

## Prática 8

- ❖ Copie o banco de dados atividade.mdf para a pasta Debug do projeto
- ❖ Adicione a connectiostring no arquivo app.config

```
<connectionStrings>
  <add name="conexao"
    connectionString="Data
Source=(LocalDB)\v11.0;AttachDbFilename=|DataDirectory|\ativi
dade.mdf;Integrated Security=True;Connect Timeout=30"
    providerName="System.Data.SqlClient" />
</connectionStrings>
```

- ❖ Crie um nome projeto de biblioteca de classe chamado de "Acesso a Dados"
- ❖ Crie a classe DBCliente.cs
- ❖ Crie o método de InserirCliente

```
public void InserirCliente(string _nome, string _idade, string _telefone,
string _endereco, string _sexo)
{
    SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["conexao"].ConnectionStr
ing);
    conn.Open();
    SqlCommand cmd = new SqlCommand("insert into cliente (nome, idade,
telefone, endereco, sexo) values (@nome,@idade,@tel,@end,@sexo);");
    cmd.Connection = conn;
    cmd.Parameters.AddWithValue("@nome", _nome);
    cmd.Parameters.AddWithValue("@idade", _idade);
    cmd.Parameters.AddWithValue("@tel", _telefone);
    cmd.Parameters.AddWithValue("@end", _endereco);
    cmd.Parameters.AddWithValue("@sexo", _sexo);
    cmd.ExecuteNonQuery();
    conn.Close();
}
```

- ❖ Crie o método de AtualizarCliente

```
public void AtualizarCliente(int _codigo, string _nome, int _idade, string
_telefone, string _endereco, char _sexo)
{
    SqlConnection conn = new
SqlConnection(ConfigurationManager.ConnectionStrings["conexao"].ConnectionStr
ing);
    conn.Open();
    SqlCommand cmd = new SqlCommand("update cliente set nome=@nome,
idade=@idade, telefone=@tel, endereco=@end, sexo=@sexo where id=@codigo;");
    cmd.Connection = conn;
    cmd.Parameters.AddWithValue("@codigo", _codigo);
    cmd.Parameters.AddWithValue("@nome", _nome);
    cmd.Parameters.AddWithValue("@idade", _idade);
    cmd.Parameters.AddWithValue("@tel", _telefone);
    cmd.Parameters.AddWithValue("@end", _endereco);
    cmd.Parameters.AddWithValue("@sexo", _sexo);
    cmd.ExecuteNonQuery();
    conn.Close();
}
```

- ❖ Crie o método de ConsultarClientePorNome

```
public DataTable ConsultarTodosClientes()
{
    SqlConnection conn = new
    SqlConnection(ConfigurationManager.ConnectionStrings("conexao").Connection
    String);
    SqlCommand cmd = new SqlCommand();
    string strSql = "select * from cliente where 1=1";
    cmd.CommandText = strSql;
    cmd.Connection = conn;
    conn.Open();
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);
    conn.Close();
    return dt;
}
```

- ❖ Crie o método de ConsultarClientePorNome

```
public DataTable ConsultarClientePorNome(string _nome)
{
    SqlConnection conn = new
    SqlConnection(ConfigurationManager.ConnectionStrings("conexao").Connection
    String);
    SqlCommand cmd = new SqlCommand();
    string strSql = "select * from cliente where nome like '%' + @nome + '%';";
    cmd.CommandText = strSql;
    cmd.Connection = conn;
    conn.Open();
    cmd.Parameters.AddWithValue("@nome", _nome);
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);
    conn.Close();
    return dt;
}
```

- ❖ Crie o método de RemoverCliente

```
public void RemoverCliente(string _nome)
{
    SqlConnection conn = new
    SqlConnection(ConfigurationManager.ConnectionStrings("conexao").Connection
    String);
    conn.Open();
    SqlCommand cmd = new SqlCommand("delete from cliente where nome=@nome;");
    cmd.Connection = conn;
    cmd.Parameters.AddWithValue("@nome", _nome);
    cmd.ExecuteNonQuery();
    conn.Close();
}
```

- ❖ Altere o evento do botão Salvar do form `InserirCliente` para chamar o método `InserirCliente` e salvar no banco

```
try
```

```
{
    DBCliente dbCli = new DBCliente();
    dbCli.InserirCliente(txtNome.Text.Trim(),
    Convert.ToInt32(nuIdade.Value.ToString()), mskTelefone.Text.Trim(),
    txtEndereco.Text.Trim(), (rbMasculino.Checked ? 'M' : 'F'));

    MessageBox.Show("Cliente cadastrado com sucesso!");
}
catch (Exception)
{
    MessageBox.Show("Erro ao cadastrar cliente!");
    throw;
}
```

- ❖ Altere o evento do botão Pesquisar do form **CadastrarCliente** para chamar o método ConsultarTodosClientes e ConsultarTodosClientesPorNome e salvar no banco

```
DBCliente dbCli = new DBCliente();
if (string.IsNullOrEmpty(txtNome.Text))
{
    gwCliente.DataSource = dbCli.ConsultarTodosClientes();
}
else
{
    gwCliente.DataSource = dbCli.ConsultarClientePorNome(txtNome.Text);
}
```

- ❖ Altere o evento do botão Remover do form **CadastrarCliente** para chamar o método RemoverCliente e salvar no banco

```
DBCliente dbCli = new DBCliente();
DataGridViewRow row = gwCliente.CurrentRow;
if (!string.IsNullOrEmpty(row.Cells[1].Value.ToString()))
{
    dbCli.RemoverCliente(row.Cells[1].Value.ToString());
}
btConsultar_Click(sender, e);
MessageBox.Show("Cliente removido com sucesso!");
```

- ❖ Sobrecarregue o Construtor do form **InserirCliente** para receber os dados dos clientes para alteração

```
public InserirCliente(Cliente cli)
{
    InitializeComponent();
    txtCodigo.Text = cli.Codigo.ToString();
    nuIdade.Value = cli.Idade;
    txtEndereco.Text = cli.Endereco;
    txtNome.Text = cli.Nome;
    mskTelefone.Text = cli.Telefone;
    if (cli.Sexo.Equals('M'))
        rbMasculino.Checked = true;
    else
        rbFeminino.Checked = true;
}
```

- ❖ Altere o evento do botão Alterar do form **CadastrarCliente** para chamar o form InserirCliente passando os dados do cliente selecionado no GridView

```
private void btAlterar_Click(object sender, EventArgs e)
```

```
{
    DataGridViewRow row = gwCliente.CurrentRow;
    Cliente cli = new Cliente();
    cli.Codigo = Convert.ToInt16(row.Cells[0].Value);
    cli.Nome = row.Cells[1].Value.ToString();
    cli.Idade = Convert.ToInt16(row.Cells[2].Value);
    cli.Telefone = row.Cells[3].Value.ToString();
    cli.Endereco = row.Cells[4].Value.ToString();
    cli.Sexo = Convert.ToChar(row.Cells[5].Value);
    try
    {
        if (!string.IsNullOrEmpty(row.Cells[1].Value.ToString()))
        {
            InserirCliente formInserirCliente = new InserirCliente(cli);
            formInserirCliente.MdiParent = this.MdiParent;
            formInserirCliente.Show();
        }
    }
    catch (Exception)
    {
        MessageBox.Show("Erro na pesquisa de Cliente");
        throw;
    }
}
```

- ❖ Altere o evento do botão Salvar do form `InserirCliente` para chamar o método `AtualizarCliente` e salvar no banco

```
private void btSalvar_Click(object sender, EventArgs e)
{
    DBCliente dbCli = new DBCliente();
    try
    {
        if (txtCodigo.Text == "")
            dbCli.InserirCliente(txtNome.Text.Trim(),
            Convert.ToInt32(nuIdade.Value.ToString()), mskTelefone.Text.Trim(),
            txtEndereco.Text.Trim(), (rbMasculino.Checked ? 'M' : 'F'));
        else
            dbCli.AtualizarCliente(Convert.ToInt16(txtCodigo.Text),
            txtNome.Text.Trim(), Convert.ToInt32(nuIdade.Value.ToString()),
            mskTelefone.Text.Trim(), txtEndereco.Text.Trim(), (rbMasculino.Checked ? 'M' : 'F'));
        MessageBox.Show("Cliente cadastrado com sucesso!");
    }
    catch (SQLException odbeEx)
    {
        MessageBox.Show("Erro ao gravar cadastrar cliente!");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Erro ao cadastrar cliente!");
    }
}
```