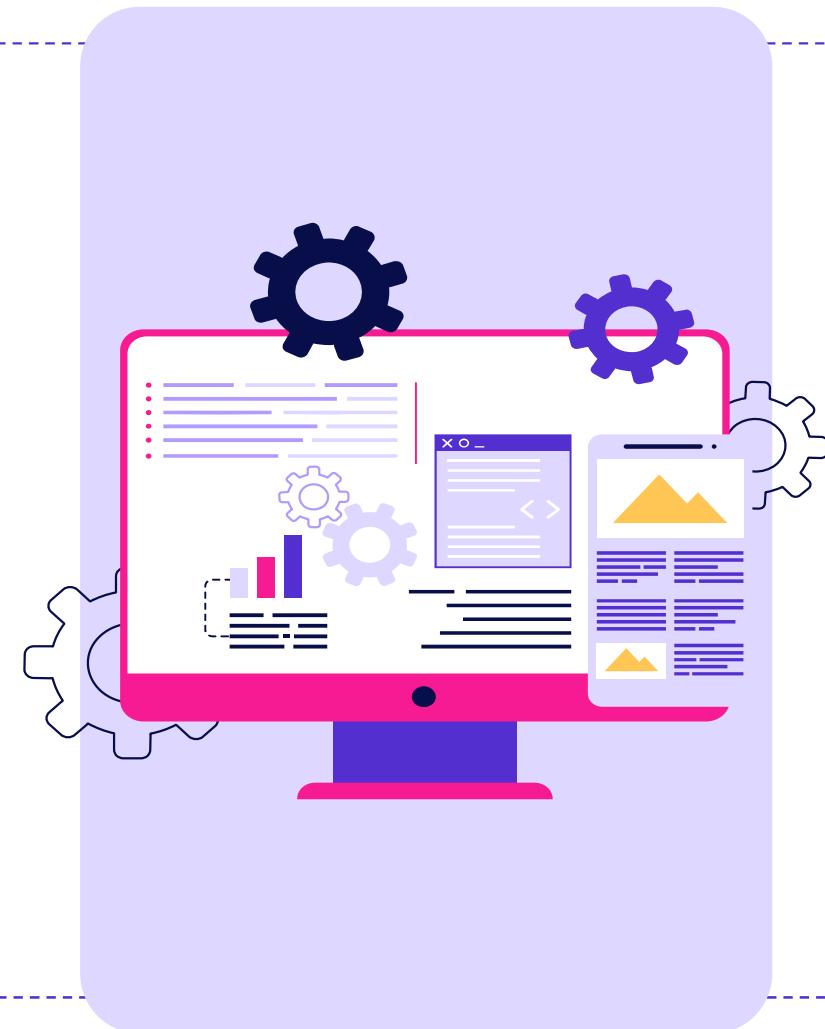


Estrutura de Dados 1

LISTA

Prof^a Juliana Franciscani



Roteiro



01

O que é

03

Utilização

05

Variações

02

Importância

04

Exemplos



Lista

- ❑ Definida como uma sequência de dados de um mesmo tipo
- ❑ Implementa as funções da lista sem se preocupar com o que está dentro dela

- ❑ Utiliza-se a nomenclatura nó para cada elemento da lista
- ❑ Uma lista pode ter n elementos $\rightarrow n \geq 0$
- ❑ Se n for 0 é dito que a lista está vazia.



Exemplos de lista

- ✓ Cadastro de funcionários
- ✓ Itens de estoque
- ✓ Pessoas na fila de um banco;
- ✓ Letras em uma palavra;
- ✓ Relação de notas dos alunos de uma turma;
- ✓ Itens em estoque em uma empresa;
- ✓ Dias da semana;
- ✓ Vagões de um trem;



Operações em lista

- Criar uma lista;
 - Remover uma lista;
 - Inserir um elemento na lista;
 - Remover elemento da lista;
 - Copiar uma lista;
-
- Acessar um elemento da lista;
 - Alterar um elemento da lista;
 - Localizar elemento através de informação;
 - Destruir uma lista;

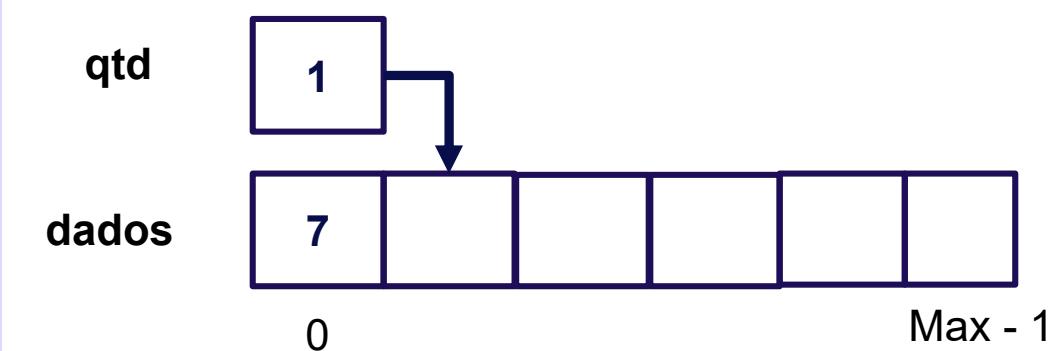


**Dependem do tipo de alocação realizada:
Alocação estática ou alocação dinâmica**

Alocação estática

- Espaço de memória é alocado no momento de compilação
- Exige a definição do número máximo de elementos da LISTA
- Acesso sequencial: elementos consecutivos na memória
- Lista é um VETOR

LISTA



Lista Sequencial Estática

Vantagens:

- Criação de um array é simples;
- Acesso rápido e direto aos elementos
- Tempo constante para acessar um elemento
- Facilidade em alterar informações
- Não há necessidade de compreender ponteiros ou referências.

Desvantagens:

- Limitações quanto ao tamanho de memória;
- Necessário definir o tamanho do vetor;
- Dificuldade para inserir e remover um elemento entre outros dois
- Custo computacional maior;
- Alocação de memória exagerada.

Lista Sequencial Estática

Quando utilizar:

- Listas pequenas
- Inserção e remoção apenas no final da lista
- Tamanho máximo bem definido
- A busca é a operação mais frequente

Implementação Lista Sequencial Estática – aula 8



Alocação Dinâmica

- Espaço de memória é alocado em tempo de execução
- A lista cresce à medida que novos elementos são armazenados e diminui à medida que removidos
- Acesso encadeado: cada elemento pode estar em uma área distinta de memória.
- Para acessar um elemento necessário percorrer os antecessores

LISTA

inicio

7

13

9

23

dado

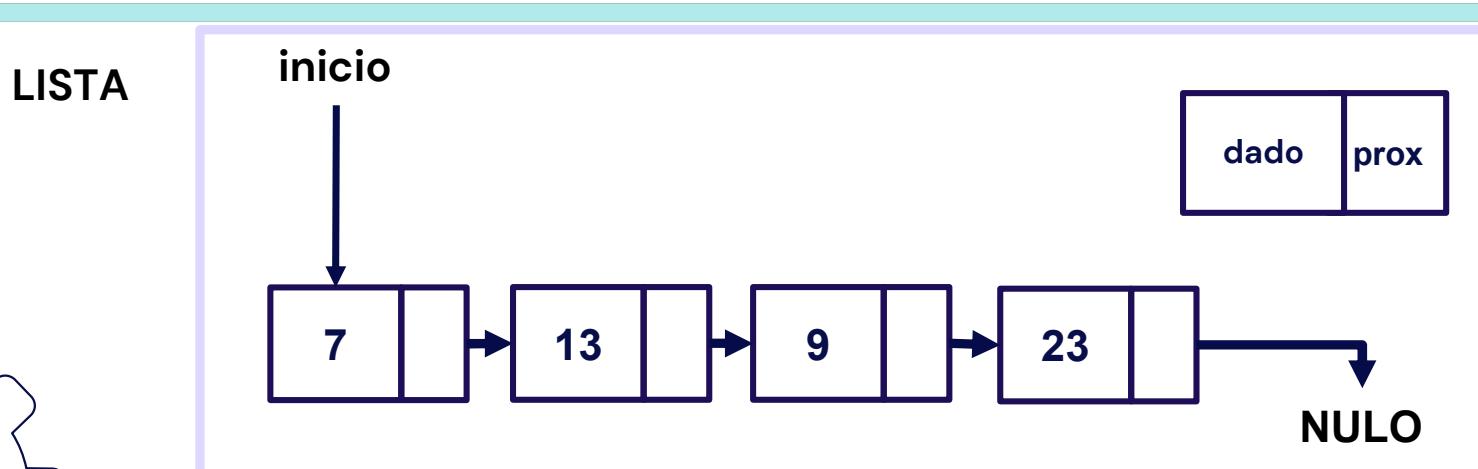
prox

NULO



Listas Dinâmicas Encadeadas

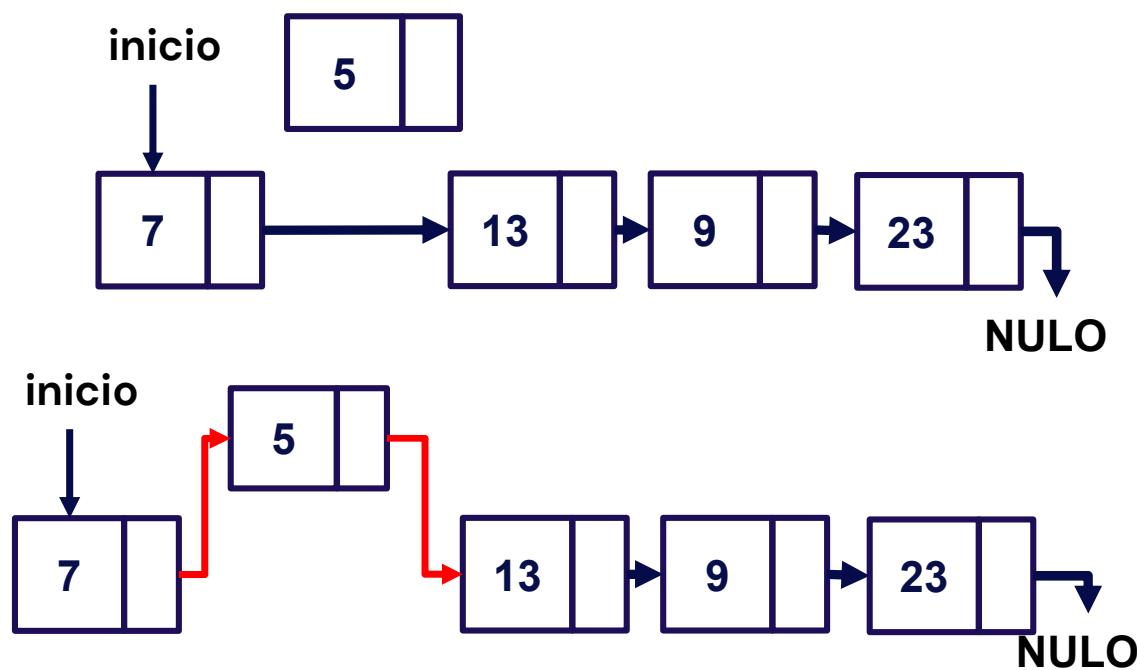
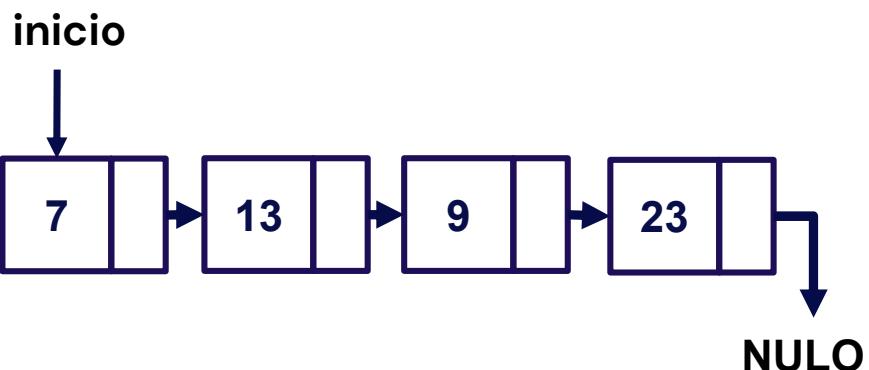
- Tipo de LISTA onde cada elemento aponta para o seu sucessor na LISTA
- Usa um ponteiro especial para o primeiro elemento da lista (inicio) e uma indicação de final de lista
- Cada elemento é tratado como um ponteiro que é alocado dinamicamente, a medida que os dados são inseridos
- Primeiro elemento: ponteiro para ponteiro (**inicio)



Lista Dinâmica Encadeada

Vantagens:

- Melhor utilização de recursos de memória
- Não precisa movimentar os elementos nas operações de inserção e remoção.



Lista Dinâmica Encadeada

Desvantagens:

- Não possui acesso direto aos elementos
- Necessidade de percorrer a lista para acessar um elemento.

Quando utilizar lista dinâmica?

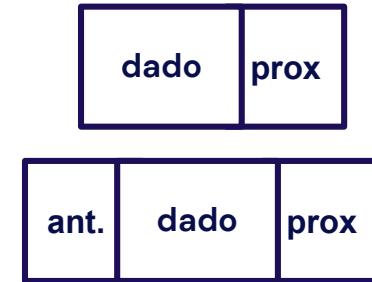
- Não há necessidade de garantir um espaço mínimo para a execução do aplicativo
- Operações mais frequentes: inserção e remoção



Listas Dinâmicas Encadeadas

Variações de Lista Dinâmica:

- Lista Simplesmente Encadeada (aula 9)
- Lista Duplamente encadeada (aula 10)
- Lista Circular
- Lista com nó descritor (nó que possui informações como inicio, final e quantidade de elementos)



Tipos especiais de Lista Dinâmica:

- Pilha
- Fila



Referências

EDELWEISS, N.; GALANTE, R.. Estruturas de dados. Porto Alegre: Bookman, 2009.

SZWARCFITER, J. L.; MARKENZON, L. Estruturas de dados e seus algoritmos. 3. ed. Rio de Janeiro: LTC, 2010.

ZIVIANI, N. Projeto de algoritmos: com implementações em Pascal e C. 3. ed. rev. e ampl. São Paulo: Cengage Learning, 2011.

Aulas e vídeo aulas do professor André Backes:
<https://www.facom.ufu.br/~backes/>

