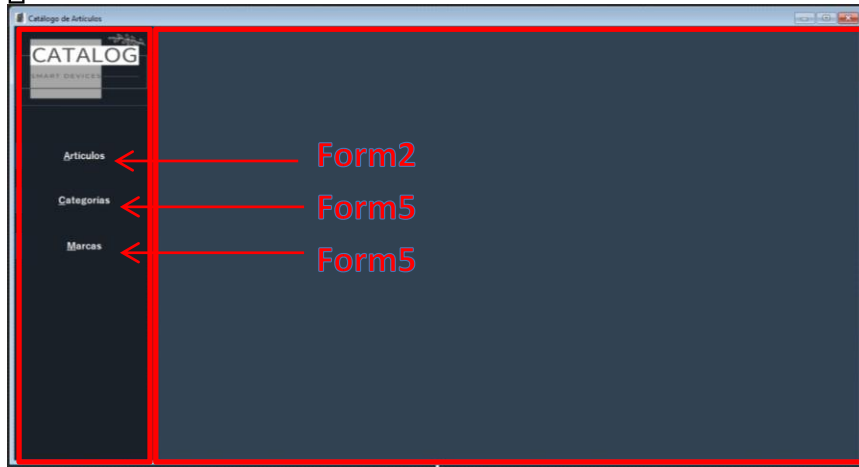


APLICACIÓN CATÁLOGO

Curso C# Nivel 2 [.Net + SQL]

Formularios

Form1:



Esta Ventana Contiene 2 Paneles de Control:

1- El Panel Menú: Tiene un Panel Logo y 3 botones

- Botón "Artículos": Se instancia un Form2 y se abre dentro del Panel Contenedor
- Botón "Categorías" y Artículos: Se instancia un Form5 y envía un parámetro el título que va a tener este Form5 para identificar de cual se trata.

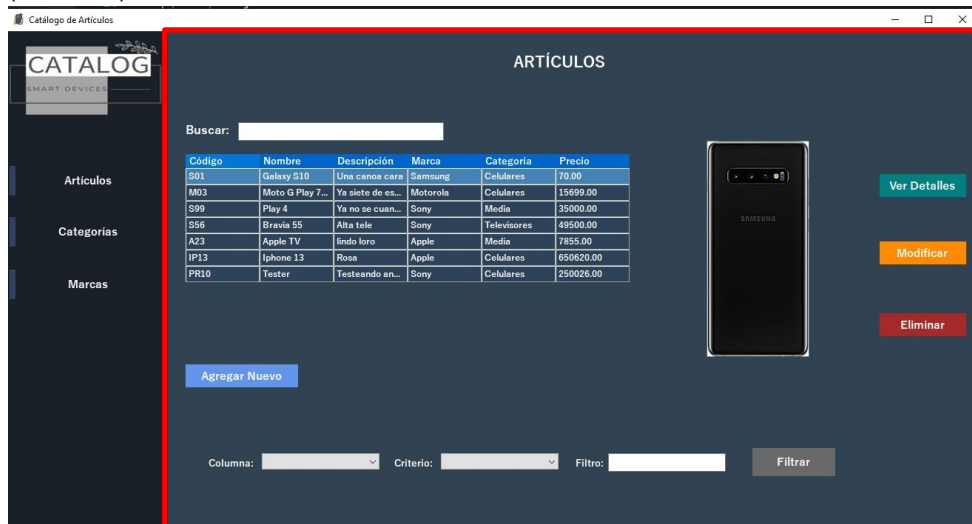
```
private void btnCategoria_Click(object sender, EventArgs e)
{
    string tituloCat = "Categoria";
    Form5 FormCategoria = new Form5(tituloCat);
    AbrirVentanaSeleccion(FormCategoria);
}

private void btnMarca_Click(object sender, EventArgs e)
{
    string tituloMar = "Marca";
    Form5 FormMarcas = new Form5(tituloMar);
    AbrirVentanaSeleccion(FormMarcas);
}
```

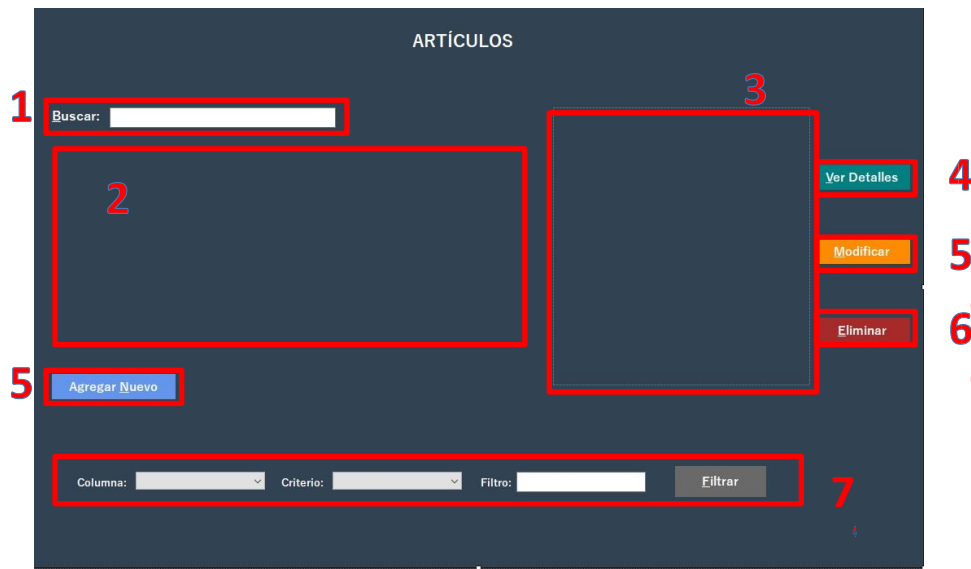
2- El Panel Contenedor: Tiene un Método que permite abrir un formulario en su interior, y otorgar todos los controles al mismo, validando si existe o no otro control

```
public void AbrirVentanaSeleccion(Form formSeleccionado)
{
    //Si existe algun control en el panel, lo eliminamos
    if (this.panelContenedor.Controls.Count > 0)
    {
        this.panelContenedor.Controls.RemoveAt(0);
    }
    //Cambiamos sus propiedades y agregamos al panel
    formSeleccionado.TopLevel = false;
    formSeleccionado.Dock = DockStyle.Fill;
    this.panelContenedor.Controls.Add(formSeleccionado);
    //Y lo mostramos
    this.panelContenedor.Tag = formSeleccionado;
    formSeleccionado.Show();
}
```

- 3- Por defecto, siempre al cargar el form1, el panel contenedor abrirá el Form2 (Artículos).



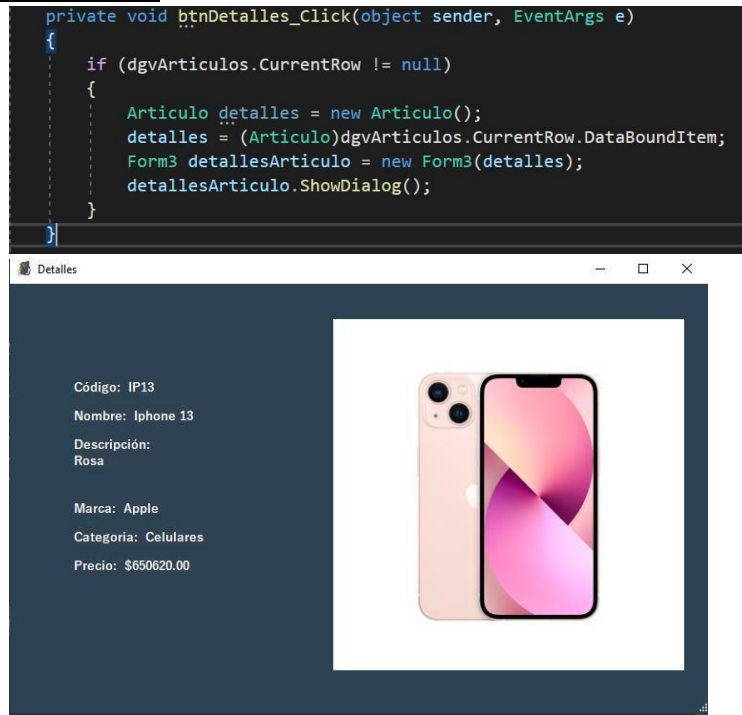
Form2:



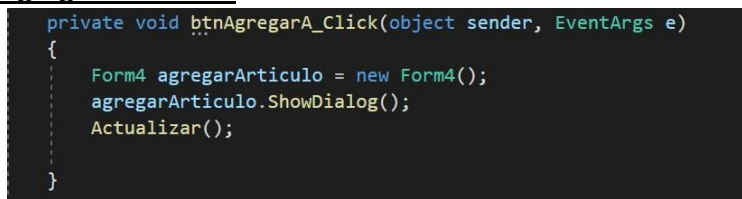
- 1- Filtro Rápido
- 2- Lista con los datos de Base de Datos
- 3- Carga de Imagen: Se utiliza el método general "ValidarImagenUrl"

```
namespace Models
{
    public class ValidarImagenUrl
    {
        public void CargarImagen(string imagen, PictureBox CajaImagen)
        {
            try
            {
                CajaImagen.Load(imagen);
            }
            catch (Exception)
            {
                CajaImagen.Load("https://media.istockphoto.com/vectors/thumbnaill-i");
            }
        }
    }
}
```

4- **Ver Detalles:** Abre el Form3.



5- **Agregar/Modificar:** Abre el Form4



Para agregar un artículo nuevo son necesarios los siguientes campos



The 'Agregar Artículo' form contains the following fields and validation messages:

- Código: (Message: Debe completar este campo)
- Nombre: (Message: Debe completar este campo)
- Descripción:
- Imagen Url: (with a '+' icon)
- Marca:
- Categoría:
- Precio: (Message: Escriba un valor Numérico)

Buttons: Aceptar, Cancelar

Código de la validación:

```

public bool ValidarCampos()
{
    bool bandera;
    //limpio los Mensajes de Validación
    lblMessageCod.Text = "";
    lblMessageName.Text = "";
    lblMessagePrecio.Text = "";

    if (txtCodigo.Text == "" || txtNombre.Text == "" || (campo.EsNumero(txtPrecio.Text) == false))
    {
        bandera = true;
    }
    else
    {
        bandera = false;
    }

    if (bandera)
    {
        if (txtCodigo.Text == "")
        {
            lblMessageCod.Text = "Debe completar este campo";
        }
        if (txtNombre.Text == "")
        {
            lblMessageName.Text = "Debe completar este campo";
        }
        if (!campo.EsNumero(txtPrecio.Text))
        {
            lblMessagePrecio.Text = "Escriba un valor Numérico";
        }
        return true;
    }
    else
    {
        return false;
    }
}

```

Para la modificación:

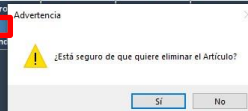
```

private void btnModificarA_Click(object sender, EventArgs e)
{
    if (dgvArticulos.CurrentRow != null)
    {
        Articulo seleccionado;
        seleccionado = (Articulo)dgvArticulos.CurrentRow.DataBoundItem;
        Form4 editarArticulo = new Form4(seleccionado);
        editarArticulo.ShowDialog();
        Actualizar();
    }
}

```

- 6- **Eliminar:** Elimina el registro seleccionado, con un mensaje de Advertencia que solicita su confirmación

| Código | Nombre | Descripción | Marca | Categoría | Precio |
|--------|------------------|-------------------|----------|-------------|----------|
| S01 | Galaxy S10 | Uno sensor cara | Samsung | Celulares | 70.00 |
| M03 | Moto G Play 7... | Ya siete de es... | Motorola | Celulares | 15999.00 |
| S99 | Play 4 | Ya no se cuan... | Sony | Media | 35000.00 |
| S56 | Bravia 55 | Alta tele | Sony | Televisores | 49500.00 |
| A23 | Apple TV | Indo for... | | | |
| IP13 | Iphone 13 | Rosa | | | |
| PR10 | Tester | Testeand... | | | |



- 7- **Filtro Avanzado:** Este filtro utiliza conexiones a la Base de Datos, tiene las siguientes validaciones:

```
private bool validarfiltro()
{
    if (cboColumna.SelectedIndex < 0)
    {
        lblMensaje1.Text = "Por favor, seleccione una columna";
        return true;
    }
    if (cboCriterio.SelectedIndex < 0)
    {
        lblMensaje2.Text = "Por favor, seleccione un criterio";
        return true;
    }
    if (cboColumna.SelectedItem.ToString() == "Precio")
    {
        if (!(campoEsNum.EsNumero(txtFiltroAvanz.Text)))
        {
            lblMensaje3.Text = "Por favor, ingrese un número";
            return true;
        }
    }
    return false;
}
```

The image shows three sequential screenshots of a web form titled 'Filtro Avanzado'. Each screenshot has a dark blue background with white text and form elements. The first screenshot shows the 'Columna' dropdown menu open with no selection, and a red error message 'Por favor, seleccione una columna' below it. The second screenshot shows 'Precio' selected in the 'Columna' dropdown, and a red error message 'Por favor, seleccione un criterio' below it. The third screenshot shows 'Precio' selected in 'Columna', 'Igual a' selected in 'Criterio', and 'abc' entered in the 'Filtro' text box, with a red error message 'Por favor, ingrese un número' below it.

- 8- **Otras Validaciones:** No se permite agregar/modificar/eliminar cuando la lista este vacía para evitar una excepción, ya que al utilizar el filtro rápido pueden quedar vacías las filas.

```
if (dgvArticulos.CurrentRow != null)
```

Esta condición (que también se encuentra en el Form5) nos permite evitar excepciones en los eventos de los Botones “Ver Detalles”, “Modificar” y “Eliminar”.

The image is a screenshot of a web application window titled 'CATALOG'. It features a sidebar with 'Artículos', 'Categorías', and 'Marcas'. The main area is titled 'ARTÍCULOS' and contains a search bar, a table with columns 'Código', 'Nombre', 'Descripción', 'Marca', 'Categoría', and 'Precio', and a 'Agregar Nuevo' button. On the right side, there are three buttons: 'Ver Detalles' (green), 'Modificar' (orange), and 'Eliminar' (red), which are grouped together in a red rectangular box. At the bottom, there is a 'Filtro Avanzado' section with dropdowns for 'Columna' and 'Criterio', and a text input for 'Filtro'.

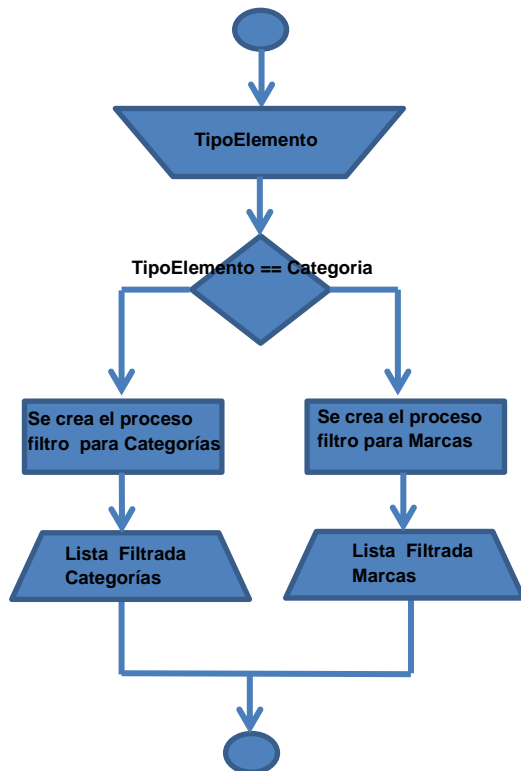
Por lo tanto, los botones no generarán ninguna acción.

Form5:

Este Form dependiendo del parámetro que recibe (campo tipoElemento) va a ir comparando en sus eventos si se trata de un objeto de tipo Categoría o un objeto de tipo de Marca. Su estructura:

The image displays two screenshots of a web application interface for managing a catalog. The top screenshot shows the 'MARCA' (Brand) management screen, and the bottom screenshot shows the 'CATEGORIA' (Category) management screen. Both screens have a sidebar with 'Articulos', 'Categorias', and 'Marcas' options. The 'MARCA' screen has a 'Filtro' input field, a table with columns 'Descripción', 'Samsung', 'Apple', 'Sony', 'Huawei', and 'Motorola', and buttons 'Modificar' and 'Eliminar'. The 'CATEGORIA' screen has a 'Filtro' input field, a table with columns 'Calculadora', 'Televisores', 'Módem', 'Audio', and 'PC', and buttons 'Modificar', 'Eliminar', and 'Agregar Nuevo'. Red boxes and numbers 1-4 highlight specific elements in the 'CATEGORIA' screen: 1 points to the 'Filtro' input field, 2 points to the table, 3 points to the 'Agregar Nuevo' button, and 4 points to the 'Eliminar' button.

1- Filtro:



En el diagrama de flujo simplifiqué el proceso y omití la línea de código que oculta el Id.
Código:

```

//Evento para el Filtro Rápido
private void txtFiltroRapido_TextChanged(object sender, EventArgs e)
{
    string filtro = txtFiltroRapido.Text.ToUpper();
    if (tipoElemento == "Categoria")
    {
        List<Categoria> listaFiltrada;
        if (filtro.Length > 2)
        {
            listaFiltrada = listaCategoria.FindAll(x => x.Descripcion.ToUpper().Contains(filtro));
        }
        else
        {
            listaFiltrada = listaCategoria;
        }
        dgvDatos.DataSource = null;
        dgvDatos.DataSource = listaFiltrada;
    }
    else
    {
        List<Marca> listaFiltradaM;
        if (filtro.Length > 2)
        {
            listaFiltradaM = listaMarca.FindAll(x => x.Descripcion.ToUpper().Contains(filtro));
        }
        else
        {
            listaFiltradaM = listaMarca;
        }
        dgvDatos.DataSource = null;
        dgvDatos.DataSource = listaFiltradaM;
    }
    dgvDatos.Columns["Id"].Visible = false;
}

```

- 2- **Lista con los datos de Base de Datos:** Con el método actualizar que recibe por parámetro el título (tipoElemento) identificamos de qué objeto se trata para mostrar su lista correspondiente.

```

private void Actualizar(string tipo)
{
    if (tipo == "Categoria")
    {
        CategoriaNegocio negocio = new CategoriaNegocio();
        listaCategoria = negocio.Listar();
        dgvDatos.DataSource = listaCategoria;
        dgvDatos.Columns["Id"].Visible = false;
    }
    else
    {
        MarcaNegocio negocioM = new MarcaNegocio();
        listaMarca = negocioM.Listar();
        dgvDatos.DataSource = listaMarca;
        dgvDatos.Columns["Id"].Visible = false;
    }
}

public Form5(string nombreObjeto)
{
    InitializeComponent();
    tipoElemento = nombreObjeto;
    lblTituloPestaña.Text = tipoElemento.ToUpper();
    Actualizar(tipoElemento);
}

```

3- Agregar/Modificar: Sobrecarga del Método "IrAForm6"

Para agregar, dependiendo del nombre del objeto (Categoría o Marca), se abre el Form6

```

//Metodo Agregar Dinámico
private void IrAForm6(string nombreObjeto)
{
    Form6 FormAgregar = new Form6(nombreObjeto);
    FormAgregar.ShowDialog();
    Actualizar(tipoElemento);
}

//Evento Para ir al Form6: "Agregar"
private void btnAgregarA_Click(object sender, EventArgs e)
{
    IrAForm6(tipoElemento);
}

```

Para modificar, se compara qué tipo de objeto hay en el registro seleccionado (Si un objeto de tipo Categoría o Marca) para crear la instancia correspondiente y llamar al método "IrAForm6" donde recibe el objeto seleccionado y el título (tipoElemento) para personalizar la ventana.


```
//Evento para ir al Form6: "Modificar"
private void btnModificarA_Click(object sender, EventArgs e)
{
    if (dgvDatos.CurrentRow != null)
    {
        if (tipoDeObjeto.ToString() == "Models.Categoria")
        {
            Categoria seleccionado = (Categoria)dgvDatos.CurrentRow.DataBoundItem;
            irAForm6(seleccionado, tipoElemento);
        }
        else if (tipoDeObjeto.ToString() == "Models.Marca")
        {
            Marca seleccionado = (Marca)dgvDatos.CurrentRow.DataBoundItem;
            irAForm6(seleccionado, tipoElemento);
        }
        Actualizar(tipoElemento);
    }
}

//Método Modificar Dinámico
private void irAForm6(object objeto, string nombreObjeto)
{
    Form6 FormModificar = new Form6(objeto, nombreObjeto);
    FormModificar.ShowDialog();
}
}
```

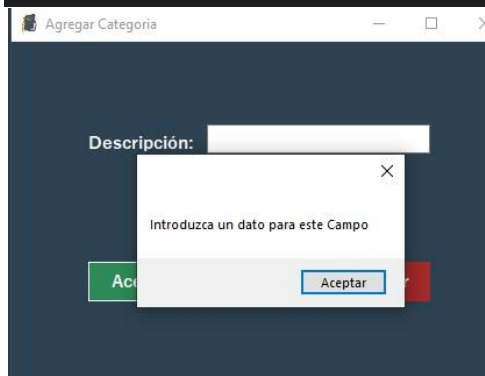
Vista Del Form6:

El tipo de Elemento que se envía en el Form5 establece el título del Form6

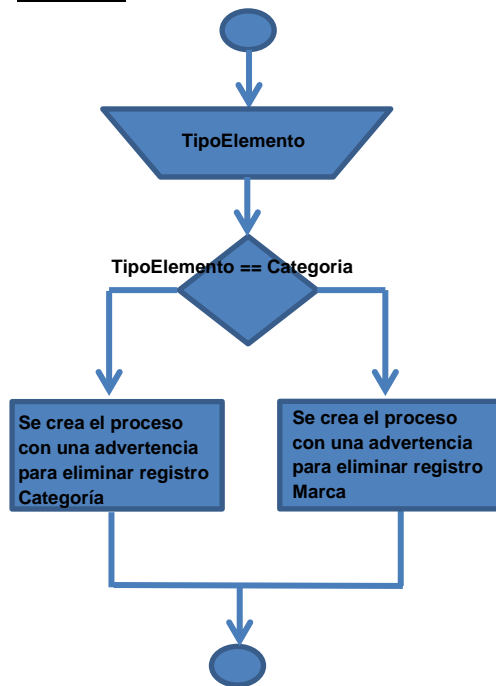


Con un método interno que valide que no se almacene vacío el campo descripción

```
//Valida que no se carguen descripciones vacias
public bool textoDescripcion(string datoIntroducido)
{
    if (datoIntroducido == "")
    {
        MessageBox.Show("Introduzca un dato para este Campo");
        return true;
    }
    return false;
}
}
```



4- Eliminar:



```
//Evento para eliminar un registro
private void btnEliminarA_Click(object sender, EventArgs e)
{
    if (dgvDatos.CurrentRow != null)
    {
        if (tipoElemento == "Categoria")
        {
            CategoriaNegocio negocio = new CategoriaNegocio();
            Categoria seleccionado;
            try
            {
                DialogResult respuesta = MessageBox.Show("¿Está seguro de Eliminar esta Categoria?", "Atención", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
                if (respuesta == DialogResult.Yes)
                {
                    seleccionado = (Categoria)dgvDatos.CurrentRow.DataBoundItem;
                    negocio.Eliminar(seleccionado.Id);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
        }
        else
        {
            MarcaNegocio negocio = new MarcaNegocio();
            Marca seleccionado;
            try
            {
                DialogResult respuesta = MessageBox.Show("¿Está seguro de Eliminar esta Marca?", "Atención", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
                if (respuesta == DialogResult.Yes)
                {
                    seleccionado = (Marca)dgvDatos.CurrentRow.DataBoundItem;
                    negocio.Eliminar(seleccionado.Id);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
        }
    }
    Actualizar(tipoElemento);
}
```