

```
1 namespace DAMLib
2 {
3     public class Stack<T>
4     {
5         private T[] _stackArray;
6
7
8         public bool IsEmpty => _stackArray.Length == 0;
9         public int Count
10        {
11            get
12            {
13                if (_stackArray == null)
14                    return 0;
15                else
16                    return _stackArray.Length;
17            }
18        }
19
20
21
22
23        public Stack()
24        {
25            _stackArray = new T[0];
26        }
27
28
29        // Funcion que introduce un elemento generico en el Stack
30        public void Push(T element)
31        {
32            if (element == null)
33                return;
34
35            int count = _stackArray.Length;
36
37            T[] stackResult = new T[count + 1];
38
39            for (int i = 0; i < count; i++)
40            {
41                stackResult[i] = _stackArray[i];
42            }
43
44            stackResult[count] = element;
45            _stackArray = stackResult;
46        }
47
48        // Funcion que extrae un elemento generico del Stack.
49        public T? Pop()
50        {
51            int count = _stackArray.Length;
52
53            T result = _stackArray[count - 1];
```

```
54         if (result == null)
55             return default(T);
56
57         T[] stackResult = new T[count - 1];
58         for(int i = 0; i < count - 1; i++)
59         {
60             stackResult[i] = _stackArray[i];
61         }
62
63         _stackArray = stackResult;
64
65         return result;
66     }
67
68     // Funcion que devuelve el elemento superior del Stack
69     public T Top()
70     {
71         T result = _stackArray[_stackArray.Length - 1];
72
73         if (result == null)
74             return default(T);
75
76         return result;
77     }
78
79     public void Clear()
80     {
81         _stackArray = new T[0];
82     }
83
84     public override string ToString()
85     {
86         string result = "";
87         int count = 0;
88
89         foreach (T element in _stackArray)
90         {
91             result += $"El elemento {count} del Stack es: {element} \n";
92             count++;
93         }
94
95         return result;
96     }
97 }
98 }
```