```
namespace DAMLib
{
    public class Stack<T>
    {
        private T[] _stackArray;

        public bool IsEmpty => _stackArray.Length == 0;
        public int Count
        {
            get
            {
                if (_stackArray == null)
                    return 0;
                else
                    return _stackArray.Length;
            }
        }

        public Stack()
        {
            _stackArray = new T[0];
        }

        // Funcion que introduce un elemento generico en el Stack.
        public void Push(T element)
        {
            if (element == null)
                return;

            int count = _stackArray.Length;

            T[] stackResult = new T[count + 1];

            for (int i = 0; i < count; i++)
            {
                stackResult[i] = _stackArray[i];
            }

           stackResult[count] = element;
            _stackArray = stackResult;
        }

        // Funcion que extrae un elemento generico del Stack.
        public T? Pop()
        {
            if (_stackArray == null)
                return default(T);

            int count = _stackArray.Length;

            T result = _stackArray[count - 1];
            T[] stackResult = new T[count - 1];

            for(int i = 0; i < count - 1; i++)
            {
                stackResult[i] = _stackArray[i];
            }

            _stackArray = stackResult;

            return result;
        }

        // Funcion que devuelve el elemento superior del Stack.
        public T? Top()
        {
            if (_stackArray == null)
                return default(T);

            T result = _stackArray[_stackArray.Length - 1];

            return result;
        }

        public T[] Clone()
        {
            int size = _stackArray.Length;
            T[] clone = new T[size];
            for (int i = 0; i < size; i++)
            {
                clone[i] = _stackArray[i];
            }
            return clone;
        }

        public void Clear()
        {
            _stackArray = new T[0];
        }

        public override string ToString()
        {
            string result = "";
            int count = 0;

            foreach (T element in _stackArray)
            {
                result += $"El elemento {count} del Stack es: {element} \n";
                count++;
            }

            return result;
        }
    }
}
```