



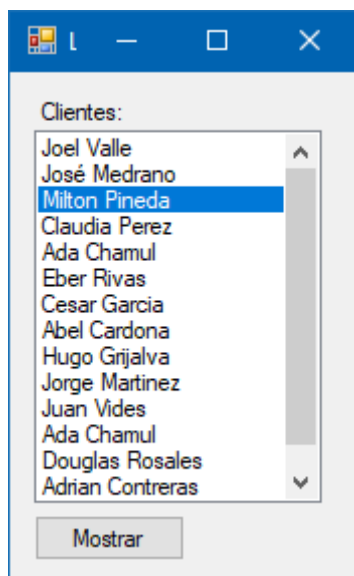
## PRÁCTICA 5: ARREGLOS

### Objetivos:

- ✓ Crear arreglos unidimensionales y bidimensionales (rectangulares y dentados)
- ✓ Realizar ordenamiento y búsqueda en arreglos
- ✓ Pasar arreglos a parámetros de funciones (métodos)

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.

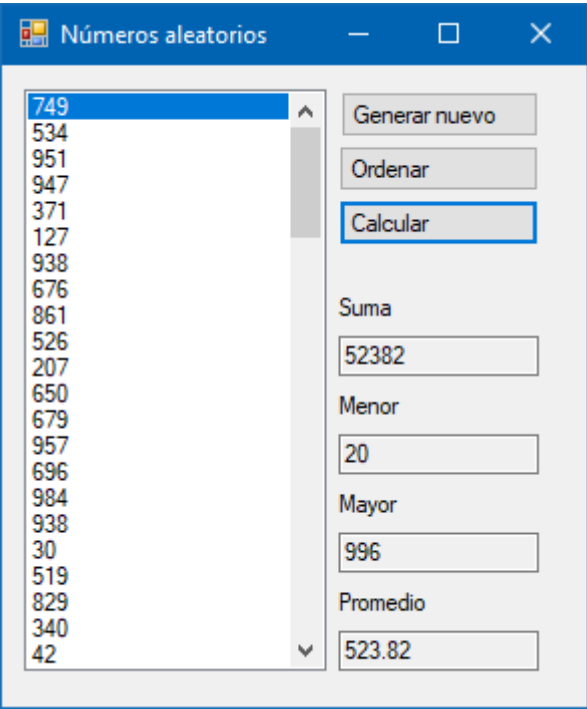
**Ejemplo 1:** mostrar arreglo unidimensional en un ListBox (usando foreach)



Control	Name	Text
ListBox	lstClientes	
Button1	btnMostrar	Mostrar

```
private void btnMostrar_Click(object sender, EventArgs e)
{
    string[] clientes = new string[15];
    clientes[0] = "Joel Valle";
    clientes[1] = "José Medrano";
    clientes[2] = "Milton Pineda";
    clientes[3] = "Claudia Perez";
    clientes[4] = "Ada Chamul";
    clientes[5] = "Eber Rivas";
    clientes[6] = "Cesar Garcia";
    clientes[7] = "Abel Cardona";
    clientes[8] = "Hugo Grijalva";
    clientes[9] = "Jorge Martinez";
    clientes[10] = "Juan Vides";
    clientes[11] = "Luisa Morales";
    clientes[12] = "Douglas Rosales";
    clientes[13] = "Adrian Contreras";
    clientes[14] = "Oscar Coreas";
    Array.Sort(clientes);
    lstClientes.Items.Clear();
    foreach (string cliente in clientes)
    {
        lstClientes.Items.Add(cliente);
    }
}
```

**Ejemplo 2:** uso de métodos de la clase array



Control	Name	Text
ListBox	lstNumeros	
Button1	btnGenerar	Generar nuevo
Button2	btnOrdenar	Ordenar
Button3	btnCalcular	Calcular
TextBox1	txtSuma	
TextBox2	txtMenor	
TextBox3	txtMayor	
TextBox4	txtPromedio	

Crear el arreglo números y la variable arregloCreado (en las declaraciones generales, fuera de los métodos)

```
int[] numeros = new int[100];
bool arregloCreado = false;

1 reference
private void btnGenerar_Click(object sender, EventArgs e)
{
    Random ale = new Random();
    for (int i = 0; i < numeros.Length; i++)
    {
        numeros[i] = ale.Next(1, 1001);
    }
    arregloCreado = true;
    lstNumeros.DataSource = null;
    lstNumeros.DataSource = numeros;
}
```

```

private void btnOrdenar_Click(object sender, EventArgs e)
{
    if (arregloCreado)
    {
        lstNumeros.DataSource = null;
        Array.Sort(numeros);
        lstNumeros.DataSource = numeros;
    }
    else
    {
        MessageBox.Show("Primero debe generar el arreglo");
    }
}

private void btnCalcular_Click(object sender, EventArgs e)
{
    if (arregloCreado)
    {
        txtSuma.Text = numeros.Sum().ToString();
        txtMenor.Text = numeros.Min().ToString();
        txtMayor.Text = numeros.Max().ToString();
        txtPromedio.Text = numeros.Average().ToString();
    }
    else
    {
        MessageBox.Show("Primero debe generar el arreglo");
    }
}

```

**Ejemplo 3:** paso de arreglo como parámetro a función

90	57	53	41	68	23	53	57	73	46
52	31	42	67	72	42	36	10	75	14
36	87	68	35	65	71	33	35	94	33
50	11	55	56	17	67	42	41	80	59
64	36	78	97	84	63	65	15	99	84
76	12	64	68	11	12	89	27	75	97
89	28	63	49	95	51	75	86	60	14
99	94	29	25	50	10	19	71	51	12
15	27	36	19	57	47	47	86	54	15
78	25	66	83	70	46	72	38	11	43

Sumar      Suma    5258

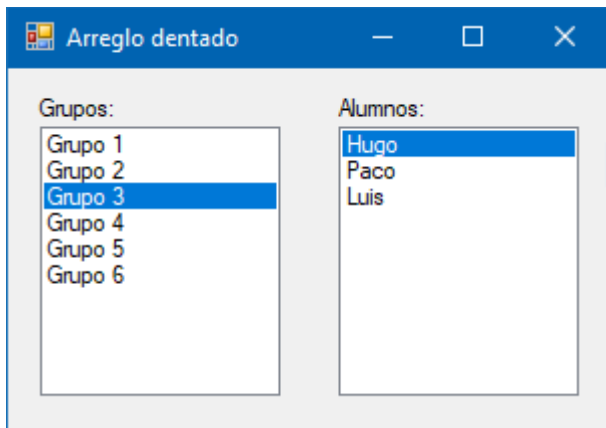
Control	Name	Text
DataGridView1	dgvNumeros	
Button1	btnSumar	Sumar
TextBox1	txtSuma	

```
int[, ] numeros = new int[10, 10];
```

```
private void frmNumeros_Load(object sender, EventArgs e)
{
    dgvNumeros.Size = new Size(300, 225);
    dgvNumeros.ColumnCount = 10;
    dgvNumeros.AllowUserToAddRows = false;
    dgvNumeros.ColumnHeadersVisible = false;
    dgvNumeros.RowHeadersVisible = false;
    dgvNumeros.AutoSizeColumnsMode =
        DataGridViewAutoSizeColumnsMode.Fill;
    Random ale = new Random();
    for (int f = 0; f < numeros.GetLength(0); f++)
    {
        dgvNumeros.Rows.Add();
        for (int c = 0; c < numeros.GetLength(1); c++)
        {
            numeros[f, c] = ale.Next(10, 100);
            dgvNumeros.Rows[f].Cells[c].Value = numeros[f, c];
        }
    }
    dgvNumeros.ClearSelection();
}
```

```
public static int sumarArreglo(int[, ] arreglo)
{
    int sum = 0;
    for (int f = 0; f < arreglo.GetLength(0); f++)
    {
        for (int c = 0; c < arreglo.GetLength(1); c++)
        {
            sum += arreglo[f, c];
        }
    }
    return sum;
}
```

```
private void btnSumar_Click(object sender, EventArgs e)
{
    int suma = sumarArreglo(numeros);
    txtSuma.Text = suma.ToString();
}
```



Control	Name
ListBox1	lstGrupos
ListBox2	lstAlumnos

```
string[][] alumnos = new string[4][];
```

```
private void frmListas_Load(object sender, EventArgs e)
{
    alumnos[0] = new string[4] { "Juan", "Mario", "Carlos", "Ana" };
    alumnos[1] = new string[2] { "Claudia", "Roberto" };
    alumnos[2] = new string[3] { "Hugo", "Paco", "Luis" };
    alumnos[3] = new string[5] { "Gabriel", "Joel", "Gladis", "Ronald", "Milton" };
    Array.Resize(ref alumnos, alumnos.Length + 2);
    alumnos[4] = new string[3] { "Mario", "Manuel", "José" };
    alumnos[5] = new string[4] { "Cristina", "Julia", "Ernesto", "Norma" };
    lstGrupos.Items.Clear();
    for (int i = 0; i < alumnos.GetLength(0); i++)
    {
        lstGrupos.Items.Add("Grupo " + (i + 1));
    }
}

private void lstGrupos_SelectedIndexChanged(object sender, EventArgs e)
{
    if (lstGrupos.SelectedIndex >= 0)
    {
        lstAlumnos.Items.Clear();
        foreach (string alumno in alumnos[lstGrupos.SelectedIndex])
        {
            lstAlumnos.Items.Add(alumno);
        }
    }
}
```



Colocar en el repositorio (GitHub) del portafolio de la **práctica5** la solución de los siguientes ejercicios.

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

1. Desarrollar una aplicación que almacene en un arreglo los nombres de 10 vendedores, que los muestre en un ListBox y provea las opciones para ordenar ascendentemente, ordenar descendientemente, quitar elementos de la lista (y del arreglo) y buscar el nombre de un vendedor (si se encuentra lo seleccionará automáticamente en la lista y si no se encuentra mostrará un mensaje)

2. Crear una aplicación que genere de manera automática una Matriz A (5x5) y una Matriz B (5x5), ambas con números aleatorios entre 10 y 90, crear una función personalizada, que reciba como parámetros: la matriz A, la matriz B y el tipo de operación a realizar (sumar o multiplicar), la función deberá retornar el arreglo resultante y será mostrado en el DataGridView Resultado.

3. Escribir una aplicación que posea un arreglo bidimensional dentado, para almacenar al menos 5 sucursales, con una cantidad variable de ventas por cada sucursal, al seleccionar una sucursal en la lista de sucursales, mostrará las ventas de esa sucursal en la lista de ventas y mostrará el total de ventas de la sucursal seleccionada.