

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN

Algoritmización y Estructuras de Datos

Profesor:

Adilson G. López

Grupo: 2M1 - CO

Autor:

Gabriel A. Ortiz

2020 - 0325U

Índice

1. Introducción	2
2. Código Fuente	3
2.1. Interfaz gráfica	3
2.2. Variables globales	4
2.3. Eventos	5
2.3.1. Botón de Imprimir	5
2.3.2. Botón de Apilar	5
2.3.3. Botón radial de Copiar	6
2.3.4. Botón radial Peek	6
2.3.5. Botón radial Pop	7
2.3.6. Evento de presionado una tecla	7
2.4. Métodos	8
2.4.1. Añadir a la pila	8
3. Repositorio	9
4. Conclusión	10

1. Introducción

Una pila (stack en inglés) es una lista ordenada o estructura de datos que permite almacenar y recuperar datos, siendo el modo de acceso a sus elementos de tipo LIFO (del inglés Last In, First Out, «último en entrar, primero en salir»). Esta estructura se aplica en multitud de supuestos en el área de la informática debido a su simplicidad y capacidad de dar respuesta a numerosos procesos.

Para el manejo de los datos cuenta con dos operaciones básicas: apilar (push), que coloca un objeto en la pila, y su operación inversa, retirar (o desapilar, pop), que retira el último elemento apilado.

En cada momento solamente se tiene acceso a la parte superior de la pila, es decir, al último objeto apilado (denominado TOS, Top of Stack en inglés). La operación retirar permite la obtención de este elemento, que es retirado de la pila permitiendo el acceso al anterior (apilado con anterioridad), que pasa a ser el último, el nuevo TOS.

Las pilas suelen emplearse en los siguientes contextos:

1. Evaluación de expresiones en notación postfija (notación polaca inversa).
2. Reconocedores sintácticos de lenguajes independientes del contexto.
3. Implementación de recursividad.

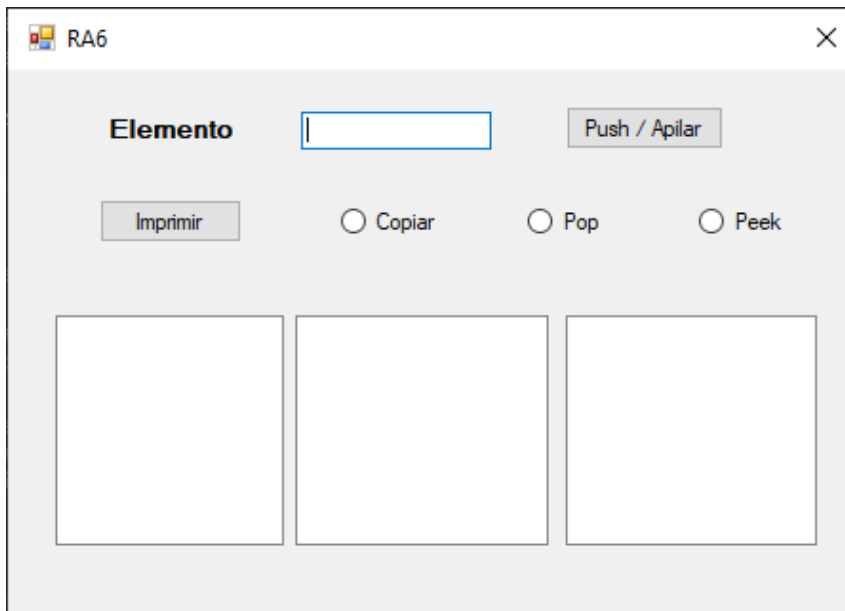
En un sistema operativo cada proceso tiene un espacio de memoria (pila) para almacenar valores y llamadas a funciones.

Una pila acotada es una pila limitada a un tamaño máximo impuesto en su especificación.


Por analogía con objetos cotidianos, una operación apilar equivaldría a colocar un plato sobre una pila de platos, y una operación retirar equivaldría a retirarlo.

2. Código Fuente

2.1. Interfaz gráfica




2.2. Variables globales



```
1 Stack<string> MyStack = new Stack<string>();
```


2.3. Eventos

2.3.1. Botón de Imprimir



```
1 private void btnPrint_Click(object sender, EventArgs e)
2 {
3     lbPrint.Items.Clear();
4
5     foreach (var Item in MyStack)
6     {
7         lbPrint.Items.Add(Item);
8         lbPrint.Items.Add("-----");
9     }
10 }
```

2.3.2. Botón de Apilar



```
1 private void btnPush_Click(object sender, EventArgs e)
2 {
3     Add();
4 }
5
```

2.3.3. Botón radial de Copiar

```
1 private void rbCopy_CheckedChanged(object sender, EventArgs e)
2 {
3     if (rbCopy.Checked == true)
4     {
5         lbRowCopy.Items.Clear();
6
7         foreach (var Item in MyStack)
8         {
9             lbRowCopy.Items.Add(Item);
10            lbRowCopy.Items.Add("-----");
11        }
12    }
13 }
```

2.3.4. Botón radial Peek

```
1 private void rbPeek_CheckedChanged(object sender, EventArgs e)
2 {
3     if (rbPeek.Checked == true)
4     {
5         lbPrint.Items.Add("Primer elemento en pila : " + MyStack.Peek());
6         lbRowCopy.Items.Clear();
7
8         foreach (var Item in MyStack)
9         {
10            lbRowCopy.Items.Add(Item);
11            lbRowCopy.Items.Add("-----");
12        }
13    }
14 }
```

2.3.5. Botón radial Pop

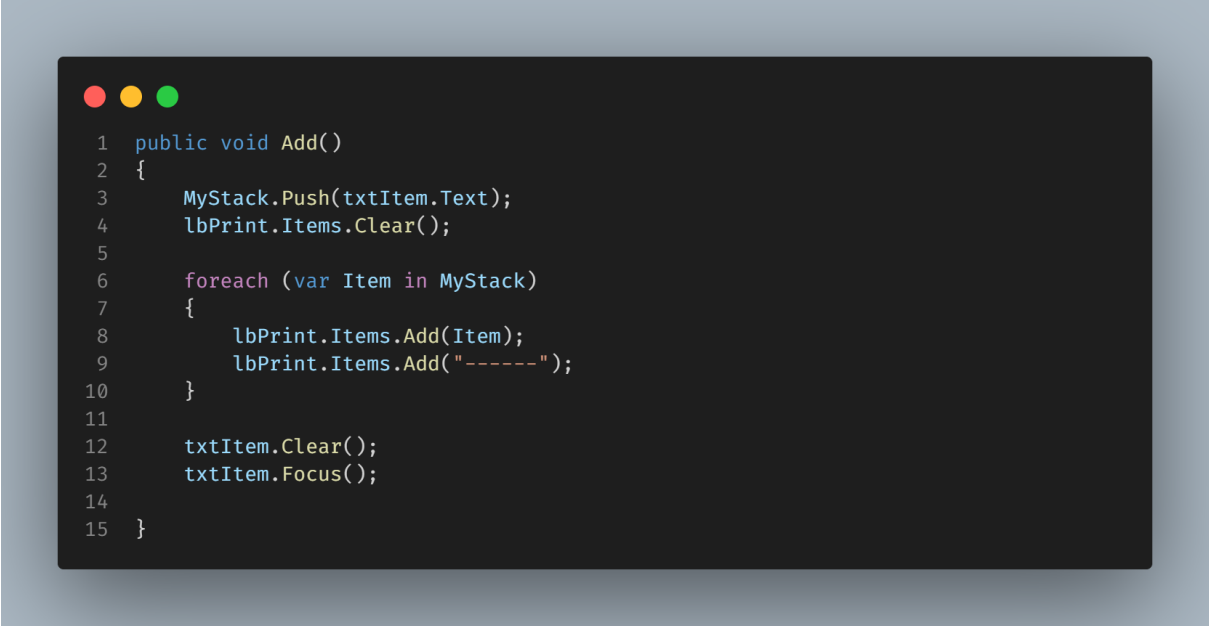
```
1 private void rbPop_CheckedChanged(object sender, EventArgs e)
2     {
3         try {
4             if (rbPop.Checked == true)
5             {
6                 lbPrint.Items.Add("Popping / Desapilar : " + MyStack.Pop());
7                 lbRowCopy.Items.Clear();
8
9                 foreach (var Item in MyStack)
10                {
11                    lbRowCopy.Items.Add(Item);
12                    lbRowCopy.Items.Add("-----");
13                }
14            }
15        }
16        catch
17        {
18            MessageBox.Show("Ha ocurrido un error");
19        }
20    }
```

2.3.6. Evento de presionado una tecla

```
1 private void txtItem_KeyPress(object sender, KeyPressEventArgs e)
2     {
3         if (e.KeyChar == 13)
4             Add();
5     }
```



2.4. Métodos

2.4.1. Añadir a la pila



```
1 public void Add()
2 {
3     MyStack.Push(txtItem.Text);
4     lbPrint.Items.Clear();
5
6     foreach (var Item in MyStack)
7     {
8         lbPrint.Items.Add(Item);
9         lbPrint.Items.Add("-----");
10    }
11
12    txtItem.Clear();
13    txtItem.Focus();
14
15 }
```

3. Repositorio

 Git: <https://github.com/include-minimaltools/AED>

4. Conclusión

Con este laboratorio se ha logrado captar la importancia de las pilas en la informática y programación, para llevar a cabo ciertos procesos en específico de una manera más sencilla. También las facilidades que nos da el lenguaje de C# con la clase especial para la pila denominada «Stack»