



# INTRODUÇÃO AO FRAMEWORK .NET

**Cairu**  
Fundação Visconde de Cairu - Desde 1905

# AULA 8 — ACESSO A BANCO USANDO ENTITY

- ❖ Conexão

- ❖ Insert, Select, Update e Delete

# APRESENTAÇÃO DIA 1 – 11/07 – PROF. ARISTÓTELES

Início	Fim	Atividade
18:30	19:20	Apresentação
19:20	20:00	Introdução ao Framework .Net e Apresentação do Visual Studio 2013 (github)
20:00	20:10	Intervalo
20:10	20:50	Introdução ao C#: tipos de dados, if, else, case, for, while, e/s console
20:50	21:30	Prática 1 – Console Application

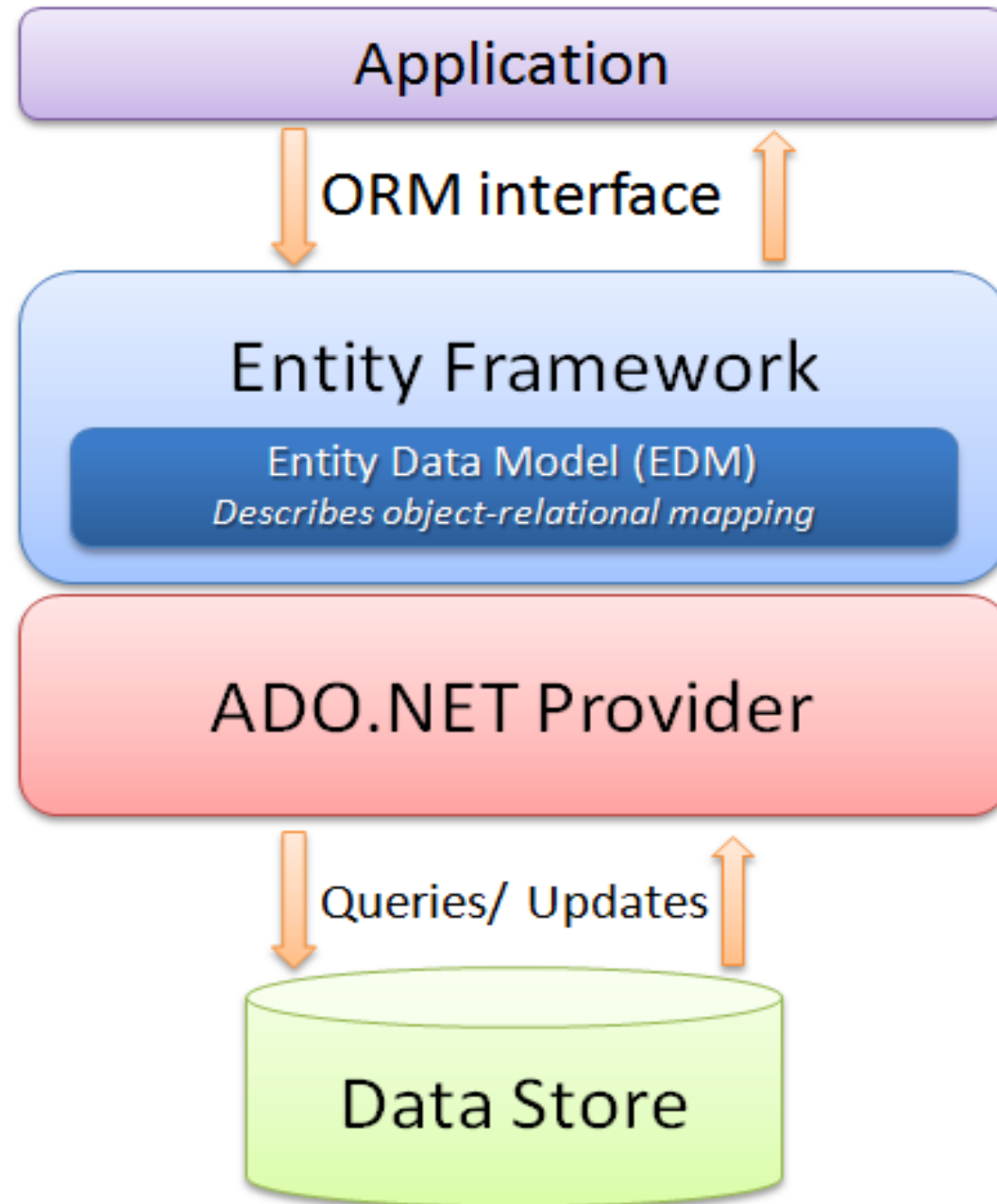
# ENTITY

O Microsoft Entity Framework é uma ferramenta de mapeamento objeto-relacional (ORM – Object Relational Management), que permite aos desenvolvedores trabalhar com classes (entidades) que correspondem a tabelas em um banco de dados, tornando transparente o acesso a estes dados e principalmente, eliminando a necessidade de escrever código de banco de dados (SELECT, INSERT, UPDATE, DELETE) na aplicação.

# ENTITY

Mapeamento do objeto-relacional (ou ORM, do inglês: Object-relational mapping) é uma técnica de desenvolvimento utilizada para reduzir a impedância da programação orientada aos objetos, utilizando bancos de dados relacionais. As tabelas do banco de dados são representadas através de classes e os registros de cada tabela são representados como instâncias das classes correspondentes.

# ENTITY

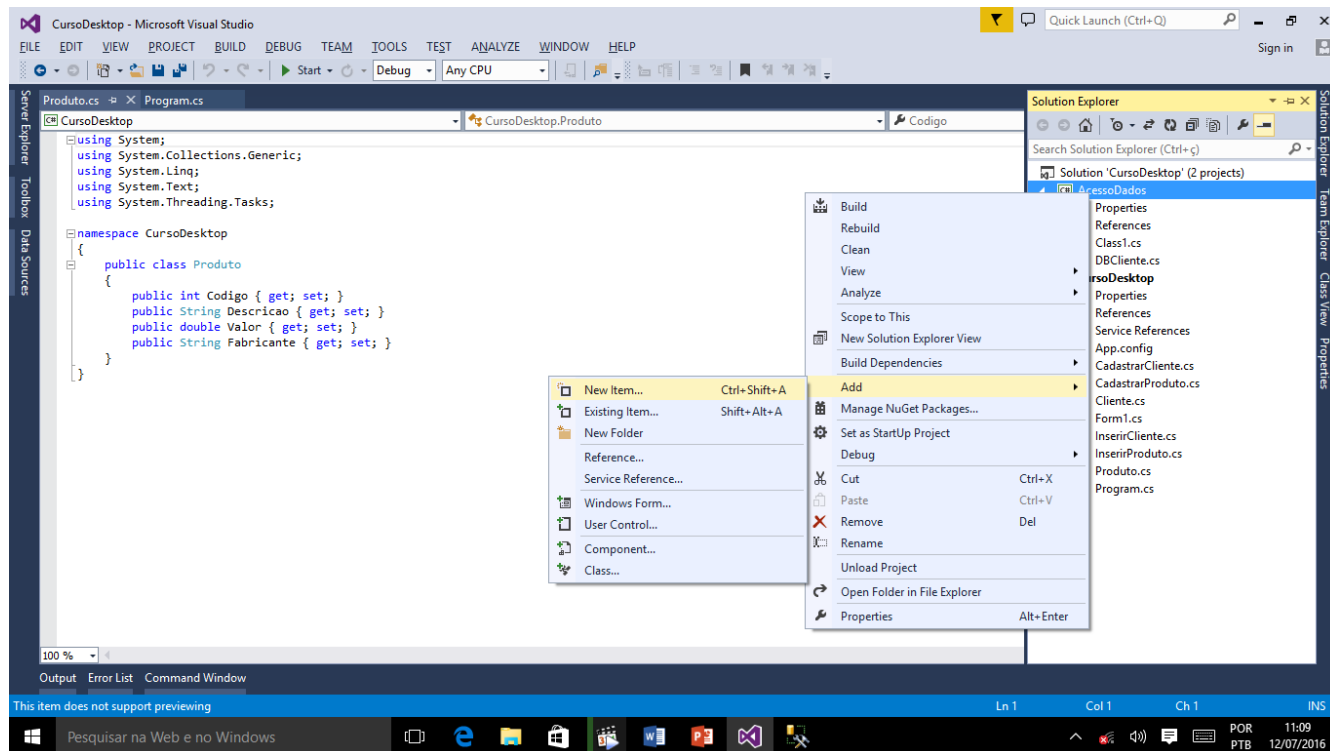


# ENTITY

- ❖ Entity Data Model (EDM), que é o coração do Entity Framework. É nele que as informações do mapeamento entre as classes e o banco de dados reside.
- ❖ O EDM é composto por três arquivos XML:
  - ❖ O primeiro descreve as classes do modelo de objetos;
  - ❖ O segundo descreve as tabelas no banco de dados;
  - ❖ O terceiro armazena o mapeamento entre os objetos e as tabelas;

# CRIANDO ENTITY DATA MODEL - EDM

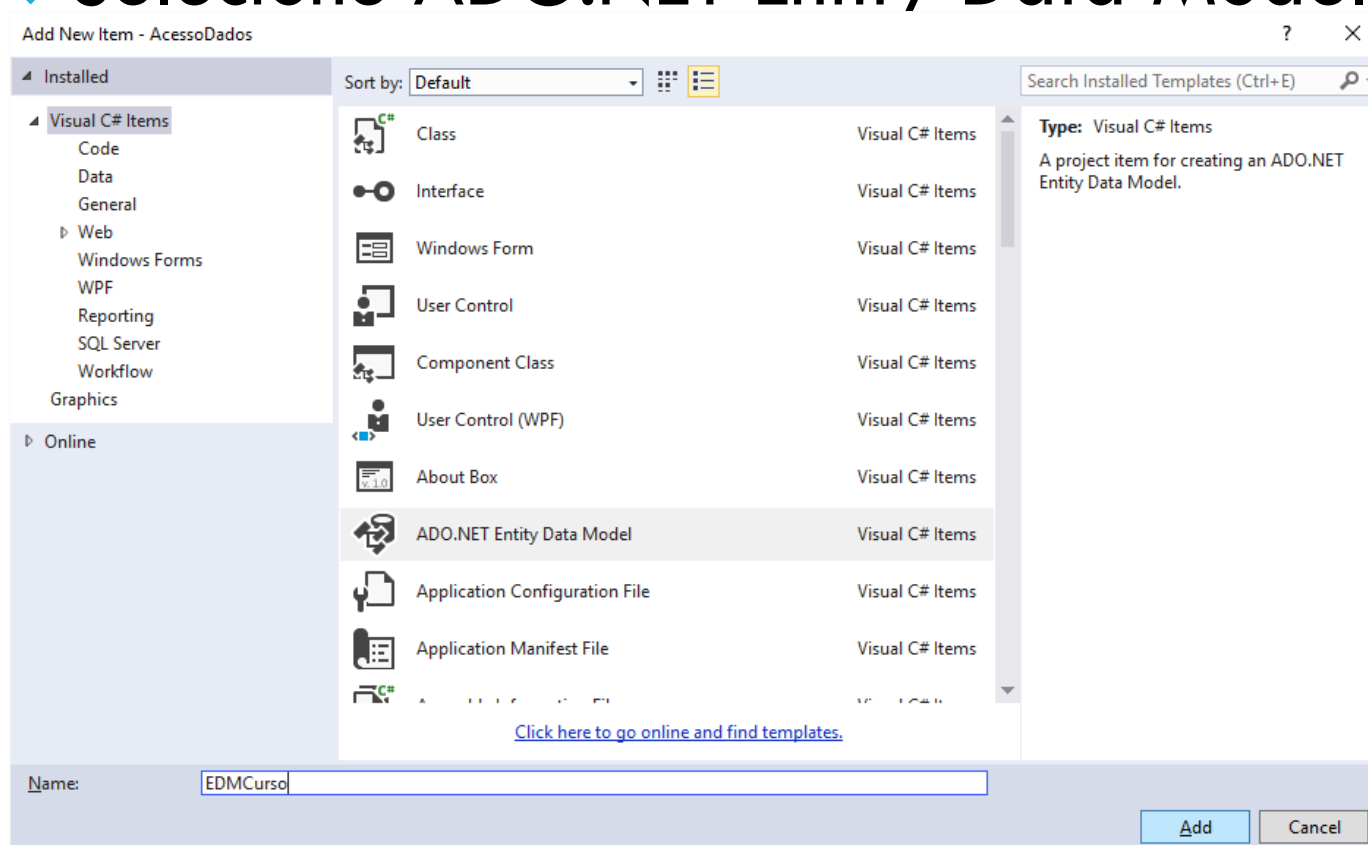
❖ Crie um novo item no projeto





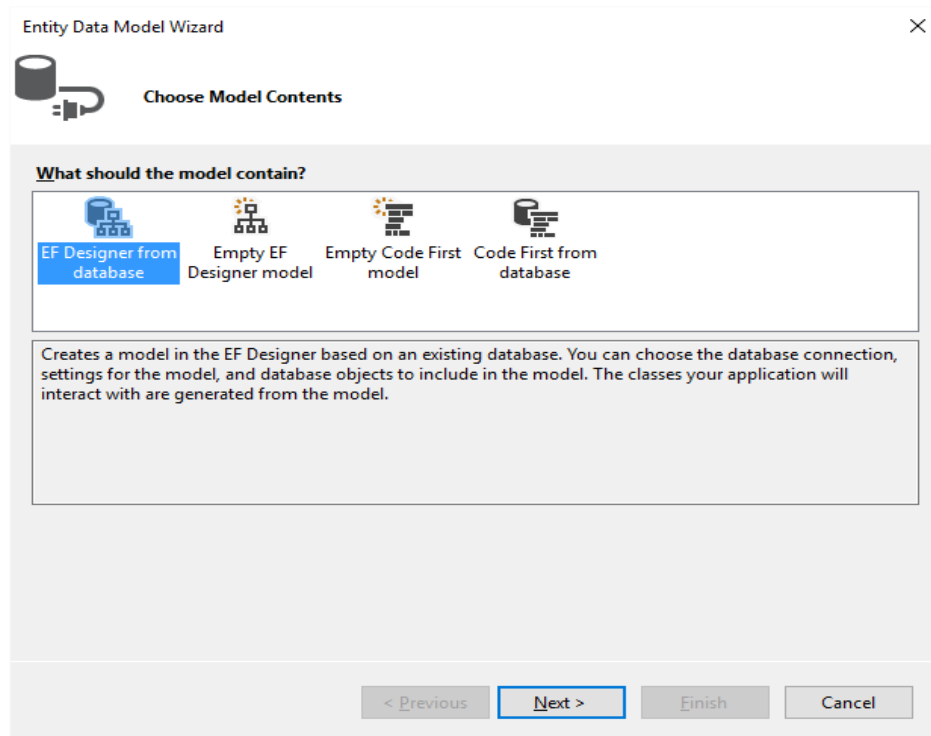
# CRIANDO ENTITY DATA MODEL - EDM

## Selezione ADO.NET Entity Data Model e adicione um nome



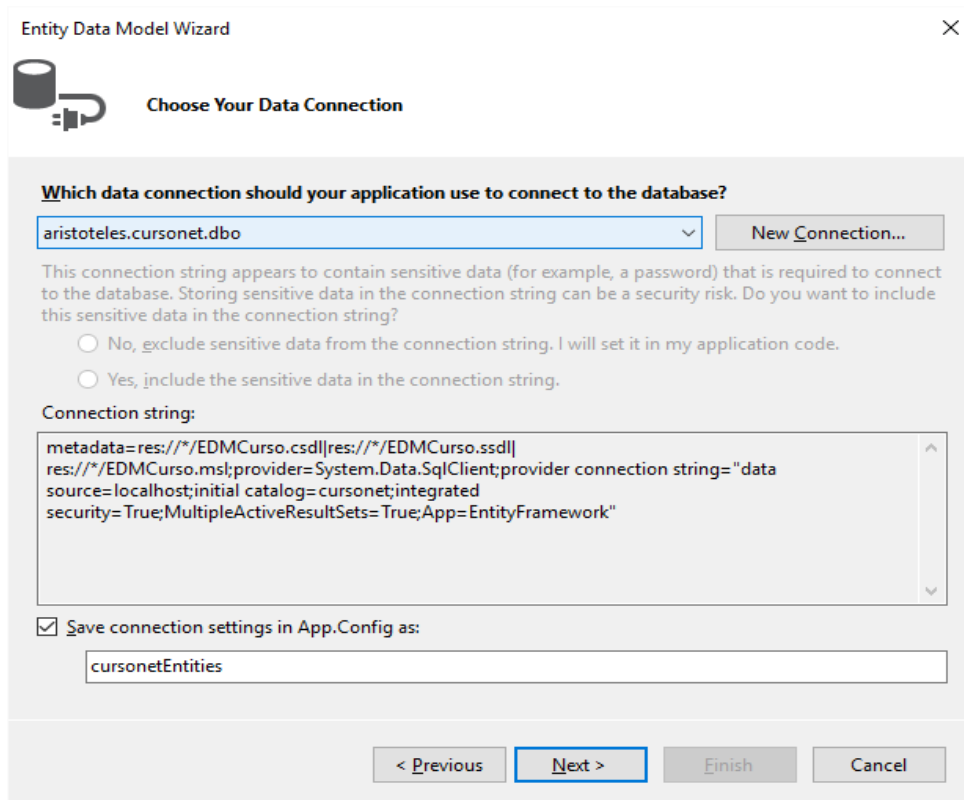
# CRIANDO ENTITY DATA MODEL - EDM

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



# CRIANDO ENTITY DATA MODEL - EDM

❖ Informe os dados de conexão com o banco de dados



The image shows a screenshot of the 'Entity Data Model Wizard' dialog box, specifically the 'Choose Your Data Connection' step. The window has a title bar with the text 'Entity Data Model Wizard' and a close button. Below the title bar is a database icon and the text 'Choose Your Data Connection'. The main area contains a question: 'Which data connection should your application use to connect to the database?'. Below this is a dropdown menu showing 'aristoteles.cursonet.dbo' and a 'New Connection...' button. A warning message states: '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?'. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' (selected) and 'Yes, include the sensitive data in the connection string.'. Below this is a text box labeled 'Connection string:' containing the following text: '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"'. At the bottom, there is a checkbox 'Save connection settings in App.Config as:' which is checked, and a text box containing 'cursonetEntities'. The bottom of the dialog has four buttons: '< Previous', 'Next >' (highlighted), 'Finish', and 'Cancel'.

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"

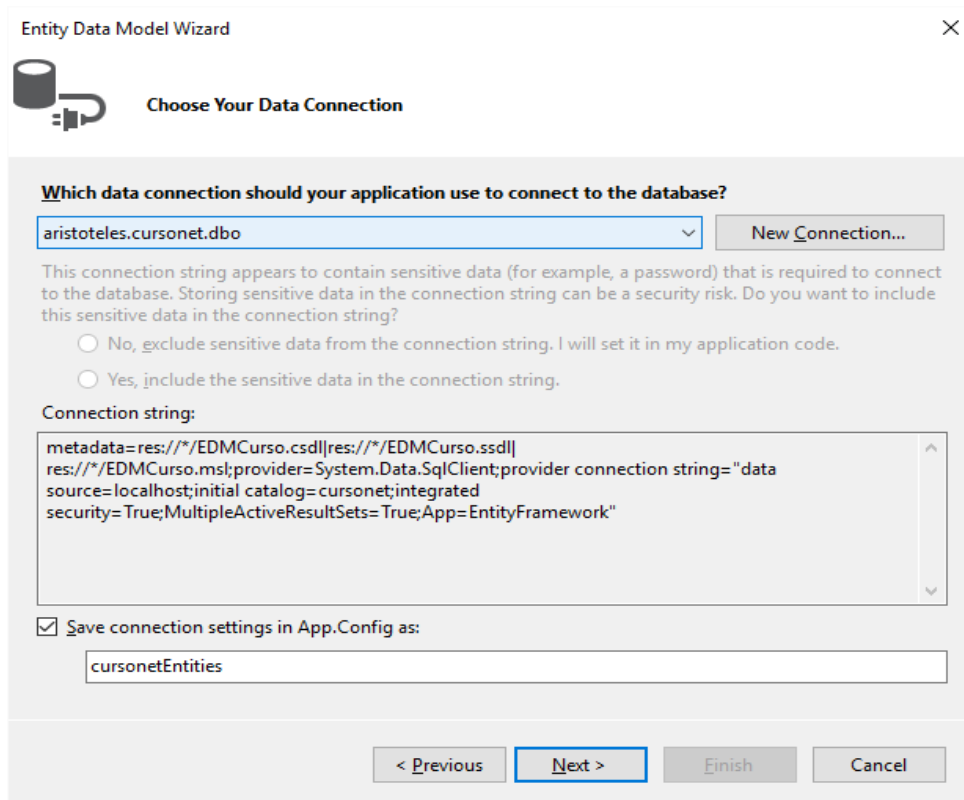
☒ Save connection settings in App.Config as:

cursonetEntities

< Previous Next > Finish Cancel

# CRIANDO ENTITY DATA MODEL - EDM

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

The image shows a screenshot of the 'Entity Data Model Wizard' dialog box. The title bar says 'Entity Data Model Wizard' with a close button. Below the title bar is a database icon and the text 'Choose Your Data Connection'. The main area has a question: 'Which data connection should your application use to connect to the database?'. Below this is a dropdown menu showing 'aristoteles.cursonet.dbo' and a 'New Connection...' button. A warning message states: '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?'. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' and 'Yes, include the sensitive data in the connection string.'. Below this is a 'Connection string:' label and a text area containing the 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"'. At the bottom, there is a checkbox 'Save connection settings in App.Config as:' which is checked, and a text box containing 'cursonetEntities'. The bottom of the dialog has four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

# CRIANDO EDM

❖ Para for criar uma nova conexão informe:

❖ Data Source

❖ Server Name

❖ Log on

❖ Database

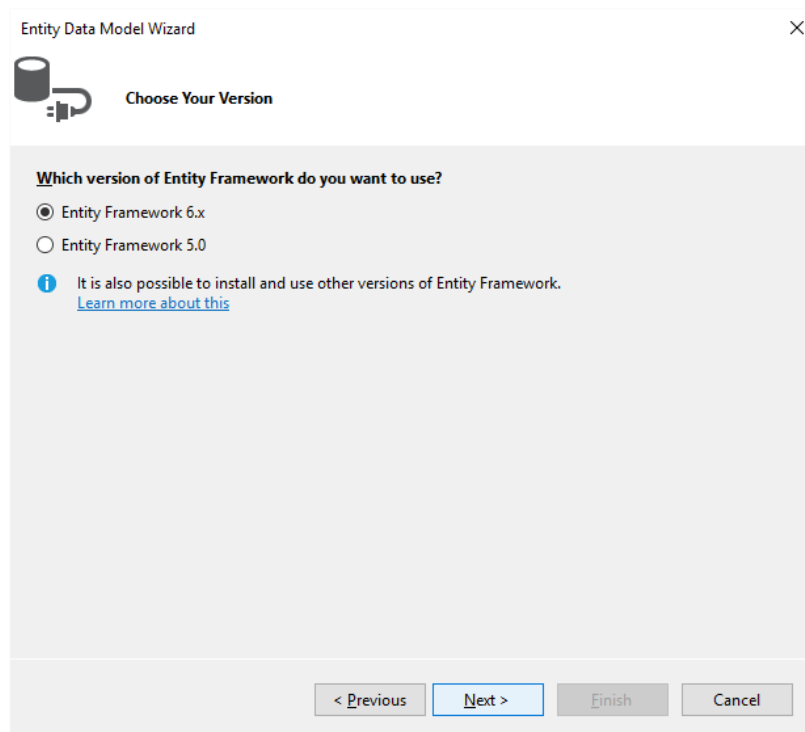
❖ Test Connection

The screenshot shows the 'Connection Properties' dialog box with the following fields and options:

- Data source:** A text box containing 'Microsoft SQL Server (SqlClient)' and a 'Change...' button.
- Server name:** A dropdown menu (currently empty) and a 'Refresh' button.
- Log on to the server:**
  - ☒ Use Windows Authentication
  - ☐ Use SQL Server Authentication
  - User name: [text box]
  - Password: [text box]
  - ☐ Save my password
- Connect to a database:**
  - ☒ Select or enter a database name: [dropdown menu]
  - ☐ Attach a database file: [text box] [Browse...]
  - Logical name: [text box]
- Buttons at the bottom:** 'Test Connection', 'OK', 'Cancel', and 'Advanced...'.

# CRIANDO ENTITY DATA MODEL - EDM


❖ Depois de informada a conexão informe a versão do Entity



# CRIANDO ENTITY DATA MODEL - EDM

 Informe os objetos do banco que serão mapeados

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

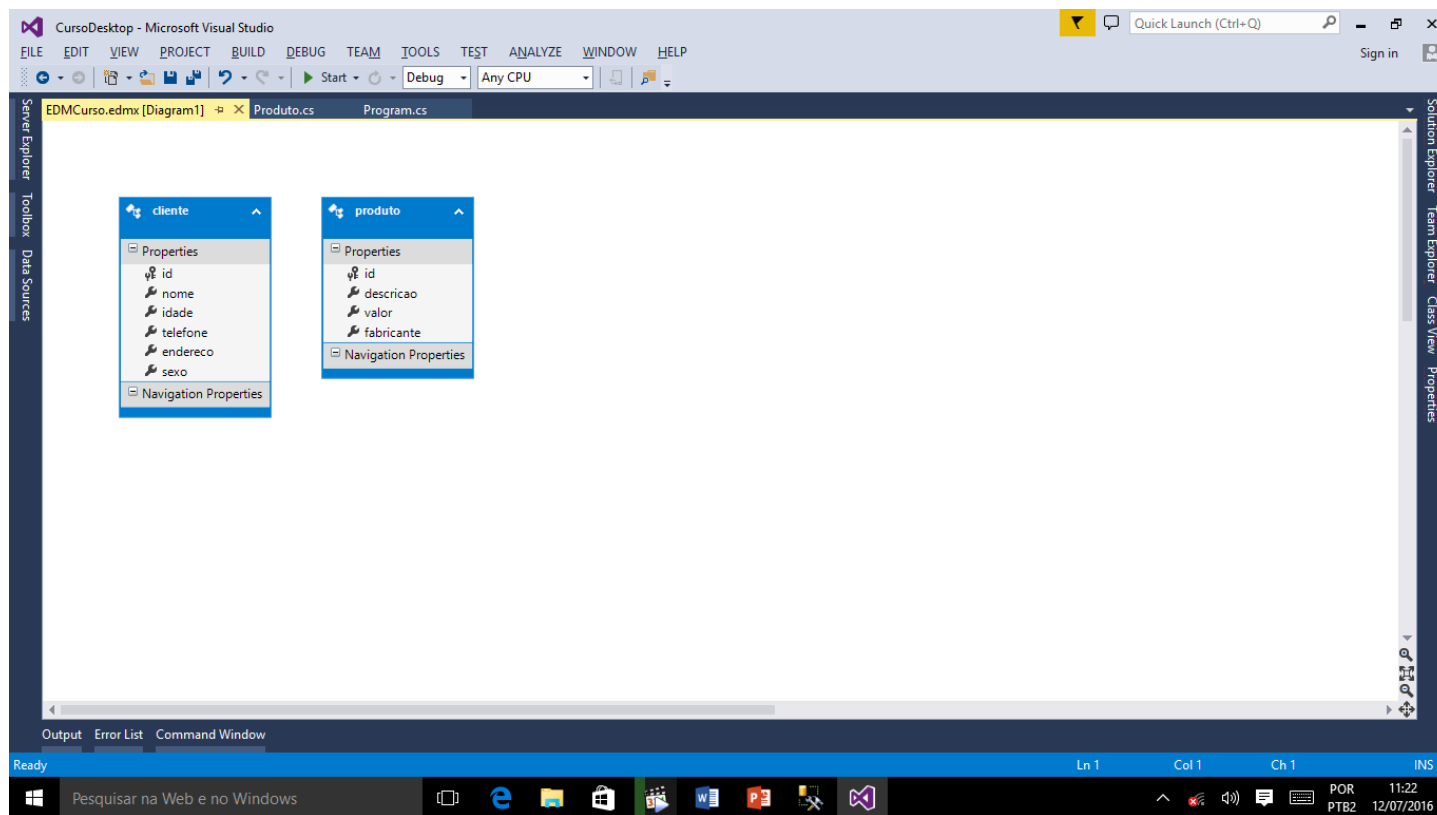
**Model Namespace:**

cursonetModel

< Previous   Next >   **Finish**   Cancel

# CRIANDO ENTITY DATA MODEL - EDM

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





# INSERT COM ENTITY

```
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();
}
```

# SELECT COM ENTITY

```
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;
}
```

# DELETE COM ENTITY

```
int id = 1;
using (var context = new cursonetEntities())
{
    produto produto = context.produto.Where(c => c.id == id).First();
    context.produto.Remove(produto);
    context.SaveChanges();
}
```

# UPDATE COM ENTITY

```
using (var context = new cursonetEntities())
{
    int id = 1;
    produto produto = context.produto.Where(c => c.id == id).First();
    produto.descricao = txtDescricao.Text.Trim();
    produto.valor = nuValor.Value;
    produto.fabricante = cbFabricante.Text.Trim();
    context.SaveChanges();
}
```



# PRÁTICA 9