

Ejercicio: Registrar los datos de los libros: Código, título, autor, año de edición y costo del libro

- Deberán ser registrados en la clase Queue (colas)
- Usar "TabControl" para los botones de mantenimiento: Registro, Listado, Búsqueda por autor y Búsqueda por año e edición.

Clase Libro

```
0 references
class Libro
{
    0 references
    public int codigo { get; set; }

    0 references
    public string titulo { get; set; }

    0 references
    public string autor { get; set; }

    0 references
    public int anoEdicion { get; set; }

    0 references
    public double costo { get; set; }
}
```

```
// Objeto de la clase Queue Cola
Queue<Libro> aLibros = new Queue<Libro>();
```

1 reference

```
public Form1()
{
    InitializeComponent();
}
```

```
// Método para generar código
```

0 references

```
private int generaCodigo()
{
    int codigo = 0;
    if (aLibros.Count < 1) return 1;
    else
    {
        foreach (Libro l in aLibros)
        {
            codigo = l.codigo;
        }
    }
    return codigo + 1;
}
```

```
// Método para listar el componente lv de la página de Listado
```

0 references

```
private void listado()
{
    lvLibros.Items.Clear();
    foreach (Libro l in aLibros)
    {
        ListViewItem fila = new ListViewItem(l.codigo.ToString());
        fila.SubItems.Add(l.titulo);
        fila.SubItems.Add(l.autor);
        fila.SubItems.Add(l.anoEdicion.ToString());
        fila.SubItems.Add(l.costo.ToString());
        lvLibros.Items.Add(fila);
    }
}
```

1 reference

```
private void Form1_Load(object sender, EventArgs e)
{
    lblCodigo.Text = generaCodigo().ToString();
}
```

// Botón para generar código

1 reference

```
private void btnGeneraCodigo_Click(object sender, EventArgs e)
{
    lblCodigo.Text = generaCodigo().ToString();
}
```

// Método para limpiar controles

0 references

```
private void limpiar()
{
    txtAutor.Clear();
    txtTitulo.Clear();
    txtAnoEdicion.Clear();
    txtConsto.Clear();
    txtTitulo.Focus();
}
```

// Método para llenar el combo autor en tiempo real

1 reference

```
private void llenarAutores()
{
    // Lista de autores
    var autores = (from Libro a in aLibros select new { a.autor }).Distinct().ToList();
    cboAutor.DataSource = autores; // Llenar el combo de autores
    cboAutor.DisplayMember = "autor"; // = que el atributo de la clase
}
```

// Método para llenar el combo año de edición en tiempo real

1 reference

```
private void llenarAno()
{
    // Lista de autores
    var anos = (from Libro a in aLibros select new { a.anoEdicion }).Distinct().ToList();
    cboAnoEdicion.DataSource = anos; // Llenar el combo de autores
    cboAnoEdicion.DisplayMember = "anoEdicion"; // = que el atributo de la clase
}
```

```

1 reference
private void Form1_Load(object sender, EventArgs e)
{
    lblCodigo.Text = generaCodigo().ToString();
}

// Botón para generar código
1 reference
private void btnGeneraCodigo_Click(object sender, EventArgs e)
{
    lblCodigo.Text = generaCodigo().ToString();
}

```

```

// Botón Registrar
1 reference
private void btnRegistrar_Click(object sender, EventArgs e)
{
    // Objeto de la clase Libro
    Libro objL = new Libro();

    // Enviando los valores al objeto
    objL.codigo = int.Parse(lblCodigo.Text);
    objL.titulo = txtTitulo.Text;
    objL.autor = txtAutor.Text;
    objL.anoEdicion = int.Parse(txtAnoEdicion.Text);
    objL.costos = double.Parse(txtConsto.Text);
    foreach (Libro l in aLibros)
    {
        if (l.titulo == objL.titulo)
        {
            MessageBox.Show("El libro ya existe en la BD");
            return;
        }
    }
}

```

```

// Agregando el libro si no existe aún
aLibros.Enqueue(objL);
listado();
limpiar();
lblCodigo.Text = generaCodigo().ToString();
llenarAutores(); // Llenar autores
llenarAno(); // Llenar años
}

```

```
// Búsqueda por autor (combo) => Listado
```

```
1 reference
```

```
private void cboAutor_SelectedIndexChanged(object sender, EventArgs e)
{
    lvLibrosAutor.Items.Clear(); // Limpiar los items
    foreach (Libro l in aLibros)
    {
        if (l.autor == cboAutor.Text)
        {
            ListViewItem fila = new ListViewItem(l.codigo.ToString());
            fila.SubItems.Add(l.titulo);
            fila.SubItems.Add(l.autor);
            fila.SubItems.Add(l.anoEdicion.ToString());
            fila.SubItems.Add(l.costo.ToString());
            lvLibrosAutor.Items.Add(fila);
        }
    }
}
```

```
// Búsqueda por año de edición (combo) => Listado
```

```
1 reference
```

```
private void cboAnoEdicion_SelectedIndexChanged(object sender, EventArgs e)
{
    lvLibrosxAno.Items.Clear(); // Limpiar los items
    foreach (Libro l in aLibros)
    {
        if (l.anoEdicion.ToString() == cboAnoEdicion.Text)
        {
            ListViewItem fila = new ListViewItem(l.codigo.ToString());
            fila.SubItems.Add(l.titulo);
            fila.SubItems.Add(l.autor);
            fila.SubItems.Add(l.anoEdicion.ToString());
            fila.SubItems.Add(l.costo.ToString());
            lvLibrosxAno.Items.Add(fila);
        }
    }
}
```