

```
using System;
using System.Windows.Forms;
//Referências às camadas
using Negocios;
using ObjetoTransferencia;

namespace Apresentacao
{
    public partial class FrmClienteCadastrarAlterar : Form
    {
        string tipoDaAcaoGloal; //Verificar se os dados serão do tipo Inserir ou Alterar
        int codClienteGlobal; //Código para consultas de telefone

        //Método Construtor Principal
        public FrmClienteCadastrarAlterar(string tipoDaAcao, Cliente cliente)
        {
            InitializeComponent();
            //Evita que o DataGridView gere as colunas automaticamente com o retorno do BD
            dgvTelefone.AutoGenerateColumns = false;

            tipoDaAcaoGloal = tipoDaAcao;

            if (cliente == null)
            {
                codClienteGlobal = 0;
            }
            else
            {
                codClienteGlobal = cliente.CodCliente;
            }

            if (tipoDaAcao == "inserir")
            {
                this.Text = "Inserir Cliente";
            }
            else
            {
                this.Text = "Alterar Cliente";

                txtCodigo.Text = cliente.CodCliente.ToString();
                txtNome.Text = cliente.Nome;
                maskCPF.Text = cliente.CPF;
                dtpNascimento.Value = cliente.Nascimento;
            }
        }

        //Atualiza o comboBox com os dados vindos do BD (dbLookup)
        private void AtualizaGridTelefoneTipo()
        {
            NegociosTelefone negociosTelefone = new NegociosTelefone();

            TelefoneTipoColecao telefoneTipoColecao =
```

```
        negociosTelefone.ConsultaTelefoneTipo();
        //Limpa a referência de DataSource
        cbxTelefoneTipo.DataSource = null;
        //Atribui um novo DataSource
        cbxTelefoneTipo.DataSource = telefoneTipoColecao;
        //Exibição na comboBox
        cbxTelefoneTipo.DisplayMember = "Descricao";
        //Valor que será gravado como Chave Estrangeira
        cbxTelefoneTipo.ValueMember = "Codigo";
        //Atualiza a lista
        dgvTelefone.Update();
        //Atualiza a exibição
        dgvTelefone.Refresh();
    }

    //Atualiza os telefones do Cliente
    private void AtualizaGridTelefone()
    {
        NegociosTelefone negociosTelefone = new NegociosTelefone();

        TelefoneColecao TelefoneColecao = negociosTelefone.ConsultaTelefone  ➤
            (codClienteGlobal);

        dgvTelefone.DataSource = null;
        dgvTelefone.DataSource = TelefoneColecao;
        dgvTelefone.Update();
        dgvTelefone.Refresh();

        if (TelefoneColecao.Count <= 0)
        {
            MessageBox.Show("Não há registros para exibição!", "Consulta",  ➤
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }

    private void btnCancelar_Click(object sender, EventArgs e)
    {
        Close();
    }

    private void btnSalvar_Click(object sender, EventArgs e)
    {
        //Cria um novo objeto
        Cliente cliente = new Cliente();
        //Preenche os dados comuns entre Inserir e Alterar
        cliente.Nome = txtNome.Text.Trim();
        cliente.CPF = maskCPF.Text;
        cliente.Nascimento = dtpNascimento.Value;

        NegociosCliente NegociosCliente = new NegociosCliente();

        if (tipoDaAcaoGlobal == "inserir")
        {
            string retorno = NegociosCliente.InserirCliente(cliente);

            try
```

```
{
    int num = Convert.ToInt32(retorno);
    MessageBox.Show("Registro inserido com sucesso!");
    codClienteGlobal = num;
    txtCodigo.Text = num.ToString();
    groupDados.Enabled = false;
}
catch
{
    MessageBox.Show("Erro ao inserir o registro!\n" + retorno);
}
}
else
{
    //Preenche o código do cliente que servirá como parâmetro na alteração
    cliente.CodCliente = Convert.ToInt32(txtCodigo.Text);

    string retorno = NegociosCliente.AlterarCliente(cliente);

    try
    {
        int num = Convert.ToInt32(retorno);
        MessageBox.Show("Registro alterado com sucesso!");
    }
    catch
    {
        MessageBox.Show("Erro ao alterar o registro!\n" + retorno);
    }
}
}

//Ao fazer a leitura do Formulário
private void FrmClienteCadastrarAlterar_Load(object sender, EventArgs e)
{
    //Atualiza a comboBox com os tipos de telefones
    AtualizaGridTelefoneTipo();
    //Se houver um código de cliente atualizar seus telefones
    if (codClienteGlobal != 0)
    {
        AtualizaGridTelefone();
    }
}

private void btnIncluirTelefone_Click(object sender, EventArgs e)
{
    Telefone telefone = new Telefone();

    telefone.CodCliente = codClienteGlobal;
    telefone.CodTipoTelefone = Convert.ToInt32
        (cbxTelefoneTipo.SelectedValue);
    telefone.Numero = maskTelefone.Text;

    NegociosTelefone negociosTelefone = new NegociosTelefone();
```

```
string retorno = negociosTelefone.InserirTelefone(telefone);

try
{
    int num = Convert.ToInt32(retorno);
    MessageBox.Show("Registro inserido com sucesso!");

    AtualizaGridTelefone();
    maskTelefone.Clear();
}
catch
{
    //Tratamento da mensagem de erro ao usuário
    if (retorno.Contains("restrição do FOREIGN KEY"))
    {
        MessageBox.Show("Erro ao inserir o registro!\n\nÉ necessário  
casar um usuário antes dos seus telefones.",  
"Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        MessageBox.Show("Erro ao inserir o registro!\n" + retorno);
    }
}

}

private void btnExcluirTelefone_Click(object sender, EventArgs e)
{
    if (dgvTelefone.SelectedRows.Count <= 0)
    {
        MessageBox.Show("Não há registro selecionado!", "Excluir",  
MessageButtons.OK, MessageBoxIcon.Information);
        return;
    }

    if (MessageBox.Show("Excluir registro?", "Excluir",  
MessageButtons.YesNo, MessageBoxIcon.Question) ==  
DialogResult.Yes)
    {
        Telefone telefoneClick = dgvTelefone.SelectedRows[0].DataBoundItem  
as Telefone;

        NegociosTelefone negociosTelefone = new NegociosTelefone();

        string retorno = negociosTelefone.ExcluirTelefone(telefoneClick);

        try
        {
            int num = Convert.ToInt32(retorno);
            MessageBox.Show("Registro excluído com sucesso!", "Excluir",  
MessageButtons.OK, MessageBoxIcon.Information);
            AtualizaGridTelefone();
        }
        catch
        {

```

```
        MessageBox.Show("Erro ao excluir o registro!", "Excluir",  
        MessageBoxButtons.OK, MessageBoxIcon.Information);  
    }  
}  
}
```