**TUTORIAL DE CRIAÇÃO DE UM CRUD COM BASE NO SQL SERVER E WINDOWS FORMS ( C# )**

**TABELA BASE: CLIENTES**

**Telinha – Parque de diversões**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**F7 – Para exibir o código C# ( Caverna do Dragão )**

Na região do “using”, acrescentar o seguinte código

using System.Configuration;

using System.Data.SqlClient;

Logo abaixo da definição da classe, inserir esse código (conexão com o banco de dados)

private string conexao =

ConfigurationManager.ConnectionStrings[2].ConnectionString;

// Variável de controle para definir INSERT ou UPDATE

bool novo;

Região dos “Métodos Criados”

private void resetaTela()

{

// somente o botão Adicionar pode ser clicado

btnAdicionar.Enabled = true;

btnSalvar.Enabled = false;

btnCancelar.Enabled = false;

btnExcluir.Enabled = false;

lblCodigo.Text = string.Empty;

foreach (Control control in this.Controls)

{

if (control is TextBox)

{

TextBox textBox = (TextBox)control;

textBox.Text = string.Empty;

textBox.Enabled = false;

}

}

txtPesquisa.Enabled = true;

}

No método carregaGrid ajustar a instrução SQL com base na tabela que você programar

private void carregaGrid()

{

string sql = "SELECT codcli, nome ,cpf ,rg, rua, telefone, cidade, cep, bairro, estado " +

"FROM Clientes WHERE (nome + cpf + rg + rua + telefone + cidade + cep + bairro + estado " +

"LIKE '%' + @PESQUISA + '%')";

// cria objeto da classe de Conexão com o BD

SqlConnection con = new SqlConnection(conexao);

// cria objeto da classe de Comandos ( executa comandos SQL no BD )

SqlCommand cmd = new SqlCommand(sql, con);

// adiciona paramento requerido na SQL criada, e o parametro vem do que foi digitado na Pesquisa

cmd.Parameters.AddWithValue("@PESQUISA", txtPesquisa.Text);

// configura o tipo de comando para texto

cmd.CommandType = CommandType.Text;

// abre conexão com o BD

con.Open();

try

{

// cria objeto clone de tabela do BD

DataTable table = new DataTable();

// carrega o objeto com o Select que a SQL pediu para o BD

table.Load(cmd.ExecuteReader());

// associa a tabela gerada ao datagrid

dgvDados.DataSource = table;

}

catch (Exception ex)

{

// só passa por aqui se der erro no bloco do Try acima

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

}

finally

{

// incondicionamente passa por aqui para fechar a conexão com o BD

con.Close();

novo = false;

}

}

private void habilitaCaixasDeTexto()

{

foreach (Control control in this.Controls)

{

if (control is TextBox)

{

TextBox textBox = (TextBox)control;

textBox.Text = string.Empty;

textBox.Enabled = true;

}

}

}

Região dos “Botões”

**Lembre-se que para criar os códigos dos botões você precisa clicar 2 vezes no botão para o Visual Studio gerar o evento click**

private void btnAdicionar\_Click(object sender, EventArgs e)

{

// Ajustes dos botões ligado/desligado

btnAdicionar.Enabled = false;

btnSalvar.Enabled = true;

btnCancelar.Enabled = true;

btnExcluir.Enabled = false;

// Habilitando todas as caixas de texto

habilitaCaixasDeTexto();

// Posiciona o cursor na caixa de texto indicada

txtNome.Focus();

// Configura o controle para inserir um novo cadastro

novo = true;

}

private void btnExcluir\_Click(object sender, EventArgs e)

{

DialogResult result = MessageBox.Show("Você confirma a exclusão do registro",

"Excluindo registro...", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

string sql = "DELETE FROM Clientes WHERE codcli = @CODCLI";

SqlConnection con = new SqlConnection(conexao);

// cria objeto da classe de Comandos ( executa comandos SQL no BD )

SqlCommand cmd = new SqlCommand(sql, con);

// adiciona paramento requerido na SQL criada, e o parametro vem do que foi digitado na Pesquisa

cmd.Parameters.AddWithValue("@CODCLI", lblCodigo.Text);

// configura o tipo de comando para texto

cmd.CommandType = CommandType.Text;

// abre conexão com o BD

con.Open();

try

{

int i = cmd.ExecuteNonQuery();

if (i > 0)

{

MessageBox.Show("Registro excluido com sucesso!",

"Informação", MessageBoxButtons.OK,

MessageBoxIcon.Exclamation

);

}

}

catch (Exception ex)

{

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

}

finally

{

con.Close();

}

carregaGrid();

resetaTela();

}

}

private void btnSalvar\_Click(object sender, EventArgs e)

{

string sql;

// Verifica se todos os campos obrigatórios foram preenchidos antes de atualizar

if (novo)

{

sql = "INSERT INTO Clientes (nome, cpf, rg, rua, telefone, cidade, cep, bairro, estado) " +

"VALUES (@NOME, @CPF, @RG, @RUA, @TELEFONE, @CIDADE, @CEP, @BAIRRO, @ESTADO)";

}

else

{

sql = "UPDATE Clientes SET nome = @NOME, cpf = @CPF, rg = @RG, rua = @RUA, " +

"telefone = @TELEFONE, cidade = @CIDADE, cep = @CEP, bairro = @BAIRRO, estado = @ESTADO " +

"WHERE codcli = @CODCLI";

}

// SQL para atualizar o registro na tabela Clientes

using (SqlConnection con = new SqlConnection(conexao))

{

using (SqlCommand cmd = new SqlCommand(sql, con))

{

// Adiciona os parâmetros necessários na SQL criada

cmd.Parameters.AddWithValue("@NOME", txtNome.Text);

cmd.Parameters.AddWithValue("@CPF", txtCpf.Text);

cmd.Parameters.AddWithValue("@RG", txtRg.Text);

cmd.Parameters.AddWithValue("@RUA", txtRua.Text);

cmd.Parameters.AddWithValue("@TELEFONE", txtTelefone.Text);

cmd.Parameters.AddWithValue("@CIDADE", txtCidade.Text);

cmd.Parameters.AddWithValue("@CEP", txtCep.Text);

cmd.Parameters.AddWithValue("@BAIRRO", txtBairro.Text);

cmd.Parameters.AddWithValue("@ESTADO", txtEstado.Text);

if (int.TryParse(lblCodigo.Text, out int n))

{

cmd.Parameters.AddWithValue("@CODCLI", n);

}

// Abre conexão com o BD

con.Open();

try

{

// Executa a consulta SQL (atualização)

int rowsAffected = cmd.ExecuteNonQuery();

if (rowsAffected > 0)

{

MessageBox.Show("Registro adicionado/atualizado com sucesso!", "Informação",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Nenhum registro adicionado/atualizado.", "Informação",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show("Erro ao atualizar o registro: " + ex.ToString());

}

}

}

// Após atualizar, recarrega o grid e reseta a tela para poder cadastrar um novo cliente ou realizar nova pesquisa

carregaGrid();

resetaTela();

}

private void btnCancelar\_Click(object sender, EventArgs e)

{

carregaGrid();

resetaTela();

}

Região dos “Form, Caixas de Texto e GridView”

**Clique 2 vezes para gerar o LOAD**

private void frmBco4\_Load(object sender, EventArgs e)

{

resetaTela();

carregaGrid();

}

**Selecione a caixa de texto, depois clique no botão do RAIO, escolha o evento “TextChanged” e clique 2 vezes**

Desenho feito a mão

Descrição gerada automaticamente com confiança média

private void txtPesquisa\_TextChanged(object sender, EventArgs e)

{

carregaGrid();

}

**Selecione a datagridview, depois clique no botão do RAIO, escolha o evento “CellClick” e clique 2 vezes**

Desenho feito a mão

Descrição gerada automaticamente com confiança média

private void dgvDados\_CellClick(object sender, DataGridViewCellEventArgs e)

{

// Ajustes dos botões ligado/desligado

btnAdicionar.Enabled = false;

btnSalvar.Enabled = true;

btnCancelar.Enabled = true;

btnExcluir.Enabled = true;

// Habilitando todas as caixas de texto

habilitaCaixasDeTexto();

// Posiciona o cursor na caixa de texto indicada

txtNome.Focus();

// Utiliza uma variável para armazenar o índice da linha selecionada

int selectedIndex = e.RowIndex;

if (selectedIndex >= 0 && selectedIndex < dgvDados.Rows.Count)

{

lblCodigo.Text = dgvDados.Rows[selectedIndex].Cells[0].Value.ToString();

txtNome.Text = dgvDados.Rows[selectedIndex].Cells[1].Value.ToString();

txtCpf.Text = dgvDados.Rows[selectedIndex].Cells[2].Value.ToString();

txtRg.Text = dgvDados.Rows[selectedIndex].Cells[3].Value.ToString();

txtRua.Text = dgvDados.Rows[selectedIndex].Cells[4].Value.ToString();

txtTelefone.Text = dgvDados.Rows[selectedIndex].Cells[5].Value.ToString();

txtCidade.Text = dgvDados.Rows[selectedIndex].Cells[6].Value.ToString();

txtCep.Text = dgvDados.Rows[selectedIndex].Cells[7].Value.ToString();

txtBairro.Text = dgvDados.Rows[selectedIndex].Cells[8].Value.ToString();

txtEstado.Text = dgvDados.Rows[selectedIndex].Cells[9].Value.ToString();

}

}