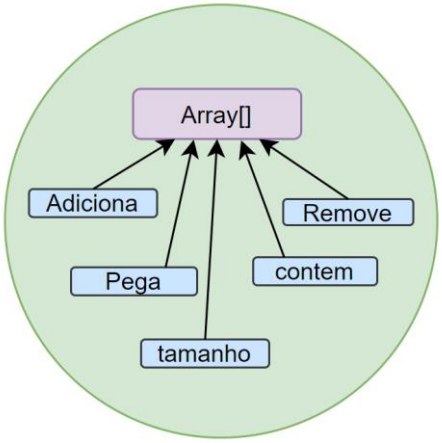
Vetor

O **vetor** é uma **estrutura de dados** que encapsula um **Array**, **ele** funciona guardando os elementos de forma sequencial *(na ordem que foram inseridos) e* dispondo de **diversos métodos** que fazem a interação com esse **array** *(como inserir, remover, pegar, deletar e etc)*:



Vetores tem uma **inserção de elementos rápida**, porem remoções, consultas e inserções através de um índice costumam ser menos performáticas por serem operações O(N). Aliás, o **ArrayList** é uma implementação de vetor.

O Array dentro do Vetor

É isso, é simplesmente um **array** dentro de uma classe que representa o **vetor**:

```
public class Vetor {

    private Aluno[] alunos = new Aluno[100];
```

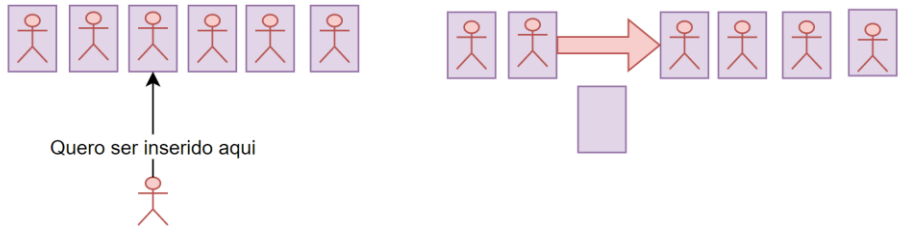
Adicionando elementos no Vetor

Existem dois **métodos** de inserção no vetor, o primeiro deles é uma **inserção direta** no **Array** que usa uma **variável** para manter a **velocidade de inserção constante**.

Basicamente essa **variável** começa em 0 *(primeira posição do **Array**)* e vai sendo **incrementada a cada inserção**, dessa maneira não importa quantos elementos o **vetor** já tem, ele sempre vai inserir na posição que está vazia, mantendo uma boa performance:

```java
public class Vetor {

    private Aluno[] alunos = new Aluno[100];

    private int totalAlunos = 0;

    public void adiciona(Aluno aluno){
        this.garanteEspaco();

        this.alunos[totalAlunos] = aluno;
        totalAlunos++;
    }
```

O outro método de inserção é feito com base em uma posição específica do **Array**, ele é um pouco mais complexo pois precisamos **alterar a estrutura do Array**. O que acontece é que **dada a posição específica**, temos que **"empurrar"** todos os elementos a partir dela para a direita, para abrir espaço para o novo elemento que vai ser inserido:



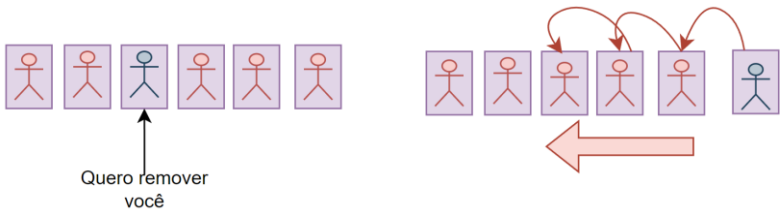
No código essa **"empurração"** funciona como um loop, onde **começamos a iterar pelo fim do Array**, **até chegarmos na posição que foi passada** e **decrementando a cada loop**.

O que vai acontecer é que **todo elemento depois da posição passada vai ser inserido uma "casa a frente" no array**, assim **liberando o espaço para o novo elemento entrar**:

```java
public void adiciona(int posicao, Aluno aluno){

    this.garanteEspaco();

    if(!posicaoValida(posicao)){
        throw new IllegalArgumentException("Posição inválida");
    }

    for(int i = totalAlunos - 1 ; i >= posicao; i-=1 ){
        alunos[i + 1] = alunos[i];
    }

    alunos[posicao] = aluno;
    totalAlunos++;
}
```

Remover elementos do Vetor

A remoção é **parecida com a inserção**, o que acontece é que agora t**odos os elementos do array a partir da posição informada vão ser empurrados para a esquerda**, e o elemento da remoção vai indo pro final do array:



Empurrando os elementos a esquerda a gente joga o **elemento indesejado** pro final do **array**.

No código isso funciona com um loop, **começamos a iteração na posição que queremos remover**, p**aramos quando for menor que o tamanho total de alunos** *(quando é menor, não vamos iterar pela última posição do array, o lugar onde o elemento indesejado está, então ele fica de fora/removido),* **e incrementando a cada loop.**

O que vai acontecer é que **todo elemento a partir da posição do elemento indesejado vai ser movida para a esquerda**, deixando o elemento indesejado por último e então simplesmente ficando de fora da iteração já que delimitados com "<" apenas:

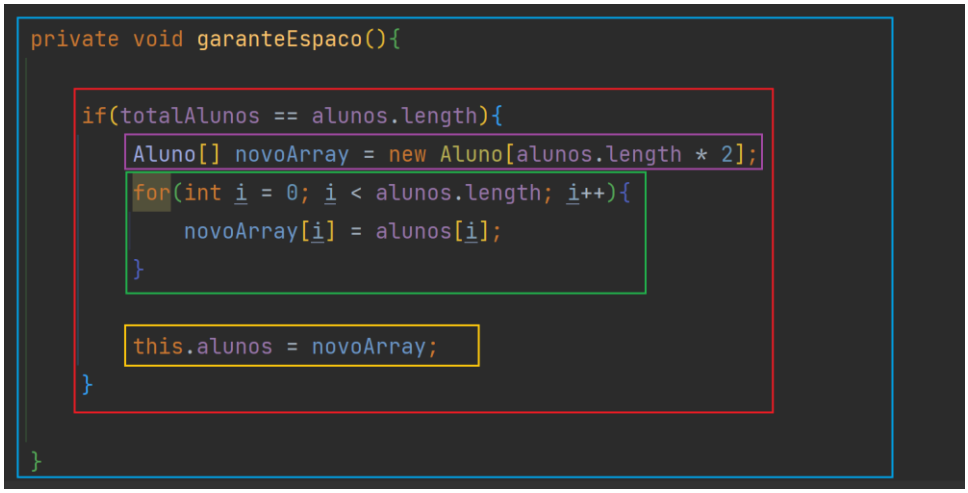
```
public void remove(int posicao){

    for (int i = posicao; i < this.totalAlunos - 1 ; i++){
        this.alunos[i] = this.alunos[i+1];
    }

    totalAlunos --;
}
```

Validações e escalonamento do Vetor

Um vetor pode ter diversos controles internos (ele deve na verdade), como por exemplo **validar se uma posição informada está dentro dos limites do Array:**

```
private boolean posicaoValida(int posicao){
    return posicao >= 0 && posicao <= totalAlunos;
}
```

Ou **garantir espaço** caso o **Array** tenha atingido seu limite pré-estabelecido na inicialização do **Vetor.** Então **se eu alcançar esse limite** vou simplesmente **criar um novo array** com o dobro do tamanho do original, **copiar os elementos do array antigo para ele** e **fazer o vetor apontar para ele:**


private void garanteEspaco(){
if(totalAlunos == alunos.length){
Aluno[] novoArray = new Aluno[alunos.length * 2];
for(int i = 0; i < alunos.length; i++){
novoArray[i] = alunos[i];
}
this.alunos = novoArray;
}
}