Newton

Quem se prepara, não para.

Estrutura de Dados

Professora: Michelle Hanne Soares de Andrade
michelle.andrade@newtonpaiva.br

Newton
Quem se prepara, não para.

Sumário:

- ✓ Lista Encadeada

Lista

Na computação, uma lista pode ser definida como um **conjunto de elementos do mesmo tipo, agrupados e identificados por um identificador único** e separados entre si em “caixas”, chamados de **nodos**, que ocupam um **endereço específico na memória.**

O relacionamento entre os nodos é definido por sua posição em relação aos demais nodos, assim como pessoas em uma fila, porém indexados na memória do computador.

Fonte: Pinto, 2020.

Lista

Toda lista apresenta um nodo inicial, o primeiro elemento da lista. A partir deste seguirá uma sequência de nodos conforme uma ordem predefinida pelo programador, assim como uma fila de banco.

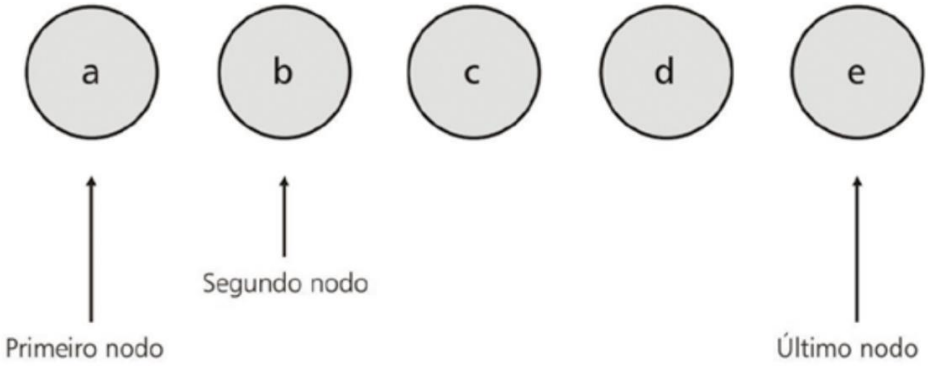


Figura 1. Lista composta por cinco elementos, nodos, sendo cada um identificado por uma letra do alfabeto.

Fonte: Edelweiss e Galante (2009, p. 50).

Fonte: Pinto, 2020.

Vetor Vantagens e Desvantagens

Newton
Quem se prepara, não para.

A seguir, vemos as vantagens de se utilizar vetores:

- Fácil implementação comparada a listas encadeadas.
- Redução da quantidade de memória utilizada, pois não requer armazenar a localização do próximo nodo.
- A sua dimensão está disponível sem a necessidade de percorrer todos os nodos.

Entretanto, há desvantagens na utilização de vetores, como mostrado a seguir.
Dificuldade na ordenação dos seus elementos.

- O aumento do seu tamanho poderá requerer o deslocamento do vetor em memória, caso não haja espaço de forma linear.
- A impossibilidade de adicionar um elemento no meio do vetor sem realizar o deslocamento de todos os demais nodos, logo, consumindo alto processamento.

Fonte: Pinto, 2020.

Estruturas de Listas Encadeadas

Newton
Quem se prepara, não para.

As estruturas dinâmicas são utilizadas para relacionar itens e nodos que **precisam ser manipulados em tempo de execução com dimensão indefinida**. A partir dessa premissa, podemos dizer que durante o tempo de execução será **possível adicionar ou remover itens de uma estrutura dinâmica de dados de acordo com a necessidade do usuário**

Fonte: Pinto, 2020.

Estruturas de Listas Encadeadas

Newton
Quem se prepara, não para.

O elemento básico de uma lista simplesmente encadeada é o nodo, cuja definição e função é importante compreender antes de tentar entender uma lista inteira.

Em Java, um nodo pode ser representado por uma classe com dois atributos, um dado e uma referência para o próximo elemento da lista.

De acordo com Knuth (1997), o armazenamento de informações em nodos e referências é mais flexível que a alocação de memória sequencial realizada por arrays, facilitando a inserção e a exclusão de informações, embora, como ponto negativo, tenha uma maior necessidade de memória.

Fonte: Rodrigues, 2021.

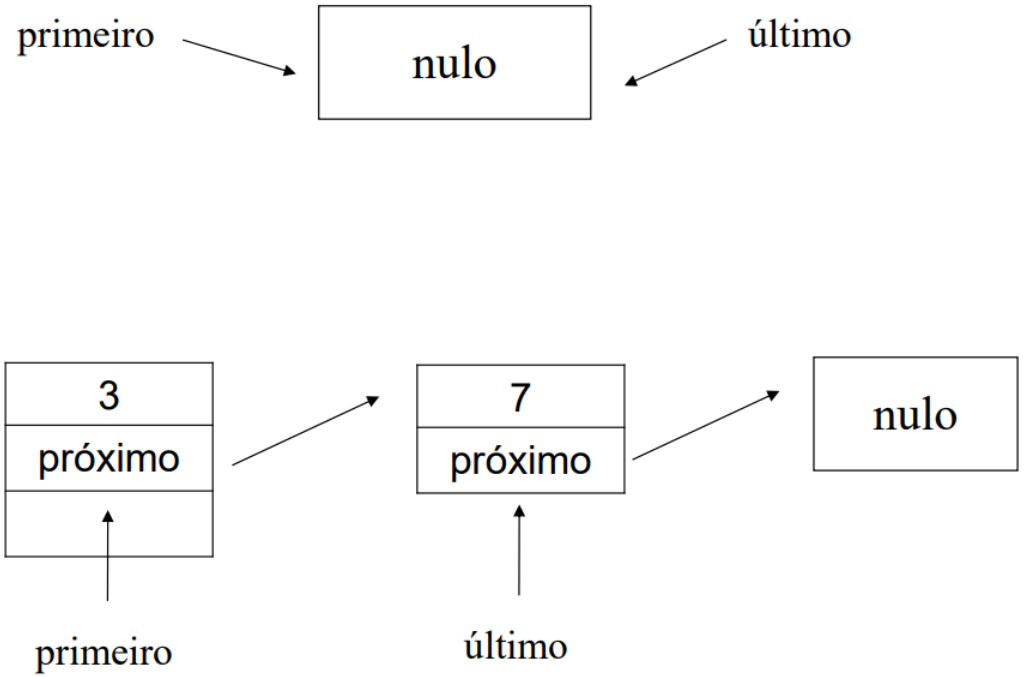
Operações de Lista Ligada

Newton
Quem se prepara, não para.

- **inserir no início** – insere um elemento no início da lista
- **inserir no final** – insere um elemento no final da lista
- **inserir no meio** – insere um elemento no meio da lista
- **lista vazia** – verifica se a lista está vazia
- **elemento do início** – mostra o elemento que está no início da lista
- **elemento do final** – mostra o elemento que está no final da lista
- **remover** – elimina um elemento que está na lista ligada
- **contar nós** – verifica quantos elementos existem na lista
- **mostrar lista** – exibe todos os elementos que a lista possui.
- **buscar** – verifica se determinado elemento pertence à lista.
- **destruir** – elimina todos os elementos da lista ligada.

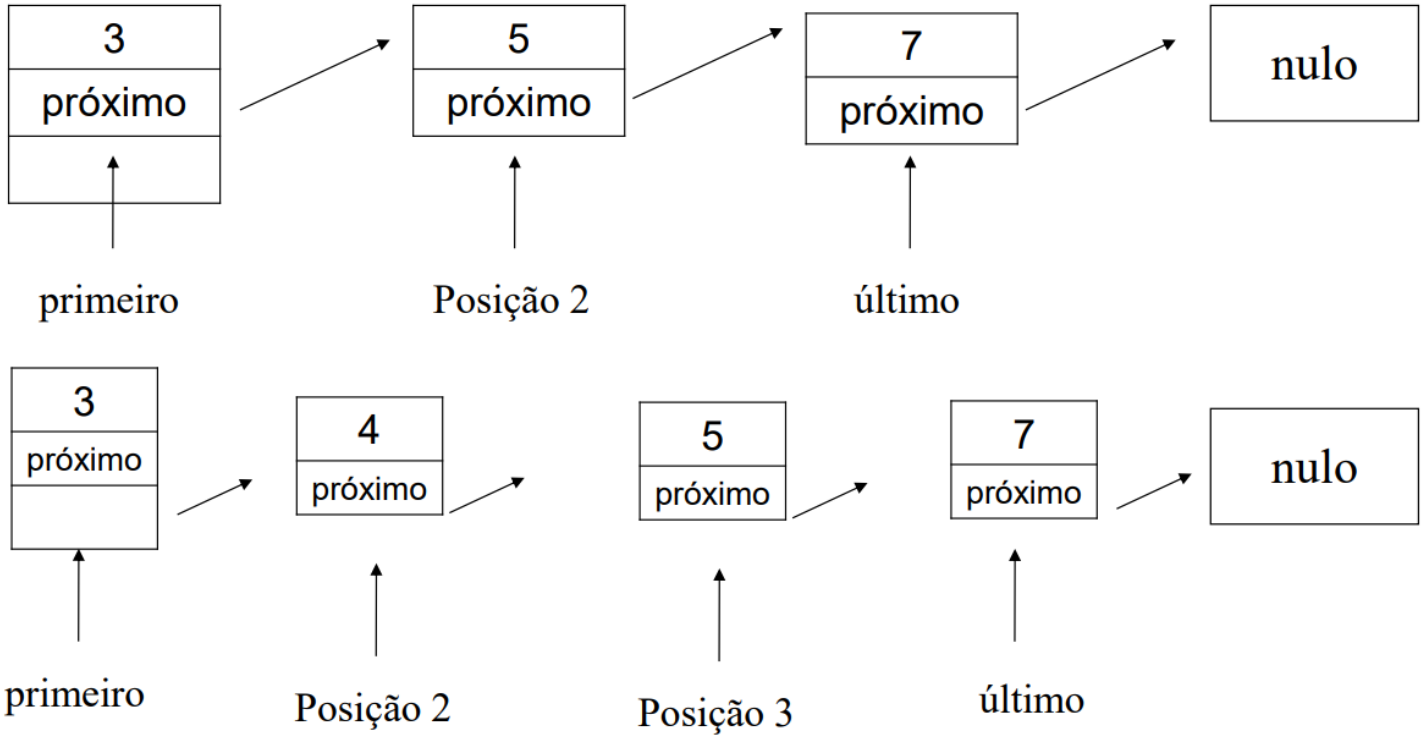
Fonte: Rodrigues, 2021.

Simulação Lista



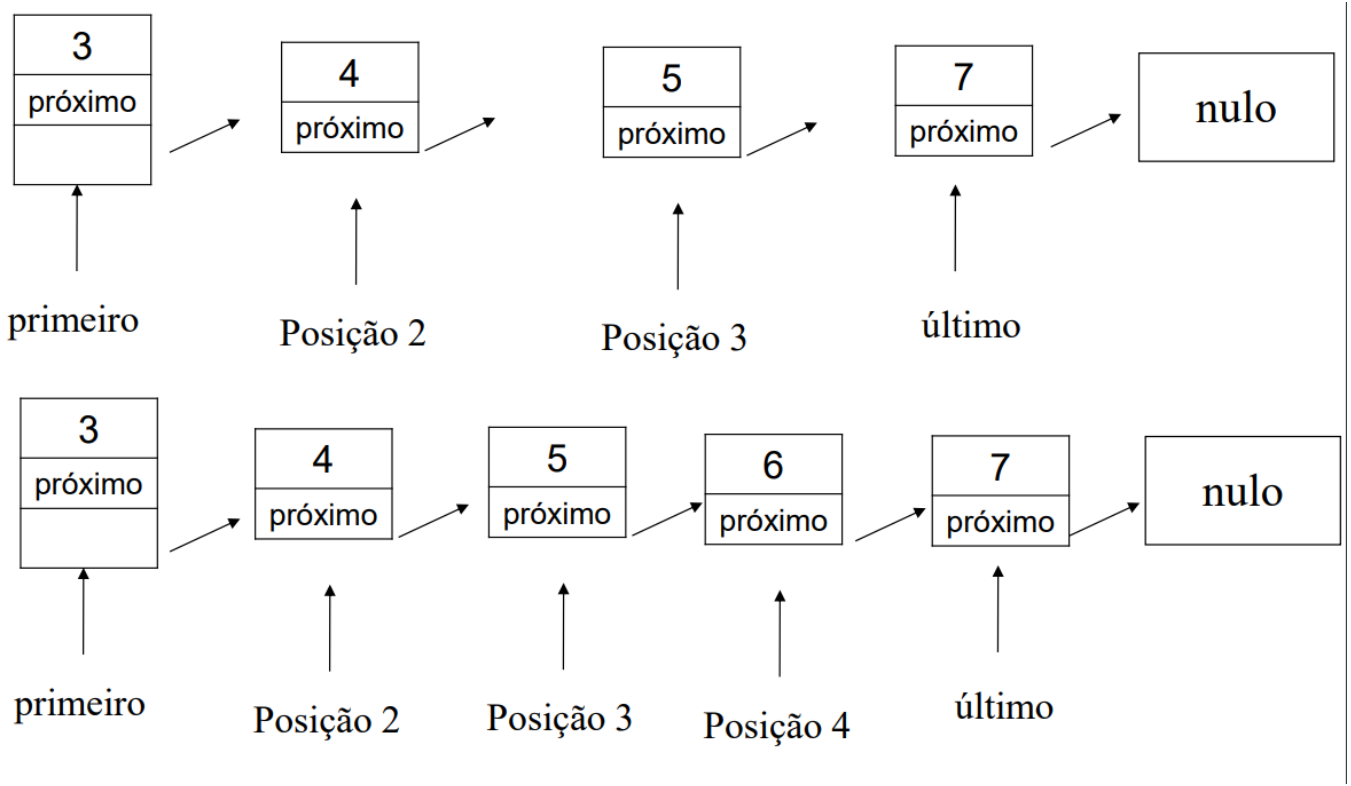
Fonte: http://walderson.com/site/wp-content/uploads/2015/02/Aula-04.pdf

Simulação Lista



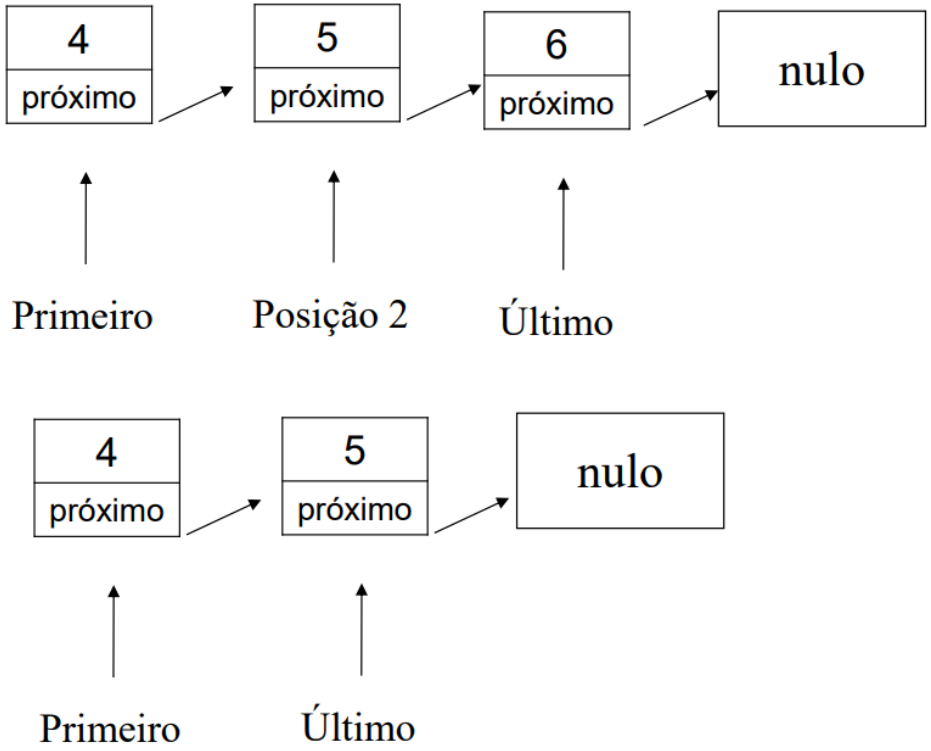
Fonte: http://walderson.com/site/wp-content/uploads/2015/02/Aula-04.pdf

Simulação Lista



Fonte: http://walderson.com/site/wp-content/uploads/2015/02/Aula-04.pdf

Simulação Lista



Fonte: http://walderson.com/site/wp-content/uploads/2015/02/Aula-04.pdf

Exemplo classe Nodo

Newton
Quem se prepara, não para.

```
public class Nodo {

private float dado;
private Nodo proximo;

public float getDado() {
return dado;
}

public void setDado(float dado) {
this.dado=dado;
}

public Nodo getProximo() {
return proximo;
}
public void setProximo(Nodo
proximo) {
this.proximo = proximo;
}
}
```

Exemplo classe Nodo

Newton
Quem se prepara, não para.

```
public class Principal {

public static void main(String[] args) {
Nodo nodo7 = new Nodo();
nodo7.setDado(7);
Nodo nodo8 = new Nodo();
nodo8.setDado(8);
nodo7.setProximo(nodo8);
System.out.println(nodo7.getDado());
System.out.println(nodo7.getProximo());

}
}
```

Outros Exemplos

Newton
Quem se prepara, não para.

Exemplos no Github: https://github.com/mihanne/ExemplosEstruturaDados

Referências

PINTO, Rafael A.; PRESTES, Lucas P.; SERPA, Matheus da S.; et al. Estrutura de dados. Editora SAGAH, 2020. ISBN 9786581492953. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9786581492953

RODRIGUES, Thiago N.; LEOPOLDINO, Fabrício L.; PESSUTTO, Lucas Rafael C.; et al. Estrutura de Dados em Java. Editora SAGAH. 2021. ISBN 9786556901282. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9786556901282