

Marcos Martinho RA 183106

Danilo Neves RA 245555

Nicolas Ruiz RA 187222

1. Para conectar o C# com o banco de dados SQL Server, primeiro eu instalei o Visual Studio e também o SQL Server. Depois, dentro do projeto no Visual Studio, usei a biblioteca System.Data.SqlClient, que já vem com o .NET. A conexão foi feita usando uma string de conexão com o nome do servidor, o banco que eu queria acessar e os dados de login. Com isso, consegui abrir a conexão e fazer as consultas ou comandos direto no banco, tudo pela aplicação em C#.

UIL

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace ParaCasa1
{
    public partial class CadLivrosUIL : Form
    {
        Livro umlivro = new Livro();
        public CadLivrosUIL()
        {
```

```

InitializeComponent();
}

private void button2_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    textBox5.Text = "";
}

private void button1_Click(object sender, EventArgs e)
{
    umlivro.setCodigo(textBox1.Text);
    umlivro.setTitulo(textBox2.Text);
    umlivro.setAutor(textBox3.Text);
    umlivro.setEditora(textBox4.Text);
    umlivro.setAno(textBox5.Text);
    LivroBLL.validaDados(umlivro,'i');
    if (Erro.getErro())
        MessageBox.Show(Erro.getMsg());
    else
        MessageBox.Show("Dados inseridos com sucesso!");
}

private void CadLivrosUIL_Load(object sender, EventArgs e)
{
    LivroBLL.conecta();
    if (Erro.getErro())
        MessageBox.Show(Erro.getMsg());
}

```

```

}

private void button3_Click(object sender, EventArgs e)
{
    umlivro.setCodigo(textBox1.Text);
    LivroBLL.validaCodigo(umlivro,'c');
    if (Erro.getErro())
        MessageBox.Show(Erro.getMsg());
    else
    {
        textBox1.Text = umlivro.getCodigo();
        textBox2.Text = umlivro.getTitulo();
        textBox3.Text = umlivro.getAutor();
        textBox4.Text = umlivro.getEditora();
        textBox5.Text = umlivro.getAno();
    }
}

private void CadLivrosUIL_FormClosing(object sender, FormClosingEventArgs e)
{
    LivroBLL.desconecta();
}

private void button4_Click(object sender, EventArgs e)
{
    umlivro.setCodigo(textBox1.Text);
    LivroBLL.validaCodigo(umlivro,'e');
    if (Erro.getErro())
        MessageBox.Show(Erro.getMsg());
    else
        MessageBox.Show("Livro Excluído!");
}

```

```
}  
  
private void button5_Click(object sender, EventArgs e)  
{  
    umlivro.setCodigo(textBox1.Text);  
    umlivro.setTitulo(textBox2.Text);  
    umlivro.setAutor(textBox3.Text);  
    umlivro.setEditora(textBox4.Text);  
    umlivro.setAno(textBox5.Text);  
    LivroBLL.validaDados(umlivro,'a');  
    if (Erro.getErro())  
        MessageBox.Show(Erro.getMsg());  
    else  
        MessageBox.Show("Dados alterados com sucesso!");  
}  
}  
}
```

BLL

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ParaCasa1
{
    class LivroBLL
    {
        public static void conecta()
        {
            LivroDAL.conecta();
        }

        public static void desconecta()
        {
            LivroDAL.desconecta();
        }

        public static void validaCodigo(Livro umlivro, char op)
        {
            Erro.setErro(false);
            if (umlivro.getCodigo().Equals(""))
            {
                Erro.setMsg("O código é de preenchimento obrigatório!");
                return;
            }
        }
    }
}
```

```
if (op == 'c')
    LivroDAL.consultaUmLivro(umlivro);
else
    LivroDAL.excluiUmLivro(umlivro);
}

public static void validaDados(Livro umlivro, char op)
{
    Erro.setErro(false);
    if (umlivro.getCodigo().Equals(""))
    {
        Erro.setMsg("O código é de preenchimento obrigatório!");
        return;
    }
    if (umlivro.getTitulo().Equals(""))
    {
        Erro.setMsg("O título é de preenchimento obrigatório!");
        return;
    }
    if (umlivro.getAutor().Equals(""))
    {
        Erro.setMsg("O autor é de preenchimento obrigatório!");
        return;
    }
    if (umlivro.getEditora().Equals(""))
    {
        Erro.setMsg("A Editora é de preenchimento obrigatório!");
        return;
    }
}
```

```
if (umlivro.getAno().Equals(""))
{
    Erro.setMsg("O ano é de preenchimento obrigatório!");
    return;
}

try
{
    int.Parse(umlivro.getAno());
}
catch (Exception)
{
    Erro.setMsg("O valor do ano deve ser numérico!");
    return;
}

if (int.Parse(umlivro.getAno()) <= 0)
{
    Erro.setMsg("O valor do Ano deve ser numérico e positivo!");
    return;
}

if (op == 'i')
    LivroDAL.inseriUmLivro(umlivro);
else
    LivroDAL.atualizaUmLivro(umlivro);
}

}

}
```

DAL

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
namespace ParaCasa1
{
    class LivroDAL
    {
        private static string strConexao = "Data Source=DESKTOP-
        3LVM070\\MSSQLSERVER1;Initial Catalog=LivrariaDB;Integrated Security=True";
        private static SqlConnection conn = new SqlConnection(strConexao);
        private static SqlCommand strSQL;
        private static SqlDataReader result;
        public static void conecta()
        {
            try
            {
                conn.Open();
            }
            catch (Exception)
            {
                Erro.setMsg("Problemas ao se conectar ao Banco de Dados");
            }
        }
    }
}
```



```

}

public static void desconecta()
{
    conn.Close();
}

public static void inseriUmLivro(Livro umlivro)
{
    string aux = "INSERT INTO TabLivro(codigo, titulo, autor, editora, ano) VALUES (@codigo,
    @titulo, @autor, @editora, @ano)";

    strSQL = new SqlCommand(aux, conn);

    strSQL.Parameters.AddWithValue("@codigo", int.Parse(umlivro.getCodigo()));
    strSQL.Parameters.AddWithValue("@titulo", umlivro.getTitulo());
    strSQL.Parameters.AddWithValue("@autor", umlivro.getAutor());
    strSQL.Parameters.AddWithValue("@editora", umlivro.getEditora());
    strSQL.Parameters.AddWithValue("@ano", int.Parse(umlivro.getAno()));

    Erro.setErro(false);

    try
    {
        strSQL.ExecuteNonQuery();
    }

    catch (Exception)
    {
        Erro.setMsg("Chave Duplicada!");
    }
}

public static void excluiUmLivro(Livro umlivro)
{
    string aux = "DELETE FROM TabLivro WHERE codigo = @codigo";

```

```

strSQL = new SqlCommand(aux, conn);

strSQL.Parameters.AddWithValue("@codigo", int.Parse(umlivro.getCodigo()));

strSQL.ExecuteNonQuery();

}

public static void atualizaUmLivro(Livro umlivro)

{

string aux = "UPDATE TabLivro SET titulo = @titulo, autor = @autor, editora = @editora,
ano = @ano WHERE codigo = @codigo";

strSQL = new SqlCommand(aux, conn);

strSQL.Parameters.AddWithValue("@titulo", umlivro.getTitulo());

strSQL.Parameters.AddWithValue("@autor", umlivro.getAutor());

strSQL.Parameters.AddWithValue("@editora", umlivro.getEditora());

strSQL.Parameters.AddWithValue("@ano", int.Parse(umlivro.getAno()));

strSQL.Parameters.AddWithValue("@codigo", int.Parse(umlivro.getCodigo()));

strSQL.ExecuteNonQuery();

}

public static void consultaUmLivro(Livro umlivro)

{

string aux = "SELECT * FROM TabLivro WHERE codigo = @codigo";

strSQL = new SqlCommand(aux, conn);

strSQL.Parameters.AddWithValue("@codigo", int.Parse(umlivro.getCodigo()));

result = strSQL.ExecuteReader();

Erro.setErro(false);

if (result.Read())

{

umlivro.setTitulo(result.GetString(1));

umlivro.setAutor(result.GetString(2));

umlivro.setEditora(result.GetString(3));

```

```
umlivro.setAno(result.GetInt32(4).ToString());  
  
}  
  
else  
  
{  
  
Erro.setMsg("Livro não cadastrado.");  
  
}  
  
result.Close();  
  
}  
  
}  
  
}
```