Listas Lineares

-> Operacoes

Criação/Inicializacao

Inserções } Inicio, meio, fim

Remocoes } Inicio, meio, fim

Destriução da Listas

Pesquisas

Ordenação

Estrutura Nodo

estrutura SNodo

INFO item;

. .

. .

. .

NODO ^prox

. .

. .

. .

fim\_estrutura

TIPO estrutura SNODO NODO;

TIPO <tipo> INFO;

Listas com Encadeamento Simples

L HEAD #100 #300 L TAIL

#200

#300

/

#100

...

nodo ponteiro nodo

Cabeça da lista elementos encadeados da lista cauda da lista

estrutura do NODO

estrutura SNodo

item : INFO

prox : ^ NODO

fim-estrutura

Operações

CriarLLEncSimples(LL:^NODO)

LL <- NULO

Fim\_CriarLLEncSimples

DestruirLLEncSimples(LL:^NODO)

aux : ^NODO;

aux <- LL;

enquanto LL != NULO faça

LL <- LL->Prox

apagar(aux)

aux <- LL

fim-enquanto

Fim-DestruirLLEncSimples

/\*

implementacao da Lista Linear (LL)

encadeamento seria

sustenido representa um endereço de memoria, uma representacao

o TIPO é um typedef

^ é um ponteiro em portugol

Algoritmo

Ao criar a lista, inicia colocando nulo na cabeça da lista

Destruir a lista é torna-la vazia e eliminar os elementos da memoria.

Apagar = Delete em C++

A lista é limpa fazendo com que a head aponte, não para o proximo, mas para o proximo do proximo, e apagando o que a cabeça anteriormente apontava

\*/

Prova

Desenvolver a lista e trazer aqui funcionando para testar no BOCA

2Prova eh prova mesmo, sem choro