

TÓPICO 04: VETORES NÃO ORDENADOS

Estrutura de Dados - Professor Ramon Venson - SATC 2026.2

Vetor Não Ordenado

0	1	2	3	4
42	15	8	73	21

Inserir

42	15	8	73	21	
----	----	---	----	----	--



42	15	8	73	21	50
----	----	---	----	----	----

Pesquisar

42	15	8	73	21
----	----	---	----	----



42	15	8	73	21
----	----	---	----	----

Deletar

42	15	8	73	21
----	----	---	----	----



42	15		73	21
----	----	--	----	----

Vetores Não Ordenados

Estrutura de dados **linear** onde cada elemento possui um **índice** único.

Características

- **Tamanho fixo:** O tamanho do vetor é definido no momento da criação.
- **Acesso direto:** É possível acessar qualquer elemento diretamente pelo seu índice.
- **Não ordenado:** Os elementos não seguem uma ordem específica, o que pode tornar algumas operações menos eficientes.

Inicialização de um Vetor Não Ordenado

```
classe VetorNaoOrdenado:  
  capacidade: inteiro  
  ultima_posicao: inteiro  
  vetor: array de inteiro  
  
  construtor(capacidade):  
    self.capacidade <- capacidade  
    self.ultima_posicao <- -1  
    self.vetor <- novo array de inteiro com tamanho capacidade
```

Imprimindo os Elementos do Vetor

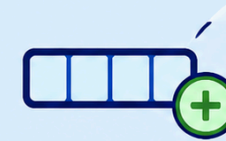
Para imprimir todos os valores de um vetor não ordenado, vamos percorrer cada um dos elementos do vetor até a posição final

```
imprimir():  
    para cada i no intervalo de 0 até ultima_posicao faça:  
        escreva(vetor[i])
```

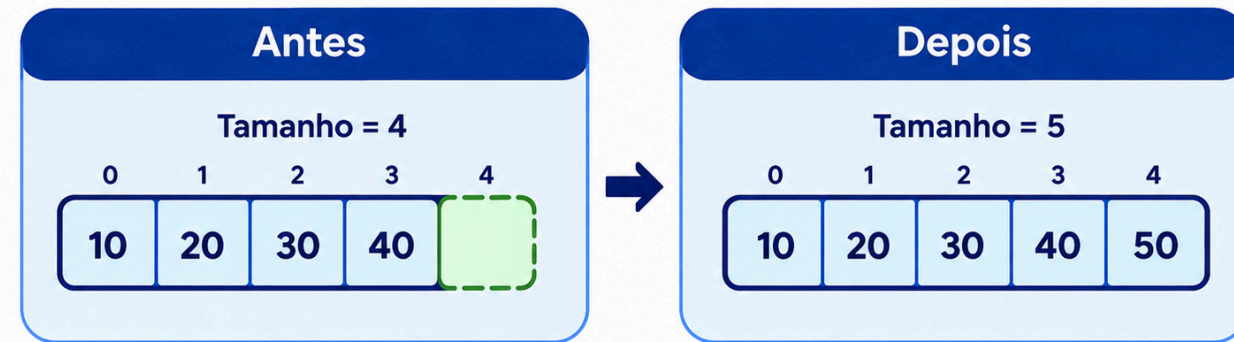
Inserção de Elementos

Inserir na **primeira posição livre** do vetor.

O algoritmo conhece o **tamanho máximo** do vetor e a **quantidade de elementos** já inseridos.



Vetor Não Ordenado Inserir



Pseudocódigo

```
SE tamanho < capacidade  
  vetor[tamanho] ← elemento  
  tamanho ← tamanho + 1
```

Pseudocódigo para Inserção

```
inserir(valor):  
    se ultima_posicao == capacidade - 1 então  
        retorne "Vetor cheio"  
    senao:  
        ultima_posicao <- ultima_posicao + 1  
        vetor[ultima_posicao] <- valor
```



Vetor Não Ordenado Pesquisar

Busca

Elemento = 73

0	1	2	3	4
42	15	8	73	21

Pseudocódigo



```
PARA i de 0 até tamanho - 1  
  SE vetor[i] = elemento  
    RETORNE i  
RETORNE -1
```



Como o vetor não é ordenado, precisamos verificar cada elemento, do início ao fim, até encontrar o valor desejado.

Pesquisa de Elementos

Em um vetor não ordenado, a pesquisa é feita **percorrendo todos os elementos** até encontrar o elemento desejado ou chegar ao final do vetor.

- **Melhor caso:** O elemento está na primeira posição.
- **Pior caso:** O elemento está na última posição ou não está presente no vetor.

Pseudocódigo para Pesquisa

```
pesquisar(valor):  
    para cada i no intervalo de ultima_posicao + 1 faça:  
        se vetor[i] = valor então  
            retorne i  
    retorne -1 // Elemento não encontrado
```

Remoção de Elementos

Para remover um elemento, o elemento a ser removido é removido e os elementos restantes são deslocados para preencher o espaço vazio.

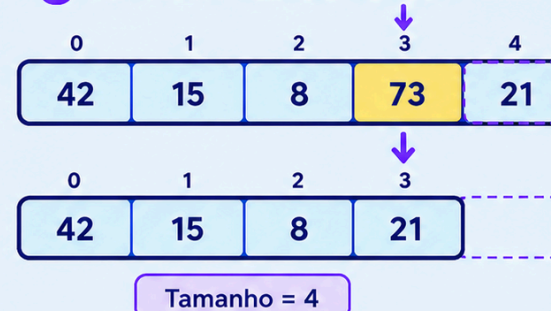
- **Melhor caso:** O elemento a ser removido está na última posição.
- **Pior caso:** O elemento a ser removido está na primeira posição.



Vetor Não Ordenado Remover

Como Remover

- 1 Pesquisar o elemento.
- 2 Se encontrado, substituir pelo último elemento do vetor.
- 3 Diminuir o tamanho do vetor.



Pseudocódigo



```
PARA i de 0 até tamanho - 1
    SE vetor[i] = elemento
        vetor[i] = vetor[tamanho - 1]
        tamanho = tamanho - 1
    RETORNE verdadeiro

RETORNE falso
```



Na remoção em vetor não ordenado, não preservamos a ordem dos elementos. A operação é eficiente, pois evita deslocamentos.

Pseudocódigo para Remoção

```
excluir(valor):  
    posicao <- pesquisar(valor)  
    se posicao = -1 então  
        retorne -1  
    senao:  
        para cada i no intervalo de posicao até ultima_posicao faça:  
            vetor[i] <- vetor[i + 1]  
        ultima_posicao <- ultima_posicao - 1
```