Universidade Eduardo Mondlane
Faculdade de Engenharia
Departamento de Engenharia Eletrotécnica
Engenharia Informática

Estrutura de Dados e Algoritmos – EDA

Por:

- **Dr. Alfredo Covele**
- **Eng. Cristiliano Maculuve**

Agosto 2023

Tópicos

No.	Designação
1	Listas
2	Pilhas
3	Filas

Tipos de dados, Estrutura de dados e Tipos Abstratos de dados.

Os tipos de dados são utilizados pelas linguagens de programação para definir o conjunto de valores que uma variável pode assumir.

Por exemplo, o tipo de dado inteiro pode assumir valores como 1, 25, 1896, etc. O tipo data pode assumir valores como “25/12/2015”, “07/09/1822”, etc

Tipos de dados, Estrutura de dados e Tipos Abstratos de dados.

Outra forma de visualizar os tipos de dados é em termos do que o usuário deseja fazer com os dados, como por exemplo, somar dois inteiros, ordenar um lista de inteiros, etc. Este tipo de conceito de tipo de dado é conhecido como TAD – Tipo Abstrato de Dado.

Conceito de função

- A função pode ser definida como um bloco de código com uma tarefa específica.

```
Tipo_do_Dado_de_Retorno Nome_da_Função(Listas de parâmetros...)
{
    Corpo da função
}
```

Tipos de estruturas de dados

As estruturas de dados podem ser classificadas como:

Lineares-São aplicações onde os objetos são representados e manipulados em uma sequência ordenada de valores.

Tipos de estruturas de dados

Não lineares -cada objeto pode ter diferentes objetos na sequência

Tipos de estruturas de dados

Lineares

- Listas
- Pilhas
- Filas

Não lineares

- Árvores
- Grafos

Listas

são estruturas que permitem representar um conjunto de dados, que de alguma forma se relacionam, de forma que os elementos fiquem dispostos em sequência.

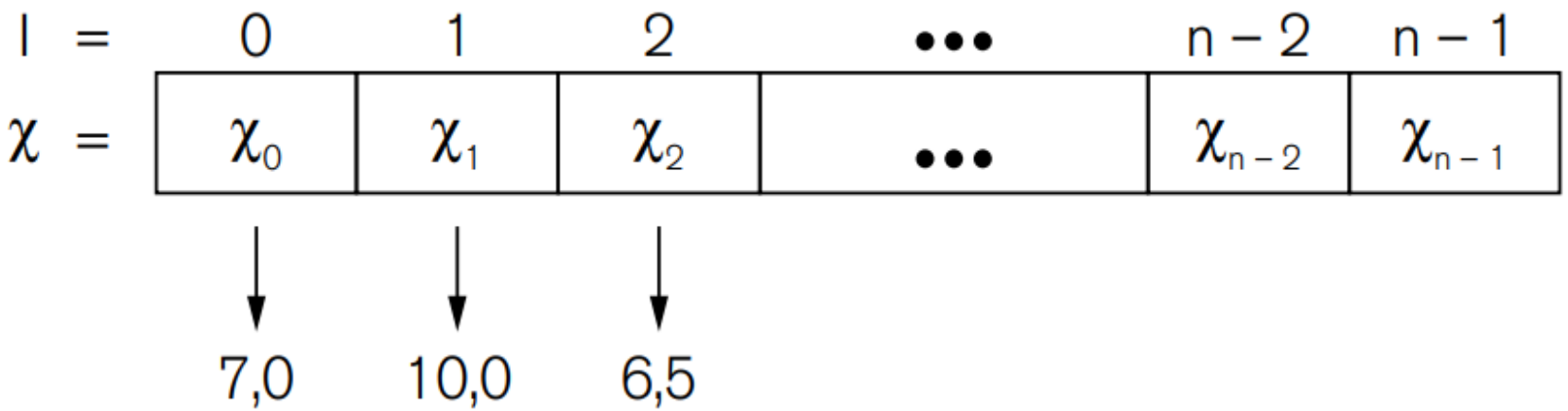
Cada elemento da lista também denominado nó, pode conter um dado primitivo (inteiro, string, etc.) ou dado composto

Listas

As listas podem ser implementadas de forma **estática** ou **dinâmica**. No desenvolvimento de um programa pode ser necessário determinar como será a inserção ou a remoção de elementos de uma lista.

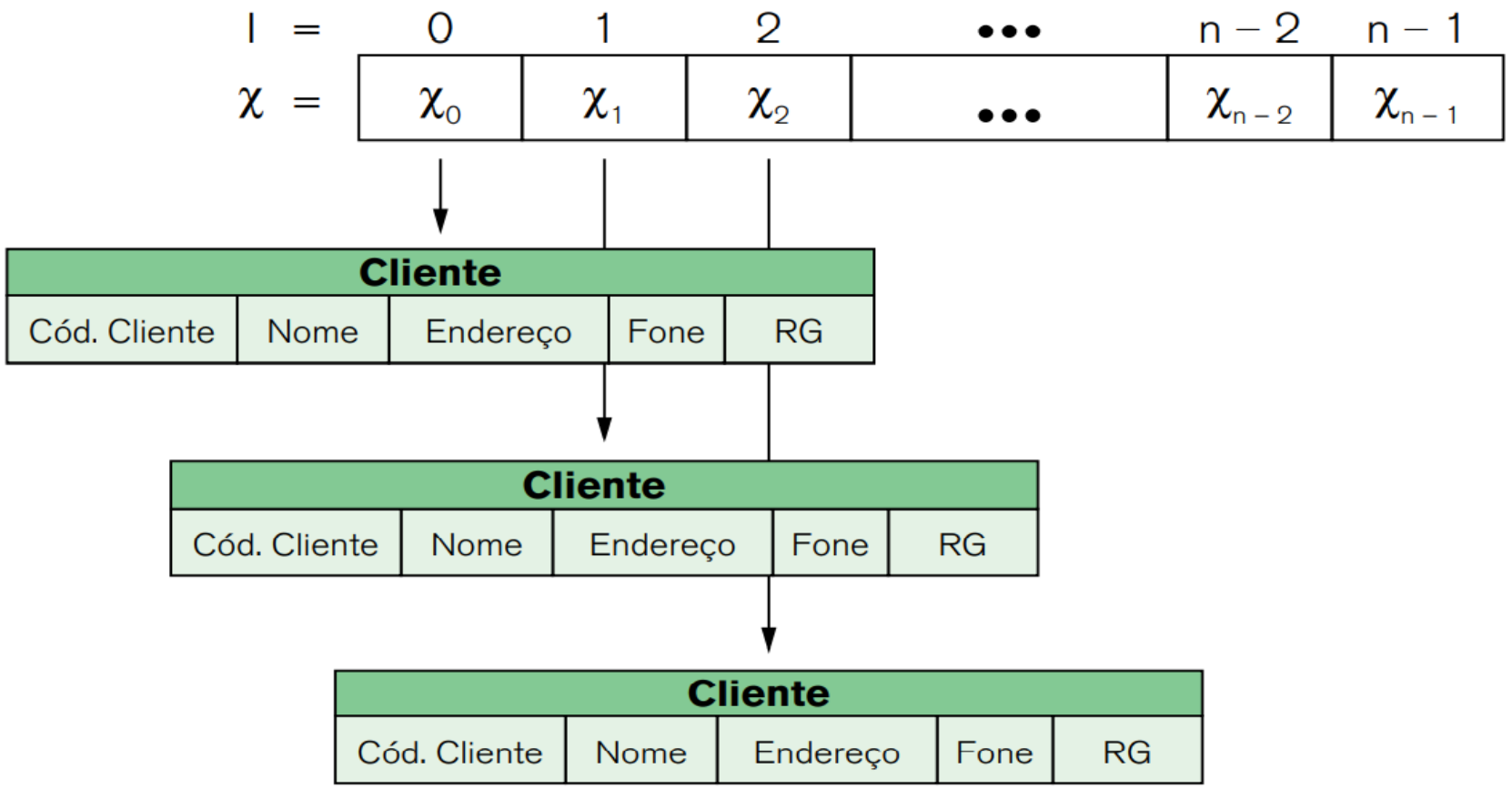
As operações implementadas em uma lista dependem do tipo de aplicação.

Listas



dado primitivo

Listas



Dado composto

Algumas operações mais comuns nas listas são:

- **Criar**: cria uma lista vazia.
- **Verificar lista vazia:** verifica se há algum elemento na lista.
- **Verificar lista cheia:** verifica se a lista esta cheia.
- **Inserir**: insere um elemento numa determinada posição ou no final da lista.
- **Alterar**: alterar algum elemento da lista.

Algumas operações mais comuns nas listas são:

- **Remover**: remove um elemento de uma determinada posição.
- **Buscar**: acessa um elemento da lista.
- **Exibir** a quantidade: retorna a quantidade de elementos da lista.
- **Combinar**: combina duas ou mais listas em uma única.
- **Dividir lista:** dividi uma lista em duas ou mais.

Algumas operações mais comuns nas listas são:

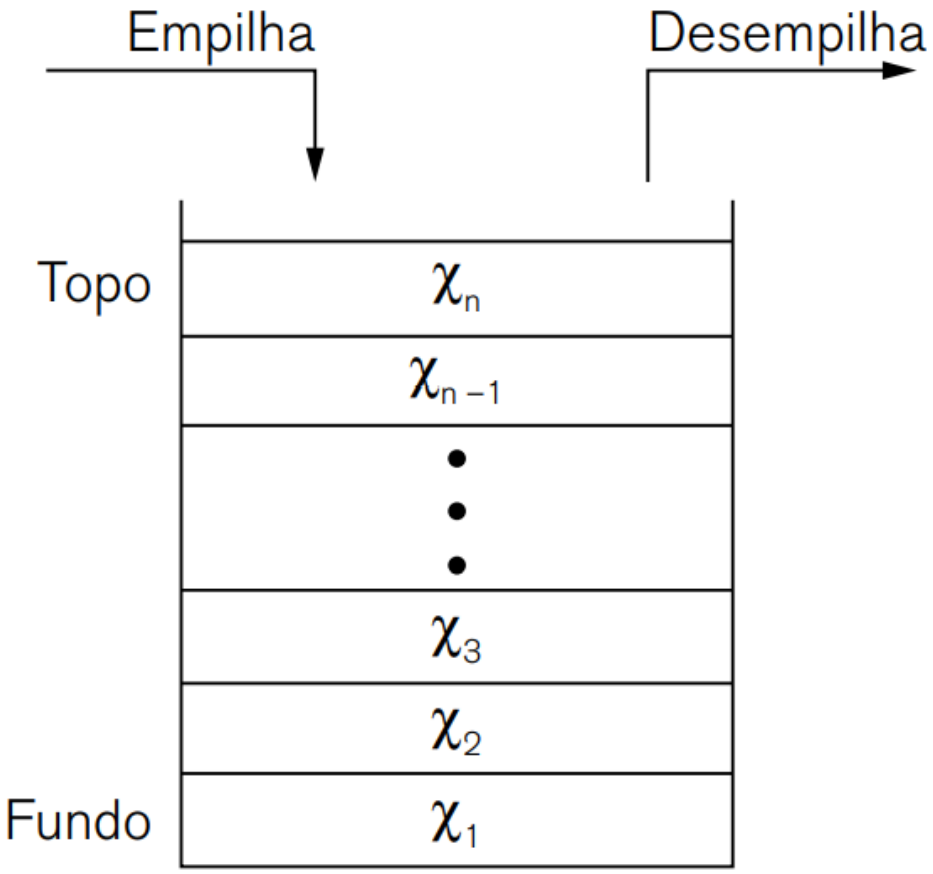
- **Ordenar**: ordena os elementos da lista de acordo com algum de seus componentes.
- **Esvaziar**: esvaziar a lista.

Pilhas

pilha é um tipo especial de lista em que os elementos a serem inseridos ou removidos ocorrem no topo da pilha.

Esta característica é conhecida como LIFO (Last In, First Out - Último a Entrar, Primeiro a Sair).

Pilhas



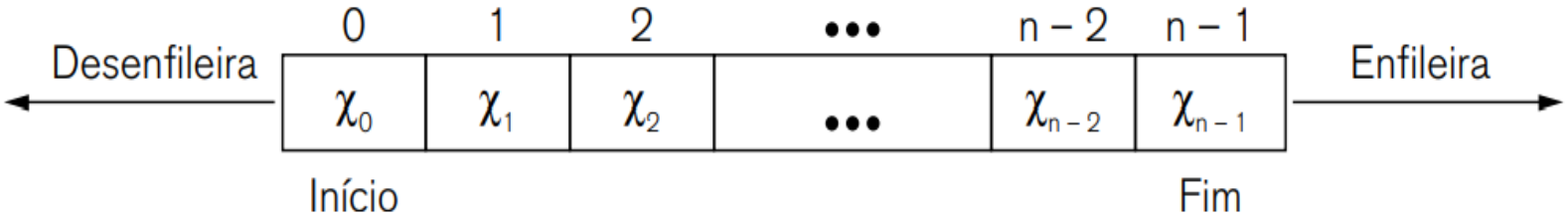
Algumas operações mais comuns nas Pilhas são:

- **Criar**: cria uma pilha vazia.
- **Empilhar**: insere um novo elemento no topo da pilha.
- **Desempilhar**: remove um elemento do topo da pilha.
- **Exibir topo:** exibe o elemento do topo da pilha.
- **Exibir** a quantidade: retorna a quantidade de elementos da pilha.
- **Esvaziar**: esvazia todos os elementos da pilha.

Fila

A fila também é um tipo especial de lista, onde os elementos são inseridos em uma extremidade, chamada início da fila, e retirados na extremidade oposta, chamada final da fila. Esta característica é conhecida como **FIFO** (First In, First Out - Primeiro a Entrar, Primeiro a Sair).

Fila



As filas são úteis em diversas aplicações, como por exemplo, os sistemas operacionais, que utilizam filas para gerenciar o escalonamento dos processos que serão executados pelo processador e a alocação de recursos.

Filas

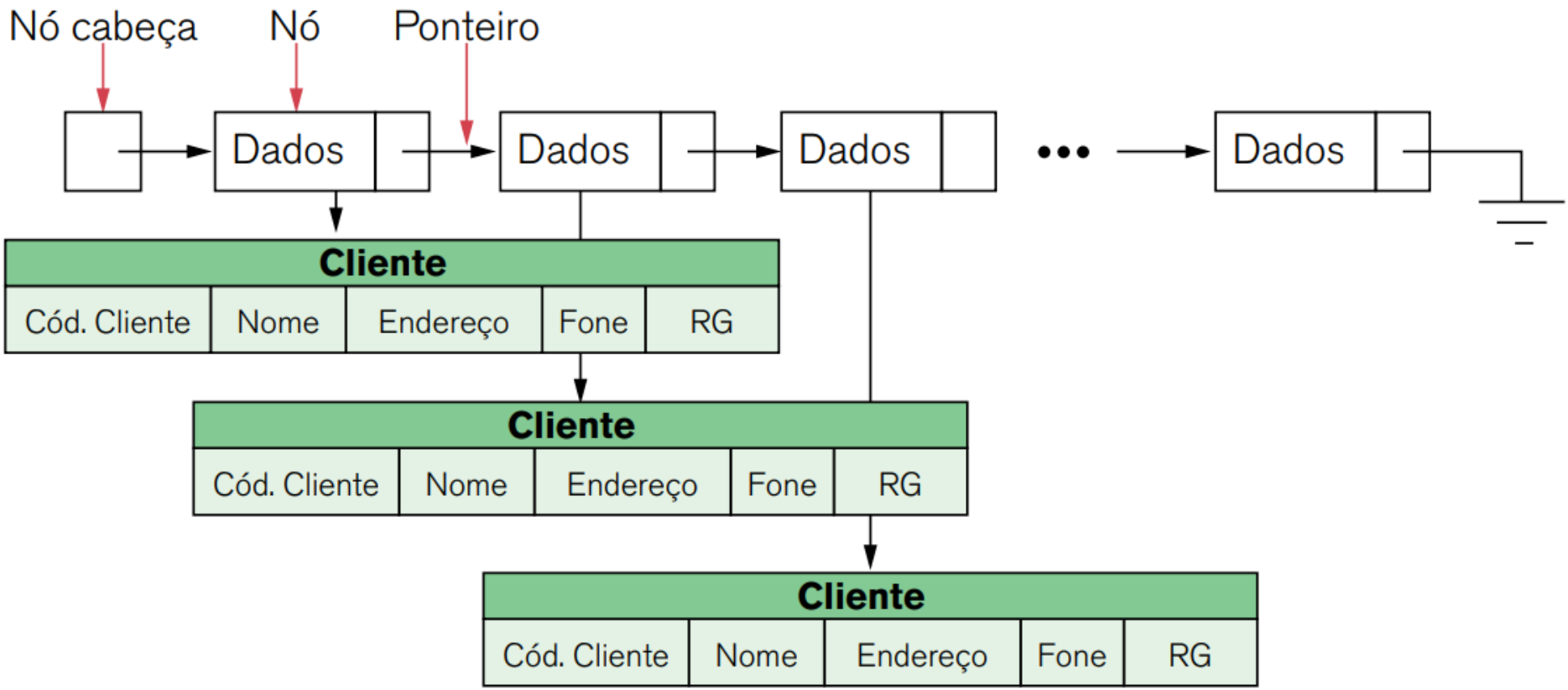
A fila também é um tipo especial de lista, onde os elementos são inseridos em uma extremidade, chamada início da fila, e retirados na extremidade oposta, chamada final da fila. Esta característica é conhecida como **FIFO** (First In, First Out - Primeiro a Entrar, Primeiro a Sair).

Algumas operações mais comuns nas Filas são:

- **Criar**: cria uma pilha vazia.
- **Enfileirar**: insere um novo elemento no topo da fila.
- **Desenfileirar**: remove um elemento do topo da fila.
- **Exibir** a quantidade: retorna a quantidade de elementos da fila.
- **Esvaziar**: esvazia todos os elementos da fila.

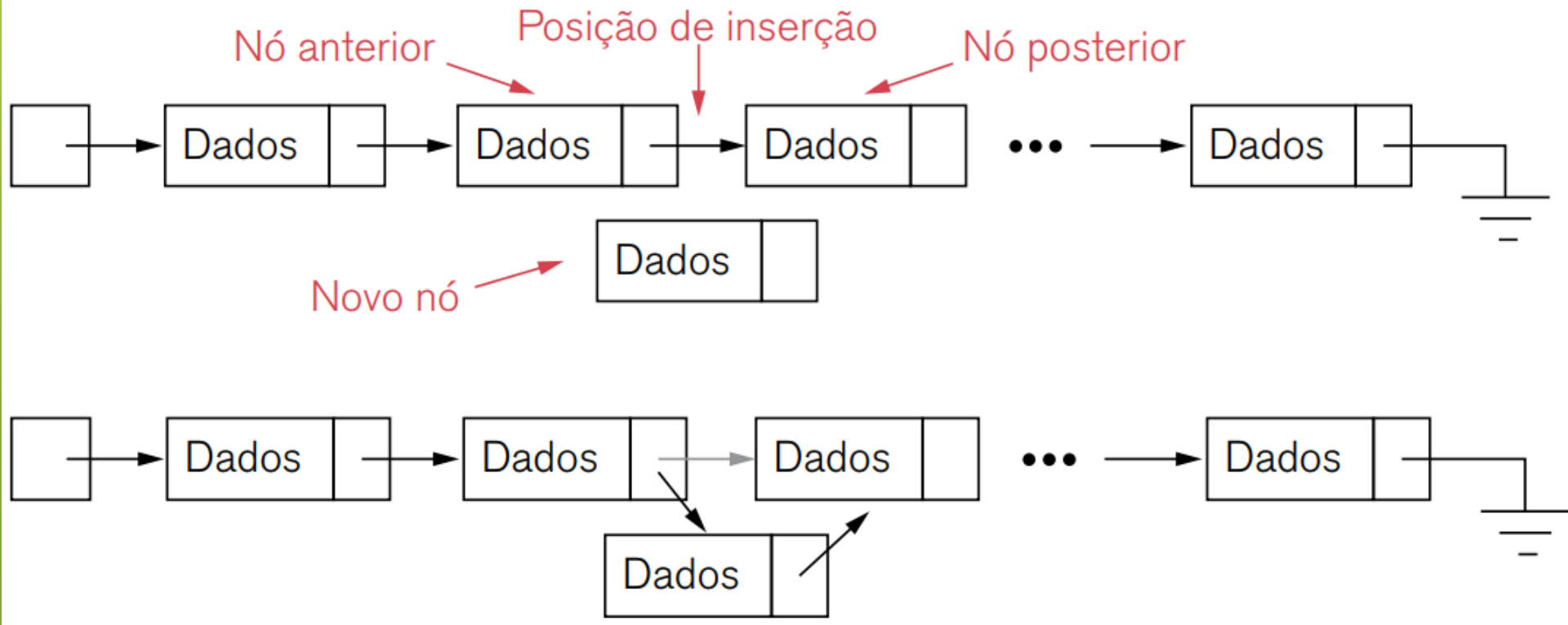
Lista Encadeada

- A lista encadeada possui um nó especial, chamado nó-cabeça. O nó-cabeça indica o início da lista encadeada e nunca pode ser removido. Dados (registros) do tipo que são armazenados nos demais nós da lista, não devem ser armazenados no nó-cabeça.


Nó cabeça
Nó
Ponteiro
Dados
Dados
Dados
Dados
Cliente
Cód. Cliente
Nome
Endereço
Fone
RG
Cliente
Cód. Cliente
Nome
Endereço
Fone
RG
Cliente
Cód. Cliente
Nome
Endereço
Fone
RG

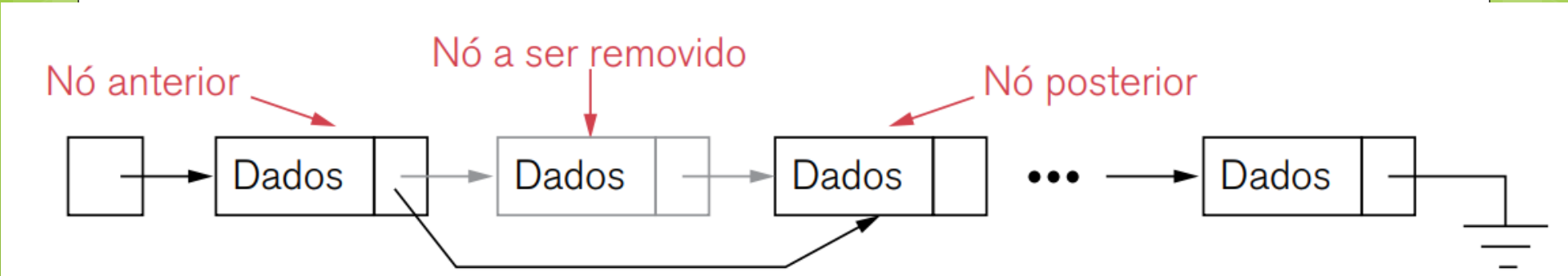
Inserir um elemento na lista:

- A partir do no-cabeça, efetua-se a busca da posição onde o novo nó deve ser inserido.
- Uma vez encontrado o local de inserção, liga o ponteiro nó anterior ao novo nó.
- Liga o ponteiro no novo nó ao nó posterior



Remover um elemento na lista:

- A partir do no-cabeça, efetua-se a busca do nó a ser removido.
- Uma vez encontrado nó a ser removido, liga o ponteiro do nó anterior ao nó posterior.



Lista Encadeada

- Vantagens da lista linear encadeada:
- • Facilidade de inserir ou remover um elemento em qualquer ponto da lista.
- • Não há a necessidade de movimentar os elementos da lista quando há uma inserção ou remoção de elemento.

Lista Encadeada

- Desvantagem da lista linear encadeada:
- • Por utilizar ponteiros, a implementação deve ser feita com muito cuidado para que não ocorra um mal encadeamento (ligação) dos nós e consequentemente a lista seja perdida.
- • Encontrar um determinado elemento na posição n da lista é necessário percorrer os n-1 anteriores.
- • Necessidade de memória extra para armazenamento dos ponteiros.