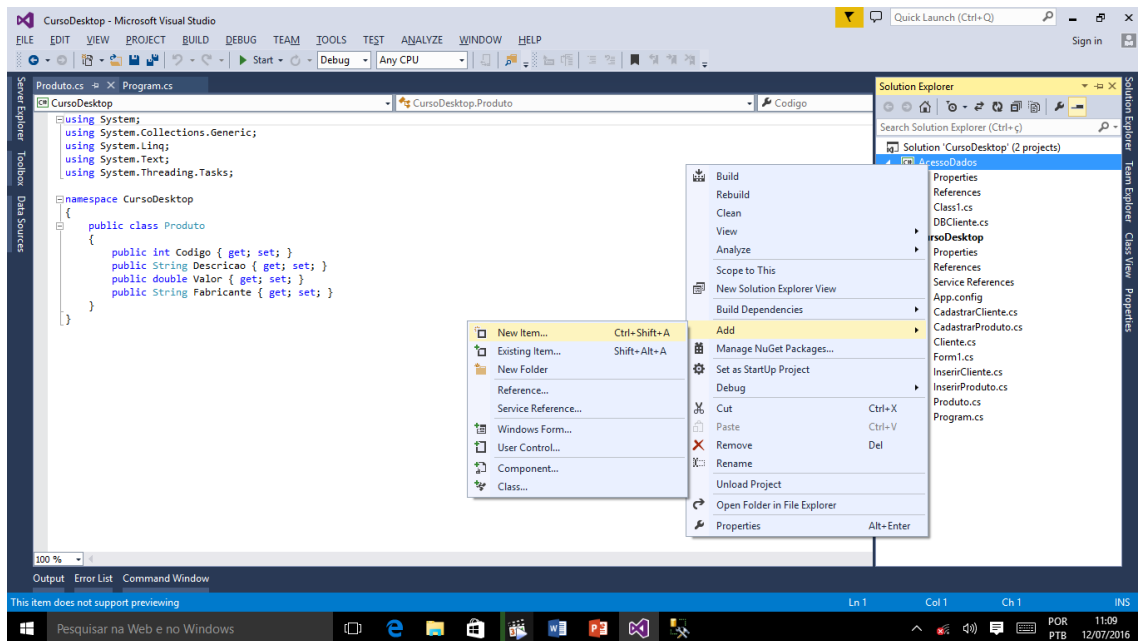
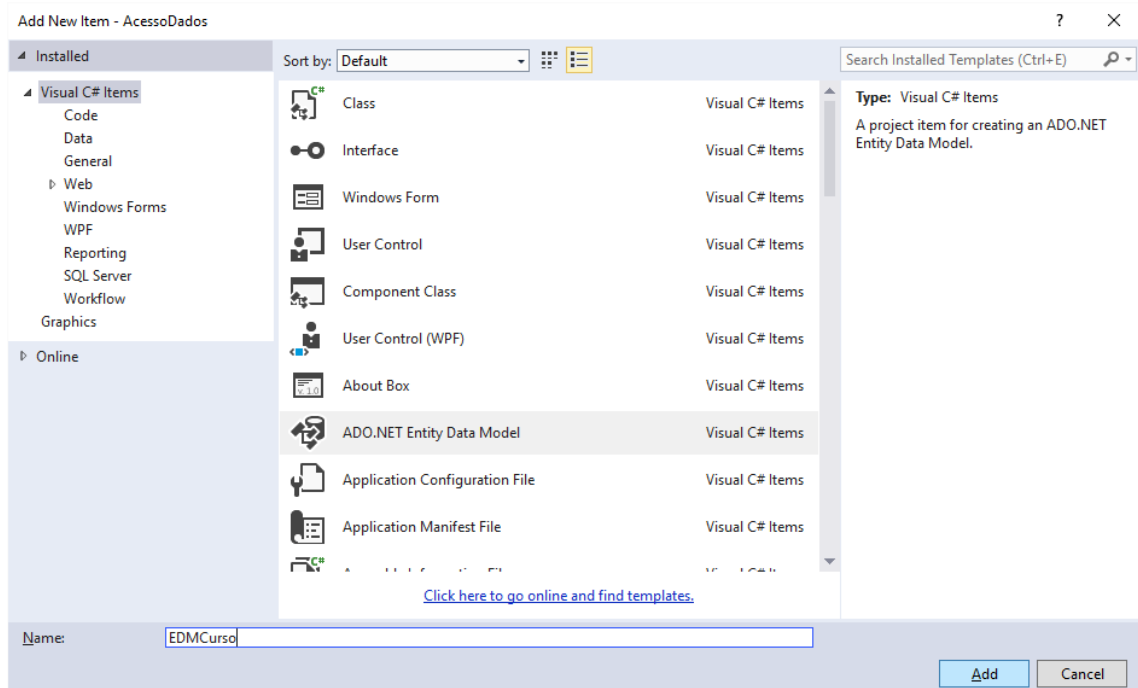


Prática 9

- ❖ Crie um novo item ao projeto




- ❖ Selecione ADO.NET Entity Data Model e adicione um nome





- ❖ Escolhe a opção gerar a partir de uma base de dados


Entity Data Model Wizard


 **Choose Model Contents**

What should the model contain?

 **EF Designer from database**

 Empty EF Designer model

 Empty Code First model


 Code First from database

Creates a model in the EF Designer based on an existing database. You can choose the database connection, settings for the model, and database objects to include in the model. The classes your application will interact with are generated from the model.

< Previous **Next >** Finish Cancel

❖ Informe os dados de conexão com o banco de dados

Entity Data Model Wizard

 **Choose Your Data Connection**

Which data connection should your application use to connect to the database?

aristoteles.cursornet.dbo New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/EDMCurso.csdl|res://*/EDMCurso.ssdl|
res://*/EDMCurso.msl;provider=System.Data.SqlClient;provider connection string="data
source=localhost;initial catalog=cursornet;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☒ Save connection settings in App.Config as:

cursornetEntities

< Previous **Next >** Finish Cancel

❖ Caso for criar uma nova conexão clique em **"New Connection"**



Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

aristoteles.cursonet.dbo New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/EDMCurso.csdl|res://*/EDMCurso.ssdl|
res://*/EDMCurso.msl;provider=System.Data.SqlClient;provider connection string="data
source=localhost;initial catalog=cursonet;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☒ Save connection settings in App.Config as:

cursonetEntities

< Previous **Next >** Finish Cancel

- ❖ Para for criar uma nova conexão informe: Data Source, Server Name, Log on, Database e Test Connection

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
▼ Refresh

Log on to the server

☒ Use Windows Authentication

☐ Use SQL Server Authentication

User name:

Password:

☐ Save my password

Connect to a database

☒ Select or enter a database name:
▼

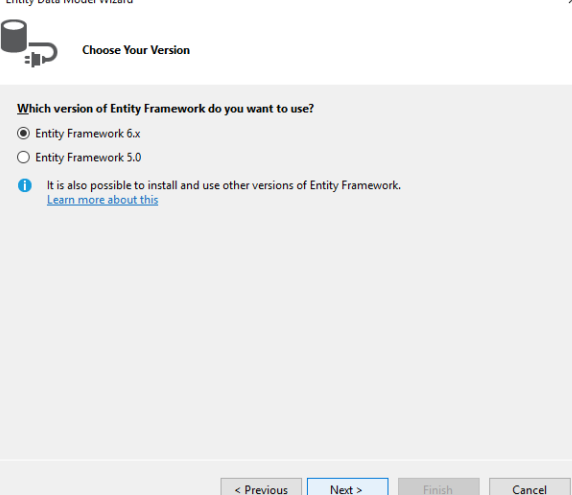
☐ Attach a database file:
 Browse...

Logical name:

Advanced...

Test Connection OK Cancel

- ❖ Depois de informada a conexão informe a versão do Entity




Entity Data Model Wizard

Choose Your Version

Which version of Entity Framework do you want to use?

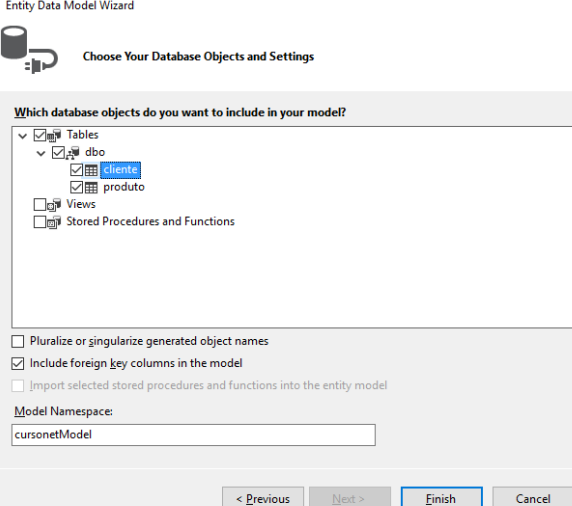
☒ Entity Framework 6.x

☐ Entity Framework 5.0

 It is also possible to install and use other versions of Entity Framework.
[Learn more about this](#)

< Previous Next > Finish Cancel

❖ Informe a tabela **produto** para ser mapeada



Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

☒ Tables

☒ dbo

☒ cliente

☒ produto

☐ Views

☐ Stored Procedures and Functions

☐ Pluralize or singularize generated object names

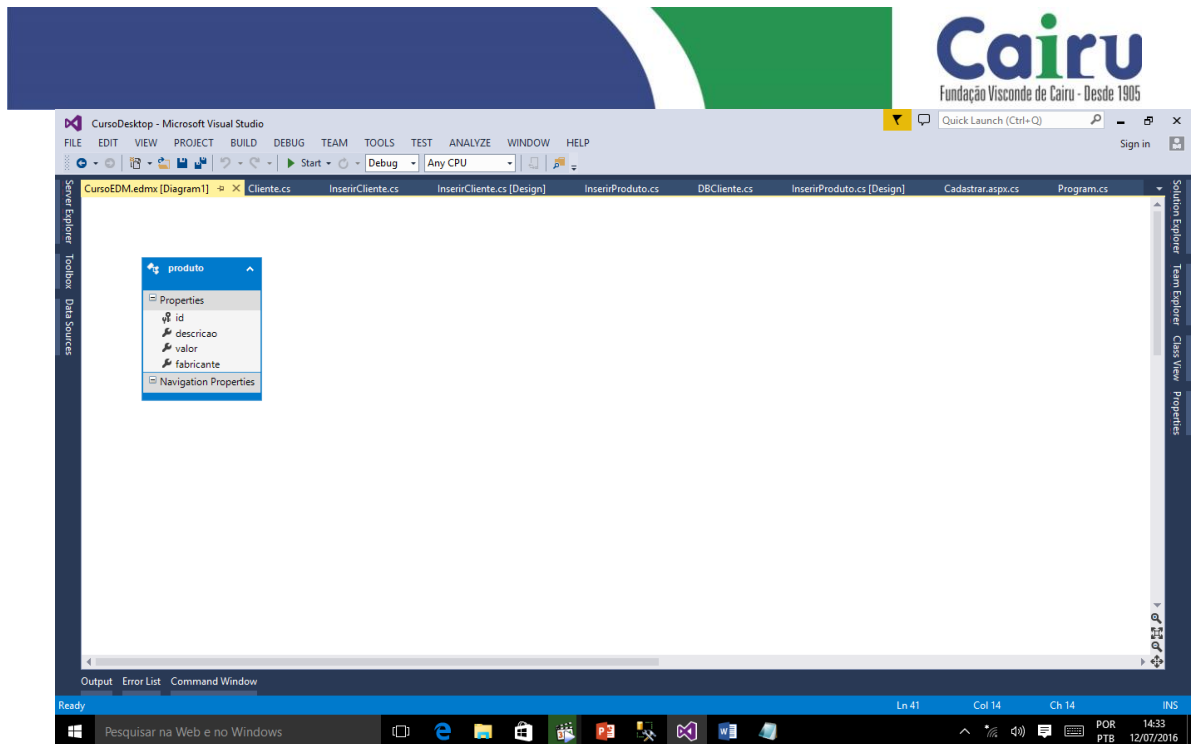
☒ Include foreign key columns in the model

☐ Import selected stored procedures and functions into the entity model

Model Namespace:
cursornetModel

< Previous Next > Finish Cancel

❖ Se tudo estiver correto deverá ser exibido o EDMCruso.edmx



- ❖ No clique do botão Salvar do form InseiriProduto cadastre os dados da tela no banco utilizando o Entity criado

```
private void btSalvar_Click(object sender, EventArgs e)
{
    try
    {
        using (var context = new cursonetEntities())
        {
            produto produto = new produto();
            produto.descricao = txtDescricao.Text.Trim();
            produto.valor = nuValor.Value;
            produto.fabricante = cbFabricante.Text.Trim();
            context.produto.Add(produto);
            context.SaveChanges();
        }
        MessageBox.Show("Produto Cadastrado com Sucesso!");
    }
    catch (Exception ex)
    {
        MessageBox.Show("Erro ao cadastrar produto!");
    }
}
```

- ❖ No clique do botão Consultar do form CadastrarProduto exiba os dados da tabela produto no gridview de pesquisa de produtos

```
private void btConsultar_Click(object sender, EventArgs e)
{
    using (var context = new cursonetEntities())
    {
        var dados = (from l in context.produto
        where ((l.descricao.Contains(txtDescricao.Text.Trim()) ||
        txtDescricao.Text.Trim().Equals(""))
        )
        select new
        {
            l.id,
            l.descricao,

```



```
l.valor,  
l.fabricante  
}).ToList();
```

```
gwProduto.DataSource = dados;
```

```
}  
}
```

- ❖ No clique do botão Remover do form CadastrarProduto remova os dados da tabela produto selecionado no gridview de pesquisa de produtos

```
private void btRemover_Click(object sender, EventArgs e)  
{  
    DataGridViewRow row = gwProduto.CurrentRow;  
    if (!string.IsNullOrEmpty(row.Cells[1].Value.ToString()))  
    {  
        int id = Convert.ToInt32(row.Cells[1].Value.ToString());  
        using (var context = new cursonetEntities())  
        {  
            produto produto = context.produto.Where(c => c.id == id).First();  
            context.produto.Remove(produto);  
            context.SaveChanges();  
        }  
    }  
    btConsultar_Click(sender, e);  
    MessageBox.Show("Cliente removido com sucesso!");  
}
```

- ❖ Sobrecarregue o Construtor do form InserirCliente para receber os dados dos clientes para alteração

```
public InserirProduto(int id)  
{  
    InitializeComponent();  
    using (var context = new cursonetEntities())  
    {  
        produto produto = context.produto.Where(c => c.id == id).First();  
        txtDescricao.Text = produto.descricao;  
        txtCodigo.Text = produto.id.ToString();  
        nuValor.Value = produto.valor;  
        cbFabricante.SelectedText = produto.fabricante;  
    }  
}
```

- ❖ Altere o evento do botão Alterar do form CadastrarCliente para chamar o form InserirCliente passando os dados do cliente selecionado no GridView

```
private void btAlterar_Click(object sender, EventArgs e)  
{  
    DataGridViewRow row = gwProduto.CurrentRow;  
    try  
    {  
        if (!string.IsNullOrEmpty(row.Cells[1].Value.ToString()))  
        {  
            int id = Convert.ToInt32(row.Cells[0].Value.ToString());  
            InserirProduto formInserirProduto = new InserirProduto(id);  
            formInserirProduto.MdiParent = this.MdiParent;  
            formInserirProduto.Show();  
        }  
    }  
    catch (Exception)
```



```
        {  
            MessageBox.Show("Erro ao alterar Cliente");  
            throw;  
        }  
    }  
}
```

- ❖ Altere o evento do botão Salvar do form `InserirProduto` para fazer a alteração do produto

```
private void btSalvar_Click(object sender, EventArgs e)  
{  
    try  
    {  
        using (var context = new cursonetEntities())  
        {  
            produto produto;  
            if (txtCodigo.Text == "")  
            {  
                produto = new produto();  
            }  
            else  
            {  
                int id = Convert.ToInt32(txtCodigo.Text);  
                produto = context.produto.Where(c => c.id == id).First();  
            }  
            produto.descricao = txtDescricao.Text.Trim();  
            produto.valor = nuValor.Value;  
            produto.fabricante = cbFabricante.Text.Trim();  
            if (txtCodigo.Text == "")  
            {  
                context.produto.Add(produto);  
            }  
            context.SaveChanges();  
        }  
        MessageBox.Show("Produto Cadastrado com Sucesso!");  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show("Erro ao cadastrar produto!");  
    }  
}
```