



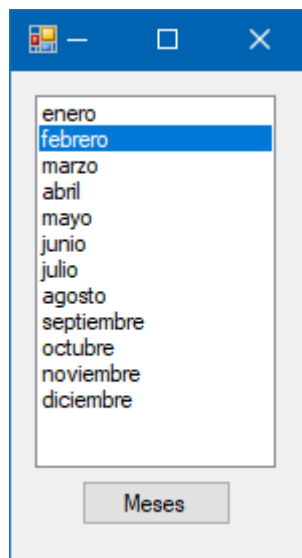
PRÁCTICA 3: ESTRUCTURAS DE CONTROL DE REPETICIÓN

Objetivos:

- ✓ Aplicar las estructuras de control repetitivas del lenguaje C#
- ✓ Utilizar las estructuras de control de repetición en aplicaciones Windows Form

1. Ejecutar Visual Studio .NET
2. Crear un nuevo proyecto de tipo Aplicación de Windows Forms (Windows Form App)
3. Crear un formulario para cada uno de los siguientes ejemplos.

Ejemplo 1: uso de **for** para agregar los meses del año al ListBox

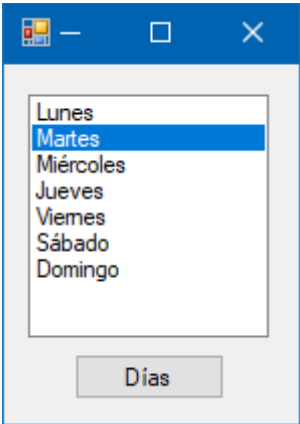


Control	Name
ListBox1	lstMeses
Button1	btnMostrar

```
using System.Globalization;
```

```
private void btnMostrar_Click(object sender, EventArgs e)
{
    lstMeses.Items.Clear();
    for (int i = 1; i <= 12; i++)
    {
        string mes = CultureInfo.CurrentCulture.DateTimeFormat.GetMonthName(i);
        mes = CultureInfo.CurrentCulture.TextInfo.ToTitleCase(mes);
        lstMeses.Items.Add(mes);
    }
}
```

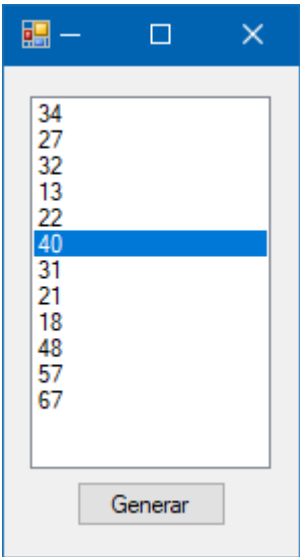
Ejemplo 2: uso de **foreach** para recorrer un arreglo



Control	Name
ListBox1	lstDias
Button1	btnMostrar

```
private void btnMostrar_Click(object sender, EventArgs e)
{
    lstDias.Items.Clear();
    string[] dias = { "Lunes", "Martes", "Miércoles", "Jueves",
                     "Viernes", "Sábado", "Domingo" };
    foreach (string dia in dias)
    {
        lstDias.Items.Add(dia);
    }
}
```

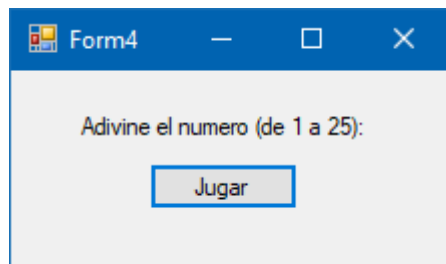
Ejemplo 3: uso de **while** para generar una cantidad específica de números aleatorios sin repetirse



Control	Name
ListBox1	lstDias
Button1	btnGenerar

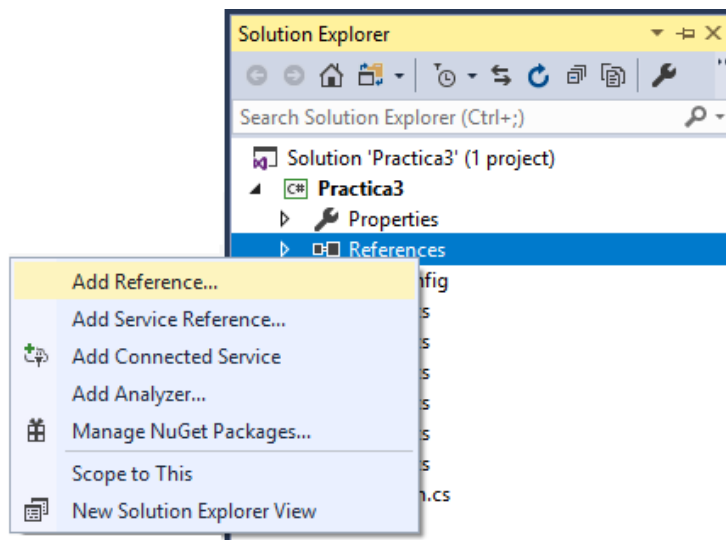
```
private void btnGenerar_Click(object sender, EventArgs e)
{
    Random r = new Random();
    const int CANTIDAD = 25;
    int num = 0, cont = 0;
    lstNumeros.Items.Clear();
    while (cont < CANTIDAD)
    {
        num = r.Next(10, 91);
        if (!lstNumeros.Items.Contains(num))
        {
            lstNumeros.Items.Add(num);
            cont++;
        }
    }
}
```

Ejemplo 4: uso de **do while** para realizar el juego de adivinar un numero entre un rango especifico



Control	Name
Label1	lblTexto
Button1	btnJugar

Agregar una referencia a Visual Basic, en el panel Add Reference...



Seleccionar Microsoft.VisualBasic

<input checked="" type="checkbox"/>	Microsoft.VisualBasic	10.0.0.0
-------------------------------------	-----------------------	----------

```

private void btnJugar_Click(object sender, EventArgs e)
{
    Random r = new Random();
    int numSecreto = 0, numIngresado = 0;
    numSecreto = r.Next(1, 26);
    bool adivinado = false;
    do
    {
        try
        {
            numIngresado = Convert.ToInt32(MessageBox.Show("Ingrese el número:"));
            if (numSecreto != numIngresado)
            {
                MessageBox.Show("Incorrecto.");
            }
            else
            {
                adivinado = true;
            }
        }
        catch (Exception)
        {
            MessageBox.Show("Ingrese un número");
        }
    } while (!adivinado);
    MessageBox.Show("Felicidades !");
}

```

Ejemplo 5: uso del control **DataGridView**

Control	Text	Name
TextBox1		txtNombre
TextBox2		txtEdad
Button1	Agregar	btnAgregar
Button2	Borrar datos	btnBorrar
Button3	Contar mujeres	btnMujeres
Button4	Contar hombres	btnHombres
Button5	Salir	btnSalir
radioButton1	Femenino	rdoFemenino
radioButton2	Masculino	rdoMasculino
dataGridView1		dgvDatos

```

private void btnAgregar_Click(object sender, EventArgs e)
{
    string nombre = txtNombre.Text;
    int edad;
    string sexo = "";
    if (nombre != String.Empty && int.TryParse(txtEdad.Text, out edad))
    {
        if (rdoFemenino.Checked)
        {
            sexo = "Femenino";
        }
        else
        {
            sexo = "Masculino";
        }
        dgvDatos.Rows.Add(nombre, edad, sexo);
        dgvDatos.ClearSelection();
    }
    else
    {
        MessageBox.Show("Ingrese nombre y edad");
    }
}

```

```

private void btnBorrar_Click(object sender, EventArgs e)
{
    dgvDatos.Rows.Clear();
    txtNombre.Clear();
    txtEdad.Clear();
    rdoFemenino.Checked = true;
    txtNombre.Focus();
}

```

```

private void btnMujeres_Click(object sender, EventArgs e)
{
    int contMujeres = 0;
    for (int f = 0; f < dgvDatos.Rows.Count; f++)
    {
        if (dgvDatos.Rows[f].Cells[2].Value.Equals("Femenino"))
        {
            contMujeres++;
        }
    }
    MessageBox.Show("Total mujeres: "+contMujeres);
}

```

```

private void btnHombres_Click(object sender, EventArgs e)
{
    int contHombres = 0;
    for (int f = 0; f < dgvDatos.Rows.Count; f++)
    {
        if (dgvDatos.Rows[f].Cells[2].Value.Equals("Masculino"))
        {
            contHombres++;
        }
    }
    MessageBox.Show("Total hombres: " + contHombres);
}

private void btnSalir_Click(object sender, EventArgs e)
{
    this.Close();
}

```



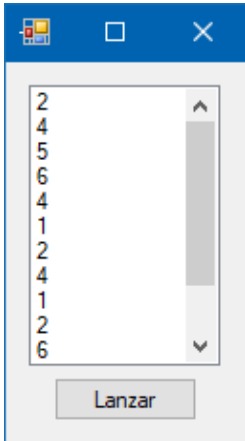
Colocar en el repositorio (GitHub) del portafolio de la **práctica3** la solución de los siguientes ejercicios.

Indicaciones: Agregar a la solución un proyecto de Visual C# de tipo Aplicación Windows Forms y programar los siguientes formularios.

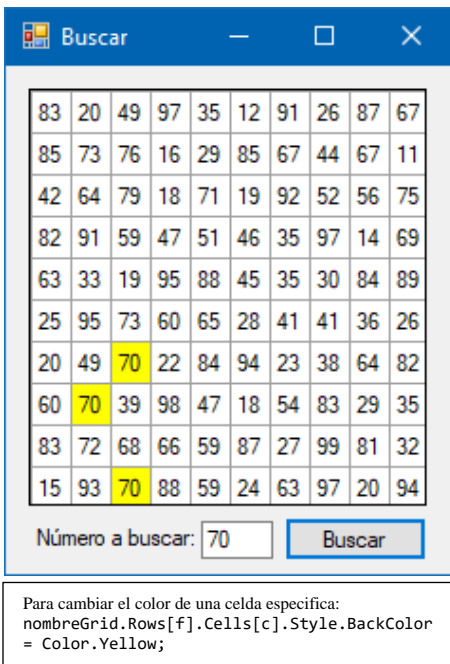
1. Elaborar una aplicación para generar en un ListBox la tabla de multiplicar especificada por el usuario.

2. Elaborar una aplicación para agregar a un ListBox el alfabeto en mayúsculas, al hacer clic en el botón Mostrar.

3. Elaborar una aplicación que realice 5000 lanzamientos de un dado, agregue cada resultado al ListBox y al final muestre en un MessageBox la cantidad de veces que logró obtener el número 6.



4. Elaborar una aplicación para generar 100 números aleatorios (entre 10 y 99) mostrarlos en un DataGridView de 10 filas x 10 columnas y programar el botón **Buscar** para que permita buscar el numero indicado en el TextBox y los resalte con otro color.



Agregar el siguiente código al evento **Load** del form para configurar el Grid:

```
private void Form7_Load(object sender, EventArgs e)
{
    dataGridView1.Size = new Size(210, 220);
    dataGridView1.AllowUserToAddRows = false;
    dataGridView1.ScrollBars = ScrollBars.None;
    dataGridView1.ColumnCount = 10;
    dataGridView1.ColumnHeadersVisible = false;
    dataGridView1.RowHeadersVisible = false;
    dataGridView1.AutoSizeColumnsMode =
        DataGridViewAutoSizeColumnsMode.AllCells;

    Random r = new Random();
    for (int f = 0; f < 10; f++)
    {
        dataGridView1.Rows.Add();
        for (int c = 0; c < 10; c++)
        {
            dataGridView1.Rows[f].Cells[c].Value = r.Next(10, 100);
        }
    }
    dataGridView1.ClearSelection();
}
```

5. Elaborar una aplicación para calcular una planilla de pagos por horas trabajadas, aplicando un impuesto del 10%. (Incluir las validaciones respectivas)

