



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA
Campus João Pessoa



CNPq
Conselho Nacional de Desenvolvimento
Científico e Tecnológico

Projeto Olímpico de Programação

[C++] STL: Container Vector

STL: Container Vector

- Container dinâmico mais utilizado da STL;
- Permite acesso aleatório aos elementos armazenados;
- **Complexidade das operações básicas:**

Operação	Complexidade
Inserção e remoção no final	$O(1)$
Inserção e remoção no meio	$O(n)$
Busca	$O(n)$

- Iteradores são válidos enquanto não houver alteração no vector;
- Inserção no código:

```
#include <vector>
```

- Declaração:

```
vector<string> rep;
```

STL: Container Vector

assign

```
#include <iostream>
#include <vector>

using namespace std;

void print(vector<int> v);

int main(){
    vector<int> v1, v2, v3;
    int numeros[] = {10, 20, 30, 40};

    v1.assign(4, 10);
    print(v1);

    v2.assign(numeros, numeros + 4);
    print(v2);

    v3.assign(v2.begin(), v2.end());
    print(v3);

    return 0;
}

void print(vector<int> v){
    for(vector<int>::iterator it = v.begin(); it != v.end(); it++){
        cout << *it << " ";
    }
    cout << endl;
}
```



STL: Container Vector

at

```
#include <iostream>
#include <vector>


using namespace std;

int main() {
    vector<int> v;
    int numeros[] = {10, 20, 30, 40};

    v.assign(numeros, numeros + 4);

    for (int i = 0; i < v.size(); ++i) {
        cout << i << " ";
        cout << v[i] << " ";
        cout << v.at(i) << endl;
    }

    return 0;
}
```



0	10	10
1	20	20
2	30	30
3	40	40

STL: Container Vector

clear e erase

```
#include <iostream>
#include <vector>

using namespace std;

void print(vector<int> v);

int main(){
    vector<int> v;
    int numeros[] = {10, 20, 30, 40, 50, 60};

    v.assign(numeros, numeros + 6);

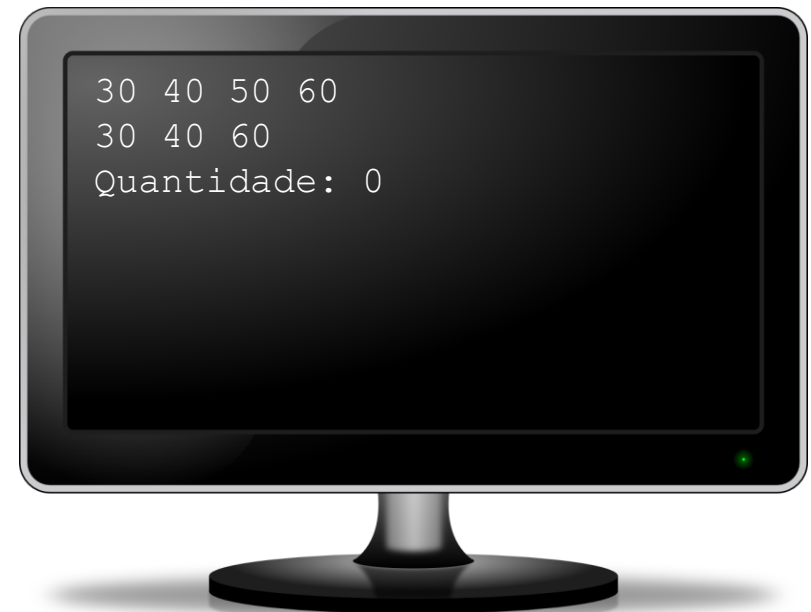
    v.erase(v.begin(), v.begin() + 2);
    print(v);

    v.erase(v.begin() + 2);
    print(v);

    v.clear();
    cout << "Quantidade: " << v.size();

    return 0;
}

void print(vector<int> v){
    for (int i = 0; i < v.size(); ++i){
        cout << v.at(i) << " ";
    }
    cout << endl;
}
```



STL: Container Vector

insert

```
#include <iostream>
#include <vector>

using namespace std;

void print(vector<int> v);

int main() {
    vector<int> v1, v2;
    int numeros[] = {30, 40, 50};

    v1.insert(v1.begin(), 10);
    print(v1);

    v1.insert(v1.begin(), 2, 20);
    print(v1);

    v2.insert(v2.begin(), v1.begin(), v1.end());
    print(v2);

    v2.insert(v2.end(), numeros, numeros + 3);
    print(v2);

    return 0;
}

void print(vector<int> v) {
    for (int i = 0; i < v.size(); ++i) {
        cout << v.at(i) << " ";
    }
    cout << endl;
}
```



STL: Container Vector

push_back

```
#include <iostream>
#include <vector>

using namespace std;

void print(vector<int> v);

int main(){
    vector<int> v;

    v.push_back(10);
    v.push_back(20);
    v.push_back(30);
    v.push_back(40);
    print(v);

    return 0;
}

void print(vector<int> v){
    for (int i = 0; i < v.size(); ++i){
        cout << v.at(i) << " ";
    }
    cout << endl;
}
```



STL: Container Vector

pop_back

```
#include <iostream>
#include <vector>

using namespace std;

int main(){
    vector<int> v;
    int numeros[] = {10, 20, 30, 40};

    v.insert(v.begin(), numeros, numeros + 4);

    while (!v.empty()){
        cout << v.back() << endl;
        v.pop_back();
    }
    return 0;
}
```



STL: Container Vector

Operadores Relacionais

```
#include <iostream>
#include <vector>

using namespace std;

int main(){
    vector<int> v1, v2;

    v1.push_back(10);
    v1.push_back(20);

    v2.push_back(10);
    v2.push_back(20);

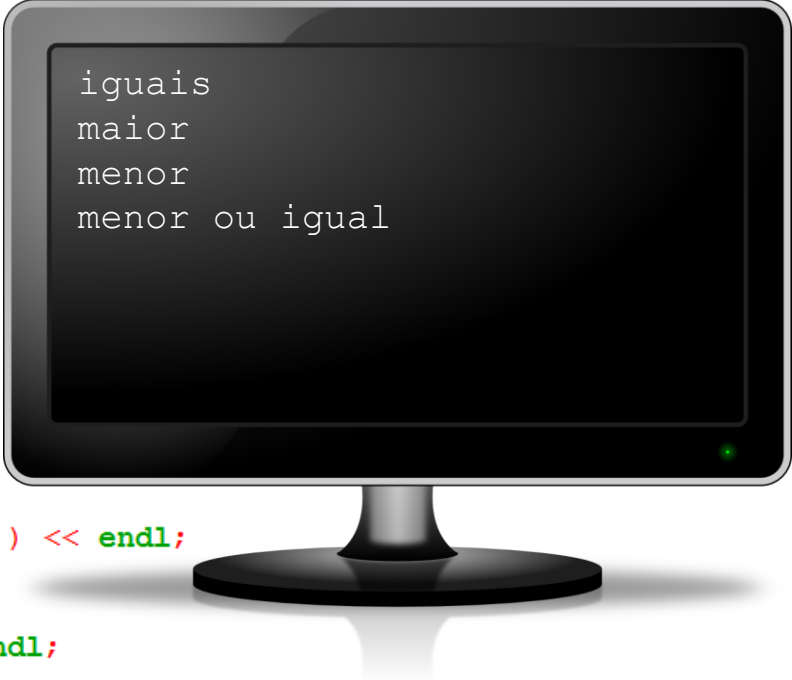
    cout << ((v1 == v2) ? ("iguais") : ("diferentes")) << endl;

    v1[0] = 20;
    cout << ((v1 < v2) ? ("menor") : ("maior")) << endl;

    v2.at(0) = 30;
    cout << ((v1 < v2) ? ("menor") : ("maior")) << endl;

    v1[0] = 30;
    cout << ((v1 <= v2) ? ("menor ou igual") : ("maior")) << endl;

    return 0;
}
```



iguais
maior
menor
menor ou igual

STL: Container Vector

Desafio!

- Desenvolva um programa para obter 1000 (mil) palavras. O programa deverá calcular e exibir qual(is) palavra(s) possui(em) maior frequência.