

# SCC 502 - ALGORITMOS E ESTRUTURAS DE DADOS I

## PRIMEIRO TRABALHO PRÁTICO

**Professores:** Debora Medeiros e Mario Gazziro

**Estagiário PAE:** Daniel Bonetti

### 1. Descrição do Trabalho

Imagine que você foi contratado para programar um ambiente para manipulação de matrizes de números reais que permita ao usuário:

1. **Declarar uma matriz:** fornecendo seu nome e dimensões desejadas.
2. **Destruir uma matriz:** fornecendo seu nome.
3. **Imprimir uma matriz:** fornecendo seu nome.
4. **Atribuir um elemento:** fornecendo o nome da matriz, as coordenadas e o valor.
5. **Atribuir uma linha a uma matriz:** fornecendo o nome da matriz o índice da linha e a sequência de elementos.
6. **Atribuir uma coluna a uma matriz:** fornecendo o nome da matriz o índice da coluna e a sequência de elementos.
7. **Transpor uma matriz:** fornecendo seu nome e o nome da matriz resultante (não precisa estar declarada).
8. **Somar duas matrizes:** fornecendo seus nomes e o nome de uma terceira matriz que deverá conter o resultado (não deve estar declarada). As 2 matrizes devem conter as mesmas dimensões.
9. **Dividir uma matriz por outra (elemento a elemento):** fornecendo seus nomes e o nome de uma terceira matriz que deverá conter o resultado (não deve estar declarada). As 2 matrizes devem conter as mesmas dimensões e a matriz que será o denominador da divisão não deve conter nenhum elemento igual a 0.
10. **Multiplicar uma matriz por outra:** fornecendo seus nomes e o nome de uma terceira matriz que deverá conter o resultado (não deve estar declarada). Sendo  $A_{m1 \times n1}$  e  $B_{m2 \times n2}$  as matrizes de entrada e  $A \times B$  a operação desejada,  $n1$  deve ser igual a  $m2$ .
11. **Multiplicar duas matrizes (elemento a elemento):** fornecendo seus nomes e o nome de uma terceira matriz que deverá conter o resultado (não deve estar declarada). As 2 matrizes devem conter as mesmas dimensões.

Para exercer essas funções, o sistema deve manter uma lista encadeada de matrizes onde cada nó deve conter o nome da matriz, o ponteiro para o próximo nó e um ponteiro onde será alocada a matriz dinamicamente (com as dimensões especificadas

pelo usuário). Os nós desta lista encadeada devem ser alocados conforme a necessidade, ou seja, quando o usuário quiser criar uma matriz ou chamar uma operação que resulte em uma matriz. A Figura 1 ilustra esta abordagem:

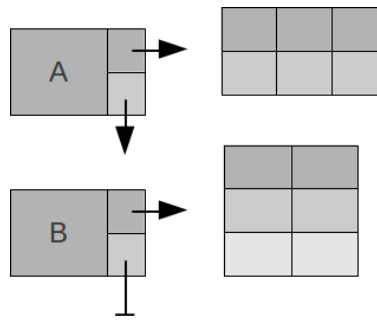


Figura 1: Lista encadeada de matrizes

Quando o usuário optar por descartar uma matriz, esta deve ser liberada da memória, assim como seu nó deve ser liberado da lista encadeada. Dois TAD's devem ser implementados, um para a lista encadeada e um para as matrizes.

## 2. Comandos e saídas

As funcionalidades do ambiente devem ser requisitadas por meio de comandos seguidos dos dados necessários. Os comandos são representados por duas letras maiúsculas. Os comandos que o sistema deve reconhecer são especificados a seguir. O símbolo `_` denota um espaço em branco. O símbolo `//` denota um comentário, não faz parte da saída ou dos comandos.

### Cria matriz

|         |  |
|---------|--|
| Comando | CM_ <code>&lt;nome sem espaços&gt;</code> _ <code>&lt;número de linhas&gt;</code> _ <code>&lt;número de colunas&gt;</code> |
| Saída   | //Vazio em caso de sucesso<br>ERRO //caso a matriz já exista ou alguma das dimensões não esteja entre 1 e 50               |

### Destroi matriz

|         |   |
|---------|---|
| Comando | DM_ <code>&lt;nome&gt;</code>                                 |
| Saída   | //Vazio em caso de sucesso<br>ERRO //caso a matriz não exista |

### Imprime matriz

|         |  |
|---------|--|
| Comando | IM_ <code>&lt;nome&gt;</code>  |
| Saída   | <code>a<sub>1,1</sub> ... a<sub>1,m</sub></code><br>... //separados por espaço, com alinhamento %4.2f<br><code>a<sub>n,1</sub> ... a<sub>n,m</sub></code><br>ERRO //caso a matriz não exista |

### Atribui um elemento da matriz

|         |   |
|---------|---|
| Comando | AE_<nome>_<linha>_<coluna>_<valor>  |
| Saída   | //Vazio em caso de sucesso<br>ERRO //caso a matriz não exista ou as coordenadas forem inválidas |

### Atribui uma linha da matriz<sup>1</sup>

|         |   |
|---------|---|
| Comando | AL_<nome>_<linha>_<val1>_..._<val_n>_#  |
| Saída   | //Vazio em caso de sucesso<br>ERRO //caso a matriz não exista, a linha seja inválida ou o número de valores não seja condizente com as dimensões da matriz. |

### Atribui uma coluna da matriz

|         |  |
|---------|--|
| Comando | AC_<nome>_<coluna>_<val1>_..._<val_m>_#  |
| Saída   | //Vazio em caso de sucesso<br>ERRO //caso a matriz não exista, a coluna seja inválida ou o número de valores não seja condizente com as dimensões da matriz. |

### Transpõe uma matriz

|         |   |
|---------|---|
| Comando | TM_<nome>_<nome-resultado sem espaços>  |
| Saída   | //Imprime matriz resultante (mesmo formato do comando IM)<br>ERRO //caso a matriz não exista ou a matriz resultante já exista |

### Somar duas matrizes

|         |  |
|---------|--|
| Comando | SM_<nome1>_<nome2>_<nome-resultado sem espaços >   |
| Saída   | //Imprime matriz resultante (mesmo formato do comando IM)<br>ERRO //caso alguma das matrizes não exista, suas dimensões não sejam as mesmas ou a matriz resultante já exista |

### Dividir uma matriz por outra (elemento a elemento)

|         |  |
|---------|--|
| Comando | DV_<nome1>_<nome2>_<nome-resultado sem espaços >   |
| Saída   | //Imprime matriz resultante (mesmo formato do comando IM)<br>ERRO //caso alguma das matrizes não exista, suas dimensões não sejam as mesmas, a segunda matriz contenha algum elemento igual a 0 ou a matriz resultante já exista |

### Multiplicar uma matriz por outra

|         |  |
|---------|--|
| Comando | MM_<nome1>_<nome2>_<nome-resultado sem espaços >   |
| Saída   | //Imprime matriz resultante (mesmo formato do comando IM)<br>ERRO //caso alguma das matrizes não exista, suas dimensões estejam incorretas (item 10 da Seção 1) ou a matriz resultante já exista |

<sup>1</sup>Sugestão: ler os valores como string com scanf, comparar com “#” e converter usando atof.