



```
Comando = new MySqlCommand("CREATE TABLE IF NOT EXISTS Categorias " +  
    "(id integer auto_increment primary key, " +  
    "categoria char(20))", Conexao);
```

```
Comando.ExecuteNonQuery();
```

```
Comando = new MySqlCommand("CREATE TABLE IF NOT EXISTS Clientes " +  
    "(Id integer auto_increment primary key, " +  
    "nome char(40), " +  
    "idCidade integer," +  
    "dataNasc date," +  
    "renda decimal(10,2), " +  
    "cpf char(14), " +  
    "foto varchar(100), " +  
    "venda boolean)", Conexao);
```

```
Comando.ExecuteNonQuery();
```

```
using MySql.Data.MySqlClient;
using System;
using System.Data;
using System.Windows.Forms;

namespace vendas.Models
{
    7 referências
    public class Cliente
    {
        4 referências
        public int id { get; set; }
        6 referências
        public string nome { get; set; }
        4 referências
        public int idCidade { get; set; }
        4 referências
        public DateTime dataNasc { get; set; }
        4 referências
        public double renda { get; set; }
        4 referências
        public string cpf { get; set; }
        4 referências
        public string foto { get; set; }
        4 referências
        public bool venda { get; set; }
    }
}
```

```
public void Incluir()
{
    try
    {
        Banco.Conexao.Open();
        Banco.Comando = new MySqlCommand
            ("INSERT INTO clientes (nome, idCidade, dataNasc, renda, cpf, foto, venda) " +
            "VALUES (@nome, @idCidade, @dataNasc, @renda, @cpf, @foto, @venda)", Banco.Conexao);
        Banco.Comando.Parameters.AddWithValue("@nome", nome);
        Banco.Comando.Parameters.AddWithValue("@idCidade", idCidade);
        Banco.Comando.Parameters.AddWithValue("@dataNasc", dataNasc);
        Banco.Comando.Parameters.AddWithValue("@renda", renda);
        Banco.Comando.Parameters.AddWithValue("@cpf", cpf);
        Banco.Comando.Parameters.AddWithValue("@foto", foto);
        Banco.Comando.Parameters.AddWithValue("@venda", venda);
        Banco.Comando.ExecuteNonQuery();
        Banco.Conexao.Close();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

```
public void Alterar()
{
    try
    {
        Banco.Conexao.Open();
        Banco.Comando = new MySqlCommand
        ("UPDATE clientes SET nome = @nome, idCidade = @idCidade, dataNasc = @dataNasc, " +
        "renda = @renda, cpf = @cpf, foto = @foto, venda = @venda where id = @id", Banco.Conexao);
        Banco.Comando.Parameters.AddWithValue("@nome", nome);
        Banco.Comando.Parameters.AddWithValue("@idCidade", idCidade);
        Banco.Comando.Parameters.AddWithValue("@dataNasc", dataNasc);
        Banco.Comando.Parameters.AddWithValue("@renda", renda);
        Banco.Comando.Parameters.AddWithValue("@cpf", cpf);
        Banco.Comando.Parameters.AddWithValue("@foto", foto);
        Banco.Comando.Parameters.AddWithValue("@venda", venda);
        Banco.Comando.Parameters.AddWithValue("@id", id);
        Banco.Comando.ExecuteNonQuery();
        Banco.Conexao.Close();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

```
public void Excluir()
{
    try
    {
        Banco.Conexao.Open();
        Banco.Comando = new MySqlCommand("delete from clientes where id = @id", Banco.Conexao);
        Banco.Comando.Parameters.AddWithValue("@id", id);
        Banco.Comando.ExecuteNonQuery();
        Banco.Conexao.Close();
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

```
public DataTable Consultar()
{
    try
    {
        Banco.Comando = new MySqlCommand("SELECT cl.*, ci.nome cidade, " +
            "ci.uf FROM Clientes cl inner join Cidades ci on (ci.id = cl.idCidade) " +
            "where cl.nome like ?Nome order by cl.nome", Banco.Conexao);
        Banco.Comando.Parameters.AddWithValue("@Nome", nome + "%");
        Banco.Adaptador = new MySqlDataAdapter(Banco.Comando);
        Banco.datTabela = new DataTable();
        Banco.Adaptador.Fill(Banco.datTabela);
        return Banco.datTabela;
    }
    catch (Exception e)
    {
        MessageBox.Show(e.Message, "Erro", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return null;
    }
}
```


Cadastro de Clientes

Código: ☒ **chkVenda** Bloqueia a venda?

Nome:

Cidade: UF:

CPF: Data de Nascimento: **dtpDataNasc**

Renda:

Incluir Alterar Cancelar Excluir Fechar

Digite o nome do Cliente para Pesquisa:

btnConsultar



ComboBox



MaskedTextBox








DateTimePicker








PictureBox

(.). MaskedTextBox

Propriedades	
mskCPF System.Windows.Forms.MaskedTextBox	
    	
ImeMode	NoControl
InsertKeyMode	Default
Mask	999.999.999-99
PasswordChar	
ReadOnly	False

 DateTimePicker

Propriedades	
dtpDataNasc System.Windows.Forms.DateTimePicker	
    	
Enabled	True
Font	Microsoft Sans Serif;
Format	Short
GenerateMember	True

```
using System;
using System.Data;
using System.Windows.Forms;
using vendas.Models;

namespace vendas.Views
{
    5 referências
    public partial class FrmClientes : Form
    {
        Cidade ci;
        Cliente cl;

        1 referência
        public FrmClientes()
        {
            InitializeComponent();
        }
    }
}
```

5 referências

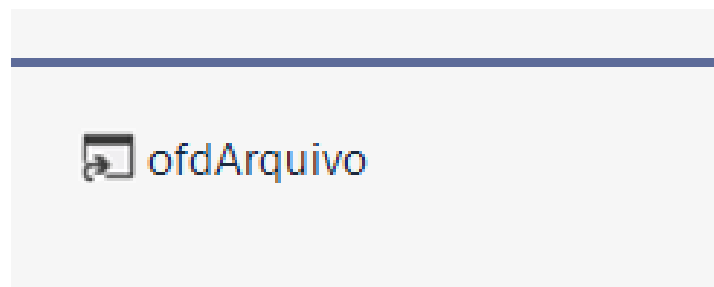
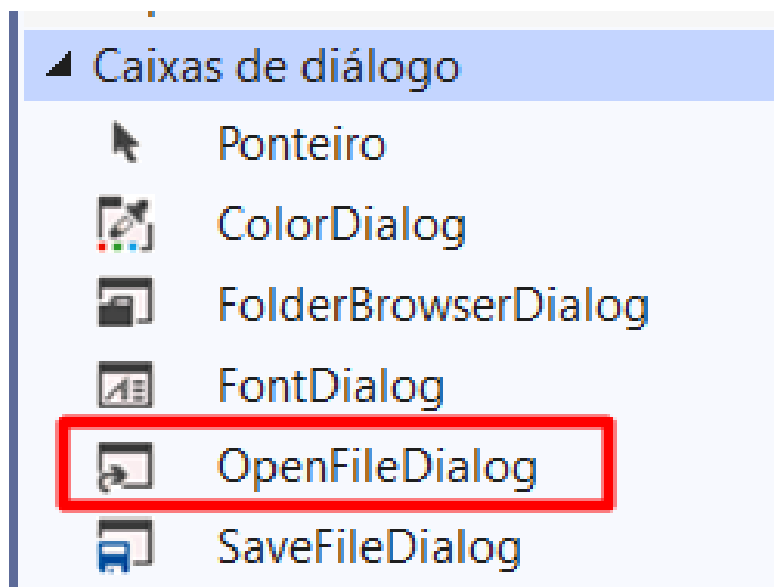
```
void limpaControles()  
{  
    txtId.Clear();  
    txtNome.Clear();  
    cboCidades.SelectedIndex = -1;  
    txtUF.Clear();  
    mskCPF.Clear();  
    txtRenda.Clear();  
    dtpDataNasc.Value = DateTime.Now;  
    picFoto.ImageLocation = "";  
    chkVenda.Checked = false;  
}
```

```
private void FrmClientes_Load(object sender, EventArgs e)
{
    // Cria um objeto do tipo cidade
    // E alimenta o comboBox
    ci = new Cidade();
    cboCidades.DataSource = ci.Consultar();
    cboCidades.DisplayMember = "nome";
    cboCidades.ValueMember = "id";

    limpaControles();
    carregarGrid("");

    // Deixa invisível colunas do Grid
    DgvClientes.Columns["idCidade"].Visible = false;
    DgvClientes.Columns["foto"].Visible = false;
}
```

```
private void cboCidades_SelectedIndexChanged(object sender, EventArgs e)
{
    if (cboCidades.SelectedIndex != -1)
    {
        DataRowView reg = (DataRowView)cboCidades.SelectedItem;
        txtUF.Text = reg["uf"].ToString();
    }
}
```



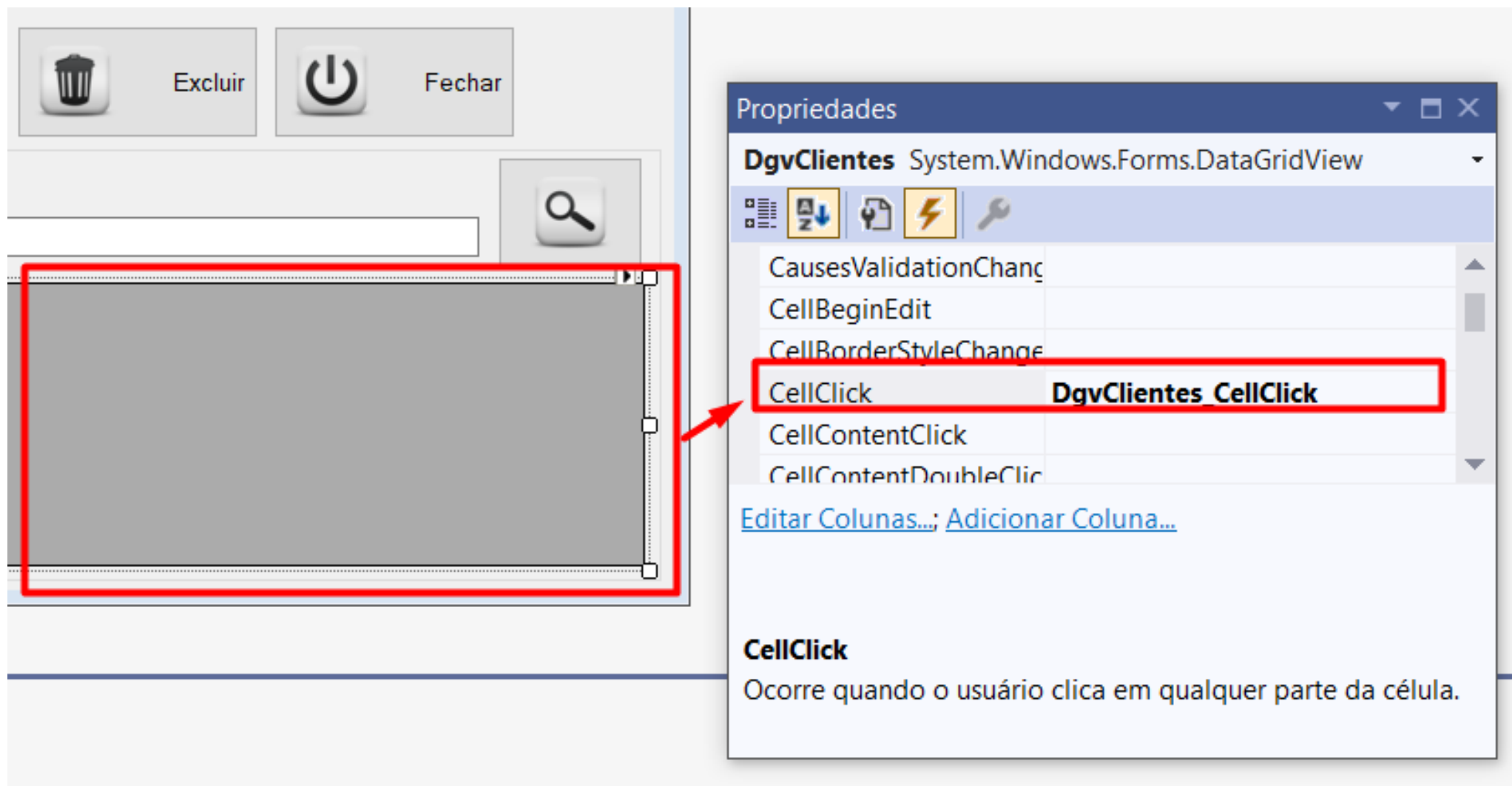
```
private void picFoto_Click(object sender, EventArgs e)
{
    ofdArquivo.InitialDirectory = "D:/fotos/clientes/";
    ofdArquivo.FileName = "";
    ofdArquivo.ShowDialog();
    picFoto.ImageLocation = ofdArquivo.FileName;
}
```



```
private void btnIncluir_Click(object sender, EventArgs e)
{
    if (txtNome.Text == "") return;

    cl = new Cliente()
    {
        nome = txtNome.Text,
        idCidade = (int)cboCidades.SelectedValue,
        dataNasc = dtpDataNasc.Value,
        renda = double.Parse(txtRenda.Text),
        cpf = mskCPF.Text,
        foto = picFoto.ImageLocation,
        venda = chkVenda.Checked
    };
    cl.Incluir();

    limpaControles();
    carregarGrid("");
}
```



The image shows a Windows Forms application interface. At the top, there are two buttons: 'Excluir' (with a trash icon) and 'Fechar' (with a power icon). Below these is a search bar with a magnifying glass icon. The main area contains a large, empty rectangular box, which is a DataGridView control, outlined with a red rectangle. To the right of the DataGridView is the 'Propriedades' (Properties) window. The 'DgvClientes' control is selected, and its type is 'System.Windows.Forms.DataGridView'. The 'CellClick' event is highlighted with a red rectangle. Below the event list, there are links for 'Editar Colunas...' and 'Adicionar Coluna...'. The 'CellClick' event description is shown at the bottom of the Properties window.

Propriedades

DgvClientes System.Windows.Forms.DataGridView

CausesValidationChang
CellBeginEdit
CellBorderStyleChange
CellClick DgvClientes CellClick
CellContentClick
CellContentDoubleClick

[Editar Colunas...](#); [Adicionar Coluna...](#)

CellClick
Ocorre quando o usuário clica em qualquer parte da célula.

```
private void DgvClientes_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if (DgvClientes.RowCount > 0)
    {
        txtId.Text = DgvClientes.CurrentRow.Cells["id"].Value.ToString();
        txtNome.Text = DgvClientes.CurrentRow.Cells["nome"].Value.ToString();
        cboCidades.Text = DgvClientes.CurrentRow.Cells["cidade"].Value.ToString();
        txtUF.Text = DgvClientes.CurrentRow.Cells["uf"].Value.ToString();
        chkVenda.Checked = (bool)DgvClientes.CurrentRow.Cells["venda"].Value;
        mskCPF.Text = DgvClientes.CurrentRow.Cells["cpf"].Value.ToString();
        dtpDataNasc.Text = DgvClientes.CurrentRow.Cells["dataNasc"].Value.ToString();
        txtRenda.Text = DgvClientes.CurrentRow.Cells["renda"].Value.ToString();
        picFoto.ImageLocation = DgvClientes.CurrentRow.Cells["foto"].Value.ToString();
    }
}
```

```
private void btnAlterar_Click(object sender, EventArgs e)
{
    if (txtId.Text == "") return;

    cl = new Cliente()
    {
        id = int.Parse(txtId.Text),
        nome = txtNome.Text,
        idCidade = (int)cboCidades.SelectedValue,
        dataNasc = dtpDataNasc.Value,
        renda = double.Parse(txtRenda.Text),
        cpf = mskCPF.Text,
        foto = picFoto.ImageLocation,
        venda = chkVenda.Checked
    };
    cl.Alterar();

    limpaControles();
    carregarGrid("");
}
```

```
private void btnExcluir_Click(object sender, EventArgs e)
{
    if (txtId.Text == "") return;

    if (MessageBox.Show("Deseja excluir o cliente?", "Exclusão",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        cl = new Cliente()
        {
            id = int.Parse(txtId.Text)
        };
        cl.Excluir();

        limpaControles();
        carregarGrid("");
    }
}
```

```
private void btnCancelar_Click(object sender, EventArgs e)
{
    limpaControles();
    carregarGrid("");
}
```

```
private void btnConsultar_Click(object sender, EventArgs e)
{
    carregarGrid(txtPesquisa.Text);
}
```

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