

# Projeto exemplo 4º ano 2013

Código da janela produto – XAML

---

```
<UserControl
xmlns:sdk="http://schemas.microsoft.com/winfx/2006/xaml/presentation/sdk"
x:Class="SilverlightApplication4.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:SilverlightApplication4"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="800">

    <UserControl.Resources>
        <local:ConversorDePreco
x:Key="ConvertePreco"></local:ConversorDePreco>
    </UserControl.Resources>

    <Grid x:Name="LayoutRoot" Background="White"
Margin="0,0,-82,0">
        <Grid.RowDefinitions>
            <RowDefinition Height="45"/>
            <RowDefinition Height="45"/>
            <RowDefinition Height="45"/>
            <RowDefinition Height="45"/>
            <RowDefinition />
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="120"/>
            <ColumnDefinition />
        </Grid.ColumnDefinitions>
        <TextBlock Grid.Row="0"
            Grid.Column="0"
            Text="ProdutoID:"
            FontSize="14"
```

```

        HorizontalAlignment="Center"
        VerticalAlignment="Center">
</TextBlock>
<TextBox Grid.Row="0"
        Grid.Column="1"
        Width="50"
        Height="35"
        HorizontalAlignment="Left"
        Margin="5"
        Text="{Binding ProdutoID}"
        Name="txtProdutoID">
</TextBox>
<TextBlock Grid.Row="1"
        Grid.Column="0"
        Text="Descrição:"
        FontSize="14"
        HorizontalAlignment="Center"
        VerticalAlignment="Center">
</TextBlock>
<TextBox Grid.Row="1"
        Grid.Column="1"
        Width="250"
        Height="35"
        HorizontalAlignment="Left"
        Margin="5"
        Text="{Binding Descricao}"
        Name="txtDescricao">
</TextBox>
<TextBlock Grid.Row="2"
        Grid.Column="0"
        Text="Preço:"
        FontSize="14"
        HorizontalAlignment="Center"
        VerticalAlignment="Center">
</TextBlock>
<TextBox Grid.Row="2"
        Grid.Column="1"
        Width="150"
        Height="35"
        HorizontalAlignment="Left"
        Margin="5"
        Text="{Binding Preco,
Converter={StaticResource ConvertePreco}}"
        Name="txtPreco">

```

```
</TextBox>
<StackPanel Orientation="Horizontal" Grid.Row="3"
Grid.ColumnSpan="2" HorizontalAlignment="Left" >
    <Button
        Content="Localizar produto"
        Width="110"
        Height="35"
        Padding="5"
        Click="Button_Click_1">
    </Button>
    <Button
        Content="Salvar produto"
        Width="110"
        Height="35"
        Padding="5"
        Margin="5"
        Click="Button_Click_2">
    </Button>
    <Button
        Content="Atualizar produto"
        Width="110"
        Height="35"
        Padding="5"
        Margin="5"
        Click="Button_Click_3">
    </Button>
    <Button Name="btnExcluir"
        Content="Excluir produto"
        Width="110"
        Height="35"
        Padding="5"
        Margin="5"
        Click="btnExcluir_Click_1">
    </Button>
    <Button
        Content="Produtos..."
        Width="300"
        Height="35"
        Padding="5"
        Margin="5"
        Click="Button_Click_4">
    </Button>
</StackPanel>
```

```

        <StackPanel Orientation="Vertical"
Grid.Column="2" Grid.Row="4">
            <TextBlock Text="Lista de produtos
cadastrados" FontFamily="Arial" FontSize="16"
Foreground="Red" FontWeight="Medium"/>
            <sdk:DataGrid x:Name="listaProduto"
AutoGenerateColumns="True"
SelectionChanged="listaProduto_SelectionChanged_1">
                <sdk:DataGrid.Background>
                    <LinearGradientBrush EndPoint="0.5,1"
StartPoint="0.5,0">
                        <GradientStop Color="#FFD69D9D"
Offset="0.111"/>
                        <GradientStop Color="#FF851212"
Offset="0.481"/>
                    </LinearGradientBrush>
                </sdk:DataGrid.Background>
            </sdk:DataGrid>
        </StackPanel>
    </Grid>
</UserControl>

```

Código do Objeto Produto – C#

---

```

namespace SilverlightApplication4.Web
{
    [DataContract()]
    public class Produto
    {
        private int produtoID;
        [DataMember()]
        public int ProdutoID
        {
            get { return produtoID; }
            set { produtoID = value; }
        }

        private string descricao;
        [DataMember()]
        public string Descricao
        {

```

```

        get { return descricao; }
        set { descricao = value; }
    }

    private Double preco;
    [DataMember()]
    public Double Preco
    {
        get { return preco; }
        set { preco = value; }
    }
}
}
}

```

Código do banco de dados - SQL

---

```

USE [Loja]
GO

```

```

CREATE procedure [dbo].[Atualizar]
@produtoID int,
@descricao varchar(50),
@preco decimal(12,2)
AS
    UPDATE Produto SET descricao = @descricao,
                        preco      = @preco
    WHERE produtoID = @produtoID

```

```

GO

```

```

CREATE PROCEDURE [dbo].[ExcluirProduto]
@produtoID int
AS
    DELETE FROM Produto where produtoID =
@produtoID
GO

```

```

create procedure [dbo].[GetProductById]

```

```
@produtoID int
AS
    Select * from Produto where produtoID =
@produtoID
GO
```

```
create procedure [dbo].[GetProdutos]
AS
    SELECT * FROM Produto
```

```
GO
```

```
create procedure [dbo].[Inserir]
@produtoID int,
@descricao varchar(50),
@preco decimal(12,2)
AS
    INSERT INTO Produto VALUES
(@produtoID,@descricao,@preco)
```

## Código do Serviço - WCF Silverlight

---

```
using System;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Activation;
using System.Web.Configuration;
using System.Data.SqlClient;
using System.Data;
using System.Collections.Generic;

namespace SilverlightApplication4.Web
{
    [ServiceContract(Namespace = "")]
    [AspNetCompatibilityRequirements(RequirementsMode =
AspNetCompatibilityRequirementsMode.Allowed)]
    public class ServiceAcessoDados
    {
```

```

        private static String connectionString =
WebConfigurationManager.ConnectionStrings["LojaDB"].Conne
ctio
nString;
        Produto produto = new Produto();
        SqlConnection con = new
SqlConnection(connectionString);

        [OperationContract]
        public List<Produto> GetProduct()
        {
            SqlCommand cmd = new
SqlCommand("GetProdutos", con);
            cmd.CommandType =
System.Data.CommandType.StoredProcedure;

            List<Produto> listaPro = new List<Produto>();

            try
            {
                con.Open();
                SqlDataReader reader =
cmd.ExecuteReader(CommandBehavior.Default);

                while (reader.Read())
                {
                    Produto produto = new Produto();
                    produto.ProdutoID =
Convert.ToInt16(reader["produtoID"]);
                    produto.Descricao =
reader["descricao"].ToString();
                    produto.Preco =
Convert.ToDouble(reader["preco"]);
                    listaPro.Add(produto);
                }
                return listaPro;
            }
            catch (SqlException ex)
            {
                throw new Exception("Erro=" +
ex.Message);
            }
            finally
            {
                con.Close();
            }
        }
    }
}

```

```

    }
}

[OperationContract]
public Produto GetProductBYID (int produtoID)
{
    SqlCommand cmd = new
SqlCommand("GetProductBYID", con);
    cmd.CommandType =
System.Data.CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("@produtoID",
produtoID);

    con.Open();
    try
    {
        SqlDataReader reader =
cmd.ExecuteReader(CommandBehavior.SingleRow);
        if (reader.Read())
        {
            produto.ProdutoID = Convert.ToInt16(
reader["produtoID"] );
            produto.Descricao =
reader["descricao"].ToString();
            produto.Preco = Convert.ToDouble(
reader["preco"]);

            return produto;
        }
        else
        {
            return null;
        }
    }
    catch(SqlException ex)
    {
        throw new Exception("Erro=" +
ex.Message);
    }
    finally
    {
        con.Close();
    }
}

```



```

    }

    [OperationContract()]
    public void InserirProduto(int produtoID, string
descricao, Double preco)
    {
        SqlCommand cmd = new
SqlCommand("Inserir", con);
        cmd.CommandType =
CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@produtoID",
produtoID);
        cmd.Parameters.AddWithValue("@descricao",
descricao);
        cmd.Parameters.AddWithValue("@preco", preco);

        con.Open();
        try
        {
            cmd.ExecuteNonQuery();
        }
        catch (SqlException ex)
        {
            throw new Exception("Erro=" +
ex.Message);
        }
        finally
        {
            con.Close();
        }
    }

    [OperationContract()]
    public void AlterarProduto(int produtoID, string
descricao, Double preco)
    {
        SqlCommand cmd = new SqlCommand("Atualizar",
con);
        cmd.CommandType =
CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@produtoID",
produtoID);
        cmd.Parameters.AddWithValue("@descricao",
descricao);
        cmd.Parameters.AddWithValue("@preco", preco);
    }
}

```

```

        con.Open();
        try
        {
            cmd.ExecuteNonQuery();
        }
        catch (SqlException ex)
        {
            throw new Exception("Erro=" +
ex.Message);
        }
        finally
        {
            con.Close();
        }
    }
    //Método para excluir produtos cadastrados
    [OperationContract]
    public void ExcluirProduto(int produtoID)
    {
        SqlCommand cmd = new
SqlCommand("ExcluirProduto", con);
        cmd.CommandType =
CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@produtoID",
produtoID);

        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
        }
        catch (SqlException ex)
        {
            throw new Exception("Erro=" +
ex.Message);
        }
        finally
        {
            con.Close();
        }
    }
}
}
}

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;

namespace SilverlightApplication4
{
    public partial class MainPage : UserControl
    {
        LojaDBReferencia.ServiceAcessoDadosClient cliente
        = new LojaDBReferencia.ServiceAcessoDadosClient();

        public MainPage()
        {
            InitializeComponent();
        }

        private void Button_Click_1(object sender,
RoutedEventArgs e)
        {
            cliente.GetProductBYIDCompleted +=
cliente_GetProductBYIDCompleted;
            cliente.GetProductBYIDAsync(Convert.ToInt16(
txtProdutoID.Text));
        }

        void cliente_GetProductBYIDCompleted(object
sender, LojaDBReferencia.GetProductBYIDCompletedEventArgs
e)
        {
            DataContext = e.Result;
        }
    }
}
```

```

        //throw new NotImplementedException();
    }

    private void Button_Click_2(object sender,
RoutedEventArgs e)
    {
        cliente.InserirProdutoCompleted +=
cliente_InserirProdutoCompleted;

cliente.InserirProdutoAsync(Convert.ToInt16(txtProdutoID.
Text),
        txtDescricao.Text,
Convert.ToDouble(txtPreco.Text));
    }

    void cliente_InserirProdutoCompleted(object
sender, System.ComponentModel.AsyncCompletedEventArgs e)
    {
        try
        {
            MessageBox.Show("Produto inserido com
sucesso!");
        }
        catch (Exception ex)
        {
            throw new Exception("Erro=" + ex.Message );
        }
    }

    private void Button_Click_3(object sender,
RoutedEventArgs e)
    {
        cliente.AlterarProdutoCompleted +=
cliente_AlterarProdutoCompleted;

cliente.AlterarProdutoAsync(Convert.ToInt16(txtProdutoID.
Text),
        txtDescricao.Text,
Convert.ToDouble(txtPreco.Text));
    }

    void cliente_AlterarProdutoCompleted(object
sender, System.ComponentModel.AsyncCompletedEventArgs e)

```

```

        {
            try
            {
                MessageBox.Show("Produto alterado com
sucesso!");
            }
            catch (Exception ex)
            {
                throw new Exception("Erro=" +
ex.Message);
            }
        }

        private void Button_Click_4(object sender,
RoutedEventArgs e)
        {
            cliente.GetProductCompleted +=
cliente_GetProductCompleted;
            cliente.GetProductAsync();
        }

        void cliente_GetProductCompleted(object sender,
LojaDBReferencia.GetProductCompletedEventArgs e)
        {
            listaProduto.ItemsSource = e.Result;
        }

        private void
listaProduto_SelectionChanged_1(object sender,
SelectionChangedEventArgs e)
        {
            DataContext = listaProduto.SelectedItem;
        }

        private void btnExcluir_Click_1(object sender,
RoutedEventArgs e)
        {
            cliente.ExcluirProdutoCompleted +=
cliente_ExcluirProdutoCompleted;

            cliente.ExcluirProdutoAsync(Convert.ToInt16(txtProdutoID.
Text));
        }

```

```
void cliente_ExcluirProdutoCompleted(object
sender, System.ComponentModel.AsyncCompletedEventArgs e)
{
    try
    {
        MessageBox.Show("Registro deletado com
sucesso!");
        txtProdutoID.Focus();
    }
    catch (Exception er)
    {
        MessageBox.Show("Erro=" + er.Message);
    }
}
}
```