



# ESTRUTURAS DA STL





# VECTOR

# VECTOR - MÉTODOS PRINCIPAIS

---

- **push\_back(valor):** método que adiciona um novo elemento (no final) do vector.
- **pop\_back(valor):** método que retira um elemento (último) do vector.
- **size():** método que retorna o tamanho do vector.
- **resize(tam):** método que redefine o vector pelo tamanho, aumentando-o ou diminuindo-o.
- **clear():** método que limpa o vector, apaga todos os elementos.

# VECTOR - EXEMPLO DE CÓDIGO

```
1  #include<bits/stdc++.h>
2  #define __ ios_base::sync_with_stdio(0);
3
4  using namespace std;
5
6  int main(){_
7      vector<int> V; //Criação da variável V do tipo vector
8
9      V.push_back(10); //Adiciono o elemento 10 no final do vector V
10     V.push_back(7); //Adiciono o elemento 7 no final do vector V
11     V.push_back(21); //Adiciono o elemento 21 no final do vector V
12     V.push_back(12); //Adiciono o elemento 21 no final do vector V
13
14     for(int i=0;i<(int)V.size();i++){ //Passo por cada elemento do vector V
15         cout << V[i] << endl; //Imprimir cada elemento do vector V
16     }
17
18     V.pop_back(); //Apaga o ultimo elemento do vector V.
19     V.clear(); //Remove todos os elementos do vector V.
20
21     return 0;
22 }
23
```

# VECTOR - ORDENAÇÃO

```
1  #include<bits/stdc++.h>
2  #define __ ios_base::sync_with_stdio(0);
3
4  using namespace std;
5
6  int main(){_
7      vector<int> V;
8
9      V.push_back(5);
10     V.push_back(10);
11     V.push_back(2);
12     V.push_back(4);
13
14     sort(V.begin(), V.end()); //Ordena todo o vector V, deixando os elementos na seguinte ordem {2, 4, 5, 10}.
15
16     for(int i=0;i<(int)V.size();i++){
17         cout << V[i] << endl;
18     }
19
20     return 0;
21 }
22
```



# QUEUE



# QUEUE - MÉTODOS PRINCIPAIS

---

- **push(valor):** método que adiciona um novo elemento (no final) da queue.
- **pop():** método que retira um elemento (primeiro) da queue.
- **front():** método que retorna o primeiro elemento (frente) da queue.
- **empty():** método que retorna se a queue está vazia.
- **size():** método que retorna a quantidade de elementos na queue.

# QUEUE - EXEMPLO DE CÓDIGO

```
1  #include<bits/stdc++.h>
2  #define _ ios_base::sync_with_stdio(0);
3
4  using namespace std;
5
6  int main(){_
7      queue<int> fila; //Declaro uma variavel fila do tipo queue.
8
9      fila.push(1); //Adiciono o elemento 1 no final da fila.
10     fila.push(2); //Adiciono o elemento 2 no final da fila.
11     fila.push(3); //Adiciono o elemento 3 no final da fila.
12     fila.push(4); //Adiciono o elemento 4 no final da fila.
13
14     while(!fila.empty()){ //Enquanto a fila não está vazia.
15         cout << fila.front() << endl; //Imprimo quem está na frente.
16         fila.pop(); //E o retiro da fila, dando a vez para o próximo.
17     }
18
19     return 0;
20 }
21
```





# SET



# SET - MÉTODOS PRINCIPAIS

---

- **insert(valor):** método que insere um novo elemento no set.
- **find(elemento):** método que serve para buscar o elemento no set.
- **erase(valor):** método que retira um elemento do set.
- **clear():** método que limpa o set, apaga todos os elementos.
- **size():** método que retorna o tamanho do set.
- **begin():** método que retorna um ponteiro para o início do set.
- **end():** método que retorna um ponteiro para o fim do set.

# SET - EXEMPLO DE CÓDIGO

```
1  #include<bits/stdc++.h>
2  #define _ ios_base::sync_with_stdio(0);
3
4  using namespace std;
5
6  int main(){_
7      set<int> S; //Cria uma variável do tipo set para armazenar números inteiros
8
9      S.insert(7); //Adiciona o elemento 10 no set
10     S.insert(10); //Adiciona o elemento 10 no set
11     S.insert(3); //Adiciona o elemento 3 no set
12     S.erase(7); //Apaga o elemento 7 do set
13
14     if(S.find(3) != S.end()){ //Se 3 está no conjunto
15         cout << 3 << " esta no set" << endl;
16     }
17
18     cout << "Elementos do set (" << S.size() << " elementos):" << endl;
19     //Pega a referencia do primeiro elemento do set e vai até a referencia do ultimo elemento
20     for(set<int>::iterator it=S.begin(); it!=S.end(); ++it){
21         cout << *it << endl; //Imprime cada elemento do set
22     }
23
24     S.clear(); //Apaga todos os elementos do set
25
26     return 0;
27 }
28
```



# MAP



# MAP - MÉTODOS PRINCIPAIS

---

- **insert(chave,valor):** método que insere um novo valor no map pela chave.
- **find(chave):** método que busca um elemento no map pela dada chave.
- **erase(chave):** método que remove um elemento de acordo com a chave dele.
- **clear():** limpa o map, apaga todos os elementos.
- **size():** método que retorna a quantidade de elementos no map.
- **begin():** método que retorna um ponteiro para o início do map.
- **end():** método que retorna um ponteiro para o fim do map.

# MAP - EXEMPLO DE CÓDIGO

```
1  #include<bits/stdc++.h>
2  #define _ ios_base::sync_with_stdio(0);
3
4  using namespace std;
5
6  int main(){_
7      map<string, int> M; //Cria uma variável do tipo map que mapeia strings em int
8
9      M["Joao"] = 1; //Relacionando o valor 1 à chave "Joao"
10     M["Alana"] = 10; //Relacionando o valor 10 à chave "Alana"
11     M.insert(make_pair("Rodrigo", 9)); //Relacionando o valor 10 à chave "Rodrigo"
12
13     if(M.find("Joao") != M.end()){ //Verifica se existe algum elemento com a chave "Joao".
14         cout << "Chave: Joao\nValor: " << M["Joao"] << endl; //Se tiver, imprime na tela.
15     }
16
17     M.erase("Joao"); //Apaga o elemento que possui a chave "Joao".
18
19     //O iterator recebe a referencia do primeiro elemento do map e vai até o último.
20     for (map<string,int>::iterator it=M.begin(); it!=M.end(); ++it){
21         cout << "(" << it->first << ", " << it->second << ")" << endl; //Imprime a chave e o valor
22     }
23
24     M.clear(); //Todos os elementos do map são apagados.
25
26     return 0;
27 }
28
```

# BIBLIOGRAFIA

---

CodCad: Vector. Site: <http://www.codcad.com/lesson/17>.

CodCad: Queue. Site: <http://www.codcad.com/lesson/19>.

CodCad: Set. Site: <http://www.codcad.com/lesson/24>.

CodCad: Map. Site: <http://www.codcad.com/lesson/25>.