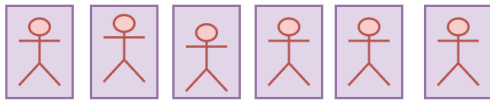
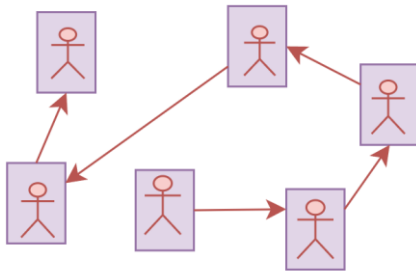


Lista ligada

O vetor guardava os elementos do **array** um do lado do outro na memória:



Já a **lista ligada** não guarda os elementos dessa forma, **ela** simplesmente os deixa **“espalhados”** na memória, porém com **cada elemento** sabendo quem é o **próximo** a ser chamado depois **dele**:

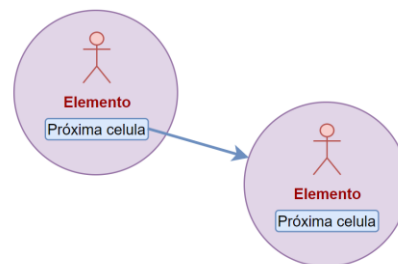


Essa estrutura tem como objetivo tornar rápida a inserção de um elemento no **“meio”** da lista. Com relação a inserir no começo e no fim da lista, também leva um tempo constante. Como essa lista não é indexada, recuperar elementos de uma posição **“x”** pode levar tempo linear.

As células de uma lista ligada

Como um **elemento** de uma lista ligada conhece seu próximo? Através da **célula/nó** que o encapsula. A **célula** é composta do **elemento** que está **nela** e **também de outra célula** (*célula essa que é a próxima*):

```
public class Celula {  
    private Object elemento;  
    private Celula proximo;  
}
```



Como uma lista ligada é formada

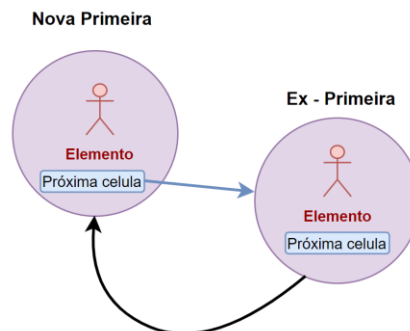
Uma **lista ligada** é um objeto composto por **células**, inicialmente ele é inicializado com a **primeira** e **última** célula com valor vazio:

```
public class ListaLigada {  
    private Celula primeira = null;  
    private Celula ultima = null ;  
    private int totalElementos = 0;  
}
```

Adicionar Elementos na lista ligada

Podemos adicionar um elemento direto no começo da lista ligada, essa operação é simples. Basta que o elemento seja atribuído a primeira célula da lista, setando nele essa **ex-primeira célula** como sua próxima, é como se tivéssemos simplesmente trocando de lugar e fazendo o elemento novo fazer referência para a antiga primeira como sua próxima célula (vale ressaltar que se o total de elementos da lista ligada for 0, então a primeira célula também é a última, já que só tem uma):

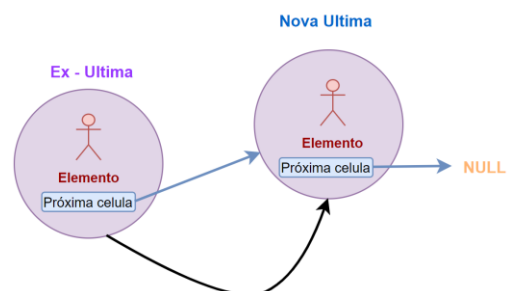
```
public void adicionaNoComeco(Object elemento){  
    this.primeira = new Celula(elemento, this.primeira);  
  
    if(totalElementos == 0){  
        this.ultima = this.primeira;  
    }  
  
    this.totalElementos++;  
}
```



Outra operação simples é adicionar um elemento no final da lista, primeiro vale a pena checar se tem algum elemento na lista ou não.

Depois disso basta criar uma célula nova que vai receber esse elemento e setar a próxima célula dela como null (já que ela vai ser a última e não aponta pra ninguém). Finalmente, antes de trocar as posições da última célula, é necessário fazer a **ex-ultima célula** fazer referência para a nova última célula como sua próxima:

```
public void adiciona(Object elemento){  
    if(this.totalElementos == 0){  
        this.adicionaNoComeco(elemento);  
    }else{  
        Celula novaCelula = new Celula(elemento, proximo: null);  
        this.ultima.setProximo(novaCelula);  
        this.ultima = novaCelula;  
        this.totalElementos++;  
    }  
}
```



Percorrer a lista ligada

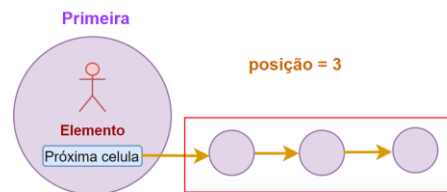
Pra navegar nas células de uma lista ligada precisamos percorrer pelas referências. Nesse caso eu estou percorrendo baseado em uma posição (eu uso a posição para iterar sobre as células) começando da primeira:

```
private Celula pegaCelula(int posicao){
    if(!posicaoOcupada(posicao)){
        throw new IllegalArgumentException("Posição inexistente");
    }

    Celula atual = this.primeira;

    for(int i = 0; i < posicao; i++){
        atual = atual.getProximo();
    }

    return atual;
}
```



Adicionar Elementos no “meio” da lista ligada

Para adicionar um elemento em uma posição específica, a ideia é que essa **nova célula** seja inserida entre **outras células já existentes**.

Pensando nisso a lógica é **pegar a célula que está na posição anterior a essa nova** (pra isso usamos o *pegaCelula* passando a posição da nova célula - 1), agora esse **próximo elemento** da **célula anterior** vai ser o **próximo elemento** da nossa **nova célula** e **nova célula** vai ser o **próximo elemento** da **anterior** (é basicamente encaixar a célula nova entre outras células):

```
public void adiciona(int posicao, Object elemento){
    if(posicao == 0){
        adicionaNoComeco(elemento);
    }else if(posicao == this.totalElementos){
        adiciona(elemento);
    }else{
        Celula anterior = this.pegaCelula(posicao: posicao - 1);
        Celula nova = new Celula(elemento, anterior.getProximo());
        anterior.setProximo(nova);
        this.totalElementos++;
    }
}
```

