

# **Fundação Hermínio Ometto**

## **Bacharelado em Sistemas de Informação**

SIF009 - Linguagem de Programação I

Prof. Dr. Sérgio Luis Antonello

**Aula 14**

03/11/2020

# Plano de ensino

1. Unidade I - Programação estruturada e Linguagem C (objetivos b, c e d).
  - 1.1. Conceitos de programação estruturada.
  - 1.2. Estrutura de um programa de computador.
  - 1.3. Códigos fonte, objeto e executável.
  - 1.4. Biblioteca de códigos.
  - 1.5. Compiladores e Interpretadores.
  - 1.6. Processos de compilação e link edição.
  - 1.7. Identificação dos tipos de erros e alertas (léxicos, sintáticos e semânticos).
  - 1.8. Depuração de código.
  - 1.9. Palavras reservadas.
  - 1.10. Tipos de dados.
  - 1.11. Constantes. Variáveis simples e estruturadas. Escopo de variáveis.
  - 1.12. Operadores e precedência.
  - 1.13. Expressões aritméticas, lógicas e relacionais.
  - 1.14. Comandos.
  - 1.15. Ambientes de desenvolvimento e programação.
2. Unidade II - Estruturas de controle (sequência, decisão e repetição), registro e arquivo.
  - 2.1. Comandos if e switch.
  - 2.2. Comandos for, while e do while.
  - 2.3. Blocos de comandos e aninhamento.
  - 2.4. Definição de tipos.
  - 2.5. Registro.
  - 2.6. Arquivo: leitura e gravação de dados em disco.
3. Unidade III - Ponteiros e Funções (objetivos a, c, d, e).
  - 3.1. Ponteiros.
  - 3.2. Funções.
  - 3.3. Passagem de parâmetro por valor.
  - 3.4. Passagem de parâmetro por referência.
4. Unidade IV - Strings e Variáveis indexadas (objetivos a, c, d).
  - 4.1. Manipulação de strings.
  - 4.2. Manipulação de caracteres.
  - 4.3. Declaração e manipulação de vetores.
  - 4.4. Declaração e manipulação de matrizes.

# Plano de ensino

Data	Atividade
04/08	Aula 01
11/08	Aula 02
18/08	Aula 03
25/08	Aula 04
01/09	Aula 05
08/09	Aula 06
15/09	Prova 1
22/09	Aula 08
29/09	Aula 09
06/10	Aula 10

Data	Atividade
13/10	Aula 11
20/10	Maratona FHO de Programação
27/10	Aula 13
03/11	Aula 14
10/11	Aula 15
17/11	Prova 2 Entrega Trabalho
24/11	Semana Científica
01/12	Prova SUB
08/12	Aula 19
15/12	Aula 20



# Sumário da aula

---

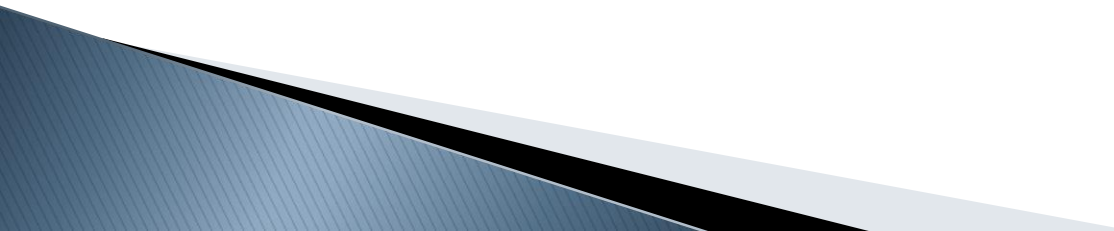
## Primeiro momento (revisão)

- ✓ Vetor
  - ✓ Declaração
  - ✓ Acesso a um elemento específico
  - ✓ Acesso a todos os elementos sequencialmente

## Segundo momento (conteúdo)

- ✓ Conceitos de variáveis indexadas
- ✓ Manipulação de matriz

## Terceiro momento (síntese)

- ✓ Retome pontos importantes da aula
- 

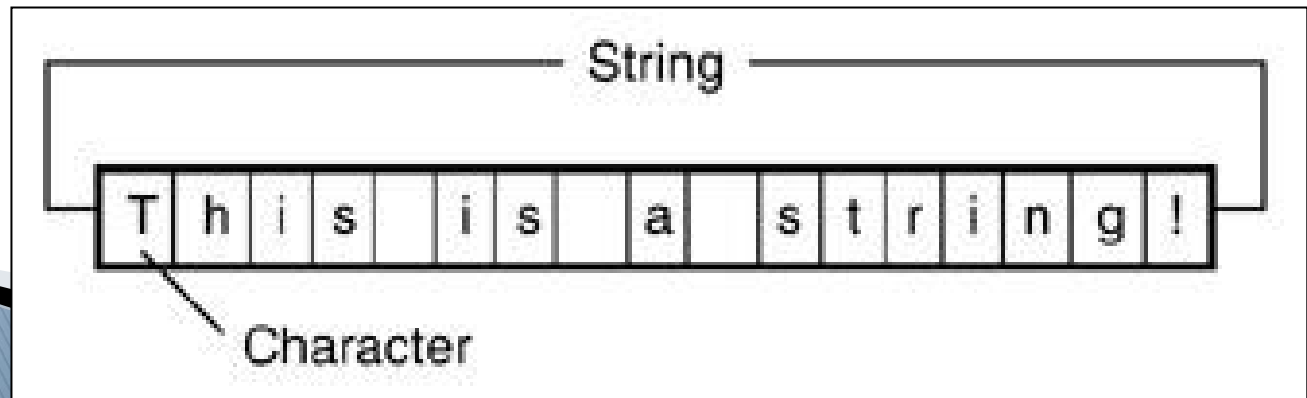
# 1. Primeiro momento: revisão

Grande parte dos dados armazenados e que circulam digitalmente estão no formato de strings.

É uma importante forma de dados, composta por caracteres alfanuméricos e caracteres especiais.

Cada caractere pode ser acessado individualmente, como em qualquer vetor, através do uso de colchetes.

Funções específicas permitem manusear tanto a strings como um todo quanto individualmente cada um de seus caracteres.



# 1. Primeiro momento: revisão

- ▶ Correção de exercícios



## 2. Segundo momento

---

- ▶ Conceitos de variáveis indexadas
- ▶ Manipulação de matriz

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

## 2. Segundo momento: motivação

---

É bastante comum a necessidade de armazenamento e manipulação de um ou vários dados de maneira sequencial.

Como por exemplo pode-se observar o cálculo das notas de um aluno, dado pela média entre 5 notas (nota da prova, nota do trabalho, nota da workshop e uma nota da entrega de tarefas).

São quatro valores numéricos de mesmo tipo para o cálculo de aluno.

E se precisarmos fazer esse processamento para todos os alunos de uma turma? Qual recurso poderemos usar?





## 2. Segundo momento: motivação

Será que ocorre a necessidade de se processar especificamente uma quantidade de dados de mesmo tipo?



RAZÃO SOCIAL DA EMPRESA					
RUA FULANO DE TAL, 123 - SÃO PAULO - SP					
CNPJ:00.000.000/0001-00 IE:000.000.000 IN:0.000.000-3					
01/01/2013 23:59:59 CCF:000001 COD:000001					
CNPJ/CPF consumidor: 000.000.000-00					
CUPOM FISCAL					
ITEM	CODIGO	DESCRICAO	QTD.	UN.	VLUNIT(R\$)
001	11202230	ACUA COCOKERO 200 ML	1.000	UN x 1,85	1,85
002	15154868	COCACOLA PET 2,5 LT	1.000	UN x 4,65	4,65
003	25220041	AZEITE CALO LT 500ML	1.000	UN x 14,95	14,95
004	25145153	SORVETE HORZS PT 1 LT	1.000	UN x 20,60	20,60
Subtotal R\$					42,05
DESCONTO-ICMS 10,00%					-4,20
TOTAL R\$					37,85
Dinheiro					37,85
2>02>hp*lpRrtbx.jhgT~bYYthgKlmmhHT\$**6Su=bugff4					
ZPM ZPM/1FIT LOGGER ECF-IF					
VERSAO:01.01.01 ECF:001 LOJA:0001 OPERADOR:001					
CCAP:INSAC 01/01/2013 23:59:59U					
FAB:ZP0000000000					



Aluno	Status Mat.	Faltas Mat.	Limit Faltas	SPA	Nota Anterior	Nota	Média
	CURSANDO	0	20	-	6,4	3,3	4,3
	CURSANDO	16	20	-	4,1	5,7	5,2
	CURSANDO	0	20	-	6,3	7,6	7,2
	CURSANDO	0	20	-	8,9	9,4	9,2
	CURSANDO	12	20	-	3,6	2,8	3,1
	CURSANDO	12	20	-	9,2	7,2	7,9
	CURSANDO	0	20	-	7,0	7,9	7,6
	CURSANDO	0	20	-	3,3	4,3	4,0
	CURSANDO	4	20	-	7,5	7,0	7,2

# 3. Variáveis indexadas

---

Variável Indexada é o conjunto de variáveis do mesmo tipo, referenciadas pelo mesmo nome e individualizadas por índices.

Uma variável indexada pode fazer uso de um ou mais índices, de acordo com a quantidade de dimensões desta variável.

Esses índices são necessários para permitir o acesso a cada elemento da variável indexada.

Vetor é uma variável indexada com uma dimensão.

Matriz é uma variável indexada de duas ou mais dimensões.



# 4. Matriz

---

Uma matriz caracteriza-se, também, por ser uma sequência de valores de um mesmo tipo.

Assim como no vetor, esses valores ficam alocados contiguamente na memória.

Cada elemento pode ser acessado a partir da sua posição.

Uma matriz necessita de **mais de um índice** para identificar cada um de seus elementos, cada índice para uma dimensão.

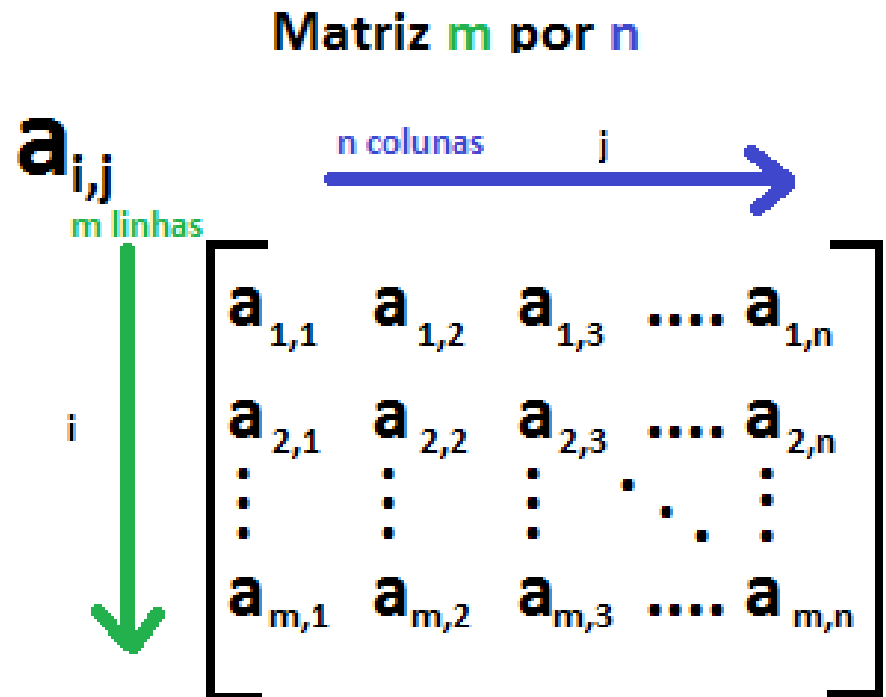


# 4. Matriz

Uma matriz bidimensional precisa de 2 índices;

Uma matriz tridimensional precisa de 3 índices; etc.

Um elemento da matriz pode ser usado no programa como qualquer outra variável.



# 4. Matriz

Para cada dimensão é necessário um índice;

O primeiro elemento da matriz tem **índice zero na linha** e **índice zero na coluna**.

```
int matriz [2] [5] = { {1, 2, 3, 4, 5} , {6, 7, 8, 9, 10} }
```

MATRIZ	coluna [0]	coluna [1]	coluna [2]	coluna [3]	coluna [4]
linha [0]	1	2	3	4	5
linha [1]	6	7	8	9	10

```
printf ("%i", matriz[0][2]);
```

3

# 4. Matriz

Alguns algoritmos precisam acessar sequencialmente todos os elementos da matriz.

Exemplo:

```
int main() {  
    int AluNotas[5][3];  
    int indalu, indnot;  
  
    for(indalu=0; indalu<5; indalu++) {  
        for (indnot=0; indnot<3; indnot++) {  
            scanf("%d", &AluNotas[indalu][indnot]);  
        }  
    }  
}
```

# 5. Exercícios

---

**Vamos  
Programar!**





# 5. Exercícios

---

- 1) Carregar duas matrizes bidimensionais A e B, cada uma com 4 linhas e 3 colunas. Construir uma matriz C de mesma dimensão, onde C é formada pela soma dos respectivos elementos da matriz A com B.
- 2) Resolva o problema URI 1181 - Linha na Matriz.  
<https://www.urionlinejudge.com.br/judge/pt/problems/view/1181>
- 3) Escrever um programa que lê uma matriz float de 5 linhas e 10 colunas e no final exiba toda a coluna onde está o menor valor da matriz. Se tiver mais de um elemento igual ao de menor valor igual, considere a coluna do primeiro deles..



## 6. Terceiro momento: síntese

---

As variáveis indexadas usam índices para acessar cada um de seus elementos.

Um vetor usa um índice.

Uma matriz bidimensional usa dois índices, um para especificar a linha e outro para especificar a coluna da matriz.

Rotinas com loops aninhados possibilitam acessar individualmente todos os elementos de uma matriz.

