



Conteúdo

1. Introdução

2. Listas

3. Pilhas e Filas

4. Árvores

5. Árvores de Pesquisa

- Árvore Binária e Árvore AVL
- Árvore N-ária e Árvore B

6. Tabelas de Dispersão (Hashing)

7. Métodos de Acesso a Arquivos

8. Métodos de Ordenação de Dados





Listas





Listas

➡ Lista - série de elementos.

Lista - é a seqüência linear de itens, cuja principal propriedade estrutural é a posição relativa dos elementos dentro da seqüência.

➡ Exemplos de aplicação de listas:

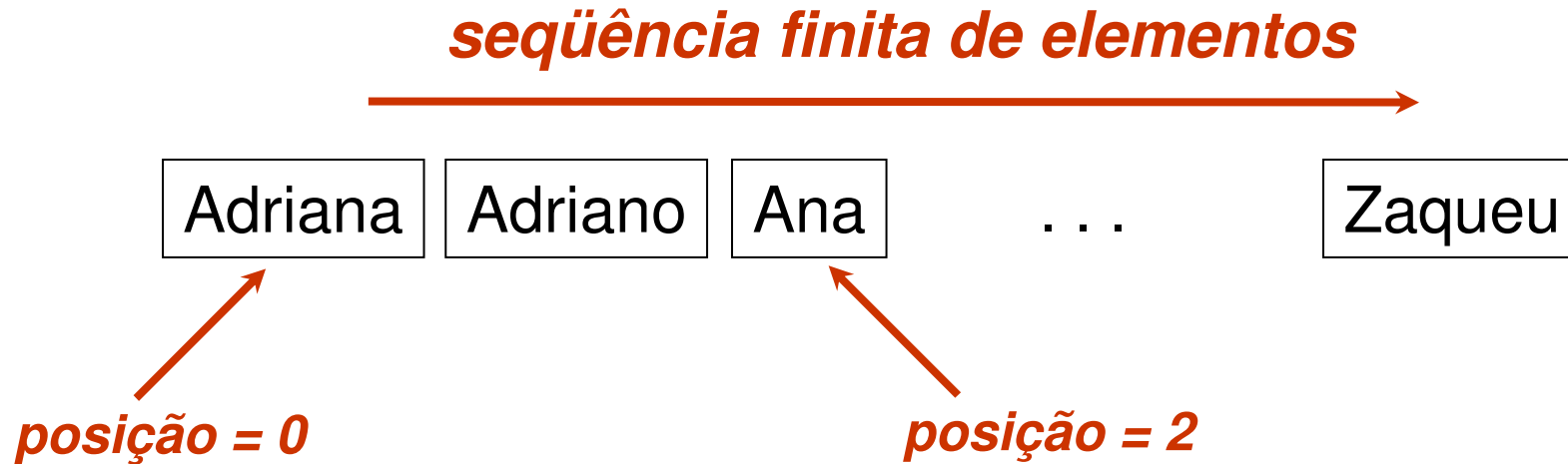
- Vôo com conexões que podem ser removidas ou incluídas
- Relação de nomes de alunos da folha de chamada, em ordem alfabética
- Relação de notas de alunos



•
•
•

Características de uma Lista

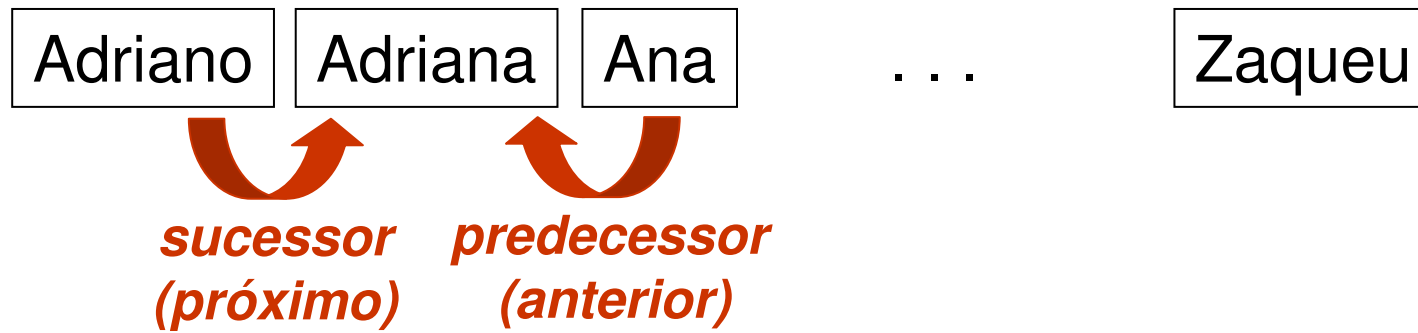
⇒ Seqüência Finita de Elementos



• • • • • • • • •

Características de uma Lista

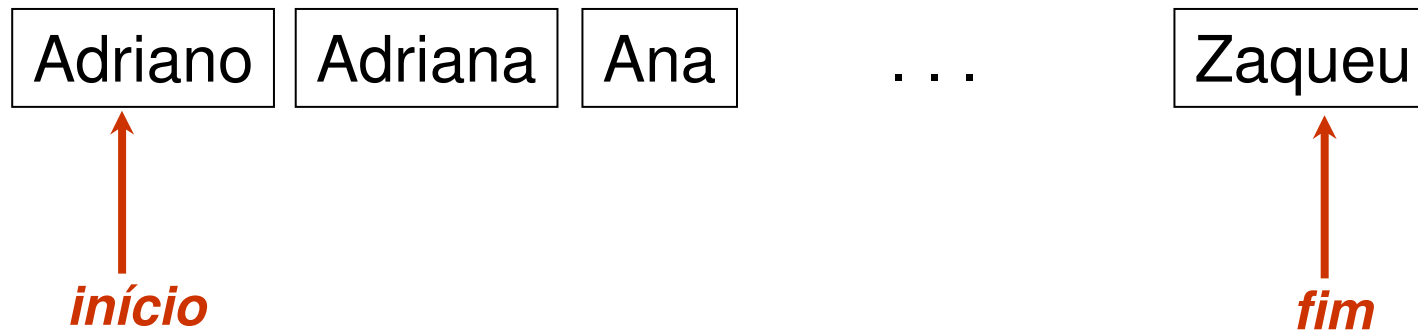
⇒ Seqüência: próximo e anterior





Características de uma Lista

➡ Sequência: início e fim





Características de uma Lista

➡ Comprimento

Adriano	Adriana	Ana	Pedro	Tânia	Zaqueu
---------	---------	-----	-------	-------	--------

Comprimento = 6

➡ Lista Vazia

Comprimento = 0





Características de uma Lista

➡ Lista homogênea



➡ Lista heterogênea (não é usual)



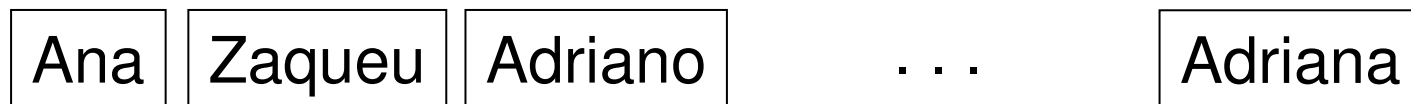


Características de uma Lista

➡ Lista ordenada



➡ Lista não-ordenada





Lista - Definição

Um conjunto de elementos $X = \{ x_1, \dots, x_n \}$, $n \geq 0$, tem uma estrutura de lista linear para uma determinada relação de precedência quando :

- i) existem n elementos na seqüência;
- ii) x_1 é o primeiro elemento;
- iii) x_n é o último elemento;
- iv) $\forall 1 \leq i, j \leq n$, se $i < j$, então x_i precede x_j ;
- v) se $i = j - 1$ então x_i é o antecessor de x_j e x_j é o sucessor de x_i .





Operações sobre uma Lista

➡ Que operações podem ser feitas sobre uma lista???

14

32

7

...

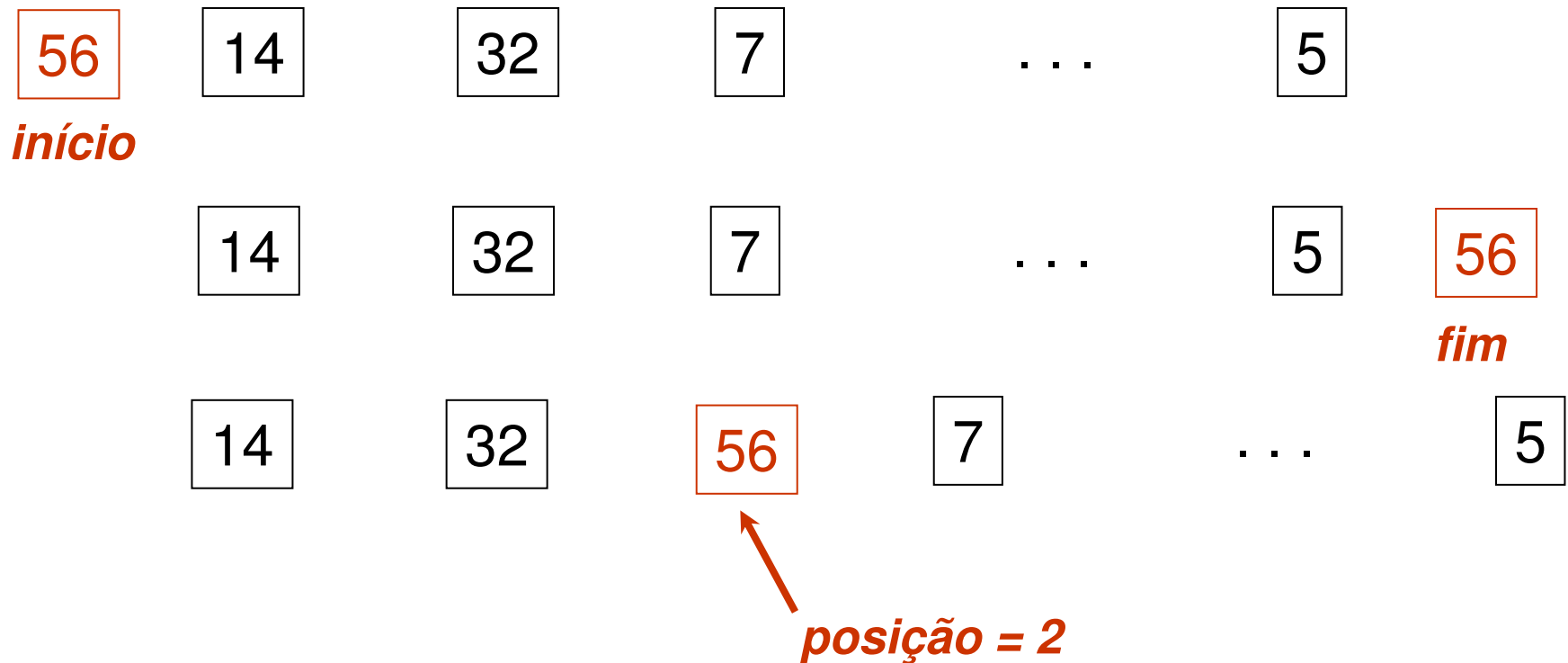
5



...

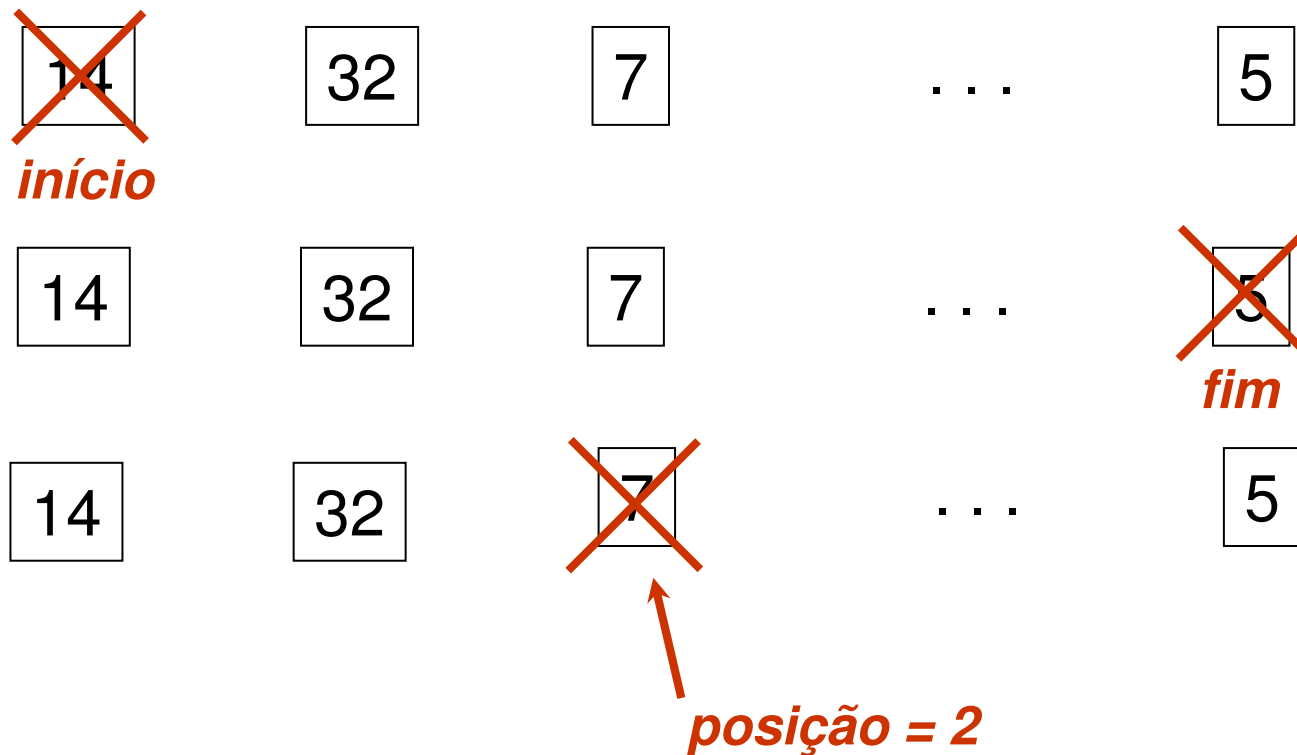
Operações sobre uma Lista

→ Inserção de elemento



Operações sobre uma Lista

→ Remoção de elemento

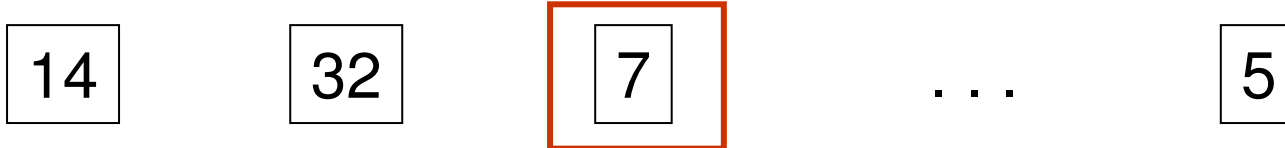


...

Operações sobre uma Lista

→ Consulta

por posição (Ex.: posição = 2)



todos os elementos





Interface da Lista

➡ Definir uma interface para a estrutura de dados Lista



Interface da Lista

```
public interface Lista <E> extends EstruturaDados{  
    public void insere (E elemento, int posicao) throws ExcecaoPosicaoInvalida;  
    public void insere (E elemento);  
    public void insere (Lista<E> l, int posicao) throws ExcecaoPosicaoInvalida;  
    public boolean contem (E elemento);  
    public E remove (int posicao) throws ExcecaoPosicaoInvalida;  
    public int remove (E elemento);  
    public void remove (Lista<E> l);  
    public E retorna (int posicao) throws ExcecaoPosicaoInvalida;  
    public int retornaPosicao (E elemento);  
    public E substitui (int posicao, E elemento) throws ExcecaoPosicaoInvalida;  
    public void trocaPosicao (int posicao1, int posicao2) throws ExcecaoPosicaoInvalida;  
    public Lista<E> retornaSubLista (int inicio, int fim) throws ExcecaoPosicaoInvalida;  
    public Iterator<E> retornaliterator();  
}
```




Interface da Lista

```
public interface EstruturaDados {  
    public boolean estaVazia();  
    public void esvazie();  
    public int numeroElementos();  
}
```





Modelagem Física de Listas

➡ Duas alternativas de implementação:

- Array
- Encadeamento





Implementação com Array

➡ Os elementos ficam justapostos na memória através da utilização de um vetor unidimensional.

0	1	2	3	4	5	6	7	8	9	...	M
e_1	e_2	e_3	e_4	e_5	e_6	e_7	...	e_n			



Implementação com Encadeamento

➡ Os elementos estão associados entre si através de referências.

