

## Disciplina Programação (CK0226) Tarefa de Laboratório nº 07 API de Vetores e Matrizes Dinâmicas

Prof. Miguel Franklin

### OBJETIVOS:

- Familiarização com a sintaxe e as estruturas da linguagem C
- Prática de projetos com múltiplos arquivos fonte
- Prática de alocação dinâmica de memória em C
- Prática de ponteiros

### ENUNCIADO:

Crie uma biblioteca (API – *Application Programming Interface*) para a criação e destruição de vetores e matrizes de tipo ponto flutuante duplo (*double*). A API deve ter as seguintes funções:

### VETOR DINÂMICO

```
void *criar_vetor(int tamanho)
```

- Esta função deve alocar o espaço necessário para armazenar o vetor da dimensão estipulada, retornando o endereço do espaço reservado.

```
double obter_vetor(void *mem, int x)
```

- Esta função deve retornar o valor armazenado na posição *x* do vetor armazenado na posição de memória *<mem>*.

```
int atribuir_vetor(void *mem, int x, double valor)
```

- Esta função deve armazenar o valor *<valor>* na posição *<x>* do vetor armazenado na posição de memória *<mem>*.

```
void liberar_vetor(void *mem)
```

- Esta função deve liberar o espaço de memória *<mem>*.

## MATRIZ DINÂMICA

```
void *criar_matriz(int tamanho_x, int tamanho_y)
```

- Esta função deve alocar o espaço necessário para armazenar uma matriz da dimensão <tamanho\_x> X <tamanho\_y>, retornando o endereço do espaço reservado.

```
double obter_matriz(void *mem, int x, int y)
```

- Esta função deve retornar o valor armazenado na posição (<x>,<y>) da matriz armazenada na posição de memória <mem>.

```
int atribuir_matriz(void *mem, int x, int y, double valor)
```

- Esta função deve armazenar o valor <valor> na posição (<x>,<y>) da matriz armazenada na posição de memória <mem>.

```
void liberar_matriz(void *mem)
```

- Esta função deve liberar o espaço de memória <mem>.

Exemplo de utilização da API:

```
#include <stdlib.h>
#include <stdio.h>
#include "../api_dinamica.h" // ← Onde estarão os protótipos da API

int main(int narg, char *argv[]) {
    void *p_m, *p_v;

    p_v = criar_vetor(16);
    atribuir_vetor(p_v, 8, 456.0) // Atribui o valor 456 na posição 8 do vetor
    printf("A pos. 8 do vetor contém %f.\n", obter_vetor(p_v, 8));
    liberar_vetor(p_v);

    p_m = criar_matriz(10, 10);
    atribuir_matriz(p_m, 5, 4, 12.0); // Atribui o valor 12 na posição (5,4)
    printf("A pos. (5,4) da matriz = %f.\n", obter_matriz(p_m, 5, 4));
    liberar_matriz(p_m);

    return EXIT_SUCCESS;
}
```

*A entrega (upload) deverá ser realizada através do Google Classroom.*