

O presente trabalho de calculadora de matrizes quadrada, triangulares superiores e inferiores e diagonal com o auxílio das estruturas lineares lista, dicionário e array. Para realizar esse código foi utilizado as bibliotecas array.array, json e ABC, que implementa classes abstratas que serão utilizadas posteriormente na manipulação das operações. O primeiro módulo define a classe matriz abstrata e suas determinadas operações como somar, subtrair e traço (matriz quadrada). A partir do segundo módulo, as classes das matrizes são definidas, matrizes diagonais, quadradas, triangulares inferiores e superiores, respectivamente. Acerca das funções, elas são definidas dentro das classes de cada matriz, utilizando as estruturas condicionais e de repetição: if, elif e else. No módulo de funções utilitárias, dá a entrada e define a matriz identidade. No próximo, tem-se a implementação das operações relacionadas ao salvamento e carregamento das matrizes incluídas diretamente pelo usuário e carregadas via arquivos para que possam ser mostradas na tela e realizadas as operações. O módulo seguinte, implementa a funções para que o usuário escolha com quais matrizes e qual operação será realizada com as matrizes escolhidas. O próximo módulo adiciona matrizes padrão, que servirão para o usuário ter exemplos e testar o funcionamento da calculadora antes de colocar seus próprios dados. Por último, é adicionada uma função que gera o menu interativo para o usuário.

A complexidade do código de tempo e espaço depende da matriz e da operação. A matriz diagonal possui $O(n)$, já as triangulares e quadradas atingem, $O(n^2)$. E, operações com multiplicação podem chegar até $O(n^3)$, no seu pior caso.

Durante o desenvolvimento do trabalho, encontramos dificuldades para conseguir encontrar as melhores formas de armazenar e operar com os dados, visto que precisávamos encontrar formas de manter os dados da matriz padrão colocada de exemplo no código e ter uma maneira de armazenar os dados que o usuário irá inserir via input direto pelo menu interativo ou por um arquivo que pode conter uma quantidade não conhecida de dados e operá-los da maneira mais dinâmica possível, sem que sua complexidade de tempo e espaço escalasse de forma inoportuna. Como resultado, obtivemos um código funcional, eficiente no uso de memória e bom desempenho nas operações básicas entre matrizes.