

# Sistema Ponto de Venda

## Turma do 3º ano

### Classe Cliente

---

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Configuration;
using System.Data;
using System.Windows;
using System.ComponentModel;

namespace SPVWPF3ano
{
    public class Cliente
    {
        private SqlConnection con = null;
        private string strcon = ConfigurationManager.AppSettings["conStr"];

        #region Properties
        public int ClienteID { get; set; }
        public string Nome { get; set; }
        public string Sexo { get; set; }
        public string Email { get; set; }
        #endregion
        #region Methods
        public Cliente GetClienteByID(Cliente pcliente)
        {
            con = new SqlConnection(strcon);

            SqlDataReader rd = null;

            try
            {
                con.Open();
                SqlCommand cmd = new SqlCommand("RetornarClientePorID", con);
                cmd.CommandType = CommandType.StoredProcedure;
                cmd.Parameters.AddWithValue("@clienteID", pcliente.ClienteID);

                rd = cmd.ExecuteReader (CommandBehavior.CloseConnection);
                Cliente cliente = new Cliente();

                while (rd.Read())
                {
                    cliente.ClienteID = Convert.ToInt16( rd["clienteID"]);
                }
            }
            catch { }
        }
    }
}
```

```

        cliente.Nome = rd["nome"].ToString();
        cliente.Sexo = rd["sexo"].ToString();
        cliente.Email = rd["email"].ToString();
    }
    return cliente;
}
catch(SqlException e)
{
    throw new Exception("Erro: " + e.Message);
}
finally
{
    con.Close();
}
}

public void SalvarCliente(Cliente pcliente)
{
    using (con = new SqlConnection (strcon))
    {
        SqlCommand cmd = new SqlCommand("InserirCliente", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@clienteID", pcliente.ClienteID);
        cmd.Parameters.AddWithValue("@nome", pcliente.Nome);
        cmd.Parameters.AddWithValue("@sexo", pcliente.Sexo);
        cmd.Parameters.AddWithValue("@email", pcliente.Email);

        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
        }
        catch (SqlException e)
        {
            throw new Exception("Erro: " + e.Message);
        }
    }
}

public void AlterarCliente(Cliente pcliente)
{
    using (con = new SqlConnection(strcon))
    {
        SqlCommand cmd = new SqlCommand("AlterarCliente", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@clienteID", pcliente.ClienteID);
        cmd.Parameters.AddWithValue("@nome", pcliente.Nome);
        cmd.Parameters.AddWithValue("@sexo", pcliente.Sexo);
        cmd.Parameters.AddWithValue("@email", pcliente.Email);

        try
        {
            con.Open();

```

```

        cmd.ExecuteNonQuery();
    }
    catch (SqlException e)
    {
        throw new Exception("Erro: " + e.Message);
    }
}
}
public void ExcluirCliente(Cliente pcliente)
{
    using (con = new SqlConnection(strcon))
    {
        SqlCommand cmd = new SqlCommand("ExcluirCliente", con);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@clienteID", pcliente.ClienteID);

        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
        }
        catch (SqlException e)
        {
            throw new Exception("Erro: " + e.Message);
        }
    }
}
#endregion
}
}

```

## XAML da janela wCliente

```

x:Class="SPVWPF3ano.View.wCliente"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="wCliente" Height="300" Width="600" WindowStartupLocation="CenterScreen">
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="40"/>
        <RowDefinition Height="40"/>
        <RowDefinition Height="40"/>
        <RowDefinition Height="40"/>
        <RowDefinition />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="100"/>
        <ColumnDefinition />
    </Grid.ColumnDefinitions>
    <StackPanel Orientation="Horizontal" HorizontalAlignment="Left" Grid.ColumnSpan="2">
        <TextBlock Text="Informe o ID do cliente:" FontSize="12" Margin="5" Padding="10"/>
        <TextBox Name="txtClienteID" Width="50" Height="35"

```

```

        Text="{Binding ClienteID, Mode=TwoWay}"/>
</StackPanel>
<TextBlock Text="Nome:"
    FontSize="12"
    Grid.Row="1"
    Grid.Column="0"
    Margin="5"
    Padding="10">
</TextBlock>
<TextBox Name="txtNomeCliente"
    Width="200"
    Height="35"
    Grid.Row="1"
    Grid.Column="1"
    Margin="46,3"
    Text="{Binding Nome, Mode=TwoWay}">
</TextBox>
<TextBlock Text="Sexo:" FontSize="12" Grid.Row="2" Grid.Column="0" Margin="5"
Padding="10"></TextBlock>
<TextBox Name="txtSexoCliente" Width="50" Height="35" Grid.Row="2"
    Grid.Column="1" Margin="46,3,196,3"
    Text="{Binding Sexo, Mode=TwoWay}"></TextBox>
<TextBlock Text="E-mail:" FontSize="12" Grid.Row="3" Grid.Column="0" Margin="5"
Padding="10"></TextBlock>
<TextBox Name="txtEmailCliente" Width="200" Height="35" Grid.Row="3"
    Grid.Column="1" Margin="46,3"
    Text="{Binding Email, Mode=TwoWay}"></TextBox>
<StackPanel Orientation="Horizontal" HorizontalAlignment="Center"
    Grid.Row="4" Grid.Column="0" Grid.ColumnSpan="2">
    <Button Name="btnNovo" Content="Novo" Width="100" Height="35" Click="btnNovo_Click"/>
    <Button Name="btnAlterar" Content="Alterar" Width="100" Height="35"/>
    <Button Name="btnExcluir" Content="Excluir" Width="100" Height="35"/>
    <Button Name="btnLocalizar" Content="Localizar" Width="100" Height="35"
        Click="btnLocalizar_Click"/>
</StackPanel>
</Grid>
</Window>

```

## Código de chamada dos eventos dos botões

```

private void btnLocalizar_Click(object sender, RoutedEventArgs e)
{
    Cliente cliente = new Cliente();
    cliente.ClienteID = Convert.ToInt16(txtClienteID.Text);
    //Chama o método para retornar um cliente pelo ID
    DataContext = cliente.GetClienteByID(cliente);
}

```

```

private void btnNovo_Click(object sender, RoutedEventArgs e)
{
    Cliente cliente = new Cliente();
    cliente.ClienteID = Convert.ToInt16(txtClienteID.Text);
    cliente.Nome = txtNomeCliente.Text;
    cliente.Sexo = txtSexoCliente.Text;
    cliente.Email = txtEmailCliente.Text;
}

```

```
//Chama o método e passa o cliente
try
{
    cliente.SalvarCliente(cliente);
    MessageBox.Show("Cliente salvo com sucesso.");
}
catch (Exception ex)
{
    MessageBox.Show("Erro na operação: " + ex.Message);
}
}
```