

# Estruturas de Dados

Prof. Patrícia Noll de Mattos

e-mail: [patricia.mattos@ulbra.br](mailto:patricia.mattos@ulbra.br)

# Competências

- Ser capaz de fazer uso eficiente da **memória** e buscar reduzir o **tempo de processamento**;
- Identificar a **melhor estrutura de dados** para determinado **contexto**;
- Ser capaz de **projetar e implementar** as principais estruturas de dados, facilitando os processos de **inclusão e exclusão** de elementos em estruturas.

# Processo Avaliativo

- Bloco de desenvolvimento 1
    - Avaliação Parcial 1 (AP1) – 1.5 pontos
  - Bloco de desenvolvimento 2
    - Avaliação Parcial 2 (AP2) – 2.5 pontos
  - Bloco de sistematização
    - Avaliação semestral (AS -cumulativa e sem consulta) – 6.0 pontos
  - Totalizando 10 da pontuação do semestre. Se o aluno atingir pontuação mínima 7.0, está aprovado.
  - Avaliação final (AF): individual, cumulativa e sem consulta, pontuação 10.
    - Frequência mínima;
    - Ter realizado a avaliação semestral.
- .Não existe mais substituição, não perder avaliações!!**

# Abordagens Temáticas

- Listas Encadeadas com e sem header e circular.
- Pilha e Fila.
- Grafos.
- Recursividade.
- Árvores e suas generalizações.

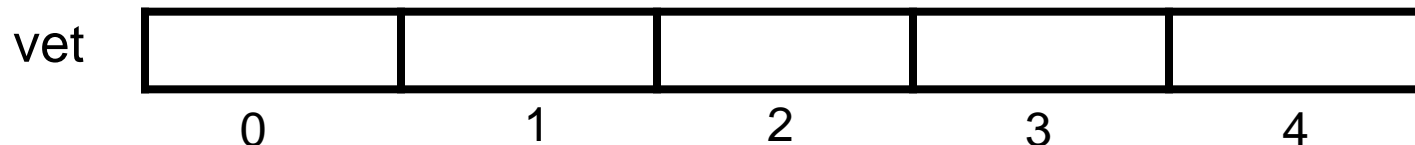
# Vetores e Matrizes

## Matrizes

Consiste em uma certa quantidade de **valores** que são referenciados pelo **mesmo nome**. Tais valores são armazenados de **forma contínua** na memória e são do **mesmo tipo**.

**Matriz Unidimensional - Vetor** (Schildt, p. 92)

**tipo nome[tamanho];**



## Exemplo 1 (vetores):

```
int vet[5];  
int i;  
for(i=0;i<5;i++)  
{  
    printf("Digite o elemento %d:\n",i);  
    scanf("%d", &vet[i]); fflush(stdin);  
}
```

//Deve-se utilizar um índice para cada dimensão! O Primeiro valor de cada índice é zero.

Exemplo 2 (matrizes) :

```
int m[10][10];  
int i,j;  
for(i=0;i<10;i++)  
    for(j=0;j<10;j++) {  
        printf("Digite o elemento %d%d:\n",i,j);  
        scanf("%d", &m[i][j]);  
    }  
}
```

//Deve-se utilizar um índice para cada dimensão! O Primeiro valor de cada índice é zero.

# Inicialização de vetores e matrizes

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

```
float umaMatriz[3][3] = {  
    {1.0, 1.1, 1.2},  
    {2.0, 2.1, 2.2},  
    {3.0, 3.1, 3.2},  
};
```

```
int i, j;           //mostrar elementos
```

```
for(i = 0; i < 10; i++)  
    printf("Vetor: %d\n", umVetor[i]);
```

```
for(i = 0; i < 3; i++)  
    for(j = 0; j < 3; j++)  
        printf("Matriz: %f\n", umaMatriz[i][j]);
```



# Estruturas Compostas

Definição de estrutura contendo campos com possibilidade de tipos diferentes de dados:

Declaração em c:

```
struct aluno {  
    inteiro codAluno;  
    float nota;  
};
```

```
struct aluno varAluno;  
struct aluno alunos[5];
```

Acesso aos campos:

```
//variável varAluno
```

```
varAluno.codAluno = 100;  
varAluno.nota = 8.5.
```

```
//vetor alunos
```

```
alunos[0].codAluno = 100;  
alunos[0].nota = 8.5;
```

```
#include <stdio.h>
#include <conio.h>>
```

# Exemplo

```
struct aluno {
    int codAluno; float nota;
};
int main() {
    struct aluno v[5];
    int i, m=0;
    printf("Digite os valores para o vetor:\n");
    for(i=0;i<5;i++){
        printf("digite o codigo:");
        scanf("%i", &v[i].codAluno); fflush(stdin);
        printf("digite a nota:");
        scanf("%f", &v[i].nota); fflush(stdin);
    }
    for(i=1; i<10; i++)
        if (vet[i].nota>vet[m].nota) m=i;
    printf("Codigo aluno com maior nota:%i\n", v[i].codAluno);
    printf("Nota aluno com maior nota:%f\n", v[i].nota);
    getch();
    return 0;
}
```

**Verifique o trecho de código em C a seguir:**

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int mat[5][5], i, j, soma=0;
    for( i=0; i<5; i++)
        for(j=0; j<5; j++) {
            printf("Digite o elemento %i%i:", i,j);
            scanf("%i", &mat[i][j]); fflush(stdin);
            soma = soma + mat[i][j];
        }
    printf("Soma=%i\n", soma);
    getch();
    return 0;
}
```

**Com base no trecho de código acima, entre no DevC++ e digite e execute esse programa, depois realize os exercícios a seguir:**

1. O que o trecho faz?

Baixar o DevC++: <https://sourceforge.net/projects/orwelldevcpp/>

Arquivo/Novo/Arquivo Fonte

Salvar

Executar/Compilar

Executar/Executar

2. Mostrar na tela os elementos da diagonal principal da matriz.

3. Passar os elementos da diagonal principal da matriz para um vetor e mostrá-los na tela.

4. Calcular e mostrar a média aritmética das notas do vetor v.