

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace BioMed
{
    static class Program
    {
        /// <summary>
        /// Ponto de entrada principal para o aplicativo.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new frmPrincipal());
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using BioMed.Apresentacao;

namespace BioMed
{
    public partial class frmPrincipal : Form
    {
        int mov;
        int movX;
        int movY;

        frmLogin login = new frmLogin();
        public frmPrincipal()
        {
            InitializeComponent();
            login.ShowDialog();

            this.Location = Screen.AllScreens[0].WorkingArea.Location;
        }

        private void picDesbloqueio_Click(object sender, EventArgs e)
        {
            tmrPrincipal.Start();
        }

        private void tmrPrincipal_Tick(object sender, EventArgs e)
        {
            pnlPrincipal.Left -= 7;
            picDesbloqueio.Left -= 7;
            if(pnlPrincipal.Left <= 0)
            {
                tmrPrincipal.Stop();
                tmrBotao.Start();
            }
        }
    }
}

```

```

    }

    private void tmrBotao_Tick(object sender, EventArgs e)
    {
        pnlBotao.Top -= 3;
        if(pnlBotao.Top <= 305)
        {
            tmrBotao.Stop();
            tmrCircular.Start();
        }
    }

    private void tmrCircular_Tick(object sender, EventArgs e)
    {
        pnlHora.Left -= 7;
        if(pnlHora.Left <= 0)
        {
            tmrCircular.Stop();
        }
    }

    private void tmrHora_Tick(object sender, EventArgs e)
    {
        lblSegundos.Text = DateTime.Now.ToString("ss");
        lblHora.Text = DateTime.Now.ToString("HH:mm");
        circularProgressBar1.Value = Convert.ToInt32(lblSegundos.Text);
    }

    private void pnlBarra_MouseDown(object sender, MouseEventArgs e)
    {
        mov = 1;
        movX = e.X;
        movY = e.Y;
    }

    private void pnlBarra_MouseMove(object sender, MouseEventArgs e)
    {
        if (mov == 1)
        {
            this.SetDesktopLocation(MousePosition.X - movX, MousePosition.Y - movY);
        }
    }

    private void pnlBarra_MouseUp(object sender, MouseEventArgs e)
    {
        mov = 0;
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using BioMed.Modelo;

namespace BioMed.Apresentacao
{
    public partial class frmLogin : Form
    {

```

```
int mov;  
int movX;  
int movY;
```

```
public frmLogin()  
{  
    InitializeComponent();  
    this.Location = Screen.AllScreens[0].WorkingArea.Location;  
    if (txbSenha != null)  
    {  
        txbSenha.UseSystemPasswordChar = true;  
        picBloquear.Hide();  
    }  
}
```

```
//=====VISIBILIDADE DO BOTÃO  
BLOQUEAR/DESBLOQUEAR=====
```

```
private void picBloquear_Click(object sender, EventArgs e)  
{  
    if (txbSenha.UseSystemPasswordChar == false)  
    {  
        txbSenha.UseSystemPasswordChar = false;  
        picDesbloquear.Show();  
        picBloquear.Hide();  
        txbSenha.UseSystemPasswordChar = true;  
    }  
}
```

```
private void picDesbloquear_Click(object sender, EventArgs e)  
{  
    if (txbSenha.UseSystemPasswordChar == true)  
    {  
        txbSenha.UseSystemPasswordChar = true;  
        picDesbloquear.Hide();  
        picBloquear.Show();  
        txbSenha.UseSystemPasswordChar = false;  
    }  
}
```

```
//=====LIMPAR CAIXA DE  
TEXTO=====
```

```
private void txbUsuario_Click(object sender, EventArgs e)  
{  
    if (txbUsuario.Text == "Usuário")  
    {  
        txbUsuario.Clear();  
    }  
    pnlUsuario.Size = new Size(195, 2);  
}
```

```
private void txbSenha_Click(object sender, EventArgs e)  
{  
    if (txbSenha.Text == "Senha")  
    {  
        txbSenha.Clear();  
    }  
    pnlSenha.Size = new Size(195, 2);  
}
```

```
private void txbUsuario_Leave(object sender, EventArgs e)  
{  
    if (txbUsuario.Text == "")  
    {  
        txbUsuario.Text = "Usuário";  
    }  
}
```

```

        pnlUsuario.Size = new Size(195, 1);
    }

    private void txbSenha_Leave(object sender, EventArgs e)
    {
        if (txbSenha.Text == "")
        {
            txbSenha.Text = "Senha";
        }
        pnlSenha.Size = new Size(195, 1);
    }

    //=====MOVER
    FORMULÁRIO=====
    private void pnlBarra_MouseDown(object sender, MouseEventArgs e)
    {
        mov = 1;
        movX = e.X;
        movY = e.Y;
    }

    private void pnlBarra_MouseMove(object sender, MouseEventArgs e)
    {
        if (mov == 1)
        {
            this.SetDesktopLocation(MousePosition.X - movX, MousePosition.Y - movY);
        }
    }

    private void pnlBarra_MouseUp(object sender, MouseEventArgs e)
    {
        mov = 0;
    }

    //=====FECHAR/MINIMIZAR
    FORMULÁRIO=====
    private void picFechar_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void picMinimizar_Click(object sender, EventArgs e)
    {
    }

    private void btnEntrar_Click(object sender, EventArgs e)
    {
        Controle controle = new Controle();
        controle.Acessar(txbUsuario.Text, txbSenha.Text);
        if (controle.mensagem.Equals(""))
        {
            if (controle.tem)
            {
                MessageBox.Show("Logado com sucesso", "Entrando", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
                frmCadastroUsu frmU = new frmCadastroUsu();
                frmU.ShowDialog();
            }
            else
            {
                MessageBox.Show("Login não encontrado, verifique login ou senha", "ERRO!",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }

```

```

        else
        {
            MessageBox.Show(controle.mensagem);
        }
    }
}
//=====MÉTODOS=====
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using BioMed.Modelo;

```

```

namespace BioMed.Apresentacao
{

```

```

    public partial class frmCadastroUsu : Form
    {
        string CpfLimpo = "";

```

```

        public frmCadastroUsu()
        {
            InitializeComponent();
        }

```

```

        private void frmCadastroUsu_Load(object sender, EventArgs e)
        {
            //btnCadastrar.Enabled = false;
        }

```

```

        private void btnCadastrar_Click(object sender, EventArgs e)
        {
            string NumeroCPF = mtxCpf.Text;
            string CpfLimpo = NumeroCPF.Replace(".", "").Replace("-", "");
            Controle controle = new Controle();
            String mensagem = controle.Cadastrar(txbNome.Text, txbSenha.Text,
            txbConfirmarSenha.Text, CpfLimpo, cmbNivel.Text, DateTime.Now);
            if (controle.tem)
            {
                MessageBox.Show(mensagem, "Cadastro", MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            }
            else
            {
                MessageBox.Show(controle.mensagem);
            }
        }

```

```

        private void mtxCpf_Validated(object sender, EventArgs e)
        {
            btnCadastrar.Enabled = false;
            if (Estaticos.VerificarCpf(mtxCpf.Text))
            {
                MessageBox.Show("CPF VÁLIDO" + CpfLimpo);
                btnCadastrar.Enabled = true;
            }
            else
            {

```

```

        MessageBox.Show("CPF INVÁLIDO");
    }
}

private void mtxCpf_TextChanged(object sender, EventArgs e)
{
    string NumeroCPF = mtxCpf.Text;
    if (Estaticos.VerificarCpf(NumeroCPF))
    {
        //MessageBox.Show("CPF VÁLIDO");

    }
    else
    {
        //MessageBox.Show("CPF INVÁLIDO");
    }
}

private void AtivarCadastro()
{
    if (txbNome.Text != "" && txbSenha.Text != "" && txbConfirmarSenha.Text != "" && CpfLimpo
    != "" && cmbNivel.Text != "")
    {
        btnCadastrar.Enabled = false;
    }
}
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace BioMed.DAL
{
    public class Conexao
    {
        SqlConnection con = new SqlConnection();

        public Conexao()
        {
            con.ConnectionString = @"Data Source=DESKTOP-PGMSA56;Initial
            Catalog=Usuarios;Integrated Security=True";
        }

        public SqlConnection Conectar()
        {
            if (con.State == System.Data.ConnectionState.Closed)
            {
                con.Open();
            }
            return con;
        }

        public void Desconectar()
        {
            if (con.State == System.Data.ConnectionState.Open)
            {
                con.Close();
            }
        }
    }
}

```



```

        cmd.ExecuteNonQuery();
        con.Desconectar();
        this.mensagem = "Cadastrado com sucesso";
        tem = true;
    }
    catch (SQLException)
    {
        this.mensagem = "Erro com Bando de dados!";
    }
}
else
{
    this.mensagem = "Senhas não correspondem";
}
}
else
{
    this.mensagem = "Já existe um usuário com esse login";
}
return mensagem;
}

public bool ChecarDados(String login)
{
    tem = false;

    cmd.CommandText = "select * from logins where usuario = @login";
    cmd.Parameters.AddWithValue("@login", login);

    try
    {
        cmd.Connection = con.Conectar();
        dr = cmd.ExecuteReader();
        if (dr.HasRows)//foi encontrado?
        {
            tem = true;
        }
        con.Desconectar();
        dr.Close();
    }
    catch (Exception)
    {
        this.mensagem = "Erro com Banco de dados!";
    }
    return tem;
}
}
}
}

```



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using BioMed.DAL;

namespace BioMed.Modelo
{
    public class Controle
    {
        public bool tem;
        public string mensagem = "";

        public bool Acessar(String login, String senha)
        {
            LoginDaoComandos loginDao = new LoginDaoComandos();
            tem = loginDao.VerificarLogin(login, senha);
            if (!loginDao.mensagem.Equals(""))
            {
                this.mensagem = loginDao.mensagem;
            }
            return tem;
        }

        public String Cadastrar(String login, String senha, String confSenha, String cpf, String dominio,
DateTime data)
        {
            LoginDaoComandos loginDao = new LoginDaoComandos();
            this.mensagem = loginDao.Cadastrar(login, senha, confSenha, cpf, dominio, data);
            if (loginDao.tem)
            {
                this.tem = true;
            }
            return mensagem;
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace BioMed.Modelo
{
    public static class Estaticos
    {
        public static bool VerificarCpf(string cpf)
        {

```

```

//=====
=====
        int[] verificarDigito1 = new int[9] { 10, 9, 8, 7, 6, 5, 4, 3, 2 };
        int[] verificarDigito2 = new int[10] { 11, 10, 9, 8, 7, 6, 5, 4, 3, 2 };
        string CPF1, CPF2;
        int soma, resultado;

```

```

//=====
=====
        cpf = cpf.Trim();
        cpf = cpf.Replace(".", "").Replace("-", "").Replace(".", "");

```

```

//=====
=====
    if (cpf.Length != 11)
    {
        return false;
    }

//=====
=====
    CPF1 = cpf.Substring(0, 10);
    soma = 0;
    for (int a = 0; a < 9; a++)
    {
        soma += Convert.ToInt32(CPF1[a].ToString()) * verificarDigito1[a];
    }
    resultado = soma % 11; //Resto da divisão

    if (resultado < 2)
    {
        resultado = 0;
    }
    else
    {
        resultado = 11 - resultado;
    }

//=====
=====
    CPF2 = resultado.ToString(); //Recebe o primeiro digito verificador
    CPF1 = CPF1 + CPF2;

//=====
=====
    soma = 0; //Reseta a variavel soma
    for (int b = 0; b < 10; b++)
    {
        soma += Convert.ToInt32(CPF1[b].ToString()) * verificarDigito2[b];
    }
    resultado = soma % 11;

//=====
=====
    if (resultado < 2)
    {
        resultado = 0;
    }
    else
    {
        resultado = 11 - resultado;
    }

//=====
=====
    CPF2 = CPF2 + resultado.ToString();
    return cpf.EndsWith(CPF2);
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;

```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.IO;

namespace Sistema_Csharp.Cadastro
{
    public partial class Paciente : Form
    {
        public Paciente()
        {
            InitializeComponent();
        }

        private void Button2_Click(object sender, EventArgs e)
        {
            SqlConnection conn = Conexao.obterConexao();

            if (comboBox1.Text != "Homem" || comboBox1.Text != "Mulher")
            {
                MessageBox.Show("Insira um gênero válido.");
                return;
            }

            if (conn == null)
            {
                MessageBox.Show("Não foi possível obter a conexão. Veja o log de erros.");
            }
            else
            {
                string inserir = "insert into PACIENTES
(NOME,RG,CPF,ENDereco,BAIRRO,CIDADE,CEP,DATA_NASCI,RACA,GENERO,IDADE,TELEFONE,PROFI
SSAO,NACIONALIDADE,ESTADO,INSTRUCAO,PAI,MAE,PRIMEIRO_ATENDIMENTO, PRONTUARIO) values
('" + txbNome.Text + "','" +
                txbRG.Text + "','" + txbCPF.Text + "','" + txbEndereco.Text + "','" + txbBairro.Text
+ "','" + txbCidade.Text + "','" + txbCEP.Text + "','" + dtnNascimento.Text + "','" + txbRaca.Text + "'"
+
                "','" + comboBox1.Text + "','" + txbIdade.Text + "','" + txbTelefone.Text + "','" +
txbProfissao.Text + "','" + txbNacionalidade.Text + "','" + txbEstado.Text + "','" + txbInstrucao.Text
+ "','" + txbPai.Text + "','" + txbMae.Text + "','" + dtnAtendimento.Text + "'" +
                "','" + txbProntuario.Text + "')";

                SqlCommand cmdInserir = new SqlCommand(inserir, conn);

                cmdInserir.ExecuteNonQuery();
                MessageBox.Show("Cadastrado com sucesso!");
                this.Close();
            }
        }

        private void pacienteBindingNavigatorSaveItem_Click(object sender, EventArgs e)
        {
            this.Validate();
        }

        private void Paciente_Load(object sender, EventArgs e)
        {
            DateTime inicio = new DateTime();
            DateTime data = DateTime.Now;

```

```

        inicio = Convert.ToDateTime( inicio);
        data = Convert.ToDateTime(data);

        TimeSpan diferenca = data.Subtract(inicio);

        txbProntuario.Text = ((data).ToString());
        // TODO: esta linha de código carrega dados na tabela 'sistemaDataSet.Paciente'. Você pode
        movê-la ou removê-la conforme necessário.
    }

    private void bindingNavigatorDeleteItem_Click(object sender, EventArgs e)
    {

    }

    private void bindingNavigatorAddNewItem_Click(object sender, EventArgs e)
    {

    }

    private void button3_Click(object sender, EventArgs e)
    {
        try
        {

            string conexao = Sistema_Csharp.Properties.Settings.Default.srtConexao;

            SqlConnection objSis = new SqlConnection(conexao);

            objSis.Open();

            string deletar = "delete from Paciente where nome = '" + txbProntuario.Text + "'";

            if (txbProntuario.Text == "")
            {
                MessageBox.Show("Informe um Paciente para deletar");
            }

            else
            {

                SqlCommand cmddeletar = new SqlCommand(deletar, objSis);

                cmddeletar.ExecuteNonQuery();

                MessageBox.Show("Deletado com sucesso !");
                txbProntuario.Clear();
                txbProfissao.Clear();
                txbNome.Clear();
                txbEndereco.Clear();
                txbEstado.Clear();
                txbBairro.Clear();
                txbCidade.Clear();
                txbCEP.Clear();
                txbCelular.Clear();
                txbProntuario.Focus();
            }
        }
    }
}

```

```
    catch (Exception)
    {
        MessageBox.Show("Não deu certo !");
    }
}
```

```
private void bindingNavigatorMoveNextItem_Click(object sender, EventArgs e)
{
}
}
```

```
private void cidadeTextBox_TextChanged(object sender, EventArgs e)
{
}
}
```

```
private void btnSair_Click(object sender, EventArgs e)
{
}
}
```

```
private void bindingNavigatorMoveLastItem_Click(object sender, EventArgs e)
{
}
}
```

```
e) private void pacienteDataGridView_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}
}
```

```
private void bindingNavigatorAddNewItem_Click_1(object sender, EventArgs e)
{
}
}
```

```
private void pacienteBindingSource_CurrentChanged(object sender, EventArgs e)
{
}
}
```

```
private void pacienteBindingNavigatorSaveItem_Click_1(object sender, EventArgs e)
{
    if (txbProntuario.Text == "")
    {
        MessageBox.Show("Informe o Prontuario !");
    }

    else
    {
        this.Validate();
        MessageBox.Show("Cadastrado com sucesso");
        Close();
    }
}
}
```

```
private void bindingNavigatorDeleteItem_Click_1(object sender, EventArgs e)
{
}
}
```

```

private void groupBox1_Enter(object sender, EventArgs e)
{
}

private void label2_Click(object sender, EventArgs e)
{
}

private void nºTextBox_TextChanged(object sender, EventArgs e)
{
}

private void rGTextBox_TextChanged(object sender, EventArgs e)
{
}

private void dtmNascimento_ValueChanged(object sender, EventArgs e)
{
    DateTime inicio = new DateTime();
    DateTime data = DateTime.Now;

    inicio = Convert.ToDateTime(dtmNascimento.Text);
    data = Convert.ToDateTime(data);

    TimeSpan diferenca = data.Subtract(inicio);

    txbIdade.Text = ((diferenca.Days / 365).ToString());
}

private void btnBuscar_Click(object sender, EventArgs e)
{
    SqlConnection conn = Conexao.obterConexao();
    if (conn == null)
    {
        MessageBox.Show("Não foi possível obter a conexão. Veja o log de erros.");
    }
    else
    {
        string sql = "SELECT * FROM PACIENTES WHERE PRONTUARIO=" + txbProntuario.Text;

        SqlConnection con = new SqlConnection(sql);
        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.CommandType = CommandType.Text;
        SqlDataReader reader;
        con.Open();

        try
        {
            reader = cmd.ExecuteReader();
            if (reader.Read())
            {

```

```

txbNome.Text = reader[0].ToString();
txbRG.Text = reader[1].ToString();
txbCPF.Text = reader[2].ToString();
dtnNascimento.Text = reader[3].ToString();
txbEndereco.Text = reader[4].ToString();
txbBairro.Text = reader[5].ToString();
txbCEP.Text = reader[6].ToString();
txbNacionalidade.Text = reader[7].ToString();
txbCidade.Text = reader[8].ToString();
txbRaca.Text = reader[9].ToString();
txbTelefone.Text = reader[10].ToString();
txbCelular.Text = reader[11].ToString();
comboBox1.Text = reader[12].ToString();
txbProfissao.Text = reader[13].ToString();
txbInstrucao.Text = reader[14].ToString();
txbPai.Text = reader[15].ToString();
txbMae.Text = reader[16].ToString();
}

```

```

else

```

```

    MessageBox.Show("Nenhum registro encontrado com o Id informado!");

```

```

}

```

```

    catch (Exception ex)

```

```

    {
        MessageBox.Show("Erro: " + ex.ToString());
    }

```

```

    finally

```

```

    {
        con.Close();
    }

```

```

}

```

```

}

```

```

private void rbHomem_CheckedChanged(object sender, EventArgs e)

```

```

{

```

```

}

```

```

private void rbMulher_CheckedChanged(object sender, EventArgs e)

```

```

{

```

```

}

```

```

private void btnSair_Click_1(object sender, EventArgs e)

```

```

{

```

```

    this.Close();

```

```

}

```

```

private void txbProntuario_TextChanged(object sender, EventArgs e)

```

```

{

```

```

    /* inicio = new DateTime();
    DateTime data = DateTime.Now;

```

```

    inicio = Convert.ToDateTime(dtnNascimento.Text + inicio);
    data = Convert.ToDateTime(data);

```

```

    TimeSpan direnrenca = data.Subtract(inicio);

```

```

    txbProntuario.Text = ((direnrenca.Days).ToString());*/

```

```

}

```

```

}

```

```

}

```

