

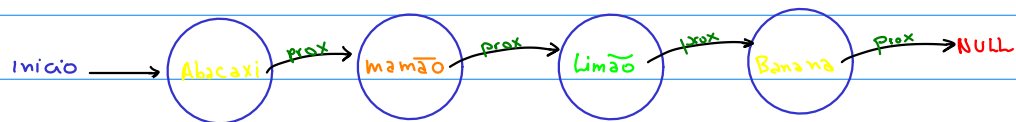
* Lista Encadeada:


↳ Linked list

• Conceito informal: sequência de elementos organizados um logo após ao outro, sem lacunas


• Conceito formal: linked list, isto é, uma lista ligada é um sequência de células/No, onde cada célula contém valores e o endereço da célula/No seguinte.

Exemplo:



Lembrando que cada No guarda seus valores e um mecanismo para unir Nos, que na lista Encadeada será utilizado para unir o próximo No da lista. Já sabemos que este mecanismo é um ponteiro denominado prox que aponta para o Endereço de  memória do próximo No.

Outra coisa importante, é saber onde está o início da lista, pois é por meio dele que você consegue acessar todos os Nos de uma lista encadeada.

Além disso, os Nos da lista encadeada são armazenados em slots de  memória arbitrários (diferente do que ocorre no Vetor)

* Vantagens:

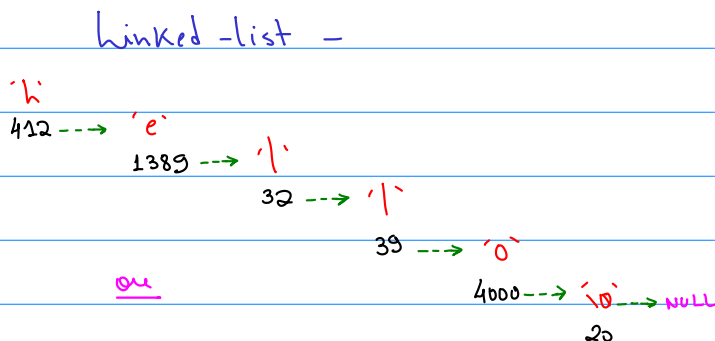
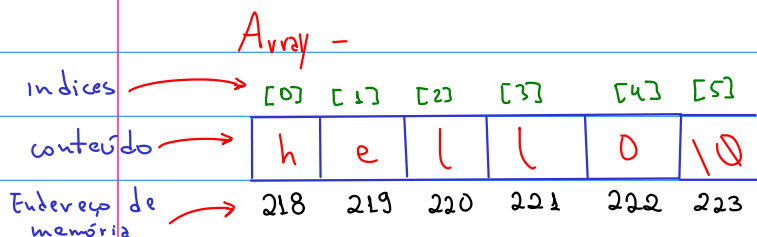
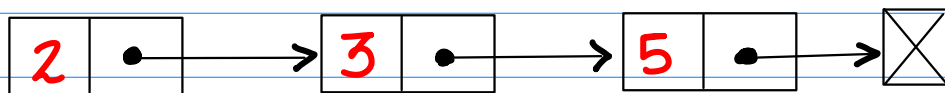
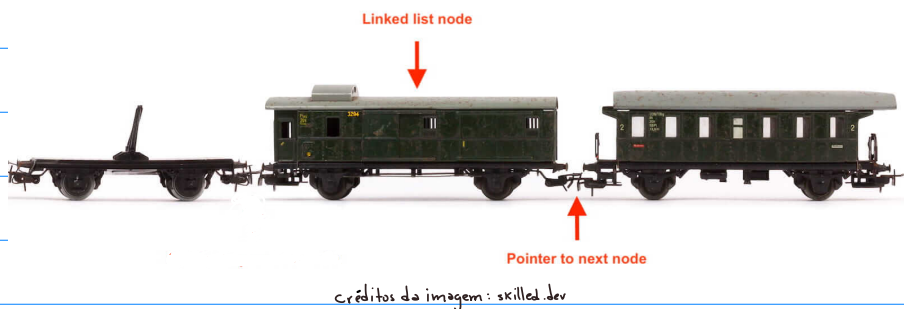
1. Tamanho flexível / distribuído - os dados não precisam ser sequenciais na memória. Podemos adicionar e remover itens em um lista encadeada desde que tenhamos uma maneira de apontar para o próximo No.
2. Rápido para adicionar / remover desde - se nós temos acesso a um No, normalmente o início ou final, nós podemos adicionar ou deletar de forma rápida, pois a única coisa necessária é atualizar um ponteiro

x Desvantagens:

1. Pesquisas lentas - não podemos acessar diretamente os dados (sem chaves/índices). Nós devemos iterar através da lista para poder acessar algum item.

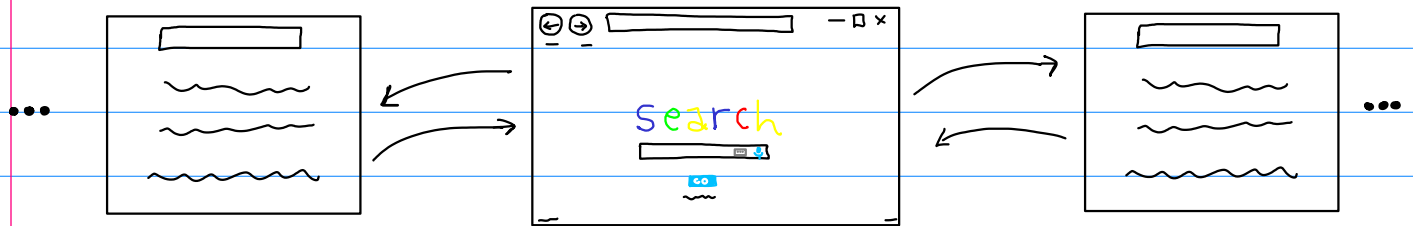
Lista Encadeada - explicação

A maneira mais simples de pensar em uma lista encadeada é imaginar ela como um Trem, onde cada vagão terá seu dado que são os itens ou pessoas que esse vagão contém. Além disso, esse vagão terá uma referência ao próximo vagão, por meio de seu acoplamento..



Curiosidade!

Podemos ver exemplos de listas encadeadas, mais especificamente lista duplamente encadeada nos navegadores de Internet (botões de voltar e avançar) onde cada página conhece aquela que visitou antes e depois



* Big O -

* Adicionar no início - $O(1)$ constante

* Adicionar no meio - $O(n)$ linear

* Adicionar no fim (sem acesso ao fim da lista) - $O(n)$ linear

* Adicionar no fim (com acesso ao fim da lista) - $O(1)$ constante

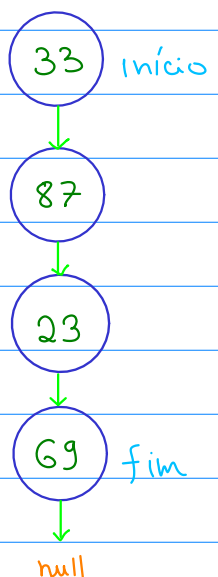
* Acessar algum elemento - $O(n)$ - linear

* Remover no início - $O(1)$ - constante

* Remover no meio - $O(n)$ - linear

* Remover no fim (sem acesso ao fim da lista) - $O(n)$ - linear

* Remover no fim (com acesso ao fim da lista) - $O(1)$ - constante



Por Paulo Henrique Diniz de Lima Alencar.

Referências:

LINKED LISTS. Skilled.dev, 2020. Disponível em: <<https://skilled.dev/course/linked-lists>>. Acesso em: 26 dez. de 2020.