GUIA N°3 - ARREGLOS EN C#

Los arreglos son un tipo de variable especial que pueden ser consideradas como estructuras de datos que agrupan varios datos de un mismo tipo; el cual puede ser uno de los tipos de datos primitivos del lenguaje o tipos aún más complejos como estructuras de datos y clases. En forma más sencilla un arreglo puede considerarse como una variable capaz de almacenar uno o más datos del mismo tipo al mismo tiempo.

1.1 Arreglos Unidimensionales.

De forma general en el lenguaje C# un arreglo se declara de la siguiente forma:

tipo_dato [ ] nombre_arreglo;

y luego el tamaño del arreglo se debe definir en una segunda línea de la siguiente forma:

nombre_arreglo = new tipo_dato[numero_entero];

Ejemplo:
int [ ] vec;
Vec= new int[5];

La declaración anterior también puede hacerse en una sola línea de la siguiente manera:

int [ ] vec = new int[5];

Los anteriores son arreglos unidimensionales comúnmente conocidos como vectores. Los arreglos se componen principalmente de dos partes: el nombre del arreglo y el subíndice que indica la posición donde se encuentra cada elemento que lo compone. Para recorrer un arreglo o vector se utiliza alguna de las instrucciones de tipo repetitivos tales como for, while. do-while, etc

Ejercicio 1:

1. Cree un nuevo proyecto tipo Windows C#, agregue un Formulario tipo “Primario MDI” y dos formularios tipo “Windows Form”.

2. En la zona correspondiente a la barra de menú, agregue una nueva opción llamada “MisFormularios” y dentro de esta dos opciones más llamadas “Formulario Vectores” y “Formulario Matrices” Respectivamente

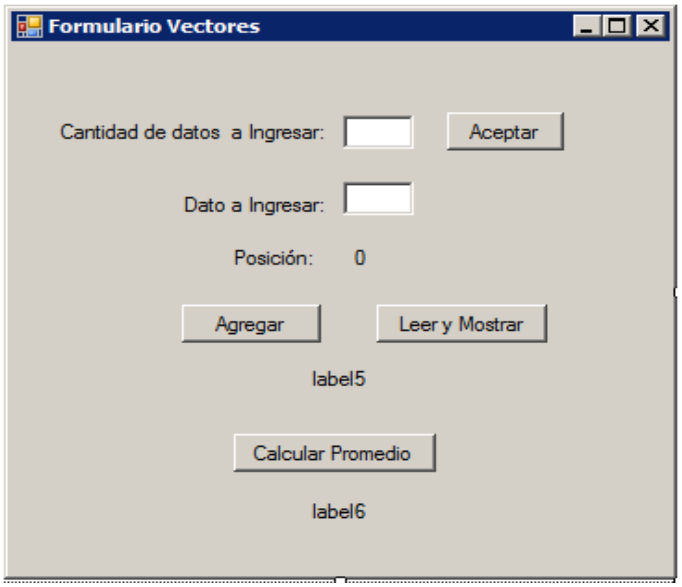
3. Presione doble click en los submenús creados y escriba el siguiente código:

```
private void form1ToolStripMenuItem1_Click(object sender, EventArgs e)
        {   Form1 formvectores = new Form1();
            formvectores.MdiParent = this;
            formvectores.Show();
        }

private void form2ToolStripMenuItem_Click(object sender, EventArgs e)
        {   Form2 formmatriz = new Form2();
            formmatriz.MdiParent = this;
            formmatriz.Show();
        }
```

Nota: Verifique en la clase “program.cs” que el formulario inicial corresponda al Primario MDI.
Ejemplo: `Application.Run(new MDIParent1());`

4. En tiempo de diseño desarrolle el formulario 1 (Formulario Vectores) tal y como se muestra a continuación:



5. Ubique cada bloque de código (Controles y Eventos) para escribir las instrucciones según se muestra a continuación:

```
public partial class Form1 : Form
    {
        private int[] vec;
        private int i;

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            i = 0;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            vec = new int[int.Parse(textBox1.Text)];
        }


        private void button2_Click(object sender, EventArgs e)
        {
           if (i < int.Parse(textBox1.Text))
            {
                vec[i] = int.Parse(textBox2.Text);
                i++;
                label4.Text = i.ToString();
```

```
            textBox2.Text = "";
        }
        else
        {
            MessageBox.Show("El vector ya esta lleno");
            textBox2.Text = "";
        }
    }


    private void button3_Click(object sender, EventArgs e)
    {
        for (i = 0; i < int.Parse(textBox1.Text); i++)
        {
            label5.Text = label5.Text + vec[i].ToString() + " ";
        }
    }


    private void button4_Click(object sender, EventArgs e)
    {
        int sumatoria = 0;
            int promedio = 0;

        for (i = 0; i < int.Parse(textBox1.Text); i++)
        {
            sumatoria = sumatoria + vec[i];

        }
        promedio = sumatoria / vec.Length;
        label6.Text = promedio.ToString();
    }
```

Nota: Tenga en cuenta identificar cada uno de los eventos donde deben colarse las instrucciones y añadir según correspondan.

1.2 Arreglos Multidimensionales

De la misma forma que en otros lenguajes de programación en C#, existen arreglos multidimencionales conocidos comúnmente bajo el nombre matrices. Los arreglos multidimensionales manejan un número de índices igual o superior a dos (2); cuando manejan dos índices se denominan arreglos bidimensionales. Para este ejemplo trabajaremos un arreglo bidimensional o matriz de dos índices.

Las matrices se declaran de forma similar que los vectores. La estructura general es:

tipo_dato [,] nombre_arreglo = new tipo_dato[numero_entero1, numero_entero2];

Ejemplo:

int mat[,] = new int[3,3];

Ejercicio 2:

1. En el mismo proyecto del ejercicio 1, En tiempo de diseño desarrolle el formulario 2 (Formulario Matrices) tal y como se muestra a continuación:

2. Ubique cada bloque de código para escribir las instrucciones según se muestra a continuación.

```
public partial class Form2 : Form
    {
        private int[,] mat;
        private int i;
        private int j;
        private int contador;

        public Form2()
        {
            InitializeComponent();
        }
        private void Form2_Load(object sender, EventArgs e)
        {
            i = 0;
            j = 0;
            contador = 0;
        }

private void button1_Click(object sender, EventArgs e)
     {
      mat = new int[int.Parse(textBox1.Text),int.Parse(textBox2.Text)];
     }
```

```
private void button2_Click(object sender, EventArgs e)
    {
       int final = int.Parse(textBox1.Text) * int.Parse(textBox2.Text);
       contador = contador + 1;

          if (i < int.Parse(textBox1.Text))
           {
               if (j < int.Parse(textBox2.Text))
               {
                   mat[i, j] = int.Parse(textBox3.Text);
                   j++;

                   textBox3.Text = "";
               }
               else
               {
                   i++;
                   j = 0;

                   if (i < int.Parse(textBox1.Text))
                   {
                       mat[i, j] = int.Parse(textBox3.Text);
                       j++;
                   }

                   textBox3.Text = "";
               }
           }

           if (contador == final)
           {
               MessageBox.Show("La matriz ya esta llena");
               textBox3.Text = "";
           }


       }

       private void button3_Click(object sender, EventArgs e)
       {
           for (i = 0; i < int.Parse(textBox1.Text); i++)
           {
               for (j = 0; j < int.Parse(textBox2.Text); j++)
               {
               label5.Text = label5.Text + mat[i, j].ToString() + "  ";
               }
               label5.Text += Char.ConvertFromUtf32(13);
           }

       }
```

```
        private void button4_Click(object sender, EventArgs e)
        {
            int sumatoria = 0;

            for (i = 0; i < int.Parse(textBox1.Text); i++)
            {
                for (j = 0; j < int.Parse(textBox2.Text); j++)
                {
                    sumatoria = sumatoria + mat[i,j];
                }
            }
            label6.Text = sumatoria.ToString();
        }
    }
```

Nota: Tenga en cuenta identificar cada uno de los eventos donde deben colarse las instrucciones y añadir según correspondan.

TALLER:

Desarrolle una única aplicación tipo Windows en lenguaje C# que satisfaga los siguientes requerimientos:

1. Se requiere un software que permita sacar algunas estadísticas sobre la penetración del servicio de banda ancha en Colombia. Dicha aplicación debe procesar y entregar la siguiente información:

a.) Número total del conexiones de banda ancha a la fecha
b.) Nombre y cantidad de conexiones de la ciudad con el mayor número de conexiones
c.) Nombre y cantidad de conexiones de la ciudad con el menor número de conexiones
d.) Promedio general de conexiones de banda ancha en el país.

El software debe solicitar como entradas: La fecha, el número de ciudades, un nombre para cada una de las ciudades y el número de conexiones de banda ancha en cada ciudad.

2. Se requiere un formulario adicional en la misma aplicación que permita conocer cómo se distribuyen el número de conexiones versus velocidades de banda ancha en el país teniendo en cuenta los siguientes aspectos:

- ✓ Se pueden ingresar N ciudades (Cantidad ilimitada)
- ✓ Se tienen 5 diferentes velocidades Banda Ancha (1MB, 2MB, 4MB, 8MB, 10MB)
- ✓ El ingreso del número de conexiones por cada velocidad se recibe en orden: primero la velocidad 1, luego la 2 y así sucesivamente hasta completar todas las ciudades inscritas

Para solucionar el problema planteado debe tener en cuenta las siguientes recomendaciones:

- ✓ Formar y mostrar en pantalla una matriz que contenga el número de conexiones reportadas en cada velocidad para cada una de los N ciudades.
- ✓ Encuentre el total de conexiones reportadas por cada ciudad incluyendo todas las velocidades y calcule el porcentaje que representa.
- ✓ Encuentre el total de conexiones reportadas por cada velocidad incluyendo todas las ciudades y calcule el porcentaje que representa