STANDARD TEMPLATE LIBRARY

VECTOR

Lucas Bucior - 2017

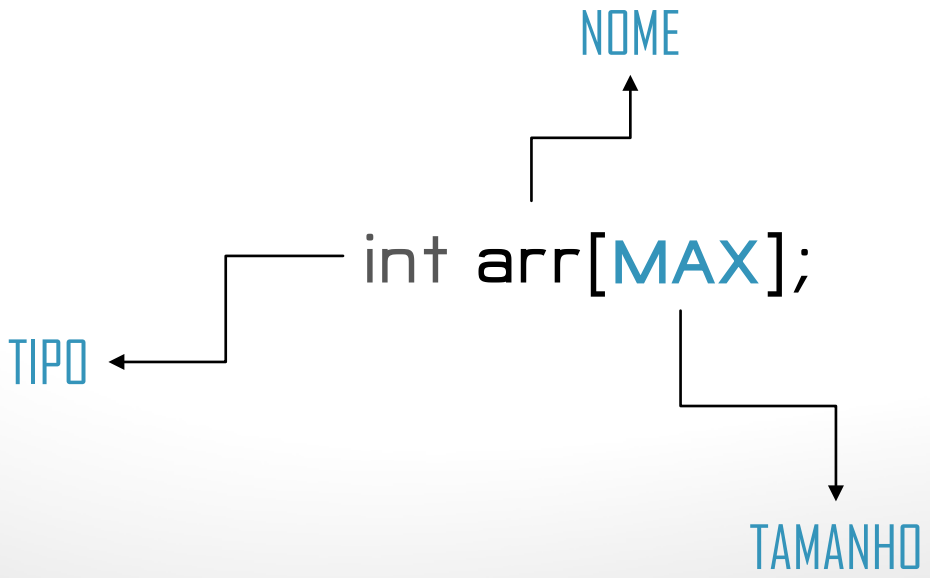
ROTEIRO

- Arrays
- Vector
  - ✓ push_back
  - ✓ size

ARRAYS

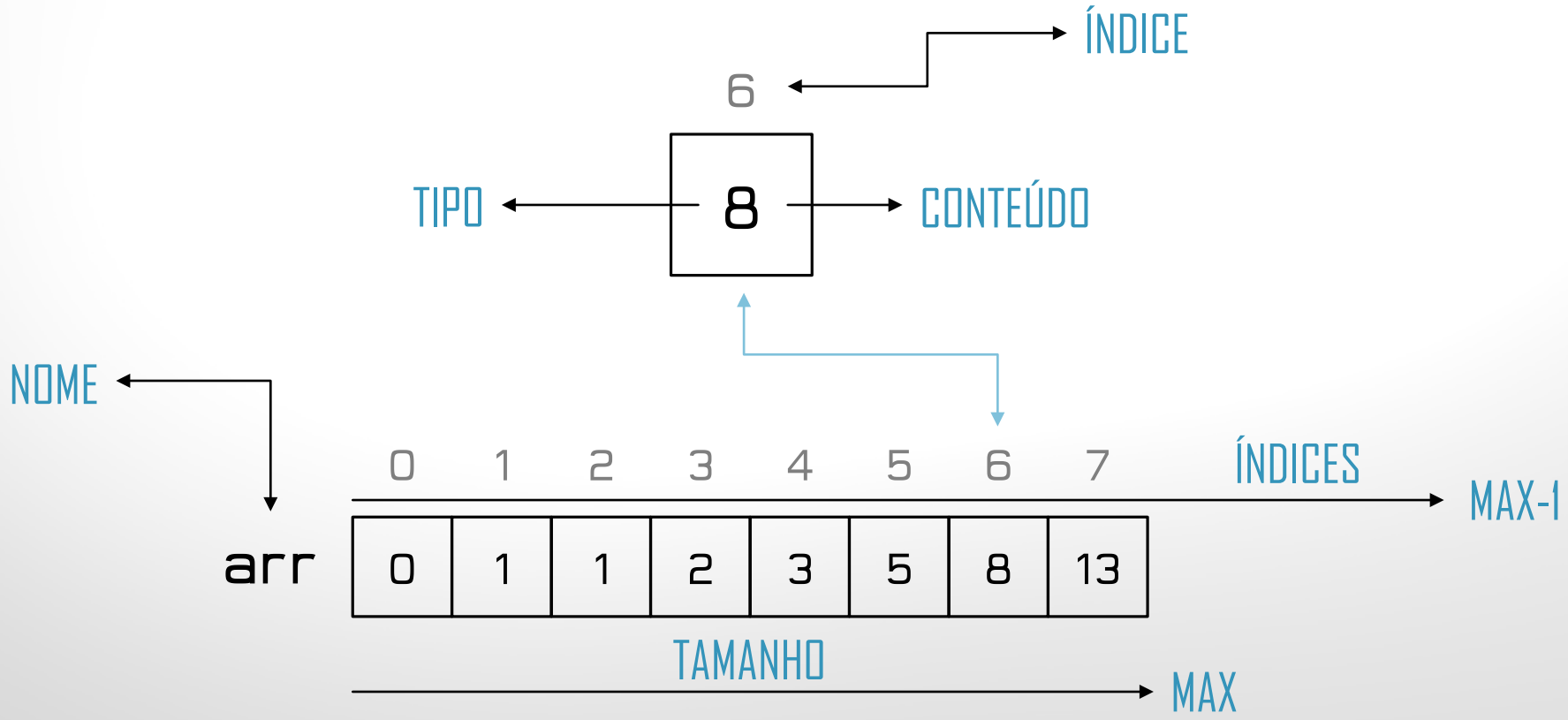
- Array de inteiros com tamanho MAX:

```
#define MAX 8
```



ARRAYS

- O que representa:



ARRAYS

- Definindo um valor para o array:

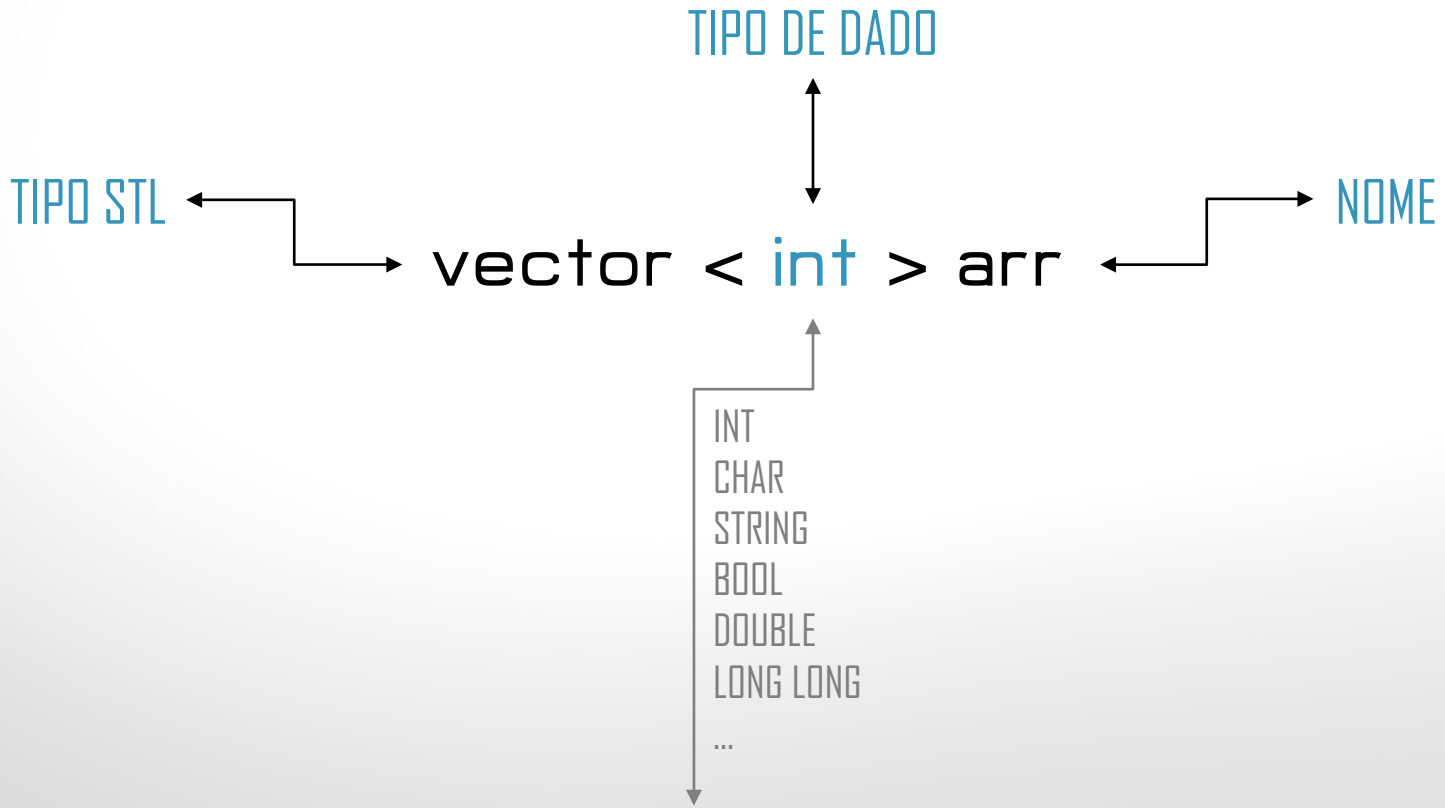
```
memset (arr, 0, sizeof (arr))
```

- Ordenando o array:

```
sort (arr, arr+N)
```

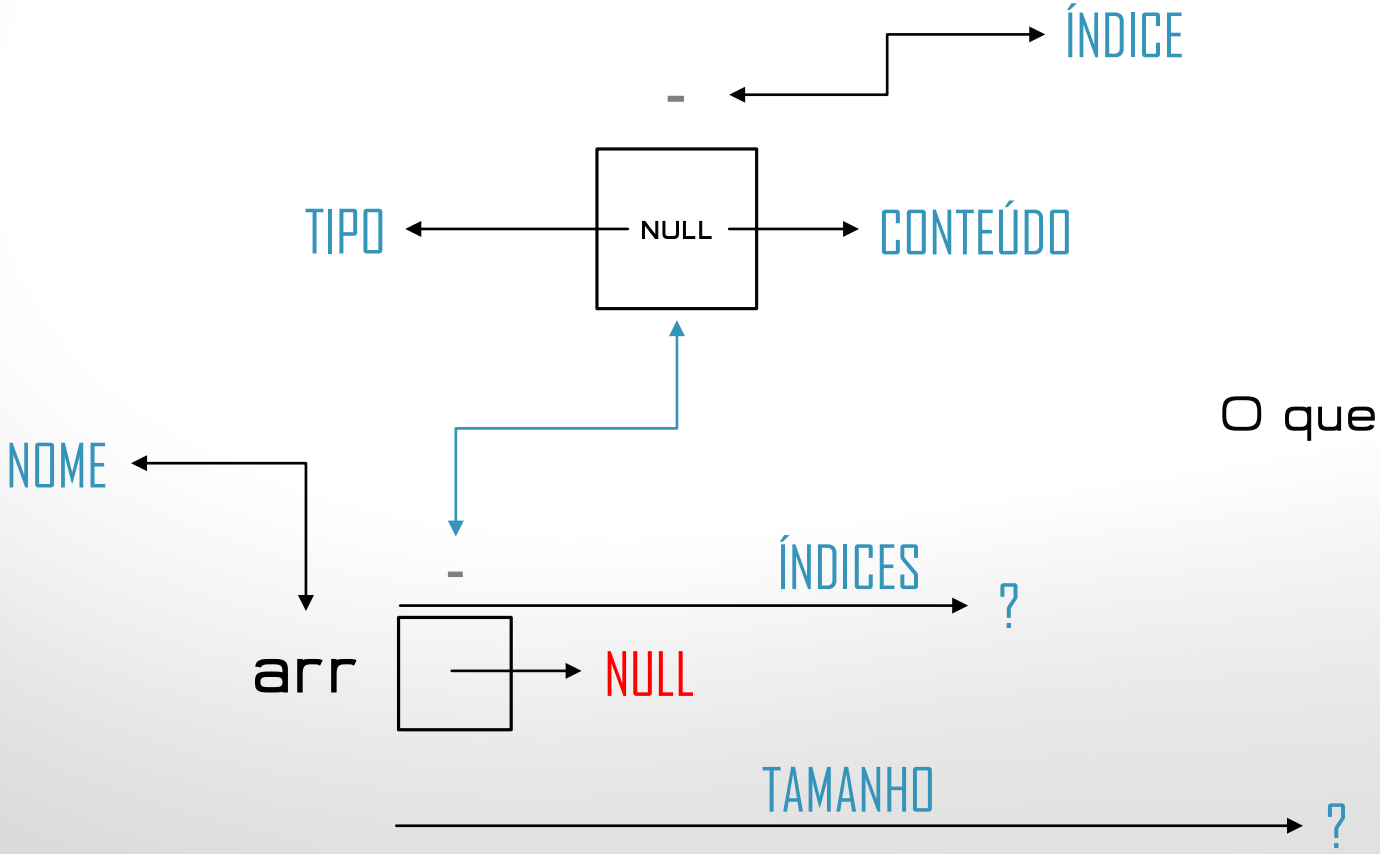
VECTOR

- Criando um vector:



VECTOR

- Como fica depois de criado:



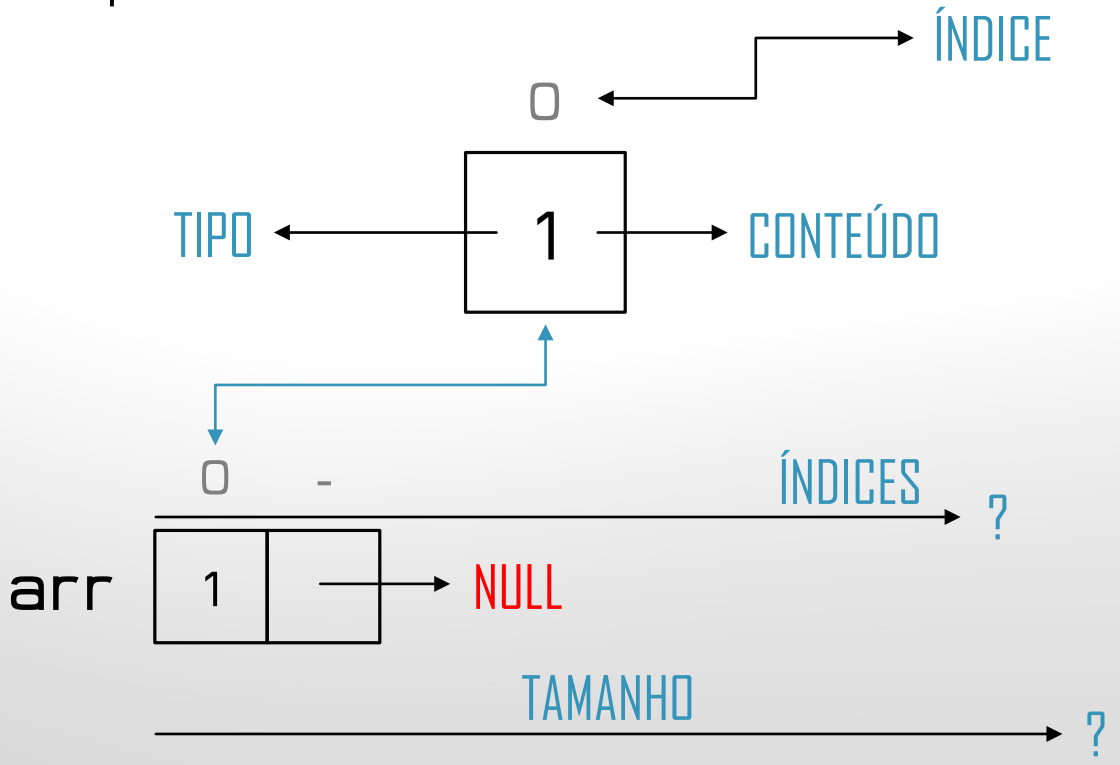
O que tem no **arr**?

VECTOR

- Inserindo o valor 1:

```
arr.push_back(1)
```

- Como fica depois de inserir 1:

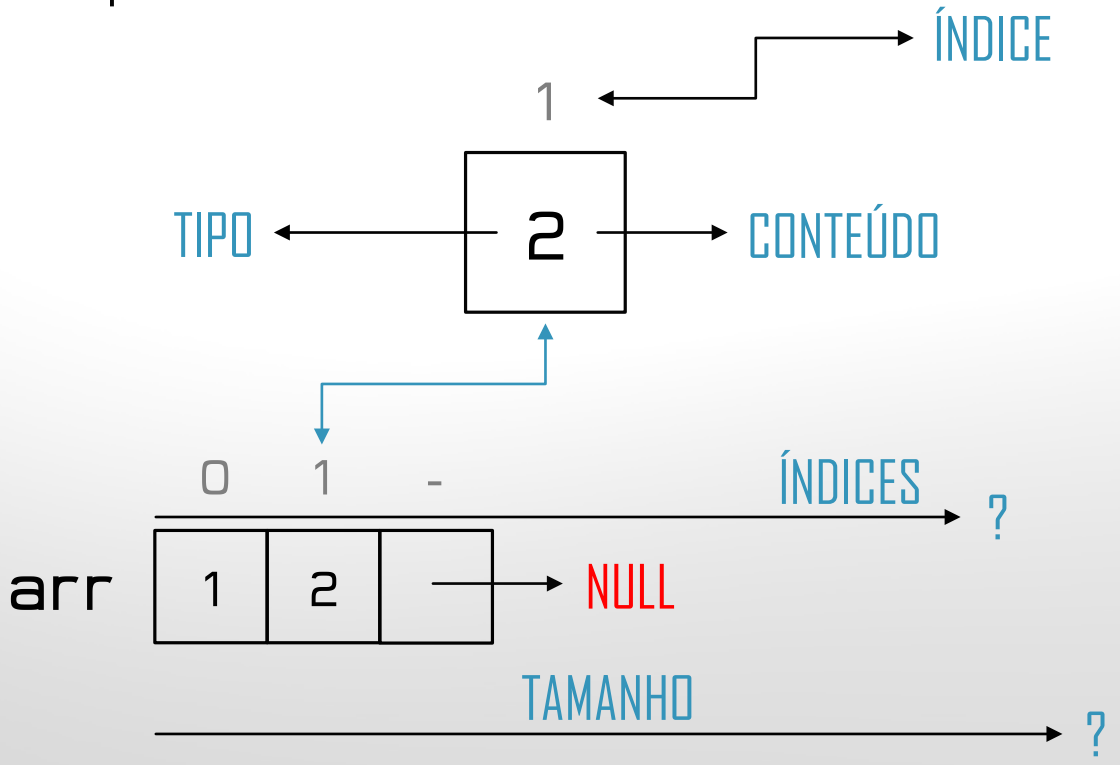


VECTOR

- Inserindo o valor 2:

```
arr.push_back(2)
```

- Como fica depois de inserir 2:

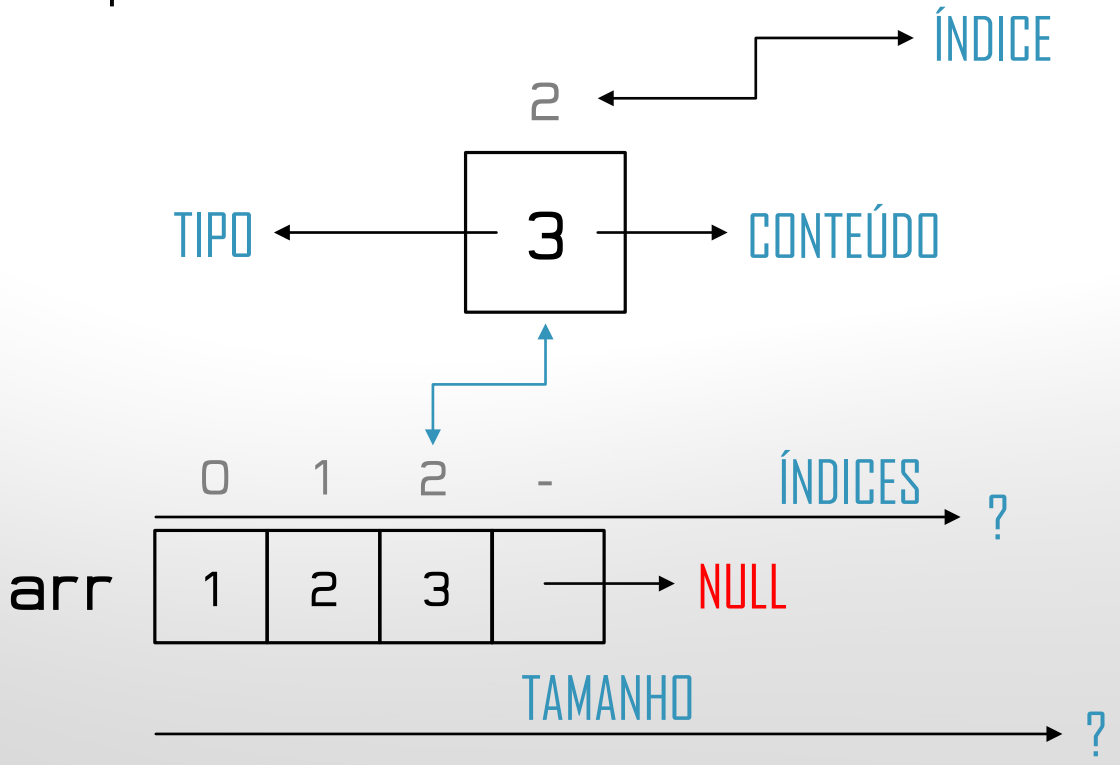


VECTOR

- Inserindo o valor 3:

```
arr.push_back(3)
```

- Como fica depois de inserir 3:

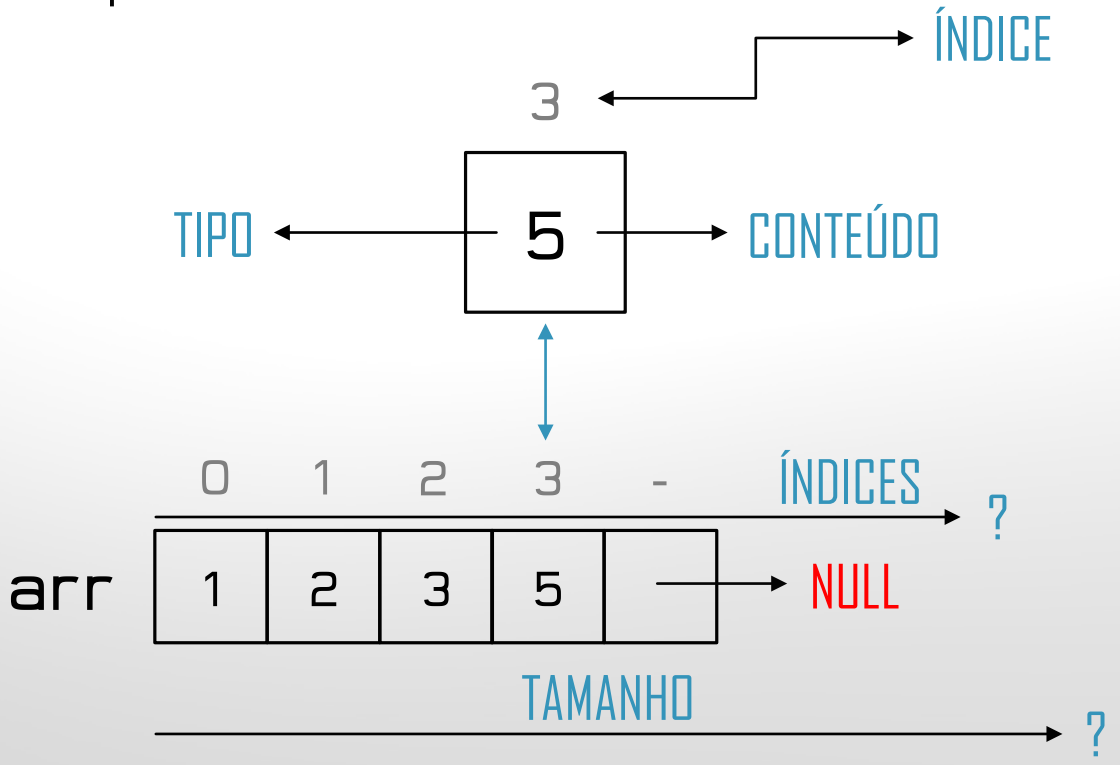


VECTOR

- Inserindo o valor 5:

```
arr.push_back(5)
```

- Como fica depois de inserir 5:



VECTOR

- Exemplo *push_back():*

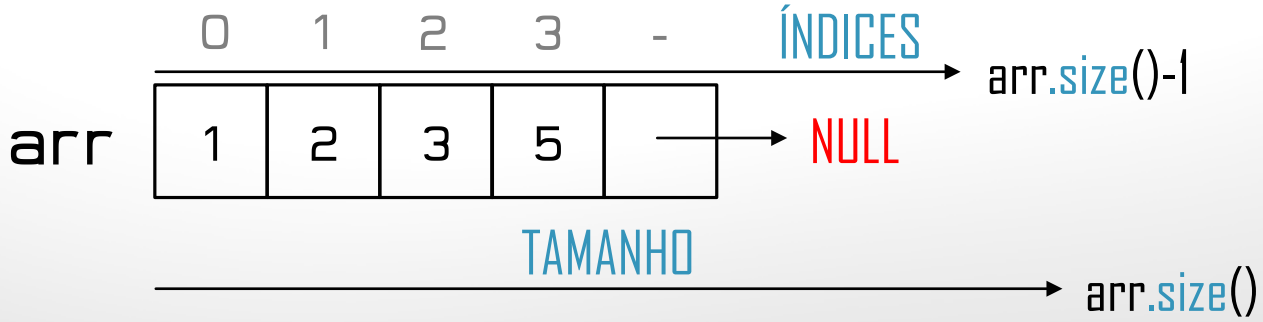
```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  int main() {
6
7      vector<int> arr; //criando o vector
8
9      arr.push_back(1); // inserindo valores
10     arr.push_back(2);
11     arr.push_back(3);
12     arr.push_back(5);
13
14     return 0;
15 }
16
17
18
```

VECTOR

- Para saber o tamanho:

```
arr.size()
```

- Exemplo:



VECTOR

- Exemplo *size()*:

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  int main() {
6
7      vector<int> arr; //criando o vector
8
9      arr.push_back(1); // inserindo valores
10     arr.push_back(2);
11     arr.push_back(3);
12     arr.push_back(5);
13
14     for (int i = 0; i < arr.size(); i++) {    // percorrendo o vector
15         printf("%d\n", arr[i]);  // imprimindo o respectivo valor
16     }
17
18     return 0;
19 }
20
```