

```
1 #include <stack>
2 #include <queue>
3 #include <iostream>
4
5 using namespace std;
6
7 template <typename T>
8
9 class stack2{
10     private:
11         queue<T> colas[2];
12     public:
13         //Vacio
14         bool empty() const{
15             return this->colas[0].empty() && this->colas[1].empty();
16         }
17         //Tamaño
18         int size() const{
19             return max(this->colas[0].size(), this->colas[1].size());
20         }
21         //Top
22         const T& top() const{
23             if(this->colas[0].empty()){
24
25                 return this->colas[1].front();
26
27             }else if(this->colas[1].empty()){
28
29                 return this->colas[0].front();
30
31             }
32
33         }
34         //Push
35         void push(const T & nuevo){
36             int i=-1,j;
37             if(this->colas[0].empty()){
38                 i=1;
39                 j=0;
40                 this->colas[0].push(nuevo);
41             }else if(this->colas[1].empty()){
42                 {
43                     i=0;
44                     j=1;
45                     this->colas[1].push(nuevo);
46                 }else
47                 {
48                     this->colas[0].push(nuevo);
49                 }
50                 if(i!=-1){
51                     while(!this->colas[i].empty()){
52                         this->colas[j].push(this->colas[i].front());
53                         this->colas[i].pop();
54                     }
55                 }
56                 //TIENE UN ORDEN DE EFICIENCIA DE N
57             }
58         //Pop
59         void pop(){
60             if(this->colas[0].empty()){
```

```
61
62         this->colas[1].pop();
63
64     }else if(this->colas[1].empty()){
65
66         this->colas[0].pop();
67
68     }
69 }
70 //TIENE UN ORDEN DE EFICIENCIA DE 1
71 };
72
73 int main(){
74     stack<int> una;
75
76     una.push(5);
77     una.push(7);
78     una.push(3);
79     una.push(5);
80     una.push(8);
81     una.push(3);
82
83     while(!una.empty())
84     {
85         cout<<una.top()<<" ";
86         una.pop();
87     }
88     cout << endl;
89     stack2<int> otro;
90
91     otro.push(5);
92     otro.push(7);
93     otro.push(3);
94     otro.push(5);
95     otro.push(8);
96     otro.push(3);
97     cout << otro.size() << endl;
98
99     while(!otro.empty())
100    {
101        cout<<otro.top()<<" ";
102        otro.pop();
103    }
104 }
```