



**USP - ICMC - SSC
SSC 0501 - 1o. Semestre 2011**

**Disciplina de
Introdução à Ciência da Computação
ICC 1 - Teoria**

Prof. Fernando Santos Osório

Email: fosorio [at] { icmc. usp. br , gmail. com }

Página Pessoal: <http://www.icmc.usp.br/~fosorio/>

Web - WIKI ICMC: <http://wiki.icmc.usp.br/index.php/SSC-501>

PAE: Daniel Sales (Mestr. CCMC – LRM)

Email: dsales [at] icmc.usp.br

Monitor: Danilo Alvares da Silva

E-mail: danilo [at] grad.icmc.usp.br

Linguagem de Programação “C”

Agenda:

- **Vetores: Coleção de dados do mesmo tipo (homogêneos)**
Unidimensional, Uso de um único índice p/acesso
- **Matrizes: Coleção de dados do mesmo tipo (homogêneos)**
Bi-dimensional, Tri-dimensional, Multi-Dimensional
Uso de um índice para cada uma das dimensões
- **Programa com Laço FOR em “C”:**
For While Do-While
Break Continue Return Exit

Informações Complementares e Atualizadas:

Consulte REGULARMENTE o material disponível na COTEIA

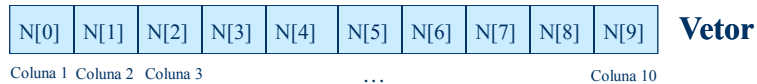
Linguagem "C": VETORES

VETORES: Agrupando Dados Iguais em Seqüência

E se eu precisasse declarar 10 Notas?
Teria que criar 10 variáveis ??!

Nota1, Nota2, Nota3, Nota4, Nota5,
Nota6, Nota7, Nota8, Nota9, Nota10 (UFA!!!)

Nestes casos podemos usar um VETOR ou uma seqüência de variáveis formando uma lista onde eu indico o seu **nome** e o seu **índice** (coluna) que eu desejo acessar



Linguagem "C": VETORES

Tipos de Dados Compostos: Estruturas HOMOGENEAS

VETORES

Exemplos Típicos:

VETOR DE CARACTERES = *STRING*

char Texto[10];

Texto[0] até Texto[9] <= São 10 posições de 1 char, lado a lado

VETOR DE INTEIROS = *TABELA*

int Tabela[10];

Tabela[0] até Tabela[9] <= São 10 posições de 1 int, lado a lado

VETOR DE DOUBLES = *DADOS*

double Dados[10];

Dados[0] até Dados[9] <= São 10 posições de 1 double, lado a lado

VETORES: Agrupando Dados Iguais em Seqüência

Criando e usando um vetor de 10 valores do tipo double...

```
main()
{
    double valor[10];  int i;

    printf ("Entre com a nota 1: ");
    scanf ("%lf",&valor[1]);
    printf ("Entre com a nota 2: ");
    scanf ("%lf",&valor[2]);

    /* Laço... */
    for ( i=0; i < 10; i++)
    {
        printf ("Entre com a nota nro. %d: ", i);
        scanf ("%lf",&valor[i]);
    }
}
```

VETORES: Agrupando Dados Iguais em Seqüência

Criando e usando um vetor de 10 valores do tipo double...

```
main()
{
    double valor[10];  int i;

    printf ("Entre com a nota 1: ");
    scanf ("%lf",&valor[1]);
    printf ("Entre com a nota 2: ");
    scanf ("%lf",&valor[2]);

    /* Laço... */
    i=0;
    do { printf ("Entre com a nota nro. %d: ", i);
        scanf ("%lf",&valor[i]); i++;
    } while (i < 10);
}
```

VETORES: Agrupando Dados Iguais em Seqüência

Criando e usando um vetor de 10 valores do tipo double...

```
main()
{
    double valor[10];  int i;

    printf ("Entre com a nota 1: ");
    scanf ("%lf",&valor[1]);
    printf ("Entre com a nota 2: ");
    scanf ("%lf",&valor[2]);

    /* Laço... */
    i=0;
    while (i < 10)
    {
        printf ("Entre com a nota nro. %d: ", i);
        scanf ("%lf",&valor[i]); i++;
    }
}
```

Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

VETORES

Vetores Numéricos:

```
#define MAX_NOTAS 10
double N[MAX_NOTAS]; /* Notas de até "Max_Notas" alunos */
```

```
N[0] = 10.0;
```

```
N[1] = 5.0;
```

```
Qtde_Notas = 3; /* Última = Qtde_Notas - 1 */
```

```
N[Qtde_Notas++] = 9.0; /* Nota índice 3 */
```

```
N[Qtde_Notas++] = 8.0; /* Nota índice 4 */
```

N[0]	N[1]	N[2]	N[3]	N[4]	N[5]	N[6]	N[7]	N[8]	N[9]
10.0	5.0	7.77	9.0	8.0					

↑
Qtde_Notas

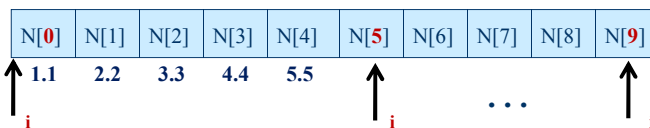
VETORES: Revisão

Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

VETORES

Vetores Numéricos:

```
#define MAX_NOTAS 10
double N[MAX_NOTAS]; /* Notas de até "Max_Notas" alunos */
int i;
for (i = 0; i < MAX_NOTAS; i++)
{
    printf("Entre com a nota %d: ", i);
    scanf("%lf", &N[i]);
}
```

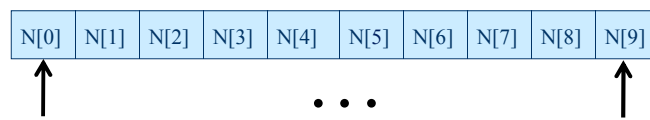


VETORES: Revisão

Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

VETORES

```
#define MAX 10
double N[MAX]; /* Notas de até 10 alunos */
int Qtde_Notas=0, Repete=1;
while (( Repete ) && ( Qtde_Notas < MAX ))
{
    scanf("%lf",& N[Qtde_Notas]);
    if (N[Qtde_Notas] < 0.0)
        Repete = 0;
    else Qtde_Notas++;
}
```



VETORES e MATRIZES

➤ O QUE MAIS PODEMOS FAZER COM VETORES...

➤ Podemos fazer mais!!!

VETORES - Vetores com UMA dimensão + UMA dimensão da strings (que são vetores de caracteres!)

```
int Codigo_Produto[10];
```

```
double Tabela_Precos[10];
```

```
char Tabela_Produtos[10] ????
```

```
>> 10 caracteres?
```

```
>> E se eu quiser guardar o nome de 10 produtos diferentes?
```

VETORES e MATRIZES

➤ O QUE MAIS PODEMOS FAZER COM VETORES...

➤ Podemos fazer mais!!!

VETORES - Vetores com UMA dimensão + UMA dimensão da strings (que são vetores de caracteres!)

```
int Codigo_Produto[10];
```

```
double Tabela_Precos[10];
```

```
char Tabela_Produtos[10] ????
```

```
>> 10 caracteres?
```

```
>> E se eu quiser guardar o nome de 10 produtos diferentes?
```

```
char TABELA_PRODUTOS [10][30];
```

```
/* 10 produtos com até 30 caractere no nome */
```

MATRIZES: Vetores bi-dimensionais

Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

MATRIZES - Vetores com mais de uma dimensão

Vetores de strings bi-dimensionais:

Armazenar em uma tabela o nome de 10 alunos (nome + '\0' com até 40 chars)

Tabela de Alunos:

```
char Nome_Alunos [10][40];
```

```
strcpy (Nome_Alunos[0], "Fulano da Silva");
```

```
strcpy (Nome_Alunos[1], "Beltrano de Oliveira");
```

```
scanf ("%s", Nome_Alunos[2] );
```

```
scanf ("%s", Nome_Alunos[3] );
```

```
printf ("Nome do aluno 0: %s \n", Nome_Alunos[0] );
```

```
printf ("Nome do aluno 1: %s \n", Nome_Alunos[1] );
```

```
printf ("Primeira letra do nome do aluno 3: %c \n", Nome_Alunos[3][0] );
```

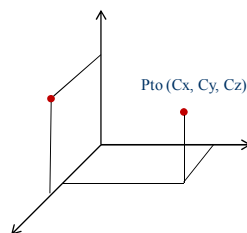
VETORES e MATRIZES

➤ O QUE MAIS PODEMOS FAZER COM VETORES...

➤ Podemos fazer mais!!!

VETORES - Vetores com UMA dimensão Múltiplos Vetores com UMA dimensão

Exemplo: Ponto representado pela Coordenada (X,Y,Z) → Cx, Cy, Cz
Como fazer para armazenar um vetor de pontos?



VETORES e MATRIZES

➤ O QUE MAIS PODEMOS FAZER COM VETORES...

➤ Podemos fazer mais!!!

VETORES - Vetores com UMA dimensão Múltiplos Vetores com UMA dimensão

Exemplo: Ponto representado pela Coordenada (X,Y,Z) → Cx, Cy, Cz
Como fazer para armazenar um vetor de pontos: vetores de Cx, Cy e Cz!
Ponto Zero => Cx[0] Cy[0] Cz[0] **Ponto Um** => Cx[1] Cy[1] Cz[1]

Cx[0]	Cx[1]	Cx[2]	Cx[3]	Cx[4]	Cx[5]	Cx[6]	Cx[7]	Cx[8]	Cx[9]
Cy[0]	Cy[1]	Cy[2]	Cy[3]	Cy[4]	Cy[5]	Cy[6]	Cy[7]	Cy[8]	Cy[9]
Cz[0]	Cz[1]	Cz[2]	Cz[3]	Cz[4]	Cz[5]	Cz[6]	Cz[7]	Cz[8]	Cz[9]

Vetores e Matrizes

VETORES

N1[0]	N1[1]	N1[2]	N1[3]	N1[4]	N1[5]	N1[6]	N1[7]	N1[8]	N1[9]	double N1[10];
N2[0]	N2[1]	N2[2]	N2[3]	N2[4]	N2[5]	N2[6]	N2[7]	N2[8]	N2[9]	double N2[10];
N3[0]	N3[1]	N3[2]	N3[3]	N3[4]	N3[5]	N3[6]	N3[7]	N3[8]	N3[9]	double N3[10];

Estas variáveis irão armazenar dados de **10 alunos** (com suas notas N1, N2, N3).

Onde cada vetor guarda um conjunto:

- (i) N1 => Nota1 de cada aluno
- (ii) N2 => Nota2 de cada aluno
- (iii) N3 => Nota3 de cada um dos 10 alunos

Entrada de Dados:

Ler os valores destas 3 notas para cada um dos 10 alunos e armazenar os dados em um vetor.
Ler os dados de N1, N2 e N3 para o primeiro aluno, depois de N1, N2 e N3 para o segundo aluno,
e assim por diante até o décimo aluno.

Os valores de N1, N2 e N3 são valores com casas após a vírgula (nros. reais)

VETORES e MATRIZES

Tipos de Dados Compostos: Estruturas HOMOGENEAS

VETORES - Vetores com UMA dimensão Sequência de Dados

N[0]	N[1]	N[2]	N[3]	N[4]	N[5]	N[6]	N[7]	N[8]	N[9]
------	------	------	------	------	------	------	------	------	------

Coluna 1 Coluna 2 Coluna 3 ... Coluna 10

Vetor

MATRIZES - Vetores com mais de uma dimensão TABELAS de Dados

N[0]	N[1]	N[2]	N[3]	N[4]	N[5]	N[6]	N[7]	N[8]	N[9]
N[0]	N[1]	N[2]	N[3]	N[4]	N[5]	N[6]	N[7]	N[8]	N[9]
N[0]	N[1]	N[2]	N[3]	N[4]	N[5]	N[6]	N[7]	N[8]	N[9]

Coluna 1 Coluna 2 Coluna 3 ... Coluna 10

Matriz

VETORES e MATRIZES

➤ O QUE MAIS PODEMOS FAZER COM VETORES...

➤ Podemos fazer muito mais!!!

VETORES - Vetores com UMA dimensão
Vetores com DUAS dimensões (matriz/tabela)
Vetores com TRÊS dimensões (cubo?)
Vetores com mais de 3 dimensões!
Vetores “N” (hiper-)dimensionais !!!!

VETORES e MATRIZES

- O QUE MAIS PODEMOS FAZER COM VETORES...
- Podemos fazer muito mais!!!

VETORES - Vetores com **UMA** dimensão
Vetores com **DUAS** dimensões (matriz/tabela)
Vetores com **TRÊS** dimensões (cubo?)
Vetores com mais de 3 dimensões!
Vetores “N” (hiper-)dimensionais !!!!



Vetores e Matrizes

VETORES

N[0]	N[1]	N[2]	N[3]	N[4]	N[5]	N[6]	N[7]	N[8]	N[9]
------	------	------	------	------	------	------	------	------	------

V[0]	V[1]	V[2]	V[3]	V[4]	V[5]	V[6]	V[7]	V[8]	V[9]
------	------	------	------	------	------	------	------	------	------

MATRIZES (ARRAYS)

M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]
M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]
M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]

VETORES e MATRIZES

Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

MATRIZES - Vetores com mais de uma dimensão

Vetores numéricos bi-dimensionais:

3 x 10

int Matriz [3][10];

Matriz[0][0] = 1;

...

Matriz [2][9] = 30;

M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]
M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]
M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]

- Inicialização de vetores:

int num [5] = { 1, 2, 3, 4, 5 };

char vogais[5] = { 'a', 'e', 'i', 'o', 'u' };

double matriz [3][2] = { { 0,0 }, { 0,1 },
 { 1,0 }, { 1,1 },
 { 2,0 }, { 2,1 } };

Matriz do Jogo da Velha

char Tabuleiro [3][3];

'O'	'X'	'X'
'.'	'O'	'.'
'.'	'.'	'O'

MATRIZES: Vetores bi-dimensionais

Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

MATRIZES - Vetores com mais de uma dimensão

Vetores numéricos bi-dimensionais:

3 x 10

int Matriz [3][10];

Matriz[0][0] = 1;

...

Matriz [2][9] = 30;

M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]
M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]
M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]

Tabela 3 x 10

MATRIZES: Vetores bi-dimensionais

VETORES

N1[0]	N1[1]	N1[2]	N1[3]	N1[4]	N1[5]	N1[6]	N1[7]	N1[8]	N1[9]	double N1[10];
N2[0]	N2[1]	N2[2]	N2[3]	N2[4]	N2[5]	N2[6]	N2[7]	N2[8]	N2[9]	double N2[10];
N3[0]	N3[1]	N3[2]	N3[3]	N3[4]	N3[5]	N3[6]	N3[7]	N3[8]	N3[9]	double N3[10];

Estas variáveis irão armazenar dados de **10 alunos** (com suas notas N1, N2, N3).
 Onde cada vetor guarda um conjunto:

- (i) N1 => Nota1 de cada aluno
- (ii) N2 => Nota2 de cada aluno
- (iii) N3 => Nota3 de cada aluno

MATRIZ 3 x 10

M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]	Linha do N1: [0][aluno]
M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]	Linha do N2: [1][aluno]
M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]	Linha do N3: [2][aluno]

23
 Set. 2009

MATRIZES: Vetores bi-dimensionais

VETORES

Estas variáveis irão armazenar dados de **10 alunos** (com suas notas N1, N2, N3).
 Onde cada vetor guarda um conjunto:

- (i) N1 => Nota1 de cada aluno
- (ii) N2 => Nota2 de cada aluno
- (iii) N3 => Nota3 de cada aluno

double N1[10];
 double N2[10];
 double N3[10];

MATRIZ 10 x 3

M[0][0]	M[0][1]	M[0][2]
M[1][0]	M[1][1]	M[1][2]
M[2][0]	M[2][1]	M[2][2]
M[3][0]	M[3][1]	M[3][2]
M[4][0]	M[4][1]	M[4][2]
M[5][0]	M[5][1]	M[5][2]
M[6][0]	M[6][1]	M[6][2]
M[7][0]	M[7][1]	M[7][2]
M[8][0]	M[8][1]	M[8][2]
M[9][0]	M[9][1]	M[9][2]

A
L
U
N
O
S

N1 N2 N3

double Notas_Turma[Max_Alunos] [Max_Notas]

Linha do N1: [aluno] [0]
 Linha do N2: [aluno] [1]
 Linha do N3: [aluno] [2]

Linha do N1: [0][aluno]
 Linha do N2: [1][aluno]
 Linha do N3: [2][aluno]

MATRIZ 3 x 10

M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]
M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]
M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]

24
 Set. 2009

MATRIZES: Vetores bi-dimensionais

Tipos de Dados Compostos: Estruturas HOMOGÊNEAS

MATRIZES - Vetores com mais de uma dimensão

Vetores numéricos bi-dimensionais:

3 x 10

int Matriz [3][10];

Matriz[0][0] = 1;

...

Matriz [2][9] = 30;

M[0][0]	M[0][1]	M[0][2]	M[0][3]	M[0][4]	M[0][5]	M[0][6]	M[0][7]	M[0][8]	M[0][9]
M[1][0]	M[1][1]	M[1][2]	M[1][3]	M[1][4]	M[1][5]	M[1][6]	M[1][7]	M[1][8]	M[1][9]
M[2][0]	M[2][1]	M[2][2]	M[2][3]	M[2][4]	M[2][5]	M[2][6]	M[2][7]	M[2][8]	M[2][9]

- Inicialização de vetores:

int num [5] = { 1, 2, 3, 4, 5 };

char vogais[5] = { 'a', 'e', 'i', 'o', 'u' };

double matriz [3][2] = { { 0.0 }, { 0.1 },
 { 1.0 }, { 1.1 },
 { 2.0 }, { 2.1 } };

Matriz do Jogo da Velha

char Tabuleiro [3][3];

'O'	'X'	'X'
'.'	'O'	'.'
'.'	'.'	'O'

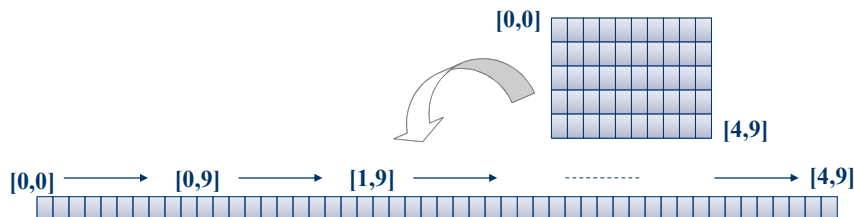
25

Set. 2009

MATRIZES: Vetores Multi-Dimensionais

Arrays Multidimensionais

- Arrays podem ter diversas dimensões, cada uma identificada por um par de colchetes na declaração.
- Ex: `int matriz[5][10];`
 - declara uma matriz de 5 linhas e 10 colunas:
 - na memória, entretanto, cada um dos inteiros são armazenados linearmente:

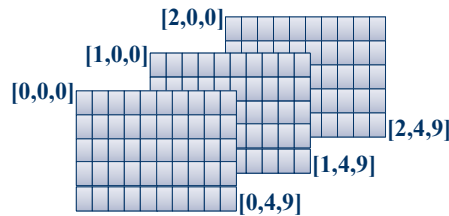


26

Set. 2009

Arrays Multidimensionais

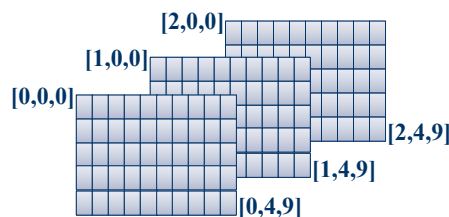
- Arrays podem ter diversas dimensões, cada uma identificada por um par de colchetes na declaração.
- Ex: `int matriz[5][10][3];`
 - declara uma matriz de 5 linhas e 10 colunas:
 - na memória, entretanto, cada um dos inteiros são armazenados linearmente:



`int matriz[3][5][10];`

Arrays Multidimensionais HOMOGÊNEOS

- Arrays podem ter diversas dimensões, cada uma identificada por um par de colchetes na declaração.
- Ex: `int matriz[5][10][3];`
 - declara uma matriz de 5 linhas e 10 colunas:
 - na memória, entretanto, cada um dos inteiros são armazenados linearmente:



`int matriz[3][5][10];`

Vetores e Matrizes

VETORES – Múltiplos vetores HETEROGÊNEOS

Cod[0]	Cod[1]	Cod[2]	Cod[3]	Cod[4]	Cod[5]	Cod[6]	Cod[7]	Cod[8]	Cod[9]	<code>int Cod[10];</code>
Val[0]	Val[1]	Val[2]	Val[3]	Val[4]	Val[5]	Val[6]	Val[7]	Val[8]	Val[9]	<code>double Val[10];</code>
Tipo[0]	Tipo[1]	Tipo[2]	Tipo[3]	Tipo[4]	Tipo[5]	Tipo[6]	Tipo[7]	Tipo[8]	Tipo[9]	<code>char Tipo[10];</code>

Estas variáveis irão armazenar dados de **10 produtos** (com seu Código, Valor e Tipo).
Onde cada vetor guarda um conjunto distinto de dados

- (i) Cod => Código do produto
- (ii) Val => Valor do produto
- (iii) Tipo => Tipo do produto ('L' = Líquido, 'S' = 'Sólido', 'G' = Gasoso)

Vetores e Matrizes

VETORES – Múltiplos vetores HETEROGÊNEOS

Cod[0]	Cod[1]	Cod[2]	Cod[3]	Cod[4]	Cod[5]	Cod[6]	Cod[7]	Cod[8]	Cod[9]	<code>int Cod[10];</code>
Val[0]	Val[1]	Val[2]	Val[3]	Val[4]	Val[5]	Val[6]	Val[7]	Val[8]	Val[9]	<code>double Val[10];</code>
Tipo[0]	Tipo[1]	Tipo[2]	Tipo[3]	Tipo[4]	Tipo[5]	Tipo[6]	Tipo[7]	Tipo[8]	Tipo[9]	<code>char Tipo[10];</code>

Estas variáveis irão armazenar dados de **10 produtos** (com seu Código, Valor e Tipo).
Onde cada vetor guarda um conjunto distinto de dados

- (i) Cod => Código do produto
- (ii) Val => Valor do produto
- (iii) Tipo => Tipo do produto ('L' = Líquido, 'S' = 'Sólido', 'G' = Gasoso)

E se o produto possuir uma descrição textual?

```
char Nome_Prod [10][100];    /* ☺ */
```



INFORMAÇÕES SOBRE A DISCIPLINA

USP - Universidade de São Paulo - São Carlos, SP
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação

Prof. Fernando Santos OSÓRIO

Web institucional: <http://www.icmc.usp.br/ssc/>

Página pessoal: <http://www.icmc.usp.br/~fosorio/>

E-mail: [fosorio \[at\] icmc. usp. br](mailto:fosorio@icmc.usp.br) ou [fosorio \[at\] gmail. com](mailto:fosorio@gmail.com)

PAE Daniel Sales – E-mail: [dsales \[at\] icmc.usp.br](mailto:dsales@icmc.usp.br)

Monitor Danilo Alvares – E-mail: [danilo \[at\] grad.icmc.usp.br](mailto:danilo@grad.icmc.usp.br)

Disciplina de Introdução a Ciência da Computação

Web disciplina: Wiki ICMC - [Http://wiki.icmc.usp.br](http://wiki.icmc.usp.br)

> Programa, Material de Aulas, Critérios de Avaliação,

> Trabalhos Práticos, Datas das Provas, Notas