



# Linguagem de Programação [C++]

[\[www.caiobarbosa.com.br\]](http://www.caiobarbosa.com.br)



Aula 03

# ESTRUTURAS PRIMITIVAS

[\[www.caiobarbosa.com.br\]](http://www.caiobarbosa.com.br)



Automação e Controle

# BÁSICO



# Visão Geral das Estruturas Básicas

- Recursos da Linguagem de Programação;
- Variáveis;
- Constantes;
- Expressões Aritméticas;
  - Operadores Aritméticos;
- Atribuições;
- Expressões Lógicas;
  - Operadores Relacionais;
  - Operadores Lógicos.
- Comandos de Entrada e Saída

# Recursos das Linguagens de Computação



Quando um programa é “**traduzido**” para linguagem de máquina pelo compilador, o mesmo precisa diferenciar:

- Variável;
- Tipo de variável;
- Instruções;
- Operadores (aritméticos e lógicos);
- Outros elementos de linguagem de comparação.

# Recursos das Linguagens de Computação (cont.)



Toda linguagem de programação possui alguns recursos que evita a possibilidade de “**dupla interpretação**” de uma linha de código pelo compilador, são elas:

- Palavras Reservadas;
- Simbologia para alguns operadores;
- Regras de nomes de variáveis;

# C++:

## Comentários



```
/*  
o comentário tradicional,  
englobando várias linhas  
*/
```

```
// já o comentário de uma única linha
```

# C++:

## Variáveis



Na programação, uma variável é um espaço da memória volátil capaz de reter e representar um valor ou expressão. Enquanto as variáveis só "existem" em tempo de execução, elas são associadas a "nomes", chamados identificadores, durante o tempo de desenvolvimento.





# C++:

## Variáveis: Tipos de Dados



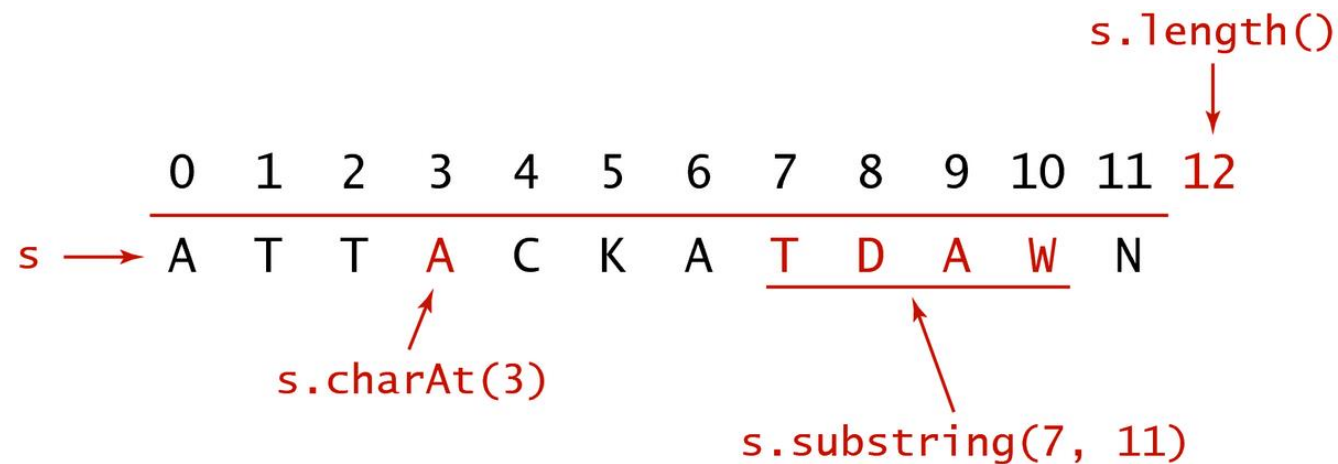
Tipo	Descrição	Rango	Exemplo
bool	Valor binário verdadeiro ou falso.	true false	bool dato = false; dato = true;
char	Valor inteiro que representa un carácter de la tabla ASCII	-128 a 127 ó 0 a 255 compilado con /J	char letra = 'A'; letra = '\n'; letra = 65;
short	Valor inteiro de 2 bytes	-32,768 32,767	short x = 94; x = -54;
int	Valor inteiro de 4 bytes	-2,147,483,648 2,147,483,647	<u>int</u> x = 1598; x = -988574;
unsigned int	Valor inteiro positivo de 4 bytes	0 4,294,967,295	unsigned int x = 9887; x = 98745;
long long	Valor inteiro de 8 bytes	-9,223,372,036,854,775,808 9,223,372,036,854,775,807	long long x = 684574; x = -998564;
float	Valor decimal de 4 bytes	3.4E +/- 38 (7 dígitos)	float x = 45.6; x = -98.58;
double	Valor decimal de 8 bytes	1.7E +/- 308 (15 dígitos)	<u>double</u> x = 9878.568; x = -98745.668;
void	Tipo de dato nulo. Representa la ausencia de valor.		

# C++:

## Variáveis: Tipos de Dados: String



É uma coleção de caracteres (char).



Fundamental constant-time String operations

# C++:

## Variáveis: Declaração de Variáveis



As variáveis têm de ser *declaradas* antes do uso:

```
int altura;  
float raio;
```

Podemos declarar múltiplas variáveis do mesmo tipo duma só vez:

```
int altura, largura, profundidade;  
float raio, massa;
```

# C++:

## Variáveis: Declaração de Variáveis



As variáveis têm de ser *declaradas* antes do uso:

```
int altura;  
float raio;
```

Podemos declarar múltiplas variáveis do mesmo tipo duma só vez:

```
int altura, largura, profundidade;  
float raio, massa;
```

# C++:

## Variáveis: Atribuição de Variáveis



Podemos definir ou modificar o valor de uma variável usando uma *atribuição*.

```
int altura;    // declaração  
altura = 8;    // atribuição
```

- A atribuição têm de ocorrer depois da declaração
- Neste caso: atribuímos a constante 8 à variável `altura`

# C++:

## Variáveis: Definição de Constantes



- É por vezes necessário usar *constantes* ou *parâmetros*
- Constantes espalhadas pelo programa podem obfuscar o sentido
- Em vez disso: podemos usar directivas `#define` para definir *macros*

```
#define INCHES_PER_METER  39.3701  
/* factor de conversão:  
   polegadas por cada metro */
```

- Convenção: nomes de constantes em maiúsculas

# C++: Include



O Comando Include, carrega uma biblioteca de funções do Framework da Linguagem.

```
#include <iostream>
```

# C++: Using



Declara um alias para um namespace. Geralmente é usado para desambiguar quando classes diferentes possuem o mesmo nome.

```
using namespace std;
```

**Sem o using:**

```
std::cout << "Exemplo de saída na tela"
```

**Com o using:**

```
cout << "Os dois primeiros nomes são "
```



# C++:

## Operador de Entrada e Saída de Dados



Operador **cout** << saída de dados

Operador **cin** >> entrada de entrada

```
1 // Example program
2 #include <iostream>
3 #include <string>
4
5 using namespace std;
6
7 int main()
8 {
9     int comprimento;
10    int largura;
11    int resposta;
12
13    cout << "Informe o comprimento e a largura do retângulo: ";
14    cin >> comprimento >> largura;
15
16    resposta = (comprimento + largura);
17
18    cout << resposta;
19 }
```

# C++:

## Alternativas Entrada e Saída de Dados



Não serão usadas nestas abordagens:

Saída:

```
printf("Altura: %d cm\n", alt);
```

Entrada:

```
int n;  
scanf("%d", &n);
```

OU

```
getline (name);
```

**Teste:**

```
printf("%.2f", 37.777779);
```

# C++:

## Main



Na maioria das linguagens é o método principal a ser executado pelo sistema operacional.

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      string nome;
8      cout << "Qual seu nome";
9      getline(cin, nome);
10     cout << "Oi, " << name << "!\n";
11 }
```

# Algumas dicas...



- Ver o meu auto completar:

Ctrl + barra



Automação e Controle

# ESTRUTURAS LÓGICAS PRIMITIVAS

# Operadores: Aritméticos



Operador	Descrição
+	(Adição)
-	(Subtração)
*	(Multiplicação)
/	(Divisão)
%	(Resto/Módulo)

# Operadores:

## Atribuição



Operador	Descrição
=	Atribuição simples
+=	Atribuição aditiva
-=	Atribuição Subtrativa
*=	Atribuição Multiplicativa
/=	Atribuição de divisão
%=	Atribuição de módulo

# Operadores: Relacionais



Operador	Descrição
==	Igualdade
>	Maior
<	Menor
<=	Menor igual
>=	Maior igual
!=	Diferente



# Operadores: Concatenação



“+” é operador de concatenação.

```
string lCaio = “Caio”;  
string lBarbosa = “Barbosa”;  
string lCaioBarbosa = lCaio + lBarbosa;
```

# Operadores:

## Incremento e Decremento

OPERADOR DE INCREMENTO: ++

OPERADOR DE DECREMENTO: --

Também é possível incrementar uma unidade antes da linha onde o cursor está posicionado como também, pode-se executar o incremento ou decremento somente após a linha em que o cursor está posicionado. Então, temos 2 formas de utilizar esse operadores:

PRÉ-INCREMENTO: ++ variavel; -- variavel;

PÓS-INCREMENTO: variavel ++; variavel --;

```
using namespace std;
```

```
int main() {
```

```
    //Incremento = ++
```

```
    //Decremento = --
```

```
    cout << "Operador de incremento" << endl;
```

```
    int i = 0, i2 = 0;
```

```
    cout << "Pre = " << ++i << endl; //pre-incrementar
```

```
    cout << "Pos = " << i2++ << endl; //pos-incrementar
```

```
    cout << "Pos = " << i2 << endl;
```

```
    cout << endl;
```

```
    cout << "Operador de decremento" << endl;
```

```
    i = 0;
```

```
    i2 = 0;
```

```
    cout << "Pre = " << --i << endl; //pre-decrementar
```

```
    cout << "Pos = " << i2-- << endl; //pos-decrementar
```

```
    cout << "Pos = " << i2 << endl;
```

```
    system("pause");
```

```
    return 0;
```

```
}
```



# Operadores: Lógicos



Operador	Descrição	Exemplo	Frase
&&	And (e) de curto circuito**	0 0 1 1 operando1 0 1 0 1 operando2 0 0 0 1 (operando1 & operando2)	Retorna Verdadeiro se A e B forem Verdadeiros
	Or (ou) de curto circuito**	0 0 1 1 operando1 0 1 0 1 operando2 0 1 1 1 (operando1   operando2)	Retorna Verdadeiro se A ou B forem Verdadeiro ou ambos
^	Xor (ou exclusivo)	0 0 1 1 operando1 0 1 0 1 operando2 0 1 1 0 (operando1 ^ operando2)	Retorna Verdadeiro se A ou B forem Verdadeiro

# Exercício



Tendo as variáveis SALARIO, IR e SALLIQ, e considerando os valores abaixo. Informe se as expressões são verdadeiras ou falsas.

SALARIO	IR	SALLIQ	EXPRESSÃO	V ou F
100,00	0,00	100	(SALLIQ >= 100,00)	
200,00	10,00	190,00	(SALLIQ < 190,00)	
300,00	15,00	285,00	SALLIQ = SALARIO - IR	

# Exercício



Sabendo que  $A=3$ ,  $B=7$  e  $C=4$ , informe se as expressões abaixo são verdadeiras ou falsas.

- a)  $(A+C) > B$  (   )
- b)  $B \geq (A + 2)$  (   )
- c)  $C = (B - A)$  (   )
- d)  $(B + A) \leq C$  (   )
- e)  $(C+A) > B$  (   )

# Exercício



Sabendo que  $A=5$ ,  $B=4$  e  $C=3$  e  $D=6$ , informe se as expressões abaixo são verdadeiras ou falsas.

- a)  $(A > C)$  **AND**  $(C \leq D)$  (   )
- b)  $(A+B) > 10$  **OR**  $(A+B) = (C+D)$  (   )
- c)  $(A \geq C)$  **AND**  $(D \geq C)$  (   )



Automação e Controle

# FUNÇÕES BÁSICAS

# Funções Numéricas: Biblioteca cmath



```
#include <cmath>
```

Para arredondamento de máquina:

`round(x)`

<code>ceil(x)</code>	Arredonda valor real para cima (ex: <code>ceil(3,2) = 4</code> )
<code>floor(x)</code>	Arredonda valor real para baixo (ex: <code>ceil(3,2) = 3</code> )
<code>sin(x)</code>	seno de um ângulo em radianos
<code>cos(x)</code>	co-seno de um ângulo em radianos
<code>tan(x)</code>	Tangente de um ângulo em radianos
<code>abs(x)</code>	Valor absoluto do número inteiro
<code>fabs(x)</code>	Valor absoluto do número real
<code>exp(x)</code>	número <b>e</b> elevado a potência <b>x</b>
<code>log(x)</code>	logaritmo natural de <b>x</b>
<code>log10(x)</code>	Logaritmo decimal de <b>x</b>
<code>sqrt(x)</code>	raiz quadrada do número
<code>pow(base,exp)</code>	Calcula <b>x</b> elevado a <b>y</b>



# Funções Numéricas:

## Biblioteca cstring

```
#include <cstring>
```

<b>strcpy(s1, s2)</b>	Para armazenar uma string literal numa variável string - ou copiar o conteúdo de uma variável string para outra
<b>strcat(s1, s2)</b>	Concatena s2 no final da s1.
<b>strcmp(s1,s2)</b>	compara, <0 se s1<s2, 0 se s1==s2, >0 se s1>s2
<b>strncpy(s1, s2,n)</b>	copia ate n caracteres
<b>strlen(s)</b>	tamanho de s sem contar com \0
<b>strchr(s,c)</b> <b>strrchr(s,c)</b>	endereço do primeiro / ultimo caractere c em s ou 0
<b>strstr(s,sub)</b>	endereço do primeiro substring em s ou 0
<b>strupr(string)</b>	converte uma string para maiúscula
<b>strlwr (string)</b>	converte uma string para minúscula
<b>toupper(char)</b>	Converte um único caractere para maiúsculo
<b>tolower(char)</b>	Converte um único caractere para minúsculo



# Conversões de Tipos



Conversão	Comando	Biblioteca
Inteiro para float	Float(<variável_inteiro>)	#include <iostream>
Float para inteiro	Int(<variável_float>)	#include <iostream>
(semelhante para todos os numéricos)	(semelhante para todos os numéricos)	#include <iostream>
...	...	
Qualquer numérico para string	to_string(<variável_numérico>);	#include <string>
String ASCII para Float (double)	double atof (const char *str);	#include <stdlib.h>
String ASCII para int	atoi (const char *str);	#include <stdlib.h>
String ASCII para long	atol (const char *str);	#include <stdlib.h>

<https://www.ti-enxame.com/pt/c%2B%2B/como-converter-um-numero-em-string-e-vice-versa-em-c/971513093/>



Automação e Controle

# ESTRUTURAS DE DECISÃO

# Estrutura de Decisão: If...else



Expression is true

// codes before if-else

```
if (number < 5)
{
    number += 5;
}
else
{
    number -= 5;
}
// codes after if-else
```

Expression is false

// codes before if-else

```
if (number < 5)
{
    number += 5;
}
else
{
    number -= 5;
}
// codes after if-else
```

# Estrutura de Decisão: If...else em sequência



Usando como referência a linguagem c#.

```
double Salario = 1200;

if (Salario < 500)
    Salario += 50;
else if ((Salario >= 500) && (Salario < 600))
    Salario += 100;
else if ((Salario >= 500) && (Salario < 700))
    Salario += 110;
else if ((Salario >= 500) && (Salario < 800))
    Salario += 120;
else
    Salario += 250;
```

# Estrutura de Decisão: Switch



É possível usar um valor default, caso nenhuma alternativa (case) esteja definida:

**default :**  
    <instruções>

```
int diaDaSemana = 3;

switch (diaDaSemana)
{
    case 1:
        MessageBox.Show("Domingo");
        break;
    case 2:
        MessageBox.Show("Segunda-Feira");
        break;
    case 3:
        MessageBox.Show("Terça-Feira");
        break;
    case 4:
        MessageBox.Show("Quarta-Feira");
        break;
    case 5:
        MessageBox.Show("Quinta-Feira");
        break;
    case 6:
        MessageBox.Show("Sexta-Feira");
        break;
    case 7:
        MessageBox.Show("Sabado");
        break;
}
```

# Exercício



Para doar sangue é necessário ter entre 18 e 67 anos. Faça um aplicativo que pergunte a idade de uma pessoa e diga se ela pode doar sangue ou não.

# Exercício



Faça um programa que receba três inteiros e diga qual deles é o maior e qual o menor.



# Exercício



Escreva um programa em que recebe um inteiro e diga se é par ou ímpar.

# Exercício



Escreva um programa que pergunte o dia, mês e ano do aniversário de uma pessoa e diga se a data é válida ou não. Caso não seja, diga o motivo.



Automação e Controle

# ESTRUTURAS DE REPETIÇÃO

# Laço: for



## Sintaxe:

```
For (variável de inicialização; condição; incremento)
{
    //Comandos
}
```

```
int answer = 0;

for (int i = 1; i < 101; i++)
{
    answer = answer + i;
}
```

# Laço: do..while



Sintaxe:

```
do
{
    Comandos;
}
While (Condição)
```

```
int x = 5;
do {
    // D: do something ...

    x++;
} while (x <= 10);
```

# Laço: while



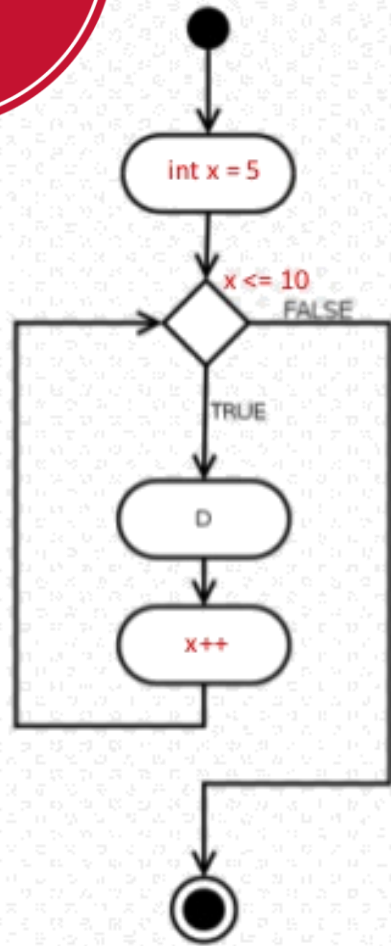
## Sintaxe:

```
While (Condição)
{
    Comandos;
}
```

```
int i = 0;
while (i < 11) {
    i+=1;
    Label1.Text += i + ", ";
}
```



# for loop => do while loop



```
for (int x = 5; x <= 10; x++)  
{  
    // D: do something ...  
}
```



```
int x = 5;  
do {  
    // D: do something ...  
  
    x++;  
} while (x <= 10);
```

# Exercício



Crie uma rotina que imprime na tela a sequência numérica de 1 até 100. Faça de duas formas distintas: usando for e while.



# Exercício



Crie uma calculadora, que realiza quatro operações, usando Java,

- a) Usando tipos inteiros;
- b) Verificando os dados de entrada e saída;
- c) Altere para tipo double;



Automação e Controle

# VARIÁVEIS COMPOSTAS

# Conceito:

## Variáveis Compostas Homogêneas



Ex: Uma **variável composta homogênea** seria uma Manada e seus **elementos** seriam os Elefantes.

Variável Composta Homogênea



Elemento



Fonte: Prof. Drndo. Anderson Elias, 2019

# Conceito:

## Variáveis Compostas Heterogêneas

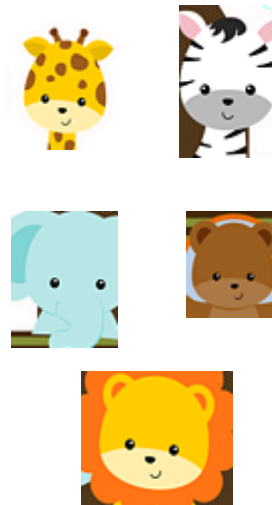


Ex: Uma **variável composta heterogênea** seria uma “Arca de Noé” e seus **elementos** seriam os diversos animais.

Variável Composta Heterogêneas



Elementos



Não abordaremos isto neste curso, pois este é um recurso típico de orientação a objetos e não de programação imperativa. Trata-se da **coleções**.

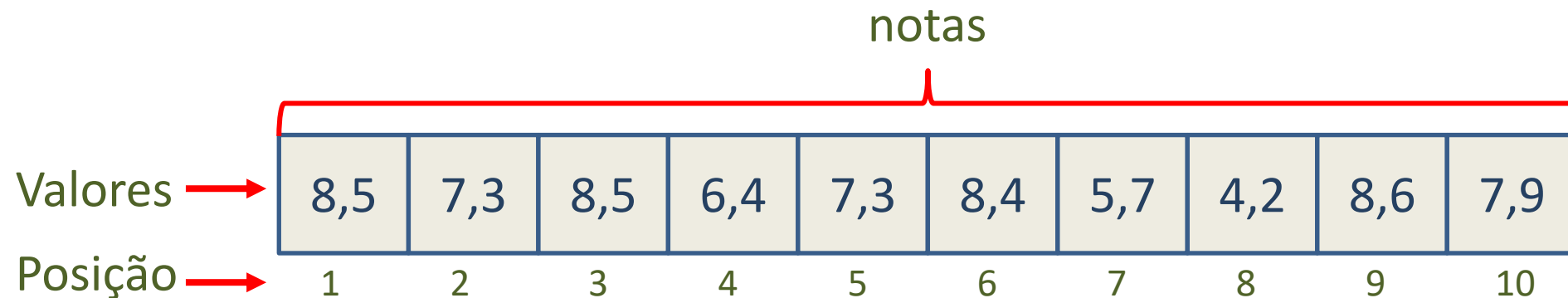
# Vetores ou Array

(Variáveis Compostas Homogêneas Unidimensionais)



**Ex:**

```
float notas[ ] = {8.5, 7.3, 8.5, 6.4, 7.3, 8.4, 5.7, 4.2, 8.6, 7.9};
```



Fonte: Prof. Dr. Anderson Elias, 2019



# Vetores ou Array

## (Variáveis Compostas Homogêneas Unidimensionais)

- Ao declaramos um vetor, os seus elementos não são inicializados.
- Mas é possível atribuir valores iniciais.
- Os valores iniciais são colocados entre chaves
- A quantidade de valores entre chaves não deve ser maior que o número de elementos
- A fim de facilitar a inicialização, C/C++ permite deixar o número de elementos em branco [].
- Neste caso, o compilador vai supor que o tamanho do vetor é igual ao número de valores especificados entre chaves

```
int vetor[] = {0, 2, 5, 3, 9}; // tamanho = 5
```

```
double notas[] = {10.0, 9.5, 7.5}; // tamanho = 3
```

# Vetores ou Array

(Variáveis Compostas Homogêneas Unidimensionais)



```
// declaração sem inicializar os valores do vetor (eles terão 'lixo')  
int v1[3];  
  
// declaração inicializando os valores do vetor  
int v2[3] = {0, 2, 5};  
  
// declaração alternativa inicializando os valores do vetor  
int v3[] = {0, 2, 5};
```

# Vetores ou Array

(Variáveis Compostas Homogêneas Unidimensionais)



## Outras informações:

- Não é permitido acessar um elemento de um array fora do seu limite => erro em tempo de execução.
- Use o `sizeof(vetor) / <número de bites>`, para saber o tamanho do array.

Fonte: Prof. Dr. Anderson Elias, 2019



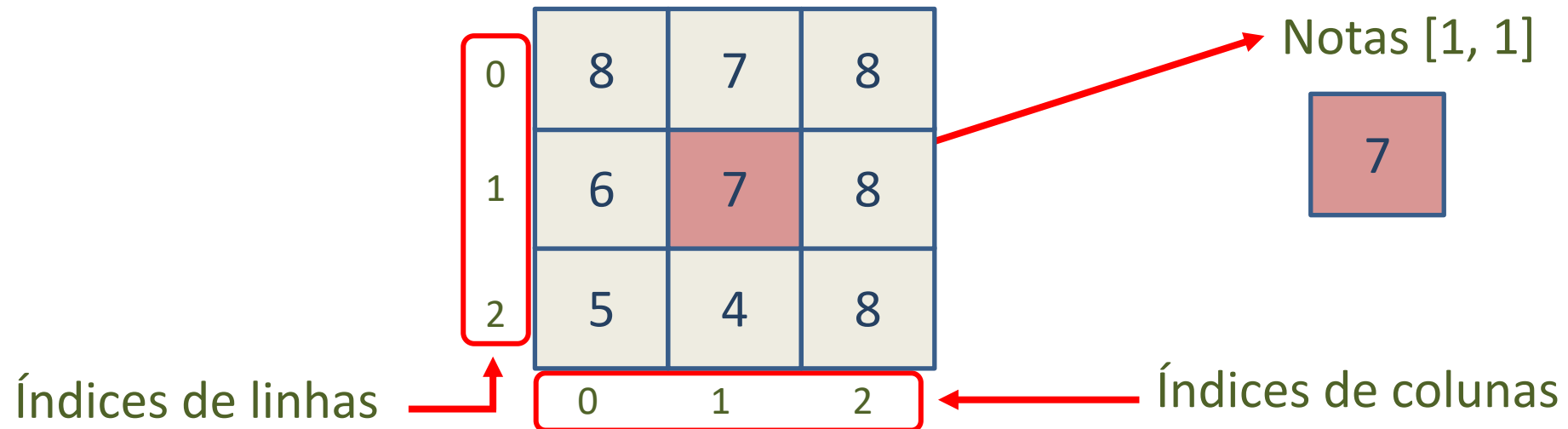
# Vetores Multidimensional ou Matriz

(Variáveis Compostas Homogêneas Multidimensionais)



Ex:

```
int notas[3][3] = {{8, 7, 8}, {6, 7, 8}, {5, 4, 8}};
```



Fonte: Prof. Drndo. Anderson Elias, 2019

# Vetores Multidimensional ou Matriz

(Variáveis Compostas Homogêneas Multidimensionais)



## Outras informações:

- Uma matriz precisa ser declarada com o número de linhas e colunas.

Fonte: Prof. Dr. Anderson Elias, 2019

# Exercício



Monte um vetor de inteiros de 121 até 1121.

- a) Qual a soma de todos os elementos?
- b) Qual a soma dos elementos pares?
- c) Qual a soma dos elementos ímpares?
- d) Qual a média?
- e) Faça um motor para encontrar o índice de um elemento determinado pelo usuário.
- f) Faça um motor para encontrar o elemento de um índice determinado pelo usuário.
- g) Faça um motor para encontrar o índice do elemento inteiro mais próximo à média.

# Exercício

$$\vec{A} = (q, r, s)$$

$$\vec{B} = (t, u, v)$$



Considere os vetores que seguem e responda.

$$A = \{1, 3, -2\}$$

$$B = \{-2, 2, 2\}$$

- a) Qual o produto escalar?
- b) Qual o módulo de A?
- c) Qual o módulo de B?
- d) Qual o ângulo de A faz como B?
- e) O ângulo é agudo, obtuso ou reto?
- f) Qual a projeção de A em B?

$$\vec{A} \cdot \vec{B} = (qt) + (ru) + (sv)$$

$$A = \sqrt{q^2 + r^2 + s^2}$$

$$\vec{A} \cdot \vec{B} = AB \cos \theta$$

$\cos \theta > 0$  é *agudo*     $\cos \theta < 0$  é *obtuso*     $\cos \theta = 0$  é *reto*

$$\text{proj}_{\vec{B}} \vec{A} = \frac{\vec{A} \cdot \vec{B}}{B^2} \vec{B}$$

```
#include <math.h>
```

```
double asin(double x);
```

```
double acos(double x);
```

(...)

$$\vec{A} = (q, r, s)$$

$$\vec{B} = (t, u, v)$$



... continuando

$$A = \{1, 3, -2\}$$

$$B = \{-2, 2, 2\}$$

$$\frac{x_1}{x_2} = \frac{y_1}{y_2} = \frac{z_1}{z_2} = k$$

Condição de paralelismo

g) A é paralelo a B?

h) A e B são linearmente dependentes?

i) Qual o produto vetorial?

j) Qual o determinante?

$L.D. = \text{paralelos}$  ou  $\det(\text{matriz\_coeficientes}) = 0$

$$\vec{A} \times \vec{B} = \begin{bmatrix} i & j & k \\ q & r & s \\ t & u & v \end{bmatrix} = (rv - us)i + (st - vq)j + (qu - tr)k$$

$$i = j = k = 1 \text{ (versor)}$$

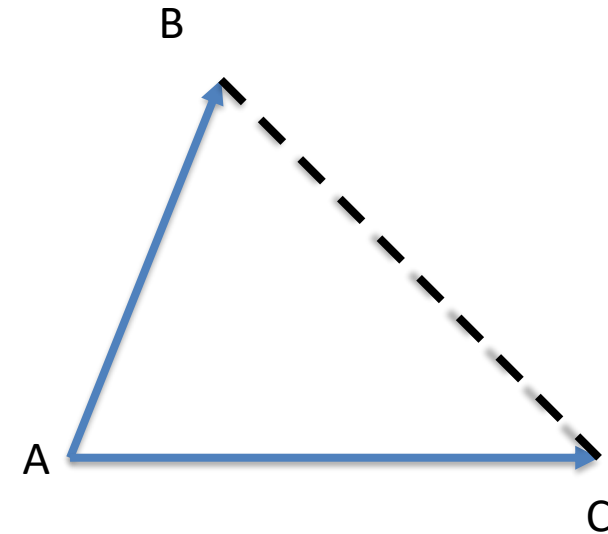
# Exercício

Calcule a área do triângulo de vértices:

$$A = \{2, 1, 3\}$$

$$B = \{6, 4, 1\}$$

$$C = \{-6, -2, 6\}$$



(...)

$$A = \{2, 1, 3\}$$

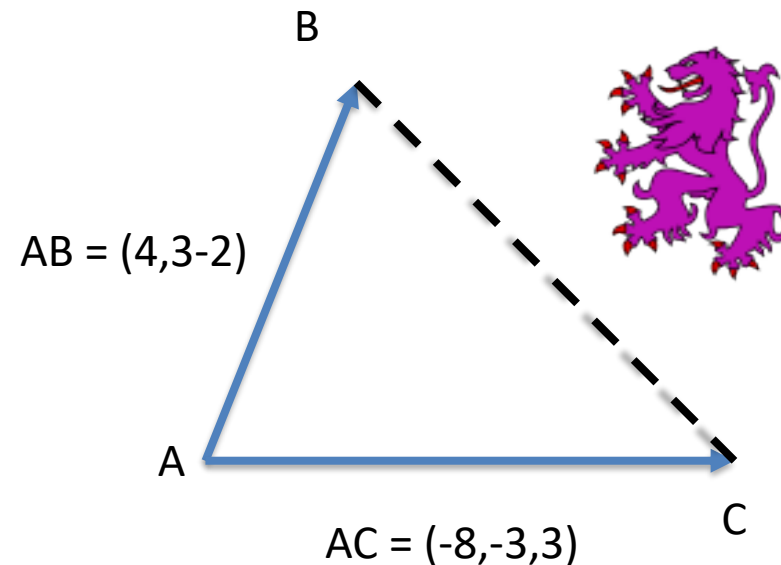
$$B = \{6, 4, 1\}$$

$$C = \{-6, -2, 6\}$$

Lembrando que o produto vetorial é a área do paralelogramo...

$$A_{\text{paralelogramo}} = |AB \times AC| = \left\| \begin{vmatrix} i & j & k \\ 4 & 3 & -2 \\ -8 & -3 & 3 \end{vmatrix} \right\| = |(3, 4, 12)| = \sqrt{3^2 + 4^2 + 12^2} = 13$$

$$A_{\text{triângulo}} = \frac{A_{\text{paralelogramo}}}{2} = \frac{13}{2}$$



# Até a próxima aula!

