



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, Junho 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.

- **Introdução:**

Neste relatório vamos analisar como foi resolvido alguns problemas, como, criação de um algoritmo para cadastrar e realizar algumas operações matemáticas com polinômios e criação de um pequeno sistema para sorteio de prêmios para clientes de uma empresa. Posto isso, para solução destes problemas foi utilizada a linguagem de programação C e foram empregados alguns conceitos importantes para resolução de tais problemas, dentre eles manipulação de estruturas de dados como listas.

- **Seções específicas:**

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

**Problema 1:**

**Resumo do problema:**

Um polinômio em três variáveis (x, y, z) pode ser representado por uma lista circular, onde cada nó deve representar um termo e deve conter: a potência do termo, o coeficiente deste termo. Faça um programa em C que possa cadastrar, somar, multiplicar polinômios.

Para solução deste problema foi desenvolvidos as seguintes funções:

**busca\_polinômio():** Esta função recebe um vetor de polinômios, onde cada elemento do vetor possui um código e uma lista simples circular contendo os termos do polinômio, então esta função vai receber um código e irá percorrer o vetor e buscar o polinômio pelo seu código e se o elemento for encontrado é devolvido a sua posição no vetor e se não for encontrado é devolvido um valor negativo.

**Cadastrar\_polinomio():** Esta função recebe uma lista e ler os dados dos termos do polinômio como coeficiente de potência do termo e armazena na lista recebida

**imprimir\_representacao():** Esta função recebe uma lista contendo um polinômio e imprime a representação do polinômio.

**soma\_polinomios():** Esta função recebe uma lista contendo um polinômio e outra lista com outro polinômio, então é calculada a soma entre os dois polinômios e é imprimido o resultado da operação para o usuário, este cálculo é feito da seguinte forma é percorrido os termos dos dois polinômios, então é verificado se os termos dos polinômios possuem variáveis e potências iguais, se sim então, mantém-se a variável e potência do polinômio e é somados os seus coeficientes, se não é repetido os termos dos polinômios.

**mult\_polinomios():** Esta função recebe uma lista contendo um polinômio e outra lista com outro polinômio, então é calculada a multiplicação entre os dois polinômios e é imprimido o resultado da operação para o usuário, este cálculo é feito da seguinte forma é percorrido os polinômios e pegamos cada termo do polinômio 1 e multiplicamos por todos os termos do polinômio 2, na multiplicação é verificado se os termos dos polinômios possuem variáveis iguais, se sim é multiplicado os seus coeficientes e somado suas potências, e se possuírem variáveis diferente é apenas multiplicado seus coeficientes e é mantido as variáveis e potências dos termos.

**derivada\_parcial():** Esta função recebe uma lista contendo um polinômio e uma variável, então é calculada a derivada parcial da expressão em relação a variável recebida e é imprimido o resultado da operação para o usuário, este cálculo é feito da seguinte forma é percorrido os termos do polinômio e é verificado a variável do termo é igual à recebida pela função se sim é calculado a derivada do termo e é mostrado para o usuário.

## **Problema 2:**

### **Resumo do problema:**

Uma empresa deseja premiar seus clientes, cada cliente tem o direito de girar uma vez uma roleta que contém vários prêmios. Faça um programa em C que crie uma roleta através de uma lista circular duplamente encadeada, ordenada pelo código do prêmio, onde o usuário irá cadastrar os prêmios. Cada prêmio deve conter um código, o prêmio(texto descrevendo o prêmio. Ex.: 10% de desconto, uma camiseta) e deve conter também a quantidade daquele prêmio. Depois que todos os prêmios forem cadastrados permite ao usuário girar a roleta, isto significa gerar um número aleatório de 1 a 100. Depois de sorteado o número, ande na lista o número de vezes do valor sorteado.

Para solução deste problema foi desenvolvidos as seguintes funções:

**insere():** Recebe um início e fim de uma lista e um NÓ contendo os dados de um prêmio, então é inserido de forma ordenada pelo o código do prêmio o prêmio na lista.

**remove():** Recebe um início e fim de uma lista e um código de um prêmio, e remove da lista o prêmio com o código igual ao recebido pela função.

**busca():** Recebe um início de uma lista e um código de um prêmio e percorre a lista buscando o prêmio com o código igual ao recebido pela função e se for encontrado é devolvido um ponteiro que contém o endereço do prêmio buscado, se não for encontrado é devolvido NULL.

**insere\_valores():** Recebe um início e fim de uma lista e lê os dados dos prêmios e chama a função **insere()** para cadastrar os prêmios, este processo é repetido até que o usuário não queira inserir prêmios na lista.

**gira\_roleta():** Recebe uma posição da lista no primeiro caso posição é a do início da lista e uma variável flag, então sorteia um valor e percorre na lista a quantidade de vezes do valor sorteado se o valor da variável flag for 0 a lista é percorrida no sentido horário e se o valor for 1 é percorrido no sentido anti-horário, por fim, imprime o prêmio que foi sorteado, decrementa o valor da quantidade do prêmio e devolve na variável posição a posição onde parou na lista, após a execução desta função é verificado se a quantidade do prêmio sorteado é igual a zero se for é chamado a função **remove()** passando o código do prêmio sorteado e a lista , então o prêmio é removido da lista.

- **Resultados da Execução do Programa:**

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

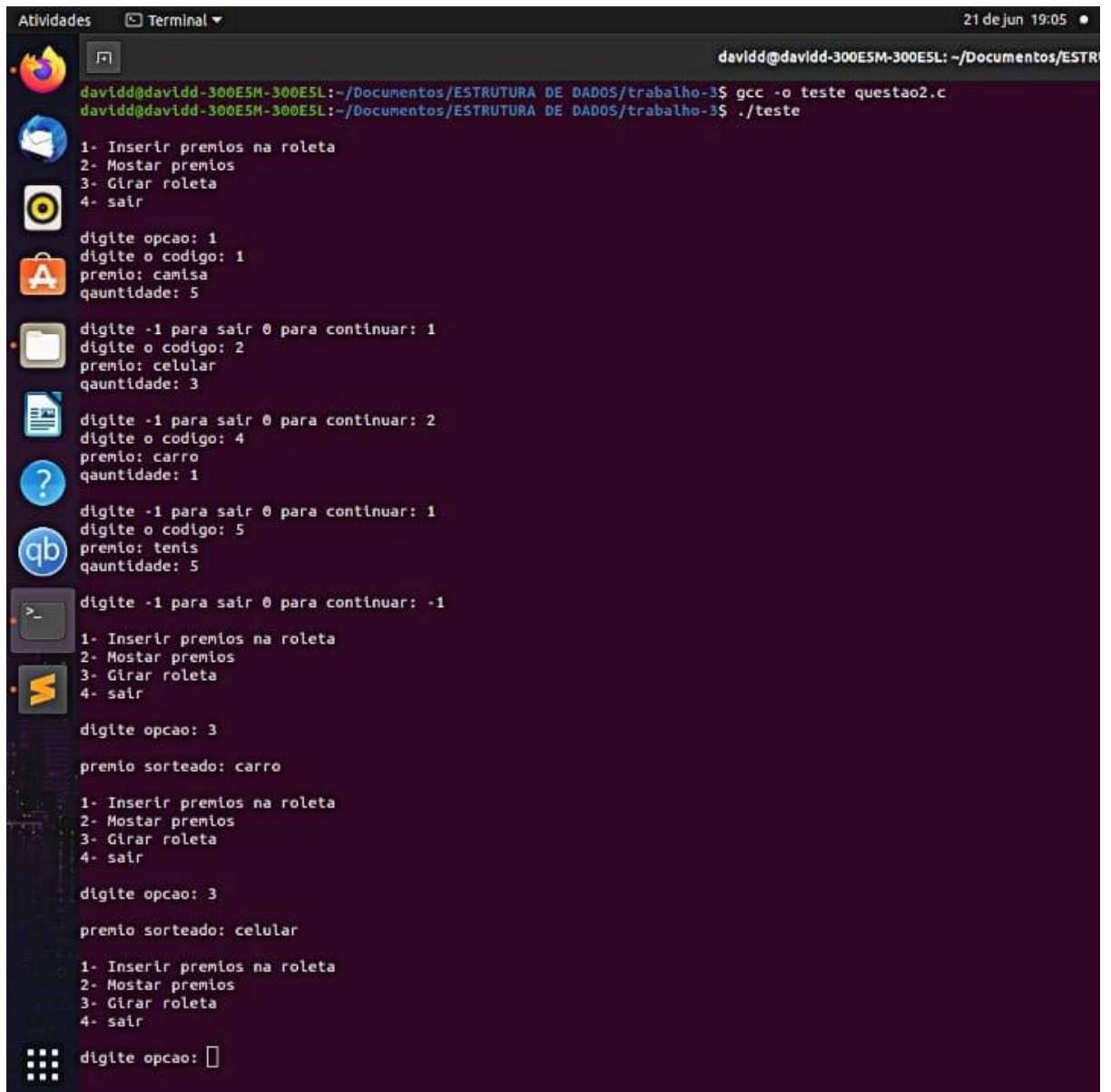
Todas as questões a seguir foram executadas pelo terminal linux.

### **Questão 1:**

```
Atividades Terminal
digite opcao: 1
digite o codigo do polinomio: 2
digite o coeficiente de x :2
digite a potencia de x :3
digite o coeficiente de y :-4
digite a potencia de y :3
digite o coeficiente de z :2
digite a potencia de z :1
1- cadastrar polinomio
2- Imprimir a representação de um polinômio
3- Imprimir a representação de todos polinômio
4- avaliar polinômio
5- somar polinomios
6- multiplicar polinomios
7- calcular derivada parcial de um polinomio
8- sair
digite opcao: 2
digite o codigo do polinomio: 2
polinomio: 2x^3 -4y^3 + 2z^1
1- cadastrar polinomio
2- Imprimir a representação de um polinômio
3- Imprimir a representação de todos polinômio
4- avaliar polinômio
5- somar polinomios
6- multiplicar polinomios
7- calcular derivada parcial de um polinomio
8- sair
digite opcao: 7
digite o codigo do polinomio: 2
digite a variavel (x, y ou z): y
derivada parcial em relacao a y : -4*3*y^2
1- cadastrar polinomio
2- Imprimir a representação de um polinômio
3- Imprimir a representação de todos polinômio
4- avaliar polinômio
5- somar polinomios
6- multiplicar polinomios
7- calcular derivada parcial de um polinomio
8- sair
digite opcao: 
```

Na imagem acima podemos ver como o código se comporta , quando usuário cadastra um polinômio, podemos ver que é pedido para que o usuário digite os valores dos coeficiente e potência de cada termo do polinômio e então o polinômio é cadastrado, podemos analisar também como o código da sua saída quando o usuário solicita imprimir a representação de um polinômio e quando é solicitado para calcular a derivada parcial.

## Questão 2:



```
davidd@davidd-300E5M-300E5L: ~/Documentos/ESTRUTURA DE DADOS/trabalho-3$ gcc -o teste questao2.c
davidd@davidd-300E5M-300E5L: ~/Documentos/ESTRUTURA DE DADOS/trabalho-3$ ./teste

1- Inserir premios na roleta
2- Mostar premios
3- Girar roleta
4- sair

digite opcao: 1
digite o codigo: 1
premio: camisa
qauntidade: 5

digite -1 para sair 0 para continuar: 1
digite o codigo: 2
premio: celular
qauntidade: 3

digite -1 para sair 0 para continuar: 2
digite o codigo: 4
premio: carro
qauntidade: 1

digite -1 para sair 0 para continuar: 1
digite o codigo: 5
premio: tenis
qauntidade: 5

digite -1 para sair 0 para continuar: -1

1- Inserir premios na roleta
2- Mostar premios
3- Girar roleta
4- sair

digite opcao: 3

premio sorteado: carro

1- Inserir premios na roleta
2- Mostar premios
3- Girar roleta
4- sair

digite opcao: 3

premio sorteado: celular

1- Inserir premios na roleta
2- Mostar premios
3- Girar roleta
4- sair

digite opcao: 
```

Na imagem acima podemos ver como o código se comporta , quando usuário insere prêmios na roleta, podemos ver que é pedido para que o usuário digite os dados dos prêmios então o prêmio é cadastrado, também podemos ver quando é solicitado girar a roleta podemos ver que o código sorteia um dos prêmios cadastrados e mostra o prêmio para o usuário.

**Conclusão:**

Por fim, podemos concluir que este trabalho teve como objetivo proporcionar para os estudantes um maior aprofundamento em conceitos importantes para a disciplina como manipulação de dados ,processamento de dados, manipulação de ponteiros e estruturas de dados como listas circulares simples e duplamente encadeada, então podemos concluir 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.