

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**  
**INSTITUTO METRÓPOLE DIGITAL**  
**DISCIPLINA: INTRODUÇÃO ÀS TÉCNICAS DE PROGRAMAÇÃO**  
**ALUNO: DÊNIS ROCHA DA SILVA**  
**RELATÓRIO - CALCULADORA DE MATRIZES**

## 1. Introdução e contexto

O presente projeto tem como objetivo desenvolver uma calculadora de matrizes em linguagem C, permitindo ao usuário realizar operações básicas entre duas matrizes. No estágio atual, o foco esteve na implementação das operações de soma e subtração entre duas matrizes, com abertura para implementação de outras operações no futuro, como a multiplicação e etc. A escolha do presente tema é em virtude de esse ser um projeto que o autor já pensava em fazer (porém, em outra linguagem) e essa pareceu a oportunidade perfeita para finalmente concretizá-lo.

## 2. Análise Técnica

O projeto foi implementado usando o VSCode, através da extensão para C (ilustrada na figura 1), que permite que arquivos com formato .c sejam executados, compilando eles inicialmente e executando no terminal acoplado ao *software*.

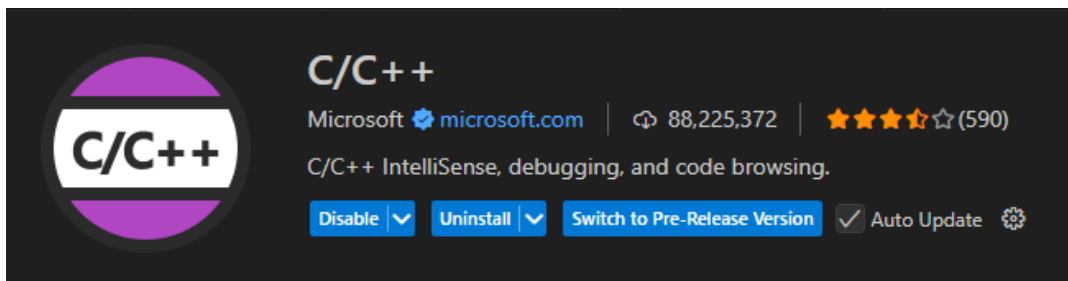


Figura 1 - Extensão que precisou ser instalada no VSCode para executar códigos em C.

Quanto ao funcionamento do programa, ele está ordenado em funções, e faz uso de declaração de variáveis, operações aritméticas e lógicas, estruturas condicionais e de repetição, além de vetores em sua forma bidimensional (matrizes), de forma a abranger o conteúdo que foi ministrado na disciplina. Quando o usuário executa o código, será apresentado ao seguinte fluxo:

- Escolha da operação:
  - A primeira entrada solicitada ao usuário é o tipo de operação que ele gostaria de realizar. Atualmente estão disponíveis as operações de soma e subtração;

- Para escolher a soma, o usuário deve inserir o dígito 1, e para escolher subtração, deve inserir o dígito 2. O programa valida a entrada com uma estrutura do...while, garantindo que apenas opções válidas sejam aceitas.
- Definição da dimensão das matrizes:
  - O usuário precisa em seguida informar o número de linhas e colunas que as matrizes terão, de forma a declarar as variáveis dinamicamente.
  - No momento, as duas matrizes usam o mesmo número de linhas e colunas, o que é conveniente com o fato de que as operações de soma e subtração só são possíveis entre matrizes de mesma ordem, porém esse ponto deverá ser alterado considerando as demais operações que serão adicionadas futuramente.
- Preenchimento das matrizes:
  - O usuário precisa fornecer, elemento por elemento, os valores das duas matrizes, que serão armazenados nos vetores bidimensionais respectivos.
  - Cada entrada é feita individualmente, com indicação da posição (linha e coluna) feitas através de laços do tipo for.
- Execução da operação escolhida:
  - De acordo com a escolha de operações que o usuário realizou, haverá uma separação por meio da estrutura condicional if...else para designar a soma ou a subtração das matrizes inseridas. Os elementos das matrizes são somados individualmente usando laços aninhados do tipo for para acessar suas posições.
- Exibição do resultado:
  - Por fim, a matriz resultante da operação é impressa de maneira a simular a formatação de uma matriz, com um espaço em branco separando os elementos da linha, e usando o recurso de quebra de linha para separar as colunas, ambos feitos com um if...else que incorporava esses elementos na impressão.

A figura 2 mostra uma captura do terminal após a execução de todas as etapas descritas acima, ilustrando todo o processo desde as entradas até o encerramento da execução.

```

Escolha a operação:
1 - Soma
2 - Subtração
1
Insira o número de linhas da matrizes: 2
Insira o número de colunas das matrizes: 2
Insira o elemento 11 da matriz 1: 2
Insira o elemento 12 da matriz 1: 2
Insira o elemento 21 da matriz 1: 1
Insira o elemento 22 da matriz 1: 1
Insira o elemento 11 da matriz 2: -5
Insira o elemento 12 da matriz 2: 4
Insira o elemento 21 da matriz 2: 2
Insira o elemento 22 da matriz 2: 14
-3 6
3 15

```

Figura 2 - Captura de tela que ilustra todas as etapas de interação via terminal.

O resultado final para a soma apresentado na figura 2 foi feita com a soma termo a termo dos elementos de ambas matrizes, conforme demonstrado abaixo:

$$\begin{vmatrix} 2 & 2 \\ 1 & 1 \end{vmatrix} + \begin{vmatrix} -5 & 4 \\ 2 & 14 \end{vmatrix} = \begin{vmatrix} -3 & 6 \\ 3 & 15 \end{vmatrix}$$

### 3. Estrutura do Código

O programa está dividido em várias funções que aumentam a organização e a legibilidade, o que foi pensado para facilitar a manutenção e futuras adaptações. São elas:

- `escolherOperacao()` → Recebe e valida a operação desejada, retornando o inteiro equivalente a operação escolhida;
- `inserirLinhasOuColunas(int tipo)` → Define as dimensões das matrizes, recebendo como entrada um int que designa o tipo da dimensão, com 0 equivalente a linha e 1 a colunas, e a distinção entre eles é feito dentro da função por meio de estrutura condicional;
- `criarMatriz(int linhas, int colunas, int matriz[linhas][colunas], int indice)` → Preenche a matriz com os valores fornecidos pelo usuário, através de estruturas de laço aninhadas, usando o número de linhas e colunas, a matriz com o tamanho das linhas e colunas e o índice que diferencia se é a primeira matriz ou a segunda matriz;
- `operacaoMatrizes(int linhas, int colunas)` → Realiza a soma ou subtração de cada um dos elementos das matrizes, que são acessados por meio de laços

aninhados. Além disso, recebe como entrada o número das linhas e das colunas da matriz;

- `imprimirMatriz(int linhas, int colunas, int matriz[linhas][colunas])` → Exibe a matriz formatada na tela, também faz uso de estruturas de repetição aninhadas, e é a última informação repassada ao usuário antes do encerramento da execução do código.

#### 4. Implementação e Reflexão

A principal dificuldade foi quanto a adaptação do pensamento para a sintaxe do C, visto que recursos acessíveis em outras linguagens não são encontrados, como, por exemplo, funções para lidar com listas ou *arrays*. Outro aspecto é quanto a dificuldade de trabalhar somente com os tipos que a linguagem oferece, diversas das funções que estão no código foram pensadas para ficar fora da função *main* e deveriam ser chamadas dentro dela, porém pela dificuldade de formatar adequadamente o retorno delas, optou-se por usar funções sem retorno (*void*) e dentro da *main*, para que pudessem só atualizar as variáveis diretamente. Ademais, no estado atual, as seguintes observações também são pertinentes acerca do código:

- Apenas operações de soma e subtração estão habilitadas para escolha;
- O programa ainda não contempla a multiplicação de matrizes, apesar de haverem alguns comentários no código que estão relacionados a essa operação e que serão melhor explorados na próxima entrega;
- Não há tratamento de erros para valores não numéricos durante a entrada de dados, porém é algo que também será implementado.

#### 5. Conclusão

O projeto encontra-se em uma versão inicial, mas já é capaz de realizar operações de soma e subtração de matrizes de dimensões arbitrárias, fornecidas pelo usuário. Esse estágio representa um passo importante para o desenvolvimento de uma calculadora de matrizes mais completa, que irá ser aperfeiçoada no decorrer da disciplina, considerando os conteúdos ministrados. Quanto às futuras operações que devem ser incluídas, a multiplicação entre duas matrizes, a inversão da matriz, e o determinante (até a terceira ordem talvez), são as mais cogitadas para serem implementadas.