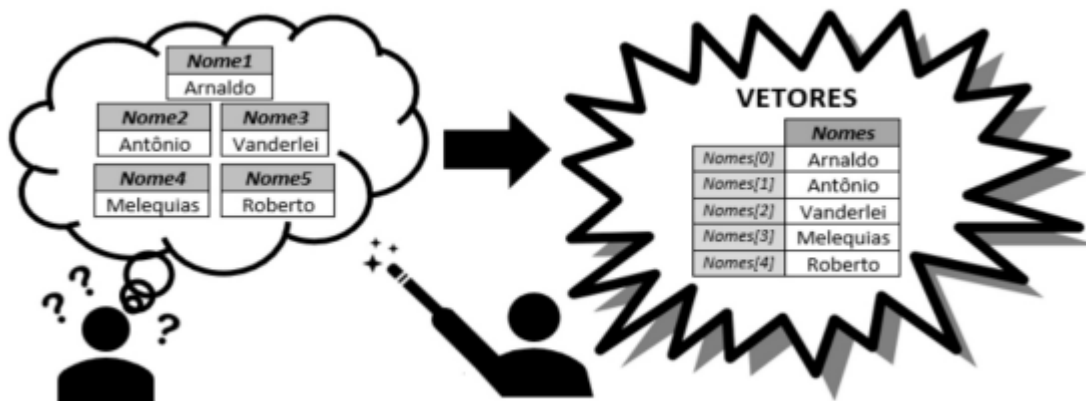


DESENVOLVIMENTO DE SISTEMAS I:

AGENDA 10

VETORES – São “casas” que armazenam vetores. São **matrizes unidimensionais**. São variáveis **indexadas (possuem índice)**. Na programação é utilizado **vetores** no lugar de **variáveis** convencionais pela praticidade (facilita acesso aos dados e economiza espaço na memória do computador). Desse modo, o processo de leitura e gravação de um dado em uma variável é simplificado graças às estruturas de repetição. A vantagem de se usar vetores em programação: facilitar acesso aos dados economizar espaço na memória do computador.

- Vetor / Array: é um tipo de variável (espaço na memória);
- Aloca um grupo de informações do **mesmo tipo**;
- Armazena quantidade **fixa** de dados (10 ints ou 10 Strings ou 20 doubles) é necessário determinar **quantidade e tipo**;



NOMENCLATURA DE UM VETOR – Nome [Índice / Posição / Endereço] == c[0], c[1], c[2], c[3] e se lê: c na posição zero, c na posição 1, ..., c na posição 3. **Índice** de um vetor = capacidade vetor - 1

A **primeira** posição de um vetor sempre será [0] zero.

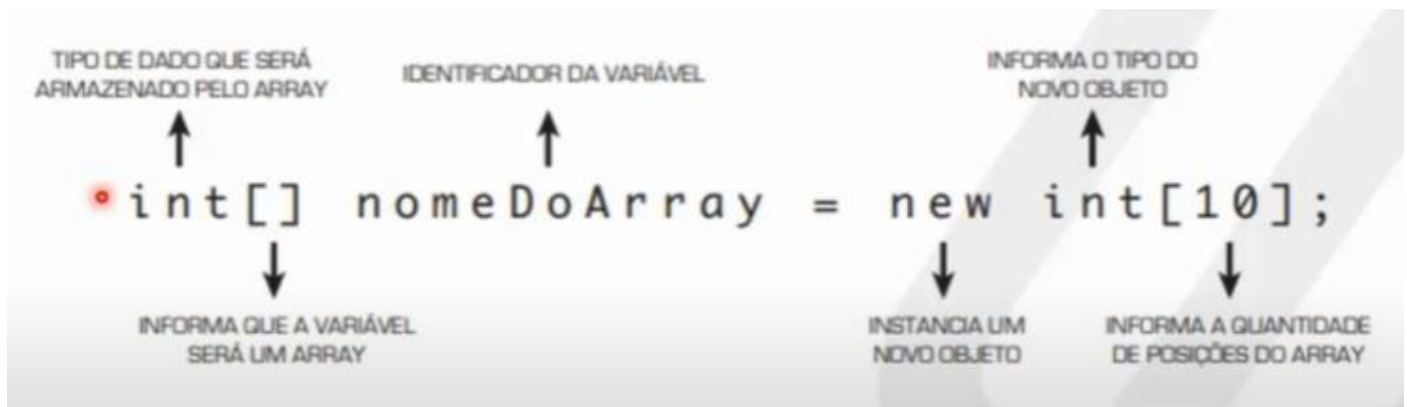
Variáveis **simples** só recebem um dado por vez.

Vetores permitem trabalhar com **agrupamento de várias informações** dentro de uma mesma variável tornando a programação mais rápida e eficiente.

EXEMPLO PRÁTICO – Ao comprar uma barra de chocolate, provavelmente você compra uma barra inteira ao invés de um único quadrado (uma unidade) pois além de ser mais barato é mais prático (apenas uma compra, desembalar apenas uma vez, armazena mais quadrados por embalagem).

Trabalhar com vetores é praticamente a mesma coisa: a ideia é juntar **vários valores** em uma única variável, cada uma com seu espaço, similar à barra de chocolate para facilitar o acesso aos dados. O processo de leitura e gravação de um dado em uma variável torna-se simplificado graças às estruturas de repetição.

VETOR EM JAVA – `int [] nomeDoArray = new int[10];`



MODIFICANDO UM VETOR – Modificamos um vetor pela posição

```
int[] numeros = new int[100];  
numeros[0] = 136;  
numeros[99] = 17;
```

Na prática, sair escrevendo cada um dos vetores não é usual, pois daria muito trabalho escrever 100 vetores ou mais, por isso é usado estruturas de repetição.

OUTRA MANEIRA DE INICIALIZAR ARRAY – `int [] números = new int[] { 100, 87};`

Entrada de dados foi manual, neste caso posição [0] = 100, posição [1] = 87,...

PERCORRENDO UM VETOR COM ESTRUTURAS DE REPETIÇÃO – Para percorrer um vetor é possível utilizar uma estrutura de repetição “for”. A melhor estrutura de repetição para percorrer um vetor é a estrutura for pois é finita, sabe-se onde começa e onde termina.

```
int[] numeros = new int[100];  
for(int i = 0; i < 100; i++) {  
    numeros[i] = i;  
}
```

CAPACIDADE MÁXIMA DO VETOR – atributo `length`: `nomeDoArray.length`

```
void imprimeArray(int[] numeros) {  
    for(int i = 0; i < numeros.length; i++) {  
        System.out.println(numeros[i]);  
    }  
}
```

→ https://www.youtube.com/watch?v=CUrcGp8NdIk&ab_channel=CursoDSCursoDS (Desenvolvimento de Sistemas 1 Agenda 09 – Vetores)

→ https://www.youtube.com/watch?v=sgvfbApx7Xw&ab_channel=GEEaDCPS (Informática – Módulo I – Agenda 16)

→ https://www.youtube.com/watch?v=7i3YfiLy1vE&ab_channel=GEEaDCPS (Informática - Módulo I - Agenda 15 - Lógica de Programação Vetores e Matrizes)

→ https://www.youtube.com/watch?v=mV4GGC5UARQ&ab_channel=GEEaDCPS (Informática - Módulo I - Revisão (Agendas 13, 14 e 15))

➔ https://www.youtube.com/watch?v=hkE9WrjpAAk&ab_channel=CursoemV%C3%ADdeo (Matrizes - Curso de Algoritmos #15 – Gustavo Guanabara)