



Iteradores

Por Ariel Parra

$[\Gamma_{\alpha} = \Omega 5]$

¿Qué es un iterador?

Los iteradores son un objeto de c++ que generalizan **punteros** de manera que permite acceder y trabajar con contenedores de c++ y rangos de estos declarado en el header `<iterator>`. Hay cinco principales tipos de iteradores: Forward, Bidirectional, Output, Input y Random access. Los primeros dos se usan como una alternativa a los índices de los ciclos for.

Declaración:

```
vector<int>::iterator it; // iterador de tipo vector STL  
string::iterator it; //iterador de tipo string STL
```

Funciones STL que devuelven iteradores

```
vector<int> v = {1, 2, 3};  
auto it = v.begin(); // it apunta al primer elemento &v[0]  
auto it = v.end(); // it apunta al ultimo elemento &v[v.size()]  
  
auto it = v.rbegin(); // it apunta al ultimo elemento &v[v.size()]  
auto it = v.rend(); // it apunta al primer elemento &v[0]  
  
auto it = next(v.begin(), 2); // it apunta al tercer elemento &v[2]  
auto it1 = prev(v.end(), 1); // it1 apunta al último elemento &v[v.size() - 1]
```

Funciones STL que reciben iteradores

```
#include <algorithm>
auto [minIt, maxIt] = minmax_element(vec.begin(), vec.end());
if (minIt != vec.end() && maxIt != vec.end()) {
    cout << "El valor mínimo es " << *minIt << endl;
    cout << "El valor máximo es " << *maxIt << endl;
}

vec.insert(vec.begin(), value); // insertar `value` en iterador

vector<int> subVec(vec.begin() + 1, vec.end() - 1); //subvectores
vector<int> copyVec(subVec.size());
copy(subVec.begin(), subVec.end(), copyVec.begin());

vector<int> vec1 = {1, 2, 3};
vector<int> vec2 = {4, 5, 6};
swap(vec1, vec2); // ahora vec1= vec2 y biceversa
reverse(vec.begin(), vec.end());
```

Ciclos y rangos con iteradores

Ejemplo de un ciclo `for` regular:

```
vector<int> numeros = {1, 2, 3, 4, 5};  
for(size_t i=0; i<numeros.size(); ++i){  
    cout<<i<<" ";  
}
```

Iterador implícito en un `for` de rango (Forward)

```
vector<int> numeros = {1, 2, 3, 4, 5};  
for(int numero : numeros){  
    cout<<numero<<" ";  
}
```

Iteradores explicitos de vectores STD (Bidirectional)

```
vector<int> numeros = {1, 2, 3, 4, 5};  
for(vector<int>::iterator it=numeros.begin(); it!=numeros.end(); ++it){  
    cout << *it << " ";  
}
```

```
vector<int> numeros = {1, 2, 3, 4, 5};  
for(auto it=numeros.begin(); it!=numeros.end(); ++it){  
    cout << *it << " ";  
}
```

```
for(auto it = lista.end(); it != lista.begin();) {  
    --it; // recorrido inverso  
    std::cout << *it << " ";  
}
```

Problemas

- **1919A** Wallet Exchange ↗
- **146A** Lucky Ticket ↗

Referencias

- Singh, M. (2023). *Iterators in C++ STL*. Recuperado de <https://www.geeksforgeeks.org/iterators-c-stl/> ↗
- GeeksforGeeks. (2024). *Introduction to Iterators in C++*. Recuperado de <https://www.geeksforgeeks.org/introduction-iterators-c/> ↗
- cppreference. (s.f.). *Iterator library*. Recuperado de <https://en.cppreference.com/w/cpp/iterator> ↗