



Universidad Nacional de Ingeniería

.- Facultad de Electrotecnia y Computación -.

Ingeniería en Computación

Algoritmización y estructuras de datos

Laboratorio No. 7

Alumno:

Marcel Enmanuel Díaz Largaespada (2020-1384U)

Docente:

Adilson González López

Fecha de entrega:

Miércoles 25 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

En esta ocasión, se abordará el concepto de colas. Similarmente al trabajo práctico anterior, se trata de una estructura de datos en la que el orden de los elementos insertados dicta la entrada y salida de datos. Sin embargo, a diferencia de las pilas, las colas persiguen el concepto de **FIFO** (First Input First Output), es decir, el primer dato que haya sido ingresado a la estructura será el primero que será devuelto.

A continuación, el reporte de las capturas de pantalla del código practicado.

2. Diseño del *form*

Se trabajó la interfaz propuesta en clases.

The image shows a screenshot of a Windows application window titled "RA6". The window has a standard Windows XP-style title bar with a close button in the top right corner. The main content area of the window is light gray and contains the following elements:

- Elemento**: A label followed by a text input field.
- Encolar**: A button located to the right of the "Elemento" input field.
- Imprimir**: A button located below the "Elemento" label.
- Radio buttons**: Three radio buttons labeled "Copiar", "Dequeue", and "Peek" are arranged horizontally to the right of the "Imprimir" button.
- Output Panels**: Three rectangular panels are arranged horizontally at the bottom of the window. The first panel is labeled "lbCola", the second "lbCopia", and the third "lbImprimir". Each panel is currently empty.

3. Código Fuente

3.1. Declaraciones y métodos iniciales.

Se declara la estructura de tipo **Queue** de forma similar a como se declaraba una fila. Sólo que, esta vez, solamente el nombre del tipo de dato es lo que cambia:

```
Queue<string> ColaNombre = new Queue<string>();
```

La siguiente función nos permitirá agregar a variable de tipo cola los elementos que vayamos insertando. Entonces, los muestra en pantalla por medio de **lbImprimir**:

```
2 referencias
public void Agregar()
{
    ColaNombre.Enqueue(txtElemento.Text);
    lbCola.Items.Clear();

    foreach(var Item in ColaNombre)
    {
        lbCola.Items.Add(Item);
        lbCola.Items.Add("-----");
    }

    txtElemento.Clear();
    txtElemento.Focus();
}
```

Cada vez que se presione la tecla *enter* al estar escribiendo en la caja de texto **txtElemento**, se ingresará el dato llamando a la función anterior:

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

3.2. Métodos de interacción con la interfaz.

Radiobotón Copiar - checado: Cuando sea checado el radiobotón, se imprimirán en el panel **lbCopia** los elementos existentes en la cola:

```
1 referencia
private void rbCopiar_CheckedChanged(object sender, EventArgs e)
{
    if (rbCopiar.Checked == true)
    {
        lbCopia.Items.Clear();

        foreach (var Item in ColaNombre)
        {
            lbCopia.Items.Add(Item);
            lbCopia.Items.Add("-----");
        }
    }
}
```

Radiobotón Dequeue – checado: Evalúa si se ha checado el radiobotón. Si lo fue, elimina de la cola el primer elemento insertado:

```
1 referencia
private void rbPop_CheckedChanged(object sender, EventArgs e)
{
    if (rbDequeue.Checked == true)
    {
        lbImprimir.Items.Add("Sacar de cola:" + ColaNombre.Dequeue());
        lbCopia.Items.Clear();
        foreach (var Item in ColaNombre)
        {
            lbCopia.Items.Add(Item);
            lbCopia.Items.Add("-----");
        }
    }
}
```

Radiobotón peek - checado: Si ha sido checado, indica en el panel correspondiente el primer elemento en la cola:

```
1 referencia
private void rbPeek_CheckedChanged(object sender, EventArgs e)
{
    if (rbPeek.Checked == true)
    {
        lbImprimir.Items.Add("Primer elemento en cola: " + ColaNombre.Peek());
        lbCopia.Items.Clear();

        foreach (var Item in ColaNombre)
        {
            lbCopia.Items.Add(Item);
            lbCopia.Items.Add("-----");
        }
    }
}
```

Botón Imprimir: Imprime en el panel **lbImprimir** los elementos dentro de la cola:

```
1 referencia
private void btnImprimir_Click(object sender, EventArgs e)
{
    lbImprimir.Items.Clear();

    foreach (var Item in ColaNombre)
    {
        lbImprimir.Items.Add(Item);
        lbImprimir.Items.Add("-----");
    }
}
```

4. Ejecución del programa

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

The screenshot shows a Java Swing window titled "RA6" with a close button (X) in the top right corner. The window contains the following elements:

- Elemento:** A text input field with the label "Elemento" to its left.
- Encolar:** A button labeled "Encolar" located to the right of the "Elemento" input field.
- Imprimir:** A button labeled "Imprimir" located below the "Elemento" input field.
- Operation Selection:** Three radio buttons labeled "Copiar", "Dequeue", and "Peek". The "Peek" radio button is selected.
- Queue Display:** Two text areas. The left text area contains a list of numbers 1 through 5, each followed by a dashed line (e.g., "1-----"). The right text area contains the number 5 followed by a dashed line ("5-----").
- Log:** A text area on the right side of the window containing a log of operations:
 - Sacar de cola: 1
 - Primer elemento en cola: 2
 - Primer elemento en cola: 2
 - Sacar de cola: 2
 - Primer elemento en cola: 3
 - Primer elemento en cola: 3
 - Sacar de cola: 3
 - Sacar de cola: 4
 - Primer elemento en cola: 5

Ingresando datos, eliminando y obteniendo el primer dato.

5. Conclusiones

Con esta práctica se culmina con un nuevo tipo de estructura de datos. A mi parecer, la utilidad de las mismas se encuentra en circunstancias bastante particulares, pero no cabe duda que, para manejar los datos en esas precisas circunstancias, lo harían con efecto.