

Estrutura de dados do tipo Lista linear

Lista linear

Uma lista linear é uma estrutura de dados que possui as seguintes características:

- A estrutura pode armazenar vários dados
- Os dados armazenados possuem o mesmo tipo de dado
- Os dados estão organizados logicamente de forma sequencial, onde cada dado tem um único predecessor e um único sucessor.
 - Exceção são o primeiro e último dado, que não possuem predecessor e sucessor, respectivamente.

Lista linear

- Exemplos de dados que podem ser armazenados numa lista:
 - Pessoas que estão numa fila de banco
 - Relação de alunos de uma turma
 - Lista de funcionários de uma empresa
- Existe duas categorias de implementação de listas lineares:
 - Implementação estática de lista
 - Implementação dinâmica de lista

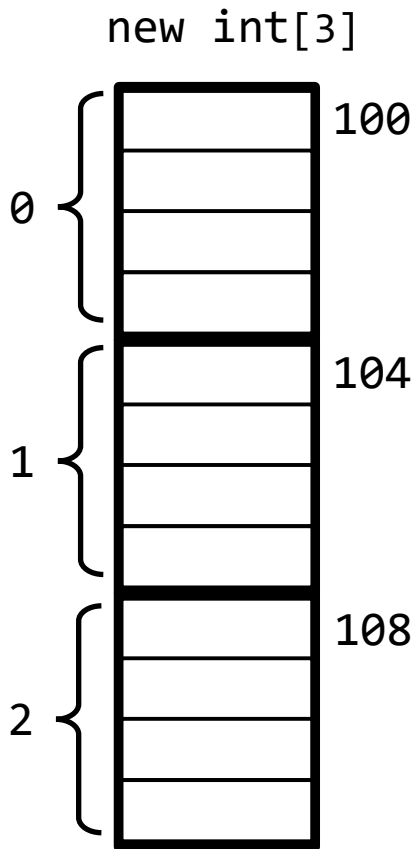
Implementação estática de lista

Característica principal da implementação estática

A **lista linear** de implementação estática utiliza um vetor para armazenar os dados

Características da implementação estática

Sobre o vetor:



- Ocupa espaço contíguo na memória
- Permite acesso randômico aos elementos
 - Posição dos elementos pode ser calculada

$$P(Indice) = \text{Endereço Inicial} + (\text{Indice} \times \text{Tamanho Elemento})$$

- Exemplo:

$$P(2) = 100 + (2 \times 4)$$

$$P(2) = 108$$

- Deve ser dimensionado com número máximo de elementos
- Remoção de elementos ainda consome espaço

Características da implementação estática

Estas são as características de uma lista com implementação estática:

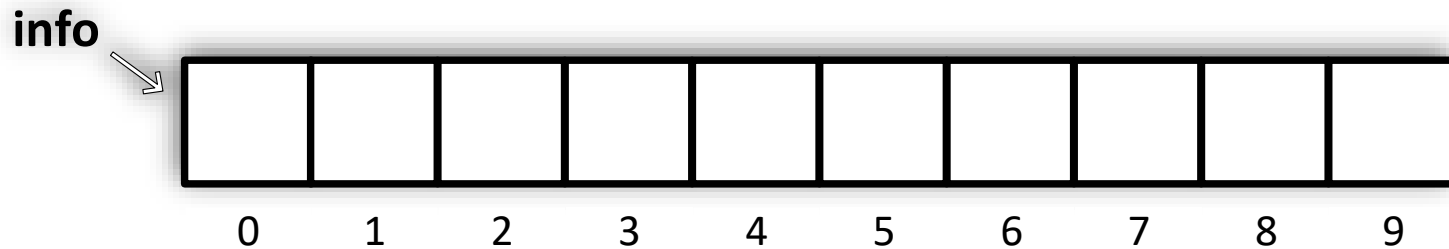
- Utiliza vetor
 - Permite acessar aleatoriamente qualquer elemento
- Inserção ocorre após o último elemento
- Retirada requer reposicionar os elementos
- Quando esgotar a capacidade de armazenamento do vetor ocorre um “pseudo redimensionamento”

Implementação estática de lista

ListaEstatica
<ul style="list-style-type: none">- info : int[]- tamanho : int
<ul style="list-style-type: none">+ ListaEstatica()- redimensionar() : void+ inserir(valor : int) : void+ exibir() : void+ buscar(valor : int) : int+ retirar(valor : int) : void+ estaVazia() : boolean

Criação de uma lista

- A criação da lista consiste em:
 - Criar um vetor com tamanho predefinido. Exemplo: 10 elementos



- Definir que a quantidade de elementos que já foram armazenados na lista é “0” (zero). Usaremos o atributo tamanho.

Algoritmo: criarLista()

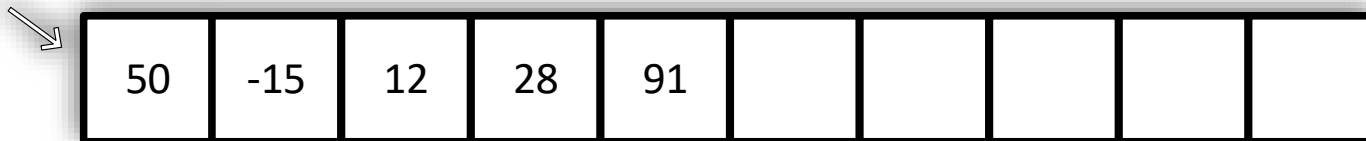
```
info ← new int[10];  
tamanho ← 0;
```

Inclusão de elementos na lista

- Os dados são adicionados ao final da estrutura:

```
lista.inserir(50);  
lista.inserir(-15);  
lista.inserir(12);  
lista.inserir(28);  
lista.inserir(91);
```

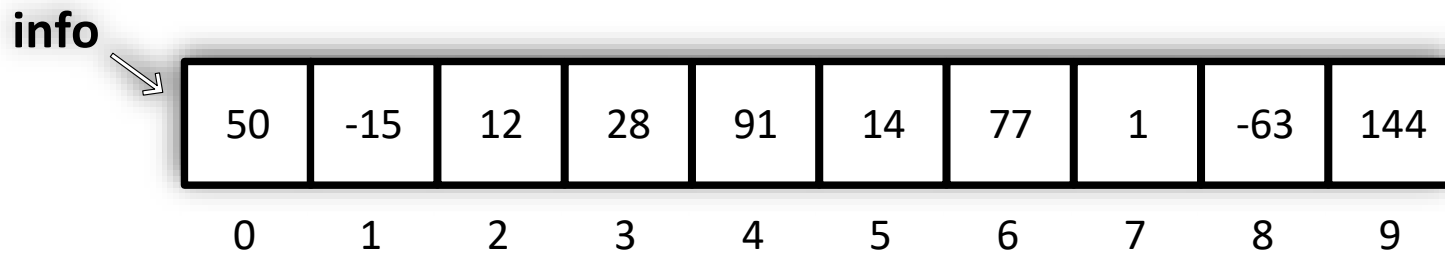
info



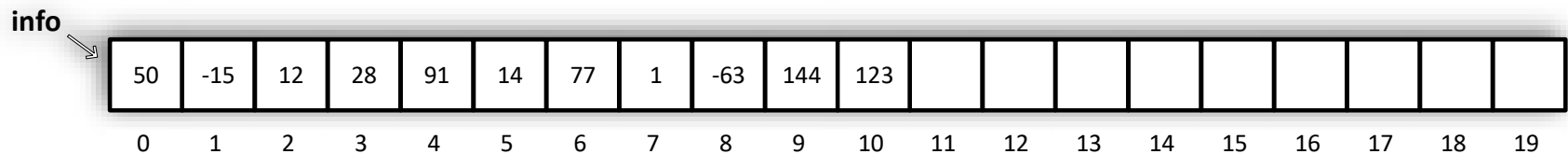
- O atributo `tamanho` é incrementado para indicar quantos dados já foram inseridos e qual a posição que deve ser utilizada para armazenar um novo dado

Inclusão de elementos na lista caso especial

- Considerar a lista constituída dos elementos abaixo



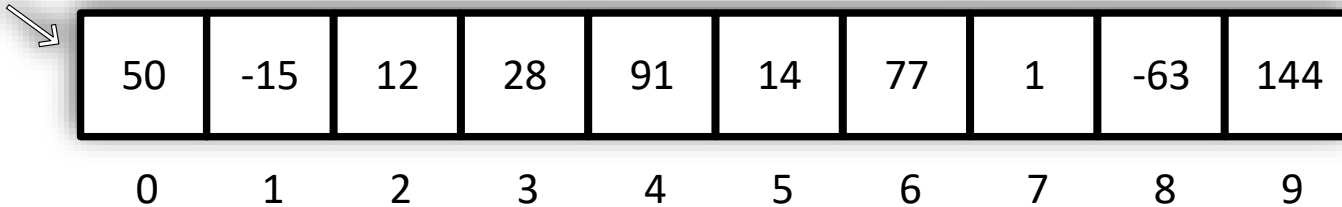
- Em seguida, é executado: `lista.inserir(123);`



Redimensionamento

Dada a lista constituída de:

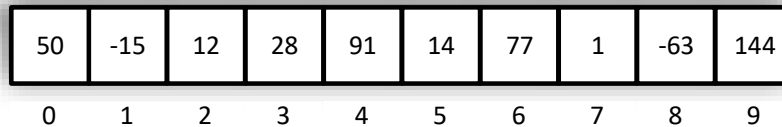
info



50	-15	12	28	91	14	77	1	-63	144
0	1	2	3	4	5	6	7	8	9

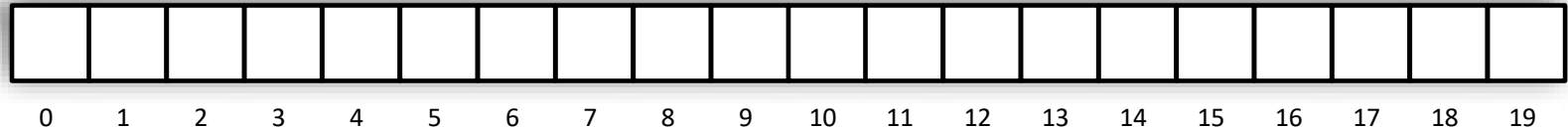
1 – Criar um novo vetor com capacidade maior

info



50	-15	12	28	91	14	77	1	-63	144
0	1	2	3	4	5	6	7	8	9

novo

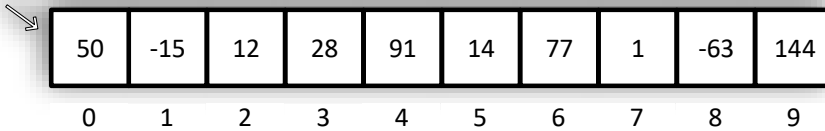


0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Redimensionamento

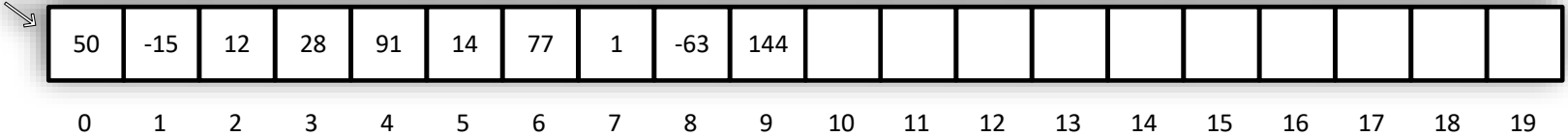
2 – Copiar os dados do vetor original para o novo vetor

info



50	-15	12	28	91	14	77	1	-63	144
0	1	2	3	4	5	6	7	8	9

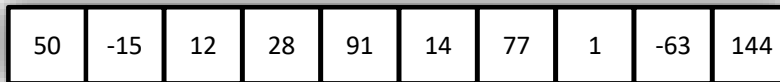
novo



50	-15	12	28	91	14	77	1	-63	144										
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

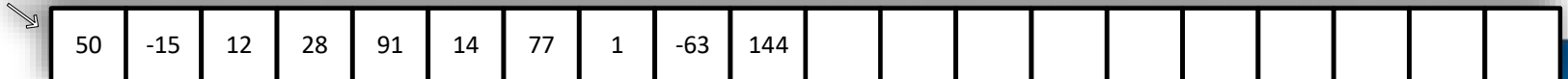
3 – Mudar a referência da variável info, para apontar para o novo vetor

info



50	-15	12	28	91	14	77	1	-63	144
----	-----	----	----	----	----	----	---	-----	-----

novo



50	-15	12	28	91	14	77	1	-63	144										
----	-----	----	----	----	----	----	---	-----	-----	--	--	--	--	--	--	--	--	--	--

Algoritmo de inclusão

Algoritmo: inserir(int valor)

```
se (tamanho = size(info)) então  
    redimensionar();  
fim-se
```

```
info[tamanho] ← valor;  
tamanho ← tamanho + 1;
```

size(info): obtém
o comprimento
do vetor "info"

Algoritmo: redimensionar()

```
int[] novo;
```

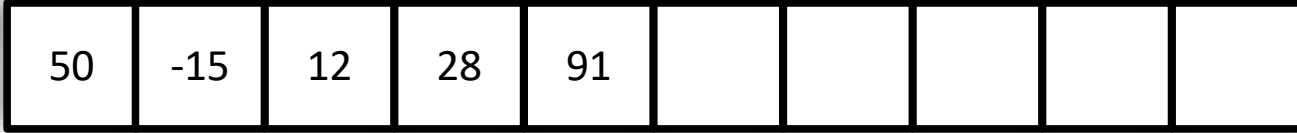
```
novoTamanho ← size(info)+10;  
novo ← new int[novoTamanho];  
para i ← 0 até tamanho-1 faça  
    novo[i] ← info[i];  
fim-para
```

```
info ← novo;
```

Buscar elementos

- O método `buscar()` deve procurar um elemento e retornar a posição encontrada no vetor. Caso não seja encontrado, deve retornar -1.
- Considerar a lista constituída do vetor abaixo:

info



50	-15	12	28	91					
0	1	2	3	4	5	6	7	8	9

O comando `lista.buscar(-15)`

Deve resultar em 1

Retirar elementos

Considerar a lista constituída do vetor abaixo:

info



50	-15	12	28	91	198	-55			
----	-----	----	----	----	-----	-----	--	--	--

Após executar: `lista.retirar(-15)`

info



50	12	28	91	198	-55				
----	----	----	----	-----	-----	--	--	--	--

Os elementos posteriores são deslocados para a esquerda.
A variável tamanho deve ser decrementada.