



Universidad Nacional de Ingeniería

.- Facultad de Electrotecnia y Computación -.

Ingeniería en Computación

Algoritmización y estructuras de datos

Laboratorio No. 6

Alumno:

Marcel Enmanuel Díaz Largaespada (2020-1384U)

Docente:

Adilson González López

Fecha de entrega:

Martes 31 de agosto del 2021

Índice

1. Introducción	3
2. Diseño del <i>form</i>.....	4
3. Código Fuente	5
3.1. Declaraciones y métodos iniciales.	5
3.2. Métodos de interacción con la interfaz.	6
4. Ejecución del programa	8
5. Conclusiones.....	9

1. Introducción

Se nos introdujo en clases el concepto de *pilas* y *columnas*. Según el entendimiento del estudiante, en pocas palabras, se trata de un tipo de estructuras de datos en las que el orden de inserción de elementos es elemental; esto es, se trabajará con los datos según el orden en que sean añadidos.

En este trabajo de laboratorio se emplean las *pilas*, estructuras cuya mayor característica se rige bajo la síntesis del **LIFO** (Last Input First Output): el último elemento en ser añadido es el primero en salir.

2. Diseño del *form*

La interfaz se realizó según el esquema propuesto en clases.

The image shows a Windows application window titled "Form1". Inside the window, there is a text input field labeled "Elemento" and a button labeled "Push/Apilar". Below these, there are three radio buttons: "Imprimir" (which is selected), "Copiar", "Pop", and "Peek". At the bottom of the form, there are three large rectangular labels: "lbPila", "lbCopiar", and "lbImprimir".

3. Código Fuente

3.1. Declaraciones y métodos iniciales.

Las *pilas* son estructuras de datos y, como tales, deberán de ser declaradas en variables para su empleo. La sintaxis es la siguiente: *Stack<tipo de dato> NombreVariable*:

```
//Declaraciones globales
Stack<string> MiPila = new Stack<string>();
```

La siguiente función nos permitirá agregar elementos a la pila denominada *MiPila*. Esto se logra por medio del método Push. A este, como parámetro, le pasamos input en el textbox *txtElemento*.

```
//Funciones
//Agregar
2 referencias
void Agregar()
{
    MiPila.Push(txtElemento.Text);
    lbPila.Items.Clear();

    foreach(var item in MiPila)
    {
        lbPila.Items.Add(item);
        lbPila.Items.Add("-----");
    }
    txtElemento.Clear();
    txtElemento.Focus();
}
```

3.2. Métodos de interacción con la interfaz.

Caja de texto Elementos – presionar una tecla: Si es la tecla **Enter**, simplemente agrega a la cola el dato ingresado.

```
1 referencia
private void txtElemento_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 13)
    {
        Agregar();
    }
}
```

Botón Push/Apilar - click: El mismo efecto que el evento anterior.

```
1 referencia
private void btnPush_Click(object sender, EventArgs e)
{
    Agregar();
}
```

Radiobotón Copiar - checado: Si el botón está ha sido checado, imprime todos los elementos la pila en el orden en que fueron añadidos.

```
1 referencia
private void rbCopiar_CheckedChanged(object sender, EventArgs e)
{
    if (rbCopiar.Checked == true)
    {
        lbCopiar.Items.Clear();
        foreach (var item in MiPila)
        {
            lbCopiar.Items.Add(item);
            lbCopiar.Items.Add("-----");
        }
    }
}
```

Radiobotón Peek - checado: Si ha sido checado, busca el último elemento de la fila mediante la función **Peek** y lo muestra en la listbox **lbImprimir**.

```
1 referencia
private void rbPeek_CheckedChanged(object sender, EventArgs e)
{
    if (rbPeek.Checked == true)
    {
        lbImprimir.Items.Add("Primero en pila: " + MiPila.Peek());
        lbCopiar.Items.Clear();
        foreach (var item in MiPila)
        {
            lbCopiar.Items.Add(item);
            lbCopiar.Items.Add("-----");
        }
    }
}
```

Radiobotón Pop – checado: Si ha sido checado, busca el último elemento de la fila mediante la función **Pop**, lo retira y lo muestra en la listbox **lbImprimir**.

```
1 referencia
private void rbPop_CheckedChanged(object sender, EventArgs e)
{
    if (rbPop.Checked == true)
    {
        lbImprimir.Items.Add("Popping/desapilar: " + MiPila.Pop());
        lbCopiar.Items.Clear();
        foreach (var item in MiPila)
        {
            lbCopiar.Items.Add(item);
            lbCopiar.Items.Add("-----");
        }
    }
}
```

4. Ejecución del programa

Finalmente, el resultado de nuestro laboratorio arrojó la siguiente interfaz:

The screenshot shows a Windows application window titled "Form1" with standard minimize, maximize, and close buttons. The interface is divided into several sections:

- Input Section:** A label "Elemento" followed by a text input field and a "Push/Apilar" button.
- Action Section:** Four radio buttons labeled "Imprimir", "Copiar", "Pop", and "Peek". The "Peek" button is selected.
- Output Panels:** Three vertical panels displaying the results of the operations:
 - Left Panel:** Shows the stack contents from top to bottom: 5, 4, 3, 2, 1. Each number is followed by a dashed line "-----".
 - Middle Panel:** Shows the element being peeked at, which is 1, followed by a dashed line "-----".
 - Right Panel:** Shows a log of operations. It lists "Popping/desapilar" for each element from 5 down to 1, followed by "Primero en pila" for each element from 4 down to 1.

Ingresando elementos en la pila; eliminando y mostrando el último.

5. Conclusiones

Como una forma de introducirnos a una nueva estructura de datos, se alcanzaron los resultados esperados. Se logró un entendimiento básico de las pilas y se espera llegar a emplearlos en futuros proyectos de forma efectiva.