

SQL

```
create database riverdan
go
use riverdan
go
create table adm
(
  Nome varchar(30)
  Senha varchar(15)
)
go
insert into adm values
('joao','123')
go
create table Aluno
(
  Nome varchar(30)
  Senha varchar(15)
)
go
insert into Aluno values
('joao','123')

create procedure sp_ExcluirValor
  @Nome varchar(40),
  @Mensagem varchar(100) output
as
begin
  delete adm where Senha = @Nome
  set @Mensagem = 'Excluido com sucesso'
end
go
```

```

create procedure sp_ExcluirValorr
@Nome varchar(40),
@Mensagem varchar(100) output
as
begin
delete Aluno where Senha = @Nome
set @Mensagem = 'Excluido com sucesso'
end

```

C#

```

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

            //Application.Run(new Apresentacao.frmCrianca());
            //Application.Run(new Apresentacao.frmCadastrarUsuario());
        }
    }
}

namespace Enfermagem
{
    public partial class frmPrimeiraEscolha : Form
    {
        public frmPrimeiraEscolha()
        {
            InitializeComponent();
        }

        private void btnADM_Click(object sender, EventArgs e)
        {
            Apresentacao.frmLoginADM frp = new Apresentacao.frmLoginADM();
            this.Hide();
            frp.ShowDialog();
        }

        private void btnAluno_Click(object sender, EventArgs e)
        {
            Apresentacao.frmLoginAluno fra = new Apresentacao.frmLoginAluno();
            this.Hide();
            fra.ShowDialog();
        }
    }
}

```

```

private void FrmPrimeiraEscolha_Load(object sender, EventArgs e)
{
    this.Hide();
    Apresentacao.frmFlash frm = new Apresentacao.frmFlash();
    frm.Show();
    frm.Update();
    System.Threading.Thread.Sleep(5100);
    frm.Close();
    this.Visible = true;
}
}
}
namespace Enfermagem.Apresentacao
{
    public partial class frmLoginADM : Form
    {
        public frmLoginADM()
        {
            InitializeComponent();
        }

        private void btnLogar_Click(object sender, EventArgs e)
        {
            // vamos obter a conexão com o banco de dados
            SqlConnection conn = Modelo.Conexao.obterConexao();

            // a conexão foi efetuada com sucesso?
            if (conn == null)
            {
                MessageBox.Show("Não foi possível obter a conexão. Veja o log de
erros.");
            }
            else
            {
                string comando = "select count (*) from adm where Nome = '" +
txbNome.Text + "' and Senha = '" + txbSenha.Text + "'";

                SqlCommand objConsultar = new SqlCommand(comando, conn);

                Modelo.Conexao.Aluno = txbNome.Text;

                //verificador de cadastro, ele conta os registros de executescalar
                if ((int)objConsultar.ExecuteScalar() > 0)
                {
                    Modelo.Estatico.logado = (true);
                    Apresentacao.frmEscolhas objMenu = new Apresentacao.frmEscolhas();
                    this.Hide();
                    Modelo.Estatico.num = "1";
                    objMenu.ShowDialog();
                }
                else
                {
                    MessageBox.Show("Nome ou senha Incorreto !");
                    txbNome.Text = "";
                    txbSenha.Text = "";
                }
            }
        }
    }
}

```

```

        private void btnVoltar_Click(object sender, EventArgs e)
        {
            frmPrimeiraEscolha frp = new frmPrimeiraEscolha();
            this.Hide();
            frp.ShowDialog();
        }
    }
}

namespace Enfermagem.Apresentacao
{
    public partial class frmLoginAluno : Form
    {
        public frmLoginAluno()
        {
            InitializeComponent();
        }

        private void btnLogarAluno_Click(object sender, EventArgs e)
        {
            // vamos obter a conexão com o banco de dados
            SqlConnection conn = Modelo.Conexao.obterConexao();

            // a conexão foi efetuada com sucesso?
            if (conn == null)
            {
                MessageBox.Show("Não foi possível obter a conexão. Veja o log de
erros.");
            }
            else
            {
                string comando = "select count (*) from Aluno where Nome = '" +
txbNomeAluno.Text + "' and Senha = '" + txbSenhaAluno.Text + "'";

                Modelo.Conexao.Aluno = txbNomeAluno.Text;

                SqlCommand objConsultar = new SqlCommand(comando, conn);

                //verificador de cadastro, ele conta os registros de executescalar
                if ((int)objConsultar.ExecuteScalar() > 0)
                {
                    Modelo.Estatico.logado = (true);
                    frmEscolhas objMenu = new frmEscolhas();
                    Modelo.Estatico.num = "2";
                    this.Hide();
                    objMenu.ShowDialog();
                }
                else
                {
                    MessageBox.Show("Usuario ou senha Incorreto !");
                    txbNomeAluno.Text = "";
                    txbSenhaAluno.Text = "";
                }
            }
        }
    }
}

```

```

    }

    private void btnVoltar_Click(object sender, EventArgs e)
    {
        frmPrimeiraEscolha frp = new frmPrimeiraEscolha();
        this.Hide();
        frp.ShowDialog();
    }
}
namespace Enfermagem.Apresentacao
{
    public partial class frmEscolhas : Form
    {
        public frmEscolhas()
        {
            InitializeComponent();
        }

        private void loginToolStripMenuItem1_Click(object sender, EventArgs e)
        {

        }

        private void tsmAlterarLogin_Click(object sender, EventArgs e)
        {

        }

        private void aDMToolStripMenuItem_Click(object sender, EventArgs e)
        {

        }

        private void alunoToolStripMenuItem_Click(object sender, EventArgs e)
        {

        }

        private void tsmMulher_Click(object sender, EventArgs e)
        {
            Apresentacao.frmMulher frm = new Apresentacao.frmMulher();
            this.Hide();
            frm.ShowDialog();
        }

        private void tsmHomem_Click(object sender, EventArgs e)
        {
            Apresentacao.frmHomem frm = new Apresentacao.frmHomem();
            this.Hide();
            frm.ShowDialog();
        }

        private void idosoToolStritsmIdosopMenuItem_Click(object sender, EventArgs e)
        {
            Apresentacao.frmIdoso frm = new Apresentacao.frmIdoso();
            this.Hide();
            frm.ShowDialog();
        }

        private void tsmCriança_Click(object sender, EventArgs e)
        {

```

```

        Apresentacao.frmCrianca frm = new Apresentacao.frmCrianca();
        this.Hide();
        frm.ShowDialog();
    }

    private void tsmGestante_Click(object sender, EventArgs e)
    {
        Apresentacao.frmGestante frm = new Apresentacao.frmGestante();
        this.Hide();
        frm.ShowDialog();
    }

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

    private void alteracoesToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        if (Modelo.Estatico.num != "1")
        {
            MessageBox.Show("Você não tem permissão");
        }
        else
        {
            Apresentacao.frmAlteracao frm = new Apresentacao.frmAlteracao();
            this.Hide();
            frm.ShowDialog();
        }
    }

    private void pacienteToolStripMenuItem1_Click(object sender, EventArgs e)
    {
    }

    private void ADMToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        if (Modelo.Estatico.num != "1")
        {
            MessageBox.Show("Você não tem permissão");
        }
        else
        {
            Apresentacao.frmCadastroADM frm = new Apresentacao.frmCadastroADM();
            this.Hide();
            frm.ShowDialog();
        }
    }

    private void AlunoToolStripMenuItem1_Click(object sender, EventArgs e)
    {
        if (Modelo.Estatico.num != "1")
        {
            MessageBox.Show("Você não tem permissão");
        }
        else
        {

```

```

        Apresentacao.frmCadastrarUsuario frm = new
Apresentacao.frmCadastrarUsuario();
        this.Hide();
        frm.ShowDialog();
    }
}

private void LoginToolStripMenuItem_Click(object sender, EventArgs e)
{
    frmPrimeiraEscolha frm = new frmPrimeiraEscolha();
    this.Hide();
    frm.ShowDialog();
}

private void PacienteToolStripMenuItem_Click(object sender, EventArgs e)
{
    Apresentacao.frmPaciente frm = new Apresentacao.frmPaciente();
    this.Hide();
    frm.ShowDialog();
}

private void FrmEscolhas_Load(object sender, EventArgs e)
{
    lblNome.Text = Modelo.Conexao.Aluno;
}

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

```

```

namespace Enfermagem.Apresentacao
{
    public partial class frmCadastrarUsuario : Form
    {
        public frmCadastrarUsuario()
        {
            InitializeComponent();
        }

        private void btnCadastrar_Click(object sender, EventArgs e)
        {
            // vamos obter a conexão com o banco de dados
            SqlConnection conn = Modelo.Conexao.obterConexao();

            // a conexão foi efetuada com sucesso?
            if (conn == null)
            {
                MessageBox.Show("Não foi possível obter a conexão. Veja o log de
erros.");
            }
            else
            {
                string inserir = "insert into Aluno (Nome, Senha) values ('" +
txbNome.Text + "', '" + txbSenha.Text + "')";

                SqlCommand cmdInserir = new SqlCommand(inserir, conn);

                cmdInserir.ExecuteNonQuery();
            }
        }
    }
}

```

```

        MessageBox.Show("Cadastrado com sucesso!");
        txbNome.Text = "";
        txbSenha.Text = "";
        {
            Listar();
        }

        void Listar()
        {
            {
                DataTable dt = new DataTable();
                SqlDataAdapter da = default(SqlDataAdapter);
                try
                {
                    Modelo.Conexao.obterConexao();
                    da = new SqlDataAdapter("SELECT * FROM Aluno",
Modelo.Conexao.connString);
                    da.Fill(dt);
                    dgvALUNO.DataSource = dt;
                }

                catch (Exception)
                {
                    MessageBox.Show("Erro.");
                    Modelo.Conexao.FecharConexao();
                }
            }
        }
    }

    private void btnVoltar_Click(object sender, EventArgs e)
    {
        frmEscolhas frm = new frmEscolhas();
        this.Hide();
        frm.ShowDialog();
    }

    private void frmCadastrarUsuario_Load(object sender, EventArgs e)
    {
        {
            Listar();
        }

        void Listar()
        {
            {
                DataTable dt = new DataTable();
                SqlDataAdapter da = default(SqlDataAdapter);
                try
                {
                    Modelo.Conexao.obterConexao();
                    da = new SqlDataAdapter("SELECT * FROM Aluno",
Modelo.Conexao.connString);

```



```

        da.Fill(dt);
        dgvALUNO.DataSource = dt;
    }

    catch (Exception)
    {
        MessageBox.Show("Erro.");
        Modelo.Conexao.FecharConexao();
    }
}

}

private void BtnExcluir_Click(object sender, EventArgs e)
{
    SqlCommand cmd = default(SqlCommand);

    if (txbNome.Text != "")
    {
        DialogResult msgSN = MessageBox.Show("Deseja realmente excluir?",
"Aviso", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button3);
        //SE O ESCOLHER SIM FAÇA
        if (msgSN == DialogResult.Yes)
        {
            try
            {
                Modelo.Conexao.obterConexao();
                cmd = new SqlCommand("sp_ExcluirValorr", Modelo.Conexao.conn);
                cmd.CommandType = CommandType.StoredProcedure;

                cmd.Parameters.AddWithValue("@Nome", txbSenha.Text);

                cmd.Parameters.Add("@Mensagem", SqlDbType.VarChar,
100).Direction = (System.Data.ParameterDirection)2;
                cmd.ExecuteNonQuery();

                string msg = cmd.Parameters["@Mensagem"].Value.ToString();
                MessageBox.Show(msg, "", MessageBoxButtons.OK,
MessageBoxIcon.Information, MessageBoxDefaultButton.Button3);

                {
                    Listar();
                }

                void Listar()
                {
                    {
                        DataTable dt = new DataTable();
                        SqlDataAdapter da = default(SqlDataAdapter);
                        try
                        {
                            Modelo.Conexao.obterConexao();
                            da = new SqlDataAdapter("SELECT * FROM Aluno",
Modelo.Conexao.connString);
                            da.Fill(dt);

```

```

        dgvALUNO.DataSource = dt;
    }

    catch (Exception)
    {
        MessageBox.Show("Erro.");
        Modelo.Conexao.FecharConexao();
    }

    }
}

catch (Exception ex)
{
    MessageBox.Show("Erro ao Excluir os dados " + ex.Message);
    Modelo.Conexao.FecharConexao();
}
txbNome.Text = "";
txbSenha.Text = "";

}

}

private void DgvALUNO_DoubleClick(object sender, EventArgs e)
{
    txbNome.Text = dgvALUNO.CurrentRow.Cells[0].Value.ToString();
    txbSenha.Text = dgvALUNO.CurrentRow.Cells[1].Value.ToString();
}

}

namespace Enfermagem.Apresentacao
{
    public partial class frmCadastroADM : Form
    {

        public frmCadastroADM()
        {
            InitializeComponent();
        }

        private void btnCadastrar_Click(object sender, EventArgs e)
        {
            // vamos obter a conexão com o banco de dados
            SqlConnection conn = Modelo.Conexao.obterConexao();

            // a conexão foi efetuada com sucesso?
            if (conn == null)
            {
                MessageBox.Show("Não foi possível obter a conexão. Veja o log de
erros.");
            }
            else
            {

```

```

        string inserir = "insert into adm (Nome, Senha) values ('" +
txbNome.Text + "','" + txbSenha.Text + "')";

        SqlCommand cmdInserir = new SqlCommand(inserir, conn);

        cmdInserir.ExecuteNonQuery();
        MessageBox.Show("Cadastrado com sucesso!");
        txbNome.Text = "";
        txbSenha.Text = "";
        {
            Listar();
        }

        void Listar()
        {
            {
                DataTable dt = new DataTable();
                SqlDataAdapter da = default(SqlDataAdapter);
                try
                {
                    Modelo.Conexao.obterConexao();
                    da = new SqlDataAdapter("SELECT * FROM adm",
Modelo.Conexao.connString);
                    da.Fill(dt);
                    dgvADM.DataSource = dt;

                }

                catch (Exception)
                {
                    MessageBox.Show("Erro.");
                    Modelo.Conexao.FecharConexao();
                }

            }

        }

        private void btnVoltar_Click(object sender, EventArgs e)
        {
            frmEscolhas frm = new frmEscolhas();
            this.Hide();
            frm.ShowDialog();
        }

        private void frmCadastroADM_Load(object sender, EventArgs e)
        {
            {
                Listar();
            }

            void Listar()
            {
                {

```

```

        DataTable dt = new DataTable();
        SqlDataAdapter da = default(SqlDataAdapter);
        try
        {
            Modelo.Conexao.obterConexao();
            da = new SqlDataAdapter("SELECT * FROM adm",
Modelo.Conexao.connString);
            da.Fill(dt);
            dgvADM.DataSource = dt;

        }

        catch (Exception)
        {
            MessageBox.Show("Erro.");
            Modelo.Conexao.FecharConexao();
        }

    }

}

private void BtnExcluir_Click(object sender, EventArgs e)
{
    SqlCommand cmd = default(SqlCommand);

    if (txbNome.Text != "")
    {
        DialogResult msgSN = MessageBox.Show("Deseja realmente excluir?",
"Aviso", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button3);
        //SE O ESCOLHER SIM FAÇA
        if (msgSN == DialogResult.Yes)
        {
            try
            {
                Modelo.Conexao.obterConexao();
                cmd = new SqlCommand("sp_ExcluirValor", Modelo.Conexao.conn);
                cmd.CommandType = CommandType.StoredProcedure;

                cmd.Parameters.AddWithValue("@Nome", txbSenha.Text);

                cmd.Parameters.Add("@Mensagem", SqlDbType.VarChar,
100).Direction = (System.Data.ParameterDirection)2;
                cmd.ExecuteNonQuery();

                string msg = cmd.Parameters["@Mensagem"].Value.ToString();
                MessageBox.Show(msg, "", MessageBoxButtons.OK,
MessageBoxIcon.Information, MessageBoxDefaultButton.Button3);

                {
                    Listar();
                }

                void Listar()
                {

```

```

        {
            DataTable dt = new DataTable();
            SqlDataAdapter da = default(SqlDataAdapter);
            try
            {
                Modelo.Conexao.obterConexao();
                da = new SqlDataAdapter("SELECT * FROM adm",
Modelo.Conexao.connString);

                da.Fill(dt);
                dgvADM.DataSource = dt;

            }

            catch (Exception)
            {
                MessageBox.Show("Erro.");
                Modelo.Conexao.FecharConexao();
            }

        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Erro ao Excluir os dados " + ex.Message);
        Modelo.Conexao.FecharConexao();
    }
    txbNome.Text = "";
    txbSenha.Text = "";

}

}

}

private void DgvADM_DoubleClick(object sender, EventArgs e)
{
    txbNome.Text = dgvADM.CurrentRow.Cells[0].Value.ToString();
    txbSenha.Text = dgvADM.CurrentRow.Cells[1].Value.ToString();
}

private void DgvADM_CellContentClick(object sender, DataGridViewCellEventArgs
e)
{
}

}

}

```