

Universidade da Beira Interior

Departamento de Informática



Departamento de
Informática

Nº 20 - 2023/2024: *[Relatório Técnico]*

Elaborado por:

Benedita Rodrigues [52008]

Daniel Cardoso [52218]

Francisco Pereira [52129]

João Rodrigues [51779]

Rodrigo Almeida [51597]

Uberaldo Kill [51784]

Orientador:

Professor Doutor Tiago Roxo

Professor Doutor Tiago Simões

7 de janeiro de 2024

Agradecimentos

Após a elaboração do trabalho apenas falta agradecer e reconhecer a disponibilidade e preocupação dos nossos orientadores da Unidade Curricular de Laboratórios de Programação, Professor Doutor Tiago Roxo e Professor Doutor Tiago Simões visto que, sem eles, a realização dos objetivos do projeto não teriam sido alcançados. Não podemos esquecer de referir a nossa apreciação do humor nos Guias das Aulas Práticas, que nos ajudou a motivar, a aprender e testar as nossas habilidades.

Conteúdo

Conteúdo	3
1 Introdução	1
1.1 Enquadramento	1
1.2 Elaboradores do projeto	1
1.3 Motivação	1
1.4 Objetivos	2
1.5 Organização do Documento	2
2 Estado da Arte	5
2.1 Introdução	5
2.2 MATLAB	5
2.3 WINMAT	5
2.4 Conclusões	6
3 Tecnologias e Ferramentas Utilizadas	7
3.1 Introdução	7
3.2 Tecnologias mais importantes	7
3.2.1 Linguagem C	7
3.2.1.1 <i>Visual Studio Code</i>	7
3.2.1.2 <i>GNU Compiler Collection</i>	8
3.2.1.3 <i>Doxygen</i>	8
3.3 <i>Overleaf</i>	8
3.4 Conclusões	8
4 Implementação do código	9
4.1 Introdução	9
4.2 Primeira Parte	9
4.2.1 Função <code>main()</code>	9
4.2.2 Funções das Operações	12
4.3 Parte elaborada	15
4.3.1 Função <code>main()</code>	15
4.4 Funções das Operações	15

4.5	Conclusões	18
5	Conclusões e Trabalho Futuro	19
5.1	Conclusões Principais	19
5.2	Trabalho Futuro	19
5.2.1	<i>O que é que ficou por fazer, e porquê?</i>	19
5.2.2	<i>O que é que seria interessante fazer, mas não foi feito por não ser exatamente o objetivo deste trabalho?</i>	19
5.2.3	<i>Em que outros casos ou situações é que o trabalho aqui descrito pode ter aplicações interessantes e porquê?</i> . . .	20
	Bibliografia	21

Lista de Excertos de Código

4.1	Função <code>main()</code> – Validação de entrada.	9
4.2	Função <code>main()</code> – Exibe o menu de operações.	10
4.3	Função <code>main()</code> – Solicitação da escolha do utilizador.	10
4.4	Função <code>main()</code> – Processamento da escolha do utilizador.	10
4.5	Função 1 – Multiplicação de todos os elementos do vetor.	12
4.6	Função 2 – Identificação maior elemento do vetor.	12
4.7	Função 3 – Responsável pelo cálculo da soma dos valores do vetor que são divisíveis por três.	12
4.8	Função 4 – Responsável pela ordenação crescente dos simétricos do vetor.	12
4.9	Função 5 – Responsável pela construção de uma matriz 10 por 10 em que cada linha é composta pelo vetor lido (primeira linha) e por permutações dos seus valores (outras linhas).	13
4.10	Função 6 – Responsável pelo cálculo do logaritmo (base natural) de todos os elementos no vetor.	14
4.11	Função <code>main()</code> – Exibição de uma página da ajuda.	15
4.12	Função 7 – Utilização do comando <code>—help</code>	15
4.13	Função 8 – Função responsável pela leitura de um novo vetor e devolução de um vetor que mistura metade do primeiro vetor e metade do segundo.	16
4.14	Função 9 – Responsável pelo cálculo do máximo divisor comum de cada dois números seguidos do vetor.	16
4.15	Função 10 – Responsável pela geração de um novo vetor 1x10 aleatório; cálculo e devolução da matriz 10x10 resultante do produto do vetor inicial com o novo vetor lido.	17
4.16	Função 11 – Responsável pelo cálculo e apresentação da matriz transposta referida no ponto anterior.	18

Acrónimos

LP Laboratórios de Programação

MATLAB *Matrix Laboratory*

VSC *Visual Studio Code*

GCC *GNU Compiler Collection*

HTML *Hypertext Markup Language*

Capítulo

1

Introdução

1.1 Enquadramento

Este documento resume-se ao relatório do projeto requerido na unidade curricular de Laboratórios de Programação (LP), integrada na área de Engenharia Informática. A elaboração deste trabalho consiste em implementar um programa em C que pedisse um número determinado de inteiros (num determinado intervalo) para posteriormente calcular alguns valores e fazer operações sobre esses mesmos valores. Inserindo assim, o projeto, na área da Matemática e Programação.

1.2 Elaboradores do projeto

Os responsáveis pela elaboração do código foram Daniel Cardoso [52218] e Francisco Pereira [52129]. Ubaldo Kill [51784] e João Rodrigues [51779] realizaram a documentação em HTML (*Doxygen*). Benedita Rodrigues [52008] e Rodrigo Almeida [51597] elaboraram o relatório técnico.

1.3 Motivação

Abordar o problema é bastante importante visto que, no ponto de vista prático nas áreas como Engenharia de *Software* é essencial que a análise de dados e valores seja feita de forma eficiente e de uma maneira mais precisa. Este projeto pode de alguma maneira ajudar nesse aspeto.

1.4 Objetivos

Este projeto consiste na necessidade de implementar um programa em C interativo que permita ao utilizador realizar vários cálculos com um conjunto de números inteiros. O programa requer a escolha e introdução de 10 números inteiros, garantindo (por validação de entrada) que estão no intervalo pedido, neste caso, entre -10 a 16.

O primeiro e principal objetivo a alcançar seria implementar um menu interativo, onde aparecem seis opções, desde cálculos à análise das estatísticas dos valores introduzidos. No menu estão incluídos: cálculo da multiplicação de todos os elementos; identificação do valor máximo; soma dos valores que são divisíveis por três; ordenação crescente dos valores simétricos; construção de uma matriz 10 por 10, com permutações dos valores do vetor e cálculo do logaritmo natural de cada elemento.

Numa versão mais elaborada do projeto e como segundo objetivo, o programa oferece novas características e funcionalidades como a leitura de um novo vetor e várias operações com o mesmo como: mistura de metade do primeiro vetor com a metade do segundo; cálculo do máximo divisor comum de cada dois números seguidos do vetor; geração de um novo vetor (1x10) aleatório; cálculo da matriz (10x10) resultante do produto inicial com o novo vetor lido; geração de uma página de ajuda no menu e suporte à execução a partir da linha de comandos.

1.5 Organização do Documento

De modo a ter uma melhor percepção do relatório está, a seguir, incluída uma breve estrutura do mesmo:

1. O primeiro capítulo – **Introdução** – Apresenta e explica o que é este documento, faz o seu enquadramento quanto às áreas em que este está inserido, bem como uma Motivação e seus Objetivos.
2. O segundo capítulo – **Estado da Arte** – Apresenta plataformas e linguagens com funções semelhantes ao projeto, podendo concluir a sua importância.
3. O terceiro capítulo – **Tecnologias Utilizadas** – descreve e apresenta as plataformas e tecnologias mais importantes para o alcance dos objetivos deste projeto.
4. O quarto capítulo – **Implementação do Código** – Apresenta trechos de código do projeto, apresentando a sua função.

5. O quinto capítulo – **Conclusões e Trabalho futuro** – Apresenta melhorias e componentes que poderiam ser interessantes adicionar ao projeto e conclusão do projeto.

Capítulo

2

Estado da Arte

2.1 Introdução

No projeto abordado foram utilizados e realizados cálculos e manipulação de dados para alcançar todas as metas propostas. Como tal, já existiam algumas plataformas com a mesma função, possibilitando também que o processo de cálculo de estatísticas fique mais eficiente.

2.2 MATLAB

O *Matrix Laboratory* (MATLAB) é uma ferramenta que é utilizada em várias áreas como a engenharia. Permite realizar muito mais além de cálculos numéricos e operações entre matrizes, oferecendo várias ferramentas para a análise de dados. Fazendo assim com que o processo de cálculo se torne mais eficiente e preciso.

2.3 WINMAT

O WINMAT é um *Software* que permite fazer cálculos que envolvem matrizes como determinar a matriz inversa, transposta, entre outras. Tornando também o processo de cálculo e elaboração de matrizes mais rápido.

2.4 Conclusões

Ao concluir o capítulo verifica-se que a existência das várias ferramentas para cálculos estatísticos e numéricos são bastante importantes no que toca à praticidade e perspicácia.

Capítulo

3

Tecnologias e Ferramentas Utilizadas

3.1 Introdução

Para a concretização deste projeto estiveram presentes duas ferramentas. Na parte da implementação do código foi utilizada linguagem C e na elaboração do relatório técnico a plataforma *Overleaf*.

3.2 Tecnologias mais importantes

3.2.1 Linguagem C

Como foi referido anteriormente, a Linguagem C foi utilizada na elaboração de funções que permitiram o funcionamento do programa e a criação do documento em *Hypertext Markup Language* (HTML) (*Doxygen*). Para isto ser possível foram necessários um IDE *Visual Studio Code* (VSC) e o compilador *GNU Compiler Collection* (GCC) na linha de comandos do Sistema *Linux*.

3.2.1.1 Visual Studio Code

O VSC é um editor de código desenvolvido pela *Microsoft*, compatível com vários Sistemas Operativos. Este IDE possui várias extensões que precisam de ser instaladas de acordo com os nossos objetivos.

3.2.1.2 *GNU Compiler Collection*

O GCC é um compilador que numa fase inicial foi apenas desenvolvido para compilar programas em C, no entanto com a sua evolução ganhou a capacidade de compilar outras linguagens. Habitualmente, este compilador já vem instalado em sistemas operacionais *Linux*.

3.2.1.3 *Doxygen*

O Doxygen é basicamente uma ferramenta que utiliza as linguagens C e *Markdown*, que torna a geração de documentos através do código num processo mais eficiente. Este pode gerar documentação em vários formatos como HTML e \LaTeX .

3.3 *Overleaf*

O *Overleaf* é uma plataforma que permite fazer documentação em \LaTeX , tal como o presente relatório.

3.4 Conclusões

Neste capítulo verifica-se a importância de cada uma destas plataformas para a elaboração e conclusão do projeto.

Capítulo

4

Implementação do código

4.1 Introdução

No respectivo capítulo estão presentes os trechos das funções que possibilitaram o funcionamento do programa.

4.2 Primeira Parte

4.2.1 Função main()

```
do{
    printf("Pressione 1 para entrar no programa\n");
    scanf("%d", &validar_entrada);
}while(validar_entrada != 1);

for (i = 0; i < TAMANHO_VETOR; i++) {
    do {
        printf("Digite o %d numero inteiro entre -10 e 16: ", i + 1)
        ;
        scanf("%d", &numeros[i]);

        if (numeros[i] < -10 || numeros[i] > 16) {
            printf("Por favor, digite um numero entre -10 e 16.\n");
        }
    } while (numeros[i] < -10 || numeros[i] > 16);
} while (voltarMenu)
```

Excerto de Código 4.1: Função main() – Validação de entrada.

```
do {
    printf("Menu:\n");
    printf("1-Calculo da multiplicacao de todos os elementos no
        vetor:\n");
    printf("2-Identificacao do maximo de todos os elementos do vetor
        :\n");
    printf("3-Devolucao da soma dos valores do vetor que sao
        divisiveis por tres:\n");
    printf("4-Devolucao do vetor ordenado por ordem crescente dos
        seus valores simetricos:\n");
    printf("5-Construcao de uma matriz 10 por 10, em que cada linha
        e composta pelo vetor lido(primeira linha) e por permutacoes
        dos seus valores(outras linhas):\n");
    printf("6-Calculo do logaritmo (base natural) de todos os
        elementos no vetor:\n");
    printf("7-Pagina de Ajuda\n");
    printf("8-Leitura de um novo vetor, e devolucao de um vetor que
        mistura metade do primeiro vetor e metade do segundo;\n")
        ;
    printf("9-Calculo do maximo divisor comum de cada dois numeros
        seguidos do vetor;\n");
    printf("10-Geracao de um novo vetor 1x10 aleatorio,
        calculo e devolucao da matriz 10x10 resultante do produto
        do vetor inicial com o novo vetor lido;\n");
    printf("11-Calculo e apresentacao da matriz transposta referida
        no ponto anterior;\n");
    ...
}
```

Excerto de Código 4.2: Função main() – Exibe o menu de operações.

```
...
printf("Escolha uma opcao: ");
scanf("%d", &escolha);
...
```

Excerto de Código 4.3: Função main() – Solicitação da escolha do utilizador.

```
...
switch (escolha) {
    case 1:
        multiplicacaoVetor (numeros, TAMANHO_VETOR);
        break;
    case 2:
        valorMaximo (numeros, TAMANHO_VETOR);
        break;

    case 3:
        somaDivisiveisPorTres (numeros,TAMANHO_VETOR);
}
```

```
        break;
    case 4:
        ordenarcrescenteSimetricos(numeros, TAMANHO_VETOR);
        break;
    case 5:
        fazerpermutacaoMatriz(numeros, matriz);
        break;
    case 6:
        calcularLogNaBaseNatural(numeros, TAMANHO_VETOR);
        break;
    case 7:
        mostrarAjuda();
        break;
    case 8:
        misturarVetores(numeros, novo_vetor, mistura_vetor);
        break;
    case 9:
        calcularMDCVetor(numeros, TAMANHO_VETOR);
        break;
    case 10:
        criarVetorAleatorio(vetor2, TAMANHO_VETOR, -10, 16);
        calcularMatrizProduto(numeros, vetor2, produtomatriz);
        break;
    case 11:
        criarVetorAleatorio_semprint(vetor2, TAMANHO_VETOR, -10,
                                     16);
        calcularMatrizProduto_semprint(numeros, vetor2,
                                       produtomatriz);
        transporMatriz(produtomatriz, transposta);
    default:
        printf("Opcao invalida. Tente novamente.\n");
    }
    printf("\nDeseja voltar ao menu= (1 - Sim / 0 - Nao):\n");
    scanf("%d", &voltarMenu);

} while (voltarMenu != 0 && voltarMenu != 1);
}
```

Excerto de Código 4.4: Função main() – Processamento da escolha do utilizador.

4.2.2 Funções das Operações

```
int multiplicacaoVetor(int vetor[], int tamanho)
{
    int multiplicacao = 1;

    for (int i = 0; i < tamanho; i++) {
        multiplicacao *= vetor[i];
    }
    printf("\nMultiplicacao dos numeros: %d\n", multiplicacao);
}
```

Excerto de Código 4.5: Função 1 – Multiplicação de todos os elementos do vetor.

```
int valorMaximo(int vetor[], int tamanho)
{
    int maximo = vetor[0];

    for (int i = 1; i < tamanho; ++i) {
        if (vetor[i] > maximo) {
            maximo = vetor[i];
        }
    }
    printf("\nO valor maximo do vetor: %d\n", maximo);
}
```

Excerto de Código 4.6: Função 2 – Identificação maior elemento do vetor.

```
int somaDivisiveisPorTres(int vetor[], int tamanho)
{
    int soma = 0;

    for (int i = 0; i < tamanho; i++) {
        if (vetor[i] % 3 == 0) {
            soma += vetor[i];
        }
    }
    printf("Soma dos valores divisiveis por 3: %d\n", soma);
}
```

Excerto de Código 4.7: Função 3 – Responsável pelo cálculo da soma dos valores do vetor que são divisíveis por três.

```
void ordenarCrescenteSimetricos(int vetor[], int tamanho)
{
    int simetricos[tamanho]; // Vetor para armazenar os simetricos
    for (int i = 0; i < tamanho; i++) {
        simetricos[i] = -vetor[i];
    }
}
```

```
    }  
    // Ordena os valores simetricos em ordem crescente  
    for (int i = 0; i < tamanho - 1; i++) {  
        for (int j = 0; j < tamanho - i - 1; j++) {  
            if (simetricos[j] > simetricos[j + 1]) {  
                int temp = simetricos[j];  
                simetricos[j] = simetricos[j + 1];  
                simetricos[j + 1] = temp;  
            }  
        }  
    }  
    // Mostra o resultado  
    printf("Vetor ordenado dos valores simetricos:\n ");  
    for (int i = 0; i < tamanho; i++) {  
        printf("%d ", simetricos[i]);  
    }  
    printf("\n");  
}
```

Excerto de Código 4.8: Função 4 – Responsável pela ordenação crescente dos simétricos do vetor.

```
//Trocar a posicao de 2 valores  
void swap(int *a, int *b) {  
    int temp = *a;  
    *a = *b;  
    *b = temp;  
}  
  
// Funcao para gerar permutacoes  
void criarPermutation(int vetor[], int n) {  
    int i;  
    for (i = n - 1; i > 0; i--) {  
        int j = rand() % (i + 1);  
        swap(&vetor[i], &vetor[j]);  
    }  
}  
  
//Faz a permutacao na matriz  
void fazerpermutacaoMatriz(int vetor[], int matriz[][TAMANHO_VETOR]) {  
    int i, j, k;  
    for (i = 0; i < TAMANHO_VETOR; i++) {  
        // Se for a primeira linha, copia o vetor  
        if (i == 0) {  
            for (j = 0; j < TAMANHO_VETOR; j++) {  
                matriz[i][j] = vetor[j];  
            }  
        }  
    }  
}
```

```

// Senao, cria permutacoes nas outras linhas a partir da
// primeira
else {
    int permutacao[TAMANHO_VETOR];
    for (k = 0; k < TAMANHO_VETOR; k++) {
        permutacao[k] = matriz[i - 1][k];
    }
    criarPermutation(permutacao, TAMANHO_VETOR);
    for (j = 0; j < TAMANHO_VETOR; j++) {
        matriz[i][j] = permutacao[j];
    }
}
}
// print da matriz
for (i = 0; i < TAMANHO_VETOR; i++) {
    for (j = 0; j < TAMANHO_VETOR; j++) {
        printf("%d ", matriz[i][j]);
    }
    printf("\n");
}
}

```

Excerto de Código 4.9: Função 5 – Responsável pela construção de uma matriz 10 por 10 em que cada linha é composta pelo vetor lido (primeira linha) e por permutações dos seus valores (outras linhas).

```

void calcularLogNaBaseNatural(int vetor[], int tamanho)
{
    printf("\nCalculo dos logaritmos: \n");
    for (int i = 0; i < tamanho; ++i) {
        if (vetor[i] > 0) {
            float resultado = log(vetor[i]);
            printf("log(%d) = %.2f\n", vetor[i], resultado);
        } else {
            printf("Nao e possivel calcular o logaritmo %d \n", vetor[i]);
        }
    }
}

```

Excerto de Código 4.10: Função 6 – Responsável pelo cálculo do logaritmo (base natural) de todos os elementos no vetor.

4.3 Parte elaborada

4.3.1 Função main()

```
void mostrarAjuda() {
    printf("\n-----
           Pagina de Ajuda
           -----\\n\\n");

    printf("Este programa solicita 10 numeros inteiros no intervalo de
           -10 a 16, e organiza-os em um vetor 1x10.\\n
           Em seguida apresenta um menu que oferece varias opcoes\\n
           Sendo a primeira opcao calcular a multiplicacao dos elementos no
           vetor\\n
           A segunda opcao ver o maximo dos elementos do vetor\\n
           A terceira opcao a soma dos valores do vetor divisiveis por tres\\n
           A quarta opcao ordenadar o vetor por ordem crescente dos seus
           valores simetricos\\n
           A quinta opcao a construcão de uma matriz 10x10 onde da segunda
           linha para a frente sao permutacoes da primeira linha\\n
           A sexta opcao o calculo do logaritmo de todos os elementos no vetor\\
           n.gvg.\\n
           A oitava opcao ler um novo vetor 1x10 e misturar a metade do vetor
           inicial com o vetor criado\\n
           A nona opcao calcular o maximo divisor comum de cada dois numero
           seguidos\\n
           A decima opcao gerar um vetor novo 1x10 aleatorio e calcular o
           produto do vetor inicial com o vetor aleatorio\\n
           E por fim a decima primeira opcao o calculo da matriz transposta
           referida na decima opcao.\\n\\n");
    printf("----- Fim da
           pagina -----\\n")
    ;
}
```

Excerto de Código 4.11: Função main() – Exibição de uma página da ajuda.

4.4 Funções das Operações

```
int main(int argc, char *argv[]) {
    if (argc > 1) {
        if (strcmp(argv[1], "--help") == 0 || strcmp(argv[1], "-h") ==
            0) {
            mostrarAjuda();
            return 0;
        }
    }
}
```

Excerto de Código 4.12: Função 7 – Utilização do comando —help.

```

void misturarVetores(int vetor[], int vetor_novo[], int vetorMisturado[]) {

    for (int i = 0; i < TAMANHO_VETOR; i++) {
        do {
            printf("Digite o %d numero inteiro entre -10 e 16 do novo vetor: ", i + 1);
            scanf("%d", &vetor_novo[i]);

            if (vetor_novo[i] < -10 || vetor_novo[i] > 16) {
                printf("Por favor, digite um numero entre -10 e 16.\n");
            }
        } while (vetor_novo[i] < -10 || vetor_novo[i] > 16);
    }

    for (int i = 0; i < TAMANHO_VETOR / 2; i++) {
        vetorMisturado[i] = vetor[i];
    }

    for (int i = TAMANHO_VETOR / 2; i < TAMANHO_VETOR; i++) {
        vetorMisturado[i] = vetor_novo[(i - TAMANHO_VETOR / 2)];
    }
    printf("Vetor Misturado: ");
    for (int i = 0; i < TAMANHO_VETOR; i++) {
        printf("%d ", vetorMisturado[i]);
    }
    printf("\n");
}

```

Excerto de Código 4.13: Função 8 – Função responsável pela leitura de um novo vetor e devolução de um vetor que mistura metade do primeiro vetor e metade do segundo.

%Função para calcular MDC

```

int calcularMDC(int a, int b) {
    int temp;
    int maior;
    while (b != 0) {
        temp = b;
        b = a % b;
        a = temp;
        maior = fabs(a); // fabs= modulo
    }
}

```

```

    return maior;
}

%Funcao para calcular o MDC de cada par consecutivo no vetor
void calcularMDCVetor(int vetor[], int tamanho) {
    for (int i = 0; i < tamanho - 1; i++) {
        int mdc = calcularMDC(vetor[i], vetor[i + 1]);
        printf("MDC(%d, %d) = %d\n", vetor[i], vetor[i + 1], mdc);
    }
}

```

Excerto de Código 4.14: Função 9 – Responsável pelo cálculo do máximo divisor comum de cada dois números seguidos do vetor.

```

\texttt{void criarVetorAleatorio(int vetor[], int tamanho, int minimo,
    int maximo) {
    %Inicializa o gerador de numeros aleatorios com o tempo atual
    srand(time(NULL));

    %Preenche o vetor com numeros aleatorios entre minimo e maximo
    for (int i = 0; i < tamanho; i++) {
        vetor[i] = rand() % (maximo - minimo + 1) + minimo;
    }
    printf("Vetor Aleatorio entre %d e %d: ", minimo, maximo);
    for (int i = 0; i < 10; i++) {
        printf("%d ", vetor[i]);
    }
}

void calcularMatrizProduto(int vetor1[], int vetor2[], int
    matrizResultado[][TAMANHO_VETOR]) {
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            matrizResultado[i][j] = vetor1[i] * vetor2[j];
        }
    }
    printf("\nMatriz Resultado:\n");
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            printf("%3d ", matrizResultado[i][j]);
        }
        printf("\n");
    }
}

%Criacao de um vetor aleatorio sem que seja escrito
void criarVetorAleatorio_semprint(int vetor[], int tamanho, int minimo,
    int maximo) {
    %Inicializa o gerador de numeros aleatorios com o tempo atual

```

```
    srand ( time (NULL) ) ;

    %Preenche o vetor com numeros aleatorios entre minimo e maximo
    for (int i = 0; i < tamanho; i++) {
        vetor[i] = rand() % (maximo - minimo + 1) + minimo;
    }
}
```

Excerto de Código 4.15: Função 10 – Responsável pela geração de um novo vetor 1x10 aleatório; cálculo e devolução da matriz 10x10 resultante do produto do vetor inicial com o novo vetor lido.

```
\texttt{void transporMatriz(int matriz[][TAMANHO_VETOR], int transposta[
    [TAMANHO_VETOR]] {
    int linhas = TAMANHO_VETOR;
    int colunas = TAMANHO_VETOR;
    for (int i = 0; i < linhas; i++){
        for (int j = 0; j < colunas; j++) {
            transposta[j][i] = matriz[i][j];
        }
    }
    printf("\n Matriz Transposta:\n");
    for(int i = 0; i<10; i++){
        for(int j = 0; j<10; j++){
            printf("%3d", transposta[i][j]);
        }
    }
    printf("\n");
}
}
}
```

Excerto de Código 4.16: Função 11 – Responsável pelo cálculo e apresentação da matriz transposta referida no ponto anterior.

4.5 Conclusões

Este capítulo extenso dá a atender a importância que a implementação do código tem para todo o funcionamento do projeto.

Capítulo

5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

Com a realização deste trabalho, conclui-se que a conexão das informações dadas nas aulas pelos professores desta unidade curricular ao longo deste semestre, e toda a pesquisa feita fora das mesmas, foram preponderantes para a concretização do trabalho.

Concluindo, todo este trabalho foi bastante produtivo e uma aventura fascinante, onde ao longo do mesmo foi usado todo o tipo de comandos e código, desde o mais simples ao mais complexo. Os resultados obtidos foram ao encontro dos esperados, mostrando também a eficácia dos métodos utilizados.

5.2 Trabalho Futuro

5.2.1 O que é que ficou por fazer, e porquê?

Por todo o conhecimento retido e por toda a pesquisa dos elementos do grupo, nenhuma parte do trabalho ficou por fazer nem por acabar, o que mostra um grande empenho do grupo em questão.

5.2.2 O que é que seria interessante fazer, mas não foi feito por não ser exatamente o objetivo deste trabalho?

Java e *Python* são duas linguagens que, embora não sejam dadas com tanta importância como a linguagem C, são bastante interessantes e apresentam uma grande variedade de código. Por isso, a utilização destas no

trabalho traria benefícios pois, apesar de o trabalho ficar mais trabalhoso e complexo, ampliaria para a compreensão e aprendizagem destas duas linguagens de programação. Falando um pouco no que foi feito embora não fosse pedido, foi desenvolvido um modo noturno e diagramas na documentação gerada em HTML.

5.2.3 Em que outros casos ou situações é que o trabalho aqui descrito pode ter aplicações interessantes e porquê?

Este projeto terá uma influência significativa, pois tudo o que foi aplicado neste, para além de conhecimentos, o esforço e organização vão ter de estar presentes no futuro tanto a nível académico como profissional. A produção de um relatório em \LaTeX é sempre importante para acompanhar o projeto, pois torna a compreensão mais acessível. Concluindo, todo o trabalho envolvido, desde o projeto principal até ao relatório, serve para proporcionar uma ideia de como apresentar um trabalho coerente e acessível, sendo um dos primeiros passos para conseguir ter sucesso neste ramo.

Bibliografia

Matlab, "Matlab", 2012, [Online] https://itb.biologie.hu-berlin.de/~kempter/Teaching/2003_SS/gettingstarted.pdf . Último acesso a 03 de Janeiro de 2024

Da Silva, Cícero Félix, "ABORDAGEM TECNOLÓGICA: UTILIZANDO O SOFTWARE WINMAT PARA O ENSINO DE MATRIZES.", [Online] https://www.editorarealize.com.br/editora/anais/conedu/2015/TRABALHO_EV045_MD1_SA8_ID262_08092015193840.pdf . Último acesso a 04 de Janeiro de 2024