Relatório da Calculadora Matricial

integrantes : Felipe Rodrigues Ribeiro DRE:119052031

Igor Torres DRE:

Durante o processo de criação do programa, optamos por utilizar listas de listas em Python como a principal maneira de representar matrizes, já que temos mais afinidade com a linguagem de programação. Cada matriz foi definida como uma lista composta por outras listas que simbolizam suas linhas. Para facilitar o armazenamento e a para o gerenciamento de várias matrizes, decidimos implementar um dicionário onde as chaves são os nomes (identificadores) atribuídos às matrizes e os valores correspondem às listas bidimensionais. Com a intenção de facilitar a implementação das operações como adição, remoção, identificação e backup das matrizes.

O formato do projeto foi pensado de maneira modular, visando à manutenção da clareza e à facilidade para modificações . A lógica central foi agrupada em um módulo, o qual é encarregado de gerenciar as interações com o usuário. Em um módulo separado, colocamos todas as funções matemáticas, incluindo adição, multiplicação, transposição e cálculo de determinantes, entre outras. Também desenvolvemos um módulo dedicado a gerenciar as matrizes — incluindo inserção, remoção, listagem e backup — além de um módulo auxiliar com funções para verificação e impressão formatada.

Cada função foi criada com o objetivo de ser clara e eficiente, levando em consideração as limitações do uso exclusivo de Python. Por exemplo, a multiplicação de matrizes foi realizada assegurando que as dimensões das matrizes em questão fossem compatíveis O( para matrizes quadradas que é bom para matrizes pequenas mas ineficiente para grandes matrizes, enquanto a função de cálculo de determinantes utilizou o método de Laplace, que computacionalmente é trabalhoso com complexidade de O() , embora tranquila de usar , tem a parte do fatorial, tornando-se difícil para fazer com matrizes maiores. A implementação da função de transposição, por outro lado, foi feita de forma simples com complexidade de O().

Durante a montagem da calculadora, enfrentamos algumas dificuldades , como a realizar cópias ao salvar backups, para prevenir alterações nas matrizes originais. Tivemos também que garantir que as matrizes fossem compatíveis antes de efetuar operações, de modo a evitar erros inesperados ou resultados incorretos, além de uma complexidade de O(k\*n\*m), onde k é o número de matrizes guardadas .Recorremos ao auxílio de inteligência artificial para testar e corrigir alguns problemas no código.

Apesar das limitações a uma implementação desenvolvida do zero, sem bibliotecas otimizadas, os resultados obtidos foram bastante interessantes . A calculadora parece funcional, permitindo ao usuário executar operações básicas e complexas com matrizes de forma relativamente simples.

No geral , a criação da calculadora foi trabalhoso pelo tamanho de códigos para fazer mas com os recursos utilizados foi proveitoso. O código, no final, é eficiente, dividido em módulos, possibilitando uma fácil ampliação e reaproveitamento. Para tarefas que requerem mais desempenho, seria benéfico reescrever certas rotinas com bibliotecas otimizadas, porém, para as metas definidas, a solução está totalmente dentro da proposta.