

EXERCÍCIO 1 –

CLASSE Form1

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;

namespace Biblioteca_
{
    public partial class Form1 : Form
    {
        public static List<Livro> LivrosAdicionados = new List<Livro>();

        public Form1()
        {
            InitializeComponent();

            lblInicial.Font = new Font("Arial", 12, FontStyle.Regular);
            btnAdicionarLivro.Font = new Font("Arial", 12, FontStyle.Regular);
            btnEmprestarLivro.Font = new Font("Arial", 12, FontStyle.Regular);
        }

        private void btnAdicionarLivro_Click(object sender, EventArgs e)
        {
            this.Controls.Clear();

            AdicionarLivro adicionarLivro = new AdicionarLivro(this);
            adicionarLivro.Dock = DockStyle.Fill;

            this.Controls.Add(adicionarLivro);
        }

        private void btnEmprestarLivro_Click(object sender, EventArgs e)
        {
            this.Controls.Clear();

            EmprestarLivro emprestarLivro = new EmprestarLivro(this);
            emprestarLivro.Dock = DockStyle.Fill;

            this.Controls.Add(emprestarLivro);
            emprestarLivro.AtualizarListaLivros();
        }

        public void MostrarTelaInicial()
        {
            this.Controls.Clear();

            InitializeComponent();

            lblInicial.Font = new Font("Arial", 12, FontStyle.Regular);
            btnAdicionarLivro.Font = new Font("Arial", 12, FontStyle.Regular);
            btnEmprestarLivro.Font = new Font("Arial", 12, FontStyle.Regular);
        }
    }
}
```

CLASSE AdicionarLivro

```
using System;
using System.Drawing;
using System.Windows.Forms;

namespace Biblioteca_
{
    public partial class AdicionarLivro : UserControl
    {
        private Form1 formularioPrincipal;

        public AdicionarLivro(Form1 formPrincipal)
        {
            InitializeComponent();
            formularioPrincipal = formPrincipal;

            lblTituloAdicionar.Font = new Font("Arial", 12, FontStyle.Regular);
            lblAno.Font = new Font("Arial", 9, FontStyle.Regular);
            lblAutor.Font = new Font("Arial", 9, FontStyle.Regular);
            lblGenero.Font = new Font("Arial", 9, FontStyle.Regular);
            lblTitulo.Font = new Font("Arial", 9, FontStyle.Regular);
            btnLimpar.Font = new Font("Arial", 8, FontStyle.Regular);

            cmbGenero.Items.Add("Ação");
            cmbGenero.Items.Add("Aventura");
            cmbGenero.Items.Add("Cinema de arte");
            cmbGenero.Items.Add("Chanchada");
            cmbGenero.Items.Add("Comédia");
            cmbGenero.Items.Add("Comédia de ação");
            cmbGenero.Items.Add("Comédia de terror");
            cmbGenero.Items.Add("Comédia dramática");
            cmbGenero.Items.Add("Comédia romântica");
            cmbGenero.Items.Add("Dança");
            cmbGenero.Items.Add("Documentário");
            cmbGenero.Items.Add("Docuficção");
            cmbGenero.Items.Add("Drama");
            cmbGenero.Items.Add("Espionagem");
            cmbGenero.Items.Add("Faroeste");
            cmbGenero.Items.Add("Fantasia");
            cmbGenero.Items.Add("Fantasia científica");
            cmbGenero.Items.Add("Ficção científica");
            cmbGenero.Items.Add("Filme épico");
            cmbGenero.Items.Add("Filmes com truques");
            cmbGenero.Items.Add("Filmes de guerra");
            cmbGenero.Items.Add("Filme policial");
            cmbGenero.Items.Add("Mistério");
            cmbGenero.Items.Add("Musical");
            cmbGenero.Items.Add("Romance");
            cmbGenero.Items.Add("Terror");
            cmbGenero.Items.Add("Thriller");
            cmbGenero.Items.Add("Não especificado");
        }

        private void btnLimpar_Click(object sender, EventArgs e)
        {
            LimparFormulario();
        }

        private void LimparFormulario()
        {
            txtAno.Clear();
            txtAutor.Clear();
            txtTitulo.Clear();
            cmbGenero.SelectedIndex = -1;
        }
    }
}
```

```

private void btnAdicionar_Click(object sender, EventArgs e)
{
    try
    {
        if (string.IsNullOrEmpty(txtTitulo.Text))
        {
            MessageBox.Show("Título é obrigatório!");
            return;
        }

        if (string.IsNullOrEmpty(txtAutor.Text))
        {
            MessageBox.Show("Autor é obrigatório!");
            return;
        }

        if (!int.TryParse(txtAno.Text, out int ano))
        {
            MessageBox.Show("Ano inválido! Digite um número.");
            return;
        }

        if (cmbGenero.SelectedIndex == -1)
        {
            MessageBox.Show("Gênero é obrigatório!");
            return;
        }

        string generoSelecioneado = cmbGenero.SelectedItem.ToString();

        Livro livro = new Livro(txtTitulo.Text, txtAutor.Text, ano, generoSelecioneado);

        Form1.LivrosAdicionados.Add(livro);

        MessageBox.Show("Livro adicionado com sucesso!");
        LimparFormulario();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Erro: {ex.Message}");
    }
}

private void btnVoltar_Click(object sender, EventArgs e)
{
    formularioPrincipal.MostrarTelaInicial();
}
}

```

CLASSE EmprestarLivro

```

using System;
using System.Drawing;
using System.Windows.Forms;

namespace Biblioteca_
{
    public partial class AdicionarLivro : UserControl
    {
        private Form1 formularioPrincipal;
    }
}

```

```

public AdicionarLivro(Form1 formPrincipal)
{
    InitializeComponent();
    formularioPrincipal = formPrincipal;

    lblTituloAdicionar.Font = new Font("Arial", 12, FontStyle.Regular);
    lblAno.Font = new Font("Arial", 9, FontStyle.Regular);
    lblAutor.Font = new Font("Arial", 9, FontStyle.Regular);
    lblGenero.Font = new Font("Arial", 9, FontStyle.Regular);
    lblTitulo.Font = new Font("Arial", 9, FontStyle.Regular);
    btnLimpar.Font = new Font("Arial", 8, FontStyle.Regular);

    cmbGenero.Items.Add("Ficção");
    cmbGenero.Items.Add("Romance");
    cmbGenero.Items.Add("Aventura");
    cmbGenero.Items.Add("Mistério");
    cmbGenero.Items.Add("Fantasia");
    cmbGenero.Items.Add("Não especificado");
}

private void btnLimpar_Click(object sender, EventArgs e)
{
    LimparFormulario();
}

private void LimparFormulario()
{
    txtAno.Clear();
    txtAutor.Clear();
    txtTitulo.Clear();
    cmbGenero.SelectedIndex = -1;
}

private void btnAdicionar_Click(object sender, EventArgs e)
{
    try
    {
        if (string.IsNullOrEmpty(txtTitulo.Text))
        {
            MessageBox.Show("Título é obrigatório!");
            return;
        }

        if (string.IsNullOrEmpty(txtAutor.Text))
        {
            MessageBox.Show("Autor é obrigatório!");
            return;
        }

        if (!int.TryParse(txtAno.Text, out int ano))
        {
            MessageBox.Show("Ano inválido! Digite um número.");
            return;
        }

        if (cmbGenero.SelectedIndex == -1)
        {
            MessageBox.Show("Gênero é obrigatório!");
            return;
        }

        string generoSelecioneado = cmbGenero.SelectedItem.ToString();

        Livro livro = new Livro(txtTitulo.Text, txtAutor.Text, ano, generoSelecioneado);
    }
}

```

```

        Form1.LivrosAdicionados.Add(livro);

        MessageBox.Show("Livro adicionado com sucesso!");
        LimparFormulario();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Erro: {ex.Message}");
    }
}

private void btnVoltar_Click(object sender, EventArgs e)
{
    formularioPrincipal.MostrarTelaInicial();
}
}
}

```

CLASSE Livro

```

using System;

namespace Biblioteca_
{
    public class Livro
    {
        private string _titulo;
        private string _autor;
        private int _ano;
        private string _genero;

        public string Titulo
        {
            get { return _titulo; }
            private set
            {
                if (string.IsNullOrEmpty(value))
                    throw new ArgumentException("Título não pode ser vazio.");
                _titulo = value;
            }
        }

        public string Autor
        {
            get { return _autor; }
            private set
            {
                if (string.IsNullOrEmpty(value))
                    throw new ArgumentException("Autor não pode ser vazio.");
                _autor = value;
            }
        }

        public int Ano
        {
            get { return _ano; }
            private set
            {
                if (value < 1450 || value > DateTime.Now.Year)
                    throw new ArgumentOutOfRangeException("Ano deve estar entre 1450 e o ano atual.");
                _ano = value;
            }
        }
    }
}

```

```
}

public string Genero
{
    get { return _genero; }
    private set
    {
        if (string.IsNullOrEmpty(value))
            throw new ArgumentException("Gênero não pode ser vazio.");
        _genero = value;
    }
}

public Livro(string titulo, string autor, int ano, string genero)
{
    Titulo = titulo;
    Autor = autor;
    Ano = ano;
    Genero = genero;
}

public override string ToString()
{
    return $"{Titulo} - {Autor} ({Ano}) [{Genero}]";
}
}
```