito longos.
- Exemplo: para identificar 3 provas, identificadores significativos **são** “*p1,p2,p3*” ou “*prova1, prova2, prova3*” ou “*prova\_1, prova\_2, prova\_3*”. **Não** são significativos “*e1,p2,t3*” ou “*a,b,c*”.



# Declaração de Variáveis

- Declaração de Variáveis
  - São declaradas no início (alocação de memória)
    - Podem ser dos tipos: inteiro, real, literal, ou lógico
    - Permite que o algoritmo saiba os tipos de operações que cada variável pode realizar

**identificador**

:

**Tipo**



Exemplo 1:

**Dada uma lista de compras com o código, quantidade, e preço de oito produtos, crie um algoritmo que escreva o valor total da compra.**

codigo : literal (ou string)  
quantidade : inteiro  
preço, valortotal : real

Exemplo 2:

**Ao serem fornecidos um valor a ser pago e uma taxa para multa, pois o pagamento está sendo feito após o vencimento, calcule o valor da multa e o valor total a ser pago.**

vlrconta, taxamulta, vlrmulta, vlrtotal : real

Exemplo 3:

**É solicitada a um motorista recém chegado de uma viagem a quantidade de quilômetros por ele percorrida. O motorista informa o solicitado e você deverá informar a ele a quantos metros se refere à quantidade de quilômetros.**

quilômetros : real  
metros : inteiro



# Expressões Aritméticas

- São aquelas cujo resultado da avaliação é do tipo numérico, seja ele inteiro ou real. Somente o uso de operadores aritméticos e variáveis numéricas (inteiras ou reais) é permitido em expressões deste tipo.

Operador	Operação	Prioridade	Exemplo
-	Inversão de Sinal	1	$-(-1) = 1$
+	Manutenção de Sinal	1	$+1 = 1$
^	Exponenciação	2	$9^2 = 81$
*	Multiplicação	3	$6 * 7 = 42$
/	Divisão	3	$8 / 2 = 4$
+	Adição	4	$1 + 2 = 3$
-	Subtração	4	$4 - 5 = -1$



# Expressões Aritméticas

- Prioridade:
  - $1 + 4 * 3 / 6 - 3 ^ 2$
  - $1 + 4 * 3 / 6 - 9$
  - $1 + 12 / 6 - 9$
  - $1 + 2 - 9$
  - $-6$
- Prioridade:
  - $1 + (4 * 3 / (6 - 3)) ^ 2$
  - $1 + (4 * 3 / 3) ^ 2$
  - $1 + (4) ^ 2$
  - $1 + 16$
  - $17$





# Expressões Lógicas

- São aquelas cujo resultado da avaliação é um valor lógico (VERDADEIRO ou FALSO)

	CONJUNÇÃO	DISJUNÇÃO	NEGAÇÃO
x y	x E y	x OU y	NÃO(x)
V V	V	V	F
V F	F	V	F
F V	F	V	V
F F	F	F	V

Operador	Operação	Prioridade	Linguagem C
OU	Disjunção	3	
E	Conjunção	2	&&
NÃO	Negação	1	!



# Expressões Lógicas

- Operadores relacionais:
  - Operadores utilizados para realizar comparações.
  - Comparações só podem ser feitas entre variáveis do mesmo tipo.
  - O resultado de uma comparação é sempre um valor lógico.

Operador	Comparação	Linguagem C
=	Igual	==
≠	Diferente	!=
<	Menor	<
<=	Menor ou Igual	<=
>	Maior	>
>=	Maior ou Igual	>=



# Expressões Lógicas

- Exemplos (V = Verdadeiro, F = Falso):

Expressão	Resultado
$2 * 4 = 24 / 3$	V
$15 \bmod 4 < 19 \bmod 6$	F
$(2 < 5) \text{ e } (15 / 3 = 5)$	V
$(2 < 5) \text{ ou } (15 / 3 = 5)$	V
$(3 \geq 2) \text{ ou } (8 < 7)$	V
$(3 \geq 2) \text{ e } (8 < 7)$	F
$(V) \text{ e } (10 + 2 = 4 * 3)$	V
$(0) \text{ e } (5 = 5 + 0)$	F
$(F) \text{ ou } (1)$	V

- Notar que um número booleano é um bit onde  $V=1$  e  $F=0$ .