

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO

Relatório Técnico
Projeto: Calculadora de Matrizes

Josiluiz Nobre dos Santos
Matrícula: 20250063632

11 de dezembro de 2025

Conteúdo

Introdução	2
1 Metodologia	3
2 Análise do código	4
3 Dificuldades e soluções	6
Conclusão	7

Introdução

Este relatório discorre sobre a descrição de um projeto da disciplina Introdução à Técnicas de Programação, com o título: “Calculadora de Matrizes”, que relaciona o estudo dos tópicos da unidade 1 com os da unidade 2, ou seja uma melhora significativa com novas técnicas que agilizam a entrada, o pré-processamento, o processamento, a análise e a saída de dados. Uma calculadora de matrizes permite manipular os seus elementos com diversos tipos de operações, mas neste caso, para projeto são: adição, subtração, multiplicação, transporte, determinante por dois métodos distintos, matriz inversa por cofatores e um caça palavras(bônus).

Capítulo 1

Metodologia

O presente projeto contou-se com o desenvolvimento nas aulas e pesquisas com o intuito de formular um código em linguagem em c, utilizando o GCC(GNU Compiler Collection) como compilador, o IDE(Ambientes de Desenvolvimento Integrado) como o VS-code(Visual Studio Code) além de armazenar no Github. Dessa forma, o código foi sendo gerado a cada momento de conhecimento da linguagem e sendo reforçado por exercícios da disciplina. O código foi desenvolvido e organizado em: Declaração de funções e alocação de matrizes, funções para realizar a operações com matrizes e parte principal “main” que chama as funções por escolha do usuário, e com essa formação fica bem mais simples sua manutenção.

Capítulo 2

Análise do código

Geralmente os elementos de uma matriz são números, assim foi acrescida um “bônus” como um caça palavras, em que a própria função cria uma matriz aleatória com letras de A a Z, e o tamanho da matriz é fornecido pelo usuário em seguida pede-se uma palavra. Como o foco do projeto eram as matrizes, então a maioria das funções possuíam repetições aninhadas como em determinante de matrizes e o caça palavras. As matrizes foram implementadas em funções percorrendo linha e em seguida coluna, onde o usuário escrevia o tamanho da matriz em seguida os seus elementos, a partir daí começavam as operações.

Os ponteiros foram de suma importância em o todo o projeto, pois o código ficou mais eficiente no acesso de cada função em que manipulava as matrizes, sendo assim a passagem por referência de memória ajudou a alterar as variáveis iniciais, atribuindo diversos modelos, evitando cópias desnecessárias. Assim como o uso ponteiros em matrizes facilitou o seu manuseio, a alocação dinâmica permitiu reservar na memória apenas em tempo de sua execução e liberando em seguida, diferentemente em tempo de compilação o que podemos ver nesse trecho:

Alocando por linha, reserva espaço para os endereços das linha:

```
float **matriz = (float **)malloc(linhas * sizeof(int));
```

Alocando por coluna em loop:

```
matriz[i] = (float*)malloc(colunas * sizeof(int));
```

O código foi dividido em três partes:

1. Main principal: onde possui um menu para escolha de um número de 1 a 9, via comando “switch” e acessa os distintos tipos de funções para operação com matrizes.

PROGRAMA SIMULANDO UMA CALCULADORA DE MATRIZES
OPERAÇÕES:

- 1 - SOMA
- 2 - SUBTRAÇÃO
- 3 - MULTIPLICAÇÃO
- 4 - TRANSPORTE
- 5 - DETERMINANTE(MÉTODO DE LAPLACE)
- 6 - DETERMINANTE(MÉTODO LU)
- 7 - COFATOR, ADJUNTA E INVERSA
- 8 - CAÇA PALAVRAS
- 9 - SAIR

2. Funções que realizam as operações anteriores.
3. Funções auxiliares que serão usadas no item anterior como para: alocar matrizes, escrever matrizes, imprimir matrizes e para desalocá-las.

Capítulo 3

Dificuldades e soluções

No início, a principal dificuldade foi para entender e aplicar os conceitos, até um tanto abstrato, de ponteiros e de alocação dinâmica, daí assisti mais de uma vez aulas gravadas do curso e além de pesquisas na internet. Em seguida foi a continuação da unidade 1 do projeto, isto é, para implementar novos assuntos da unidade 2, adequando com ponteiros e alocando na memória, assim os desafios ficaram cada vez mais difíceis, e isso requereu muito tempo de estudo e tentativa e erro. Um desses desafios também foi implementar o determinante de matrizes usando dois métodos distintos (Laplace e decomposição LU), em que o primeiro foi necessário usar a recursividade, pois sempre será calculado o determinante de uma matriz de ordem inferior. Já o segundo, o objetivo era manipular as linhas da matriz com operações para obter uma matriz triangular superior. Por fim, pensei “como incrementar strings em matrizes?”, já que estudamos mais com matrizes numéricas, daí lembrei das sugestões dadas no curso. Dessa forma, utilizei o “caça palavra”, só que escolhi a matriz de strings aleatórias do alfabeto e após digitar uma palavra, busca-se a primeira letra na matriz, se caso for encontrado, analisa as oito direções possíveis.

Conclusão

Ao decorrer do processo de aprendizado, o projeto sintetiza muito bem os assuntos estudados durante a disciplina e aponta para uma maneira prática de reunir métodos e experiências vivenciadas saindo da teoria abstrata. A evolução foi bem significativa em relação à unidade 1, pois além dos tópicos anteriores todos da unidade 2 foram contemplados de strings à alocação dinâmica.

Acredito que ainda dá mais para enxugar o código, principalmente em encontrar uma maneira para unir as funções de soma e subtração de matrizes, e aumentando o número de operações com matrizes estendendo a quantidade de matrizes, como por exemplo, somar 2 ou mais matrizes, além trabalhar com sistema de equações lineares.