



UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI  
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS – PICOS  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO  
PROFESSOR: Juliana Oliveira de Carvalho

## Trabalho de Estruturas de dados 1

Autor:

David Marcos Santos Moura

Picos, Abril de 2021

- **Resumo do projeto:**

Inicialmente vamos falar um pouco sobre Estruturas de dados, **Estrutura de dados** é o ramo da computação que estuda os diversos mecanismos de organização de dados para atender aos diferentes requisitos de processamento. Posto isso , este relatório tem como objetivo demonstrar e analisar como é utilizados na prática conceitos de manipulação de dados para solução de alguns problemas que envolvem cálculo de valores de uma matriz , manipulação de vetores entre outros.

- **Introdução:**

Neste relatório vamos analisar como foi resolvido alguns problemas, como cálculo do determinante de uma matriz, manipulação de palavras armazenadas numa matriz de strings, além da criação de um pequeno sistema para cadastrar, buscar e imprimir cursos e suas disciplinas. Para solução destes problemas foi utilizada a linguagem de programação C e foram empregados alguns conceitos importantes para resolução de tais problemas, dentre eles estão: ponteiros , busca binária , ordenação entre outros.

- **Seções específicas:**

Neste tópico iremos abordar de maneira resumida como foi solucionado os problemas propostos:

**Problema 1:**

**Resumo do problema:** Faça um programa em C que, dado uma matriz de inteiros quadrada no máximo de ordem 4, calcule o determinante da mesma.

**Solução:**

Bom vamos dividir a resolução do problema em alguns passos:

1 passo: primeiro pedimos para que o usuário digite a ordem da matriz para que possa ser calculado seu respectivo determinante, tendo a ordem da matriz já estabelecida vamos passar para o próximo passo.

2 passo: Iremos calcular o determinante da matriz de acordo com sua ordem, para isso utilizamos a estrutura condicional if para calcular o determinante de acordo com a ordem da matriz que pode ser de ordem 1 a 4.

Então se a ordem for 1 , a matriz só poderá ter apenas um elemento e seu determinante será o próprio elemento lido.

se a ordem for 2 ,o algoritmo irá ler os valores e fazer o seu cálculo multiplicando os valores da sua diagonal principal e secundária e depois irá subtraí-los para obter o determinante da matriz de ordem dois e devolver para o usuário.

se a ordem for três, o algoritmo irá ler os valores digitados pelo usuário e inseri los na matriz, e para fazer o cálculo do determinante o código primeiramente irá tratar a matriz pegando os valores da suas colunas 0 e 1 e inserindo nas colunas 3 e 4, assim a matriz de tamanho 3x3 vira uma matriz 3x5, após isso o código terá que calcular a multiplicação dos valores da diagonal principal e secundária, para isso foi desenvolvido duas funções a primeira função é diagonal\_principal que ira receber como parâmetro a matriz, a ordem da matriz e alguns parâmetros auxiliares esta função ira percorrer a matriz e devolver num variável dp(que será passada por referência) o valor da multiplicação dos valores da diagonal principal e outra função diagonal\_secundaria que irá devolver na variável ds(que também é passada por referência) o valor da multiplicação dos valores da diagonal secundária por fim calcula-se o determinante subtraindo dp de ds e é devolvido o resultado para o usuário.

se a ordem for 4 ,o algoritmo irá ler os valores digitados pelo usuário e insere na matriz, e para fazer o cálculo do determinante o código primeiramente irá tratar a matriz iremos eliminar uma linha e uma coluna que neste caso será a linha 0 e coluna 0 então a matriz ficará como uma matriz de ordem 3x3, então o algoritmo irá calcular o determinante desta nova matriz de ordem 3 que já foi explicado como funciona o cálculo no passo acima, então será calculado o cofator dos elementos da coluna que foram eliminados, o cofator é calculado da seguinte maneira se a soma dos indice (i+j) do valor que esta sendo calculado o cofator for par o cofator sera o valor \* o determinante, se for impar o cofator será o valor \* determinante -1), por fim sera calculados a soma dos cofatores dos elementos eliminados na coluna 0 e esta soma sera o valor do resultado do determinante da matriz de ordem 4.

## **Problema 2:**

**Resumo do problema:** Faça um programa em C que tenha um menu de opções com as seguintes opções: para ler uma matriz de strings; uma opção para ordenar cada uma das colunas da matriz,outra para mostrar cada coluna da matriz antes de ordenar e outra depois de ordenar; uma opção que dado uma linha e uma coluna, conte o número de dígitos e de letras maiúsculas; outra que dado uma coluna conte o número de strings dessa coluna que iniciam com consoante.

## Solução:

Para solução deste problema dividimos o código em algumas funções, iremos analisar algumas das funções mais importantes para resolução do problema:

**ler\_mat()** : se o usuário escolher a opção um, o código pedirá que o usuário digite palavras ,então esta função irá armazenar as palavras dentro de uma matriz de strings.

**cópia() e quicksort()** : se o usuário escolher a opção 2, então o código irá ordenar a matriz de strings por suas colunas e essa ordenação será feita da seguinte forma:

primeiro a função cópia() irá as palavras da coluna da matriz e armazenar num vetor de estrutura e depois este vetor é passado para a função quicksort() que é um algoritmo de ordenação recursivo esta função irá com ajuda de outras funções como minúsculo( recebe duas string s1 e s2 e faz com que s2 receba s1 com todos seus caracteres minúsculos ) e compara(recebe duas string s1 e s2 e se o caracter da string 1 vinher antes na ordem na tabela asc retorna -1 e retorna 1 se vinher depois) irá ordena as palavras armazenadas no vetor e depois irá inserir numa matriz resultante os valores ordenados na coluna e irá fazer estes procedimento em todas as colunas da matriz e a matriz resultante irá receber as palavras da matriz de entrada ordenadas.

**imprimir\_mat()**: se o usuário escolher a opção 3, o código irá chamar a função imprimir, então a função irá receber a matriz de entrada e a matriz resultante que é a matriz de entrada ordenada e irá imprimir coluna por coluna as palavras antes e depois da ordenação.

**dígitos()**: se o usuário escolher a opção 3, então o código irá ler dois índices (coluna e linha) , que irá ser a posição de uma palavra da matriz , então a função dígitos irá receber esta palavra e irá devolver em duas variáveis (que serão passadas por referência ) e irá devolver nestas respectivas variáveis a quantidade de dígitos da palavra e a quantidade de letras maiúsculas e mostrá-las para o usuário.

**quant\_consoantes()**: se o usuário escolher a opção 2, então o código irá chamar esta função , então o código irá ler o valor de uma coluna, depois irá percorrer as strings desta coluna e a cada palavra o código irá passar o caractere da primeira letra da palavra para uma função consoantes(), esta função devolve 1 se o caracter for consoante e 0 se não for , então a função quant\_consoantes irá percorrer todas as palavras da coluna e chamar a função consoante e através de uma variável irá guardar a soma das palavras que iniciarem com consoantes.

### **Problema 3:**

**Resumo do problema:** Faça um programa em C que tenha um menu de opções para cadastrar, buscar e imprimir cursos e suas disciplinas.

### **Solução:**

Análogo a questão anterior, para solução deste problema dividimos o código em algumas funções, iremos analisar algumas das funções mais importantes para resolução do problema:

**cadastrar\_curso():** se o usuário escolher a opção 1, então será chamada a função cadastrar curso ira pedir para que o usuário digite um código, então a função chama a função busca\_cod() esta função através de um algoritmo de busca binária irá retorna 0 se o curso já foi inserido e 1 senao foi , então se o curso ainda não tiver sido inserido, a função irá ler os dados do curso e cadastrá-lo e é chamada a função quicksort1() que ira ordenar o vetor dos cursos..

**cadastrar\_disciplina():** se o usuário escolher a opção 2, então será chamada a função cadastrar disciplina ira pedir para que o usuário digite o código do curso no qual vai ser inserido a disciplina , então a função chama a função busca\_cod() esta função através de um algoritmo de busca binária irá retorna 0 se o curso já foi inserido e 1 senao foi , então se o curso já tiver sido inserido, a função irá ler o código da disciplina, então a função chama a função busca\_dic() esta função irá retornar 0 se o curso já foi inserido e 1 se não foi , se a disciplina ainda não foi inserida então será lido os dados da disciplina i será feito mais uma verificação, se o período cadastrado na disciplina corresponde a o período cadrastado no curso se sim então por fim a disciplina será cadastrada e depois é chamada a função quicksort2() que irá ordenar o vetor das disciplinas.

**imprimir\_curso():** se o usuário escolher a opção 3, então será chamada a função imprimir\_curso ler um o código do curso, então a função chama a função busca\_cod() esta função através de um algoritmo de busca binária busca o curso pelo seu código e imprime os dados do curso se ele existir.

**imprimir\_disciplina():** se o usuário escolhe a opção 4, então será chamada a função imprimir\_disciplina ler um o código da disciplina, então a função chama a função busca\_dic() esta função através de um algoritmo de busca binária busca a disciplina pelo seu código e imprime os dados da disciplina se ele existir.

**imprimir\_turno():** se o usuário escolher a opção 5, então a função ler um turno e a partir deste turno busca no vetor dos cursos do mesmo turno e imprime os dados do curso se encontrar.

**imprimir\_disciplina\_curso():** se o usuário escolher a opção 6, então a função ler o código da disciplina e chama a função busca\_cod e verifica se o curso existe, se existe a função percorre o vetor de disciplina e busca as disciplinas que estão cadastradas no curso lido, fazendo a busca através do código do curso, e quando encontra disciplinas cadastradas imprime seu dados.

**imprimir\_disciplina\_periodo():** se o usuário escolher a opção 7, então a função ler o código da disciplina e chama a função busca\_cod e verifica se o curso existe, se existir ler o período e verificar se o período é válido se for, a função ira percorrer as disciplinas e ira imprimi-las se as disciplinas estiverem cadastrada no curso lido e forem do período lido.

**remover\_dic():** se o usuário escolher a opção 8, então a função ler o código da disciplina e chama a função busca\_dic() e verifica se a disciplina existe, se existir, esta função como as demais que envolvem disciplinas também recebem um parâmetro a variável quant\_dic esta variável indica quantas disciplinas estão cadastradas no vetor, então a função remover\_dic diminuir o valor nesta variável quant\_dic (quant\_dic - 1) que indica que uma disciplina vai ser excluída, então se a disciplina não for a última do vetor copia os dados da última disciplina e armazena em aux (aux é uma estrutura auxiliar do tipo disciplina), joga os dados da disciplina que vai ser removida para a última posição do vetor e depois insere os dados guardados em aux na posição da disciplina removida, desta forma a disciplina é removida e depois da remoção é chamada a função quicksort2() para ordenar o vetor das disciplinas.

obs : a função que faz a copia e troca dos dados das estruturas, utilizado na função remover\_dic é a função **copiar2()** esta função recebe duas estruturas do tipo disciplina (x e y) e copia os dados de x e armazena em y, esta função também é utilizada na função quicksort()

**remover\_curso():** se o usuário escolher a opção 9, então a função ler o código do curso e chama a função busca\_curso() e verifica se o curso existe, se existir, esta função como as demais que envolvem curso também recebem um parâmetro a variável quant\_cursos esta variável indica quantos cursos estão cadastradas no vetor, então a função remover\_curso verifica se o curso possui disciplinas cadastradas, na estrutura cursos tem uma variável chamada dic do tipo inteiro então quando é cadastrada uma disciplina é somado 1 a esta variável (cursos[indice].dic + 1), então através dessa variável é possível verificar se o curso possui disciplina cadastradas, desta forma se cursos[indice].dic = 0 quer dizer que não há disciplinas cadastradas neste curso então é possível remover o curso, para isso o código irá diminuir o valor na variável quant\_cursos (quant\_cursos - 1) que indica que um curso vai ser excluído, então se o curso não for o último do vetor copia os dados da última disciplina e armazena em aux (aux é uma estrutura auxiliar do tipo curso), joga os dados do curso que vai ser removido para a última posição do vetor e depois insere os dados guardados em aux na posição do curso que vai ser removido, desta forma o curso é removido e depois da remoção é chamada a função quicksort() para ordenar o vetor dos cursos.

obs : a função que faz a copia e troca dos dados das estruturas, utilizado na função `remover_dic` é a função **`copia2()`** esta função recebe duas estruturas do tipo disciplina (x e v ) e copia os dados de x e armazena em y, esta função também é utilizada na função `quicksort()`

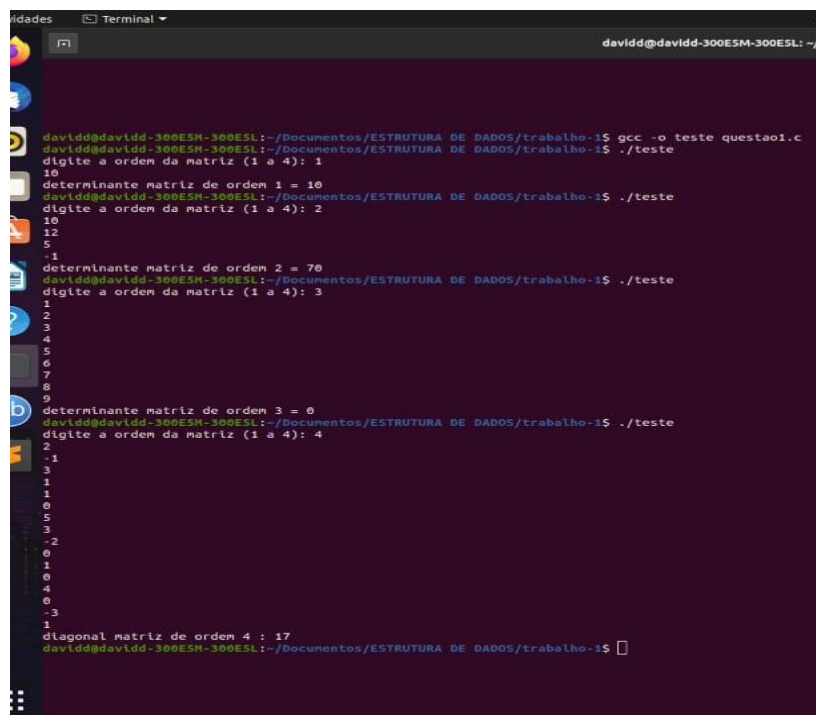
**funções auxiliares:** `quicksort1()` e `quicksort2()` estas funções ordenam os vetores de disciplinas e cursos, `busca_dic()` esta função possui um diferencial ela possui uma variável flag que é passada por parâmetro se essa variável for passada com um valor não negativo (`flag=1`) a variável flag recebe o valor do índice da disciplina se a disciplina for encontrada e se flag for negativo (`flag= -1`) a função imprime os dados da disciplina se ela existir, já na função `busca_cod()` busca código de um curso, também possui a variável flag so que quando `flag = 0` ela funciona de forma comum a função retorna 0 se o código não existir e 1 se encontrar o código , quando `flag = 1` a função imprime os dados do curso se o curso existir e se `flag < 0` e se o código existir flag devolve o índice do curso.

- **Resultados da Execução do Programa:**

Neste tópico iremos analisar dado algumas entradas como os codigos dao suas saídas:

Todas as questões a seguir foram executadas pelo terminal linux.

**Questão 1:**

A terminal window with a dark purple background. The prompt is 'daviddd@daviddd-300ESH-300ESL: ~/D'. The user runs 'gcc -o teste questao1.c' and then './teste'. The program prompts 'digite a ordem da matriz (1 a 4):'. The user enters '1', and the program outputs 'determinante matriz de ordem 1 = 10'. The user enters '2', and the program outputs 'determinante matriz de ordem 2 = 70'. The user enters '3', and the program outputs 'determinante matriz de ordem 3 = 0'. The user enters '4', and the program outputs 'diagonal matriz de ordem 4 : 17'.

```
daviddd@daviddd-300ESH-300ESL: ~/D
daviddd@daviddd-300ESH-300ESL:~/Documentos/ESTRUTURA DE DADOS/trabalho-1$ gcc -o teste questao1.c
daviddd@daviddd-300ESH-300ESL:~/Documentos/ESTRUTURA DE DADOS/trabalho-1$ ./teste
digite a ordem da matriz (1 a 4): 1
10
determinante matriz de ordem 1 = 10
daviddd@daviddd-300ESH-300ESL:~/Documentos/ESTRUTURA DE DADOS/trabalho-1$ ./teste
digite a ordem da matriz (1 a 4): 2
10
12
5
-1
determinante matriz de ordem 2 = 70
daviddd@daviddd-300ESH-300ESL:~/Documentos/ESTRUTURA DE DADOS/trabalho-1$ ./teste
digite a ordem da matriz (1 a 4): 3
1
2
3
4
5
6
7
8
9
determinante matriz de ordem 3 = 0
daviddd@daviddd-300ESH-300ESL:~/Documentos/ESTRUTURA DE DADOS/trabalho-1$ ./teste
digite a ordem da matriz (1 a 4): 4
2
-1
3
1
1
0
5
3
-2
0
1
0
4
0
-3
1
diagonal matriz de ordem 4 : 17
daviddd@daviddd-300ESH-300ESL:~/Documentos/ESTRUTURA DE DADOS/trabalho-1$
```

Na imagem acima é possível ver o teste da questão 1 , e mostra como o código da sua saída de acordo com os valores recebidos de entrada.

## Questao 2:



```
Atividades Terminal
6 - sair
digite a opcao: 1
vaca
boi
camelo
bode
1
2
3
4
camelo
1 - ler matriz
2 - ordenar matriz
3 - mostrar matriz antes e depois de ordenar
4 - verificar quantos digitos e letras maiusculas
5 - verificar numero de strings que iniciam com consoantes
6 - sair
digite a opcao: 2
Matriz Ordenada.
1 - ler matriz
2 - ordenar matriz
3 - mostrar matriz antes e depois de ordenar
4 - verificar quantos digitos e letras maiusculas
5 - verificar numero de strings que iniciam com consoantes
6 - sair
digite a opcao: 3
Antes e depois Ordenar colunas:
valores coluna [ 0 ]
vaca      3
bode      bode
3         vaca
valores coluna [ 1 ]
boi       1
1         4
4         boi
valores coluna [ 2 ]
camelo    2
2         camelo
camelo    camelo
```

Na imagem acima podemos ver como o código da questão 2 da sua saída quando o usuário escolhe as opções 1 ,2 e 3.



### Questão 3:

The image displays two side-by-side screenshots of a terminal window with a dark background and light-colored text. The terminal shows the execution of a C program that manages a database of courses and disciplines.

**Left Screenshot:**

- The program starts with a menu: "Digite 0 para ver o Menu." followed by "Digite a opcao: 0".
- A dashed line separates the menu from the list of options:
  - 1 - cadastrar curso
  - 2 - cadastrar disciplina
  - 3 - Imprimir curso
  - 4 - Imprimir disciplinas
  - 5 - Imprimir cursos de um turno
  - 6 - Imprimir disciplinas de um curso
  - 7 - Imprimir disciplinas de um periodo do curso
  - 8 - remover disciplina
  - 9 - remover curso
  - 10 - Finalizar
- Another dashed line follows.
- The user selects option 3: "Digite 0 para ver o Menu." followed by "Digite a opcao: 3" and "Codigo do curso: 1".
- The program displays the details for course 1:
  - Codigo do curso: 1
  - Nome do curso: sistemas
  - Quantidade de periodos: 8
  - Carga horaria do curso: 1000
  - Turno do curso: manhan
- Another dashed line follows.
- The user selects option 4: "Digite 0 para ver o Menu." followed by "Digite a opcao: 4" and "Codigo da disciplina: 1".
- The program displays the details for discipline 1:
  - Codigo da disciplina: 1
  - Codigo do curso onde a disciplina foi cadastrada: 1
  - Nome da disciplina: Prolog
  - Periodos da disciplina: 1
  - Carga horaria da disciplina: 40
- Another dashed line follows.
- The user selects option 0: "Digite 0 para ver o Menu." followed by "Digite a opcao: 0".

**Right Screenshot:**

- The program prompts the user to add a course: "Preencha os dados para cadastrar o curso:".
- A dashed line follows.
- The user enters "1".
- The program displays the details for the new course:
  - Nome do curso: sistemas
  - Quantidade de periodos: 8
  - Carga horaria do curso: 1000
  - Turno do curso: manhan
- Another dashed line follows.
- The program displays a success message: "Curso cadastrada com sucesso!".
- Another dashed line follows.
- The user selects option 2: "Digite 0 para ver o Menu." followed by "Digite a opcao: 2".
- Another dashed line follows.
- The program prompts the user to add a discipline: "Preencha os dados para cadastrar a disciplina: digite o codigo do curso: 1".
- A dashed line follows.
- The program displays a message: "Disciplina sera cadastrada no curso : sistemas".
- Another dashed line follows.
- The user enters the discipline code: "Digite o codigo da disciplina: 1".
- The program displays the details for the new discipline:
  - Nome da disciplina: Prolog
  - Periodo da disciplina: 1
  - carga horaria da disciplina: 40
- Another dashed line follows.
- The program displays a success message: "Disciplina cadastrada com sucesso!".
- Another dashed line follows.
- The user selects option 0: "Digite 0 para ver o Menu." followed by "Digite a opcao: 0".

Na imagem acima podemos ver como o código da questão 3 da sua saída quando o usuário cadastrar curso e disciplinas e depois pede para imprimir o curso e disciplina.

- **Conclusão:**

Por fim, podemos concluir que este trabalho teve como objetivo proporcionar para os estudantes um aprofundamento em conceitos importantes para a disciplina como manipulação de dados ,processamento de dados, manipulação de ponteiros e estruturas como matrizes e vetores, então podemos ver que através desta atividade prática foi possível aumentar o nosso conhecimento sobre a linguagem de programação C e sobre conceitos importantes para a disciplina de estruturas de dados.