Equipe: Davi Sousa Soares
Victor Kauan da Silva Miranda
João Pedro Saleh de Sousa

Professor: Raimundo Santos Moura

SETEMBRO
2024

Conteúdo

1 Introdução

Listas encadeadas são estruturas de dados que têm como finalidade criar uma sequência de elementos ordenados sem a necessidade de reservar um espaço na memória para seus elementos.A lista é composta de nódulos que apontam para o próximo elemento da lista.Chamamos de head o começo da lista e next o elemento a qual um elemento qualquer da lista aponta para. Neste relatório será discutido a elaboração de duas lista encadeadas cada uma com funcionalidades de acordo com o que foi solicitado pela atividade pela atividade

1.1 Objetivos

O objetivo de cada código é cumprir uma determinada tarefa de modo a facilitar o entendimento do aluno a respeito dessas ferramentas

1.2 Especificações

O primeiro código se trata de uma lista encadeada simples com um único descritor com as seguintes tarefas:

A. Inserir um elemento no início da lista
B. Inserir um elemento no final da lista

C. Verificar se um determinado elemento existe na lista, sendo o elemento fornecido pelo usuário
D. Excluir um determinado elemento da lista
E. Mostrar todos os elementos e dizer a quantidade de elementos na lista.
F. Tratar as informações sem considerar case-sensitive, ou seja, nomes como: RAIMUNDO, Raimundo e raimundo devem ser tratados como um único nome;
G. Garantir a unicidade dos elementos, ou seja, NÃO permitir informações duplicadas. Se a informação já estiver armazenada, uma mensagem do tipo "Nome já cadastrado" deve ser apresentada

Já o segundo trata-se de um lista duplamente encadeada(faz referência ao seu sucessor e ao seu antecessor na ordem) por finalidade:

A. Inserir os elementos ordenados (em ordem crescente);
B. Verificar se um determinado elemento existe na lista, sendo o elemento fornecido pelo usuário;
C. Excluir um determinado elemento da lista;
D. Mostrar todos os elementos e dizer a quantidade de elementos na lista.

2 Funcionalidades

Este tópico é destinado a explicar as capacidades do código realizado

2.1 Primeira tarefa

A. Para inserir um elemento no início da lista identificasse o elemento que o head da lista aponta e torna-o o next do elemento que será adicionado,feito isso o head da lista agora é o último elemento adicionado
B. Para inserir um elemento no final da lista, a lista possui um identificador do último elemento dela chamado back,o elemento adicionado se torna o next desse elemento contido no back e em seguida o back começa a apontar para o elemento adicionado
C. Para verificar se um elemento existe na lista ,iterasse pela lista procurando um nódulo com o valor igual ao fornecido caso seja achado a lista retorna True, caso contrário retorna False
D. Para excluir um elemento da lista ,iterasse pela lista procurando o nódulo cujo next aponta para o elemento que será apagado e o nódulo em si. Feito isso o primeiro nódulo mencionado deixa de apontar para o elemento que será apagado e passa a apontar para seu next.
    a. Para checar case-sensitivity todos os elemento são transformados em string e colocado em caixa alta,portando tanto raimundo quando Raimundo se tornam RAIMUNDO que são identificados como o mesmo elemento
    b. Todos dos os elemento antes de serem adicionados é verificado se eles existem na lista pelo método search, caso existam não são adicionados
E. Para mostrar todos os elemento foi usado a função repr e para contar quantos elemento tem na lista,a lista em si possui um contador que muda sempre que um item é adicionado ou retirado

2.2 Segunda tarefa

A. Para inserir os elementos ordenadamente primeiro o código checa se o head ou o tail aponta para algum node, se não o elemento adicionado assume o lugar do head.Caso a lista não esteja vazia o código itera por todos os seus elementos até achar um que seja maior que o elemento que será adicionado, assim colocando o elemento adicionado atrás desse elemento maior
B. Para verificar se um elemento existe na lista o código itera por todos os seu elementos, checa seus valores e os compara com o elemento que procura.Caso ache retorna o nodulo caso contrário retorna None
C. O código realiza o método de busca mencionado no item B.Caso a busca retorne um nódulo ele pega seu prev e seu next e faz com que apontem um pro outro.São realizadas checagens para garantir que o prev e o next não são None.
D. Para mostrar todos os seu elementos ele itera pela lista e adiciona os valores contidos nos nódulos em uma lista e retorna a mesma,junto com a contagem de elementos que é realizada sempre que ocorre uma adição ou exclusão