Análise e Desenvolvimento de Sistemas

Aula 6

Professor Ederson Cichaczewski

Conversa Inicial

Olá, alunos! Chegamos a nossa sexta aula da disciplina **Programação Visual**. Sim, é o nosso último encontro! Mas, com certeza você aprendeu bastante coisa, certo?!

Hoje, vamos aprender os conceitos básicos de bancos de dados relacionais. Conheceremos a interface ADO.NET, que permite uma página WEB acessar bancos de dados. Iremos entender como desenvolver uma aplicação com ADO.NET conectando em uma base SQL (*Server no ambiente Visual Studio 2015*). Veremos uma aplicação em linguagem ASP.NET com ADO.NET acessando uma base SQL (*Server e code behind em Visual Basic*.NET). Também faremos um exemplo com a página em ASP.NET acessando um arquivo XML como um banco de dados *e code behind* em C#. Então, para iniciar os trabalhos, ele está de volta... Sim! O professor Ederson!

Veja o que ele diz sobre o conteúdo desta aula no material online!

Contextualizando

Bancos de dados são utilizados praticamente em tudo o que fazemos atualmente, quando interagimos com sistemas computacionais. Ao fazer login no e-mail, ao preencher um cadastro, ao visualizar informações de um website bancário, postar uma foto em uma rede social, etc.

A interface de uma página WEB que acessa banco de dados possui recursos visuais de interação, como formulários com caixas de texto, caixas de seleção, botões, etc. É necessário organizar estas informações e armazená-las, de forma a conseguir recuperá-las e, inclusive, modificá-las. Para isto existem os sistemas gerenciadores de bancos de dados

(SGBDs), como o *Microsoft SQL Server*, *Oracle*, *MySQL*, entre outros. A maioria dos bancos de dados são relacionais, ou seja, armazenam as informações em tabelas. Para fazer a conexão da aplicação com o banco de dados, usa-se um elemento chamado interface. Em páginas WEB desenvolvidas em ASP.NET esta interface chama-se ADO.NET.

Com ADO.NET podemos acessar qualquer tipo de banco de dados, inclusive dados em formato XML. Mas, antes de vermos exemplos de interface gráfica para WEB em ASP.NET com interface ADO.NET acessando banco de dados SQL *Server* e XML.

O professor Ederson contextualiza melhor esse tema no material online. Confira!

Tema 01 - Conceitos de Banco de Dados

Banco de Dados

É uma coleção organizada de informações.
SGBD: sistema gerenciador de banco de dados.
Sistema relacional: modelo entidade – relacionamento.
Linguagem: SQL (*Structured Query Language*).
Interface: conecta a aplicação ao banco de dados.

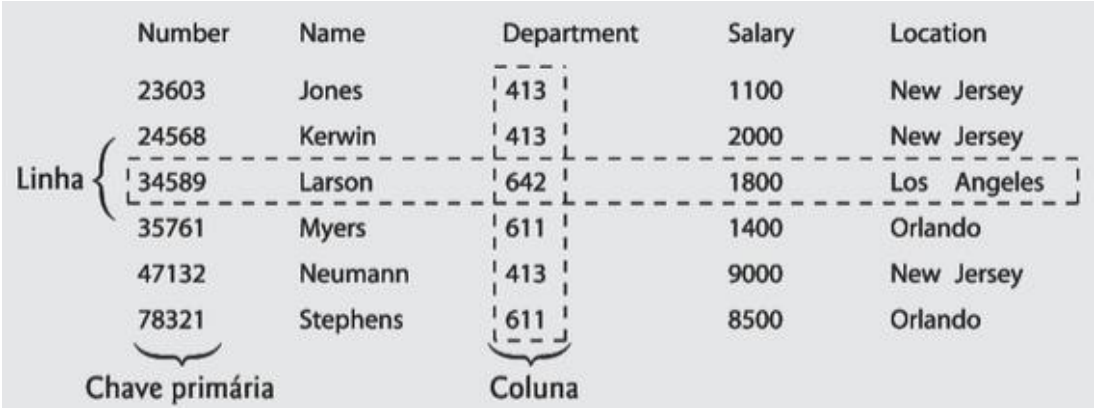
Tabela

É aonde os dados são armazenados e de onde os dados são consultados. É a entidade em si. Entidade é um objeto ou a representação de um conjunto de informações. Ex.: cliente, produto, usuário, livro, etc.
A entidade possui atributos, que são as colunas da tabela. Os valores associados aos atributos são os registros, que são as linhas da tabela.

Chave primária: é um valor exclusivo, que não pode ser repetido em outras linhas.

A consulta em uma tabela permite selecionar determinados dados de interesse.

Exemplo:



Consulta dos departamentos e localização:

Department	Location
413	New Jersey
611	Orlando
642	Los Angeles

Relacionamento

- 1 para 1. Ex.: uma pessoa possui apenas um currículo.
- 1 para muitos. Ex.: um autor pode ter vários livros.
- Muitos para muitos: um livro pode ter vários autores, assim como, um autor pode ter vários livros.

Diagrama Entidade-Relacionamento:

Comandos SQL

- Consulta

SELECT * FROM nomeTabela WERE critério

- Inserção

INSERT INTO nomeTabela(nomeColuna1, ...) VALUES (valor1, ...)

- Apagar

DELETE FROM nomeTabela WERE critérios

Bancos de Dados

Segue abaixo alguns SGBDs disponíveis no mercado:

- MySQL
- Microsoft SQL Server
- Access
- Oracle
- PostgreSQL
- InterBase
- Firebird

Agora, assista no material online, um trecho da videoaula onde o professor Ederson falará sobre cada um destes conceitos.

Tema 2- Conceitos de ADO.NET

ADO.NET (*Active Data Objects*)

ADO.NET é orientado à objetos.

Funcionalidades:

- Interface: conecta-se a um banco de dados.
- Comanda o banco de dados.
- Utiliza os resultados.

ADO.NET Integrado com XML

- XML é a base do ADO.NET.
- Qualquer tipo de dado pode ser suportado.
- XML pode ser transportado sobre o protocolo HTTP.
- Permite trocar dados entre objetos localizados em qualquer lugar.

Modo desconectado do ADO.NET

- A aplicação não precisa de uma conexão persistente com a fonte de dados.
- A classe *DataSet* funciona como um cache local de dados.
- A conexão com o banco de dados é feita apenas quando é necessário salvar dados permanentemente.

Provedores Gerenciados

São classes que estabelecem a conexão com um banco de dados específico. Basicamente existem 2 classes:

- SQL *Client*: provedor utilizado para conectar ao SQL *Server*.

- OLE DB: provedor utilizado para conectar a outras fontes de dados que tem suporte à API OLE DB.

Provedores adicionais: classes específicas, como ODBC, *Oracle Client*, entre outros. São gerenciados porque a plataforma .NET reserva ou libera memória automaticamente.

Namespaces

O *namespace System.Data* é o espaço de nomes raiz para a API ADO.NET.

Variantes:

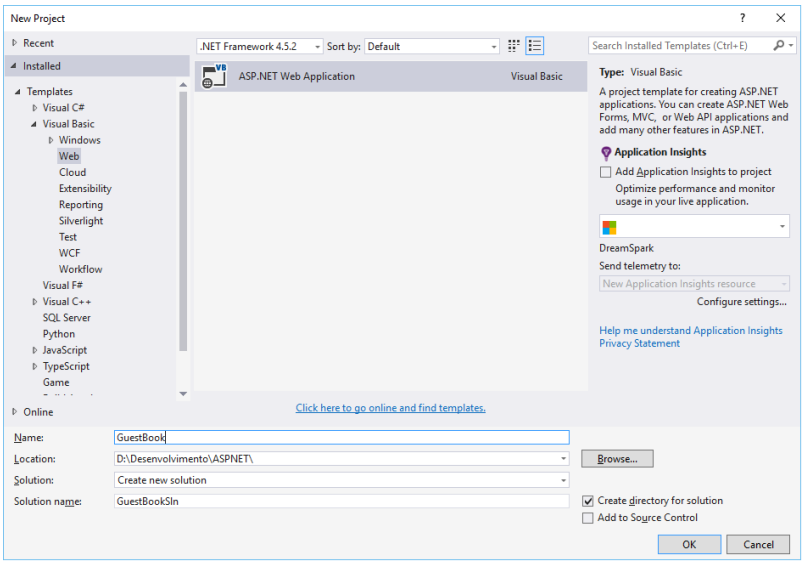
- *System.Data.OleDb*: trabalha com fontes de dados com suporte a OLE DB.
- *System.Data.SqlClient*: trabalha com a fonte de dados SQL *Server*.
- *System.Data*: representa a classe *DataTable*, que compreende as coleções *DataRows* e *DataColumns*.
- *System.Data.DataSet*: representa um conjunto de *DataTables*, consiste no cache de dados, imitando a estrutura de um banco de dados.

Agora, assista no material online um trecho de videoaula onde o professor Ederson falará mais sobre este tema.

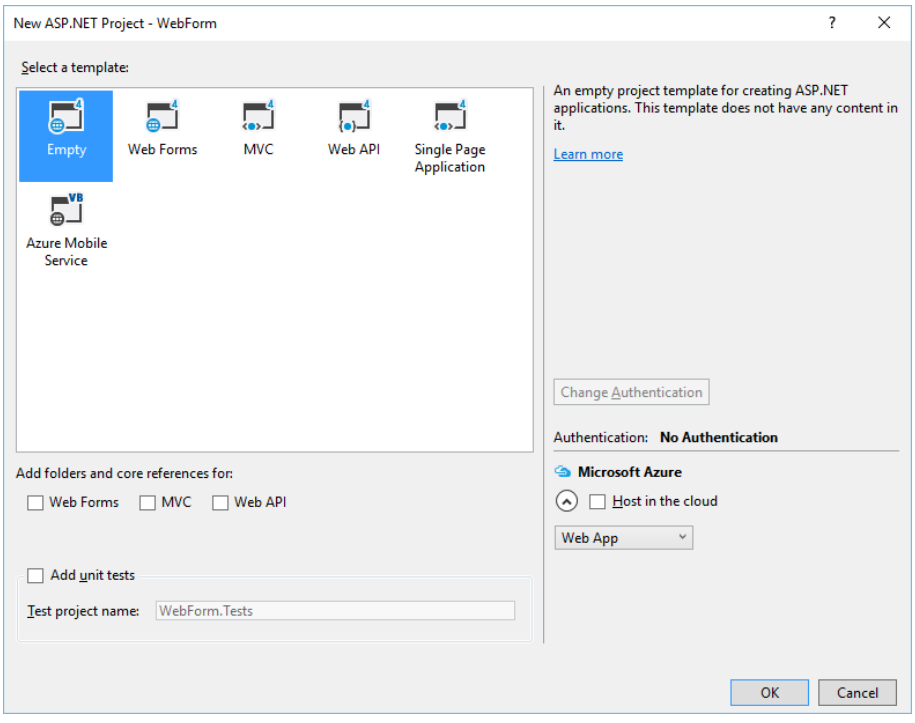
Tema 3- Criando um projeto ASP.NET no Visual *Studio Community* 2015

Criando uma aplicação ASP.NET com Banco de Dados SQL *Server* no Visual *Studio Community* 2015.

1º Passo: Menu File → New → Project...

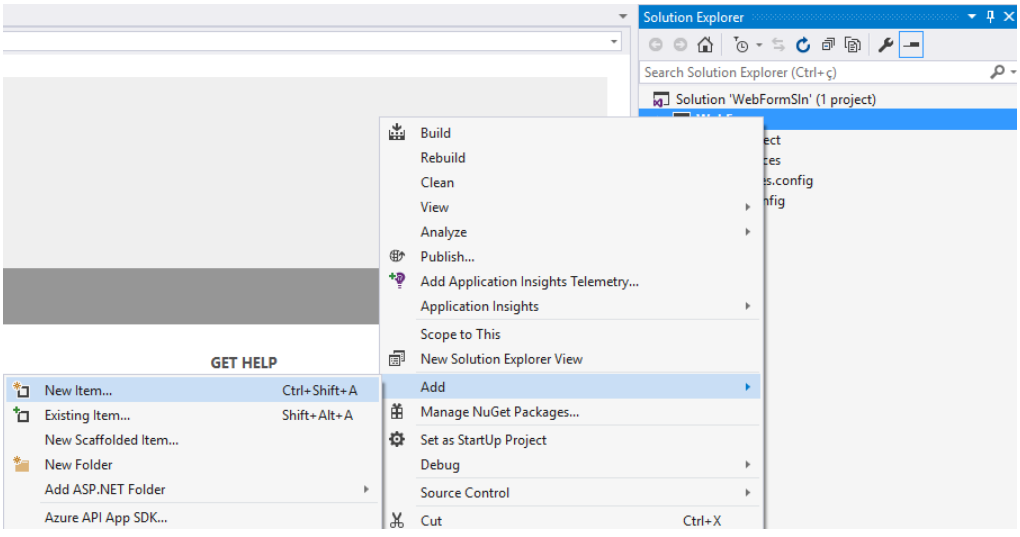


2º Passo: Especificando o template: escolher a opção *Empty*.

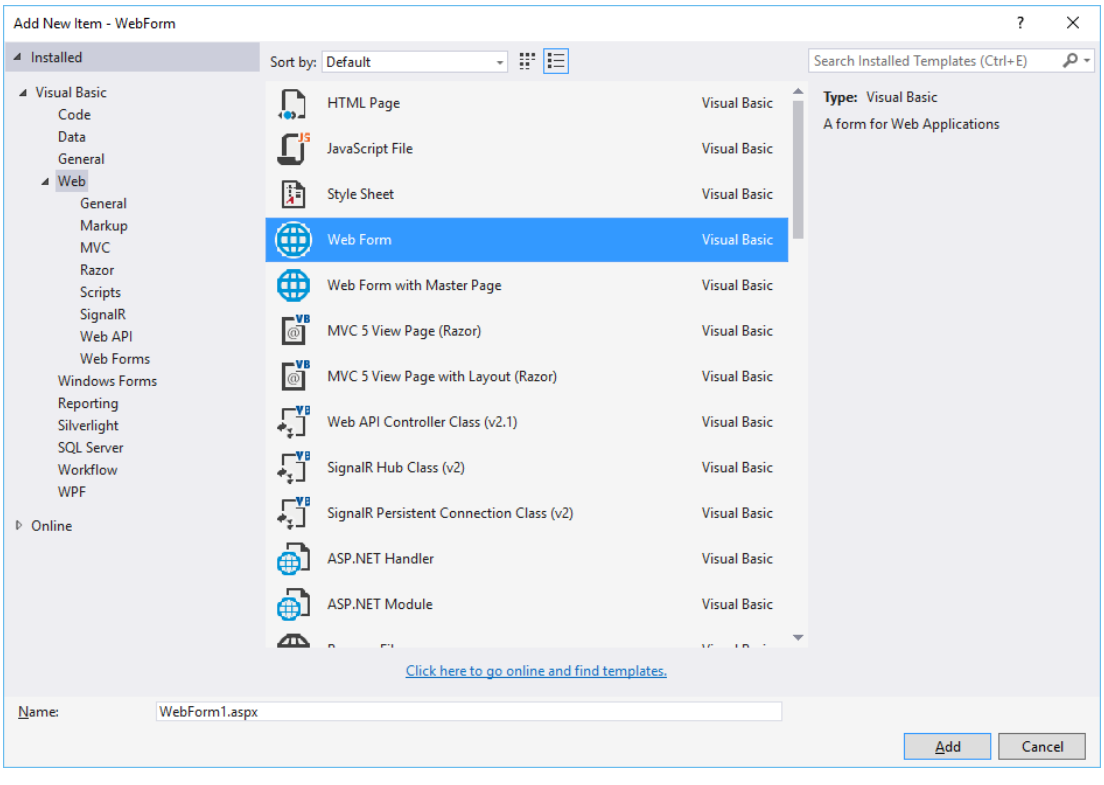


3º Passo: Adicionando um Web *Form*

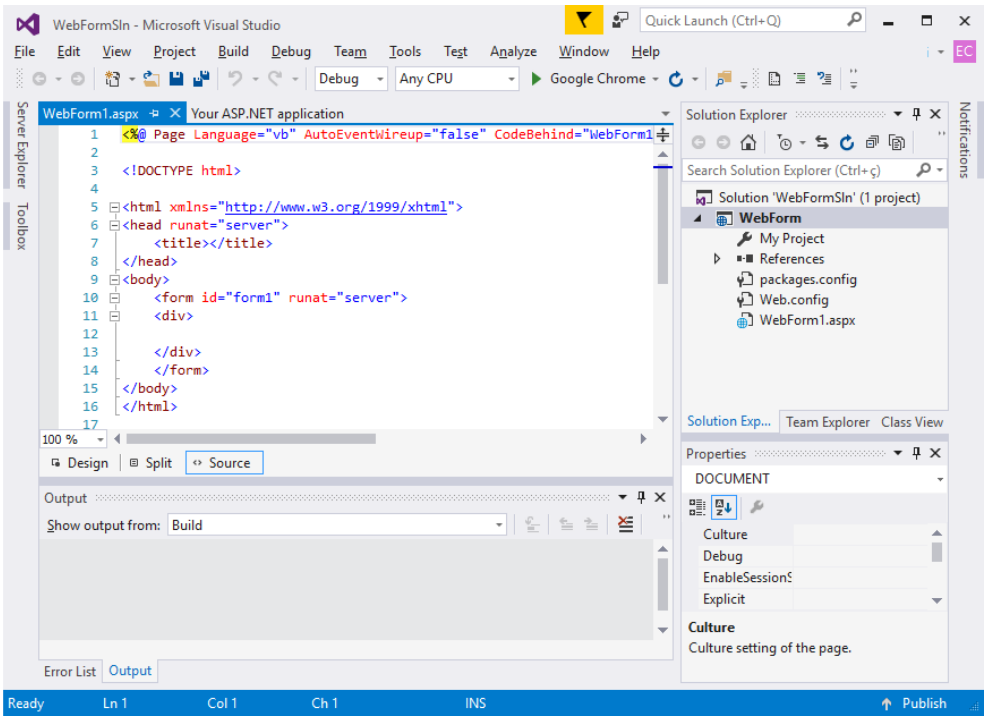
Clicar com o botão direito no nome do projeto, ir em Add → New Item...



Escolher Web *Form.*

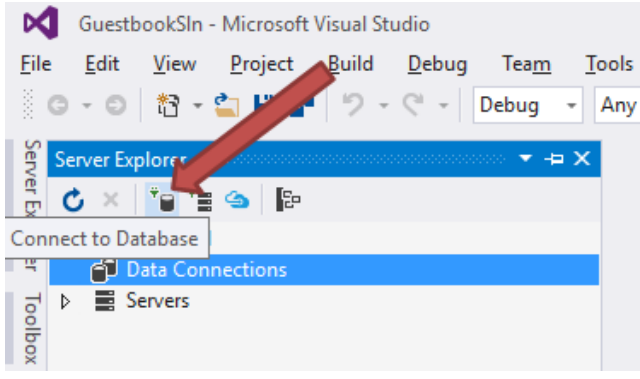


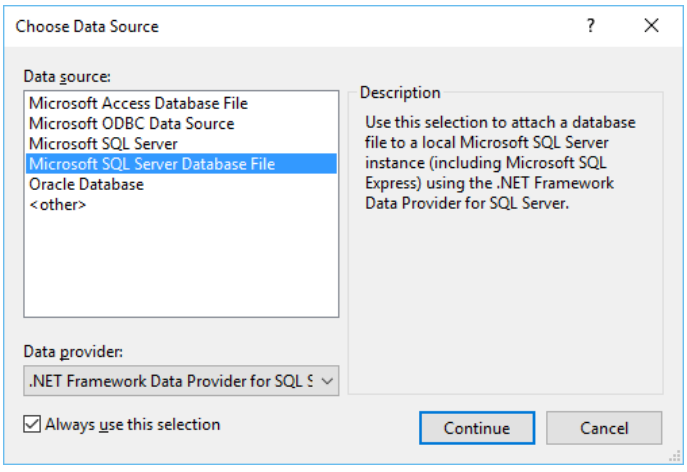
Ambiente de desenvolvimento:



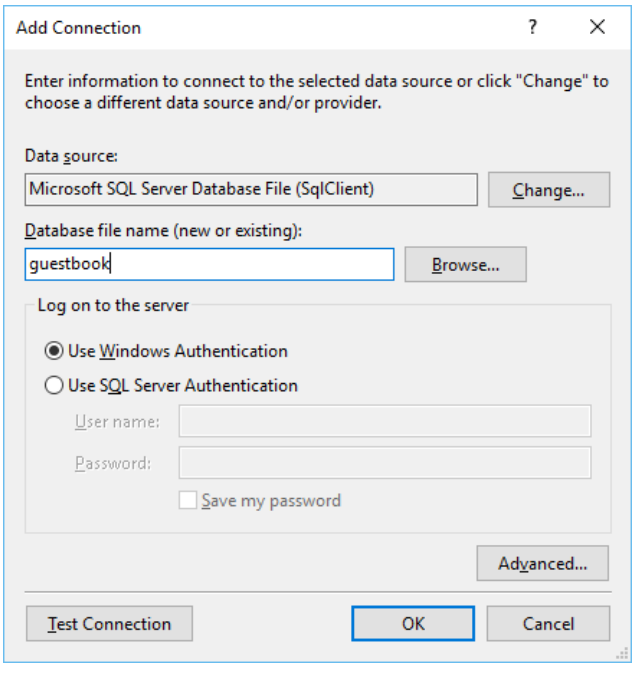
4º Passo: Server Explorer:

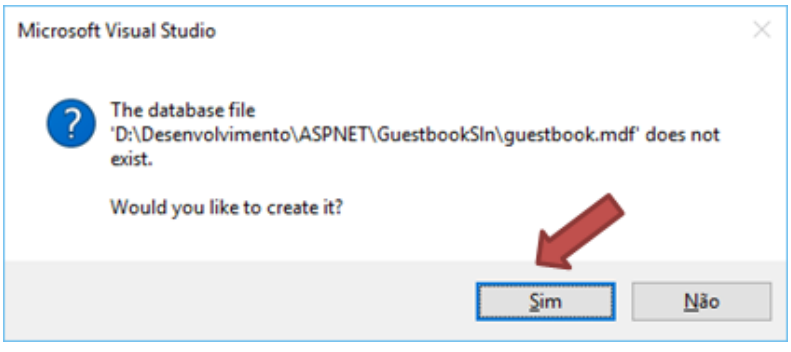
Ir na janela Server Explorer e clicar no botão "*Connect to Database*".



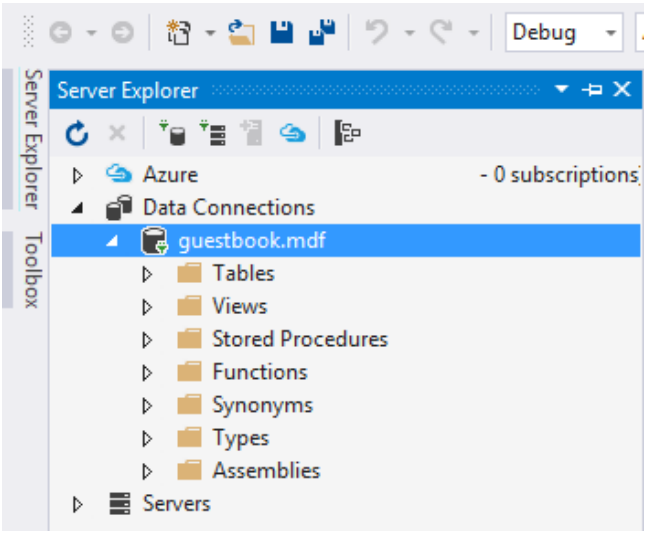


Vamos criar um banco de dados, e vamos especificar um nome para o banco de dados, em nosso exemplo, vai chamar "*guestbook*".



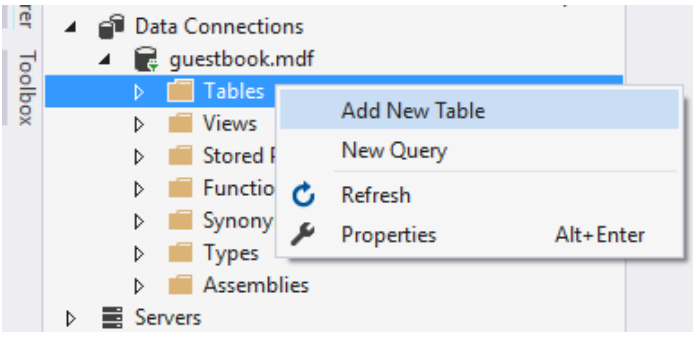


Pronto, agora o banco de dados está criado com o nome guestbook.mdf, mas vazio.



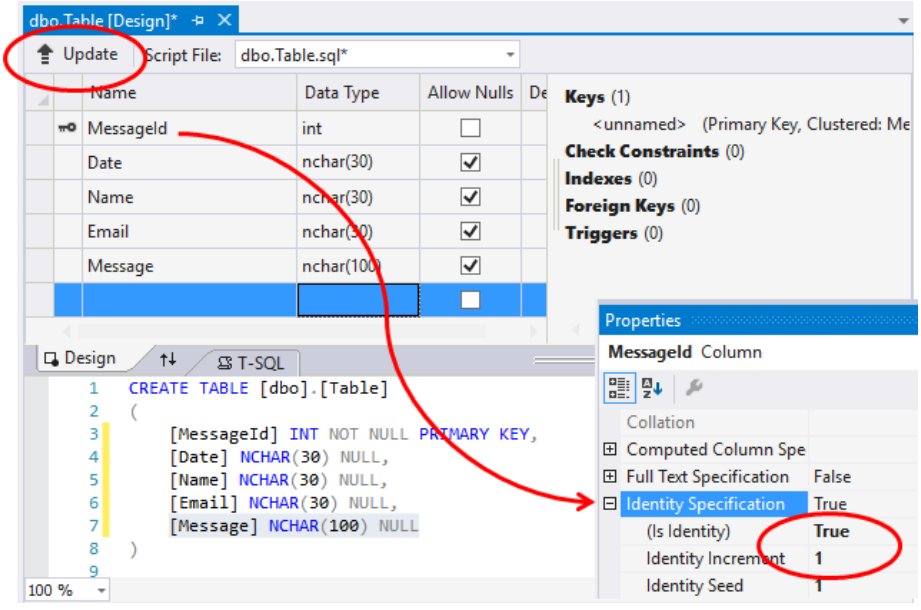
5º Passo: Criando uma Tabela

Vamos clicar com o botão direito do mouse em *Tables* e escolher a opção "*Add New Table*".

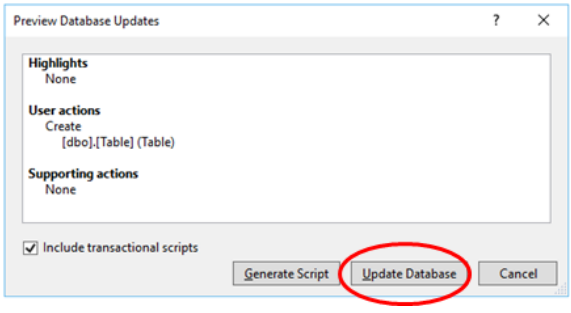


Irá abrir a janela de design dos campos da tabela, a primeira linha é a chave primária, vamos nomear de MessageId. Vamos deixar o mouse clicado nesta linha e vamos na janela *Properties*, devemos marcar a opção "*Is Identity*" como *True*. Então podemos criar os demais campos, inserindo mais linhas conforme a figura abaixo.

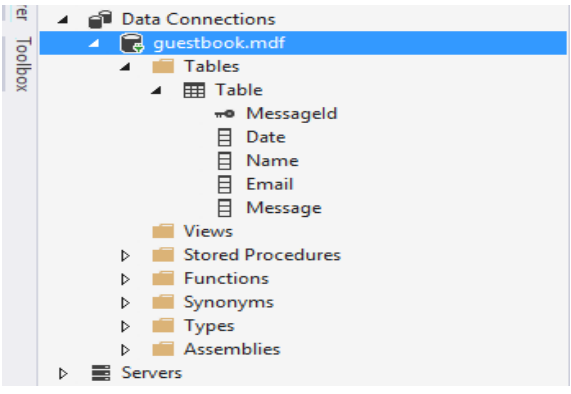
Ao finalizar de criar os campos da tabela, clicar no botão Update.



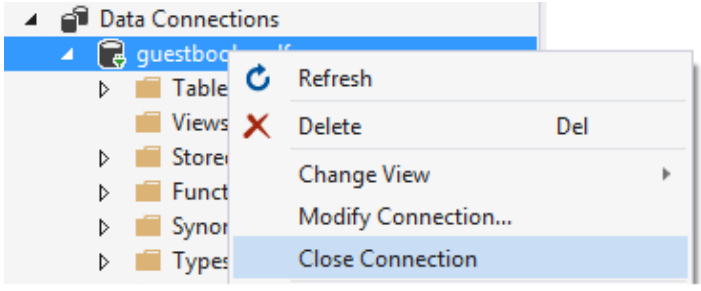
Na janela que abrir clicar em Update Database.



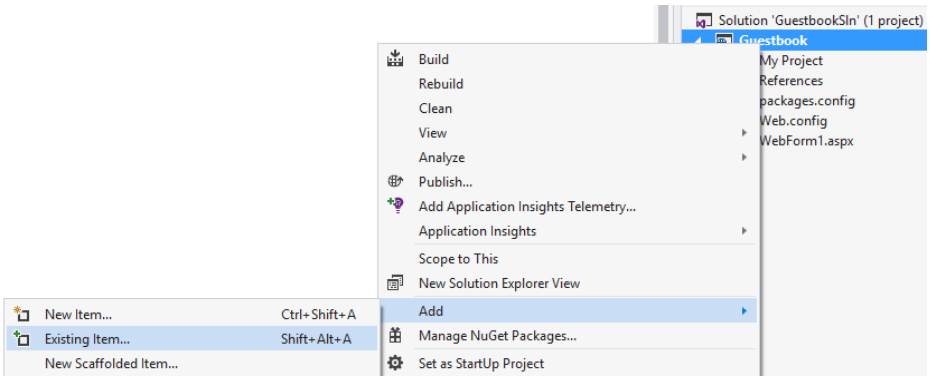
Podemos ver a tabela criada no banco de dados.



6º Passo: Agora adicionar ao projeto o arquivo guestbook.mdf: Primeiro é necessário fechar a conexão do banco de dados, conforme a figura a seguir.



Então adicionar o arquivo do banco de dados ao projeto:



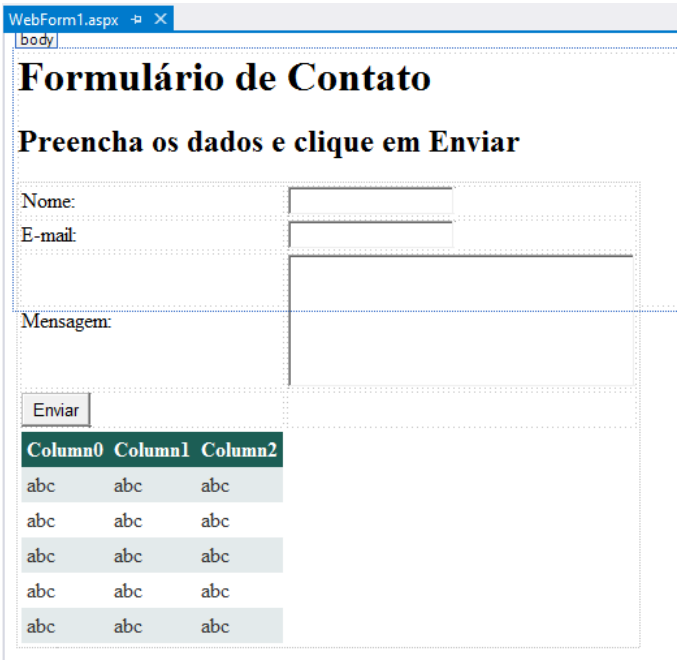
Saiba mais sobre este assunto, acessando o material online

Tema 4 - Página ASP.NET com SQL *Server*

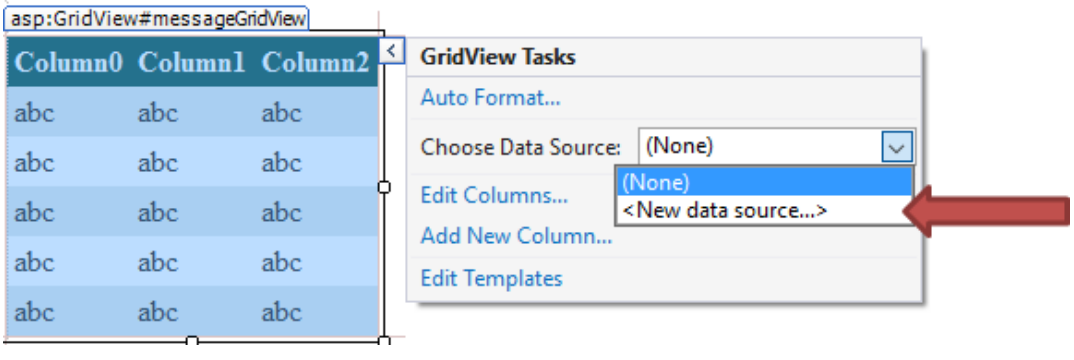
Que tal continuarmos com o projeto iniciado? Agora vamos implementar uma interface para o usuário inserir informações no banco de dados. Confira o passo a passo.

1º Passo: Inserindo um formulário na página ASP.NET. No modo design insira os componentes conforme o layout abaixo. Para os *TextBox* nomeie como *NameTextBox, emailTextBox* e *messageTextBox*. O botão nomeie *submitButton*. O *GridView* nomeie *messagesGridView*.

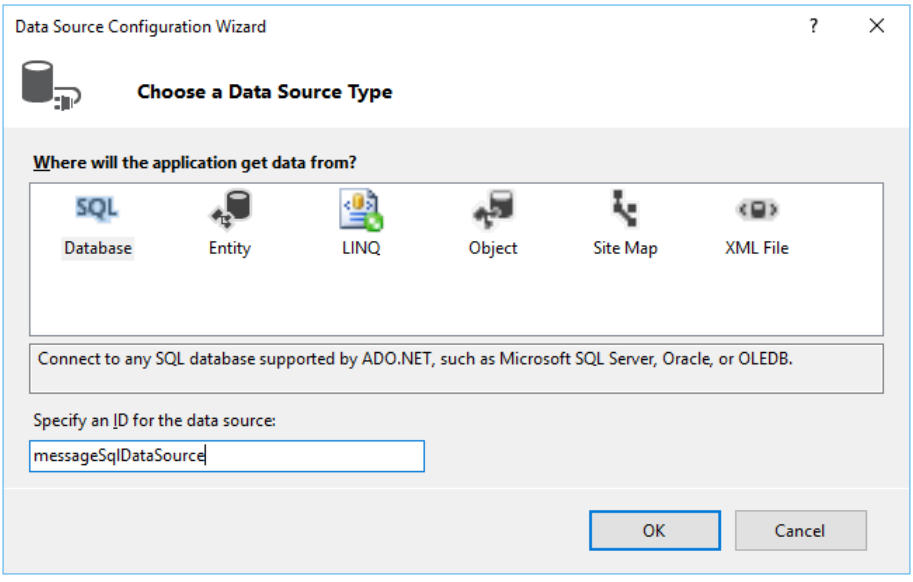
A tabela é um componente *GridView* que fica seção Data do *Toolbox*.



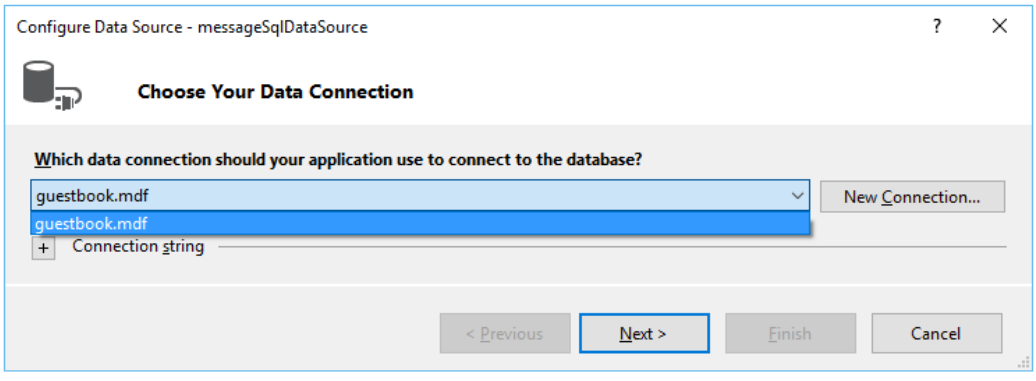
2º Passo: Vinculando a *GridView* à tabela *Messages* do banco de dados. Clique naquela pequena seta para a direita no componente *GridView*, em "*Choose Data Source*" clique em "*New data source*...".



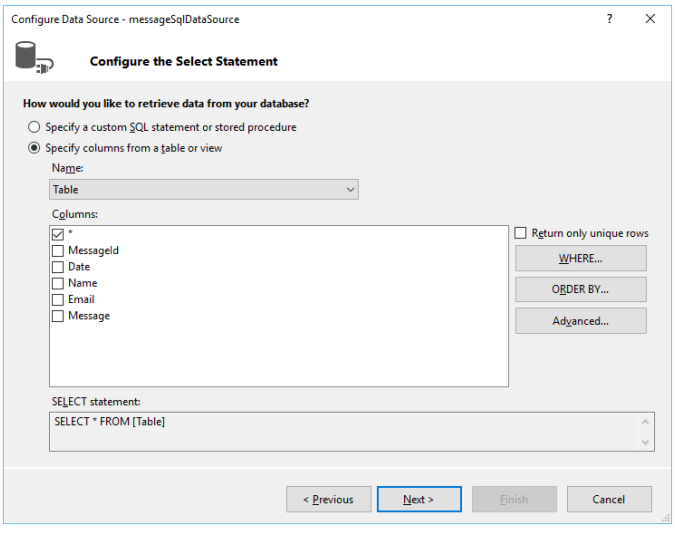
Dê um nome para a fonte de dados, neste exemplo é "messageSqlDataSource".



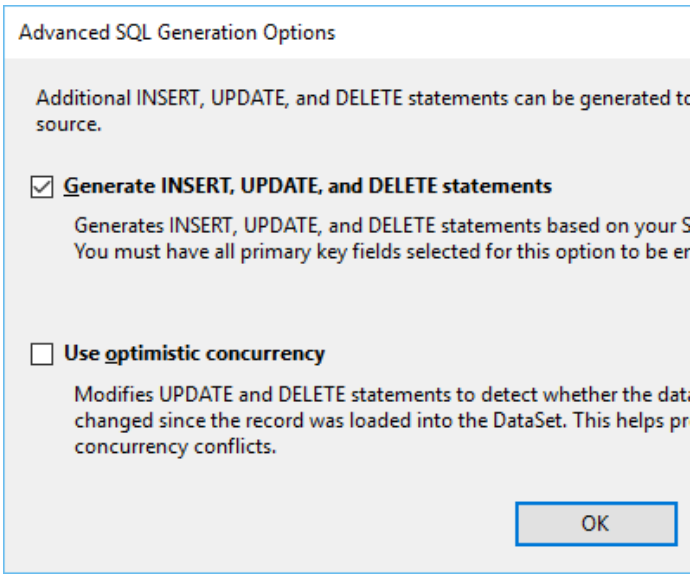
Especifique o arquivo do banco de dados guestbook.mdf.



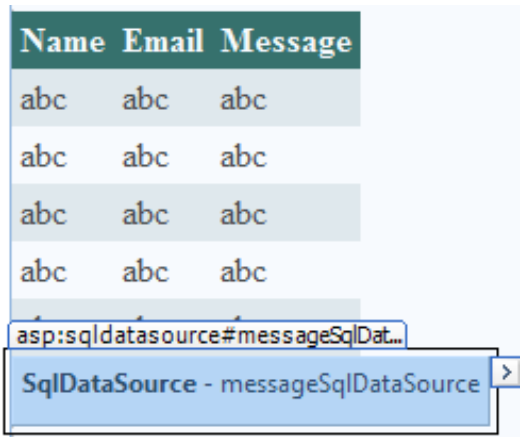
Deixe marcado o asterisco (*) para selecionar todos os campos.



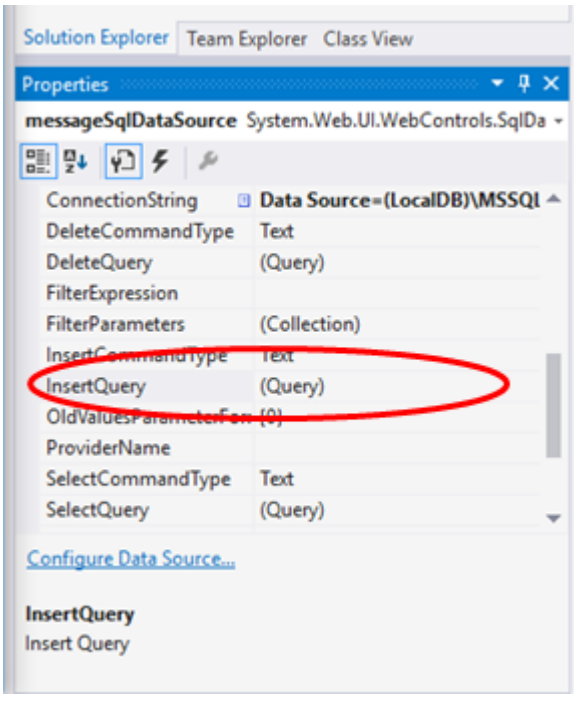
Clique em *Advanced*... e marque a primeira opção.

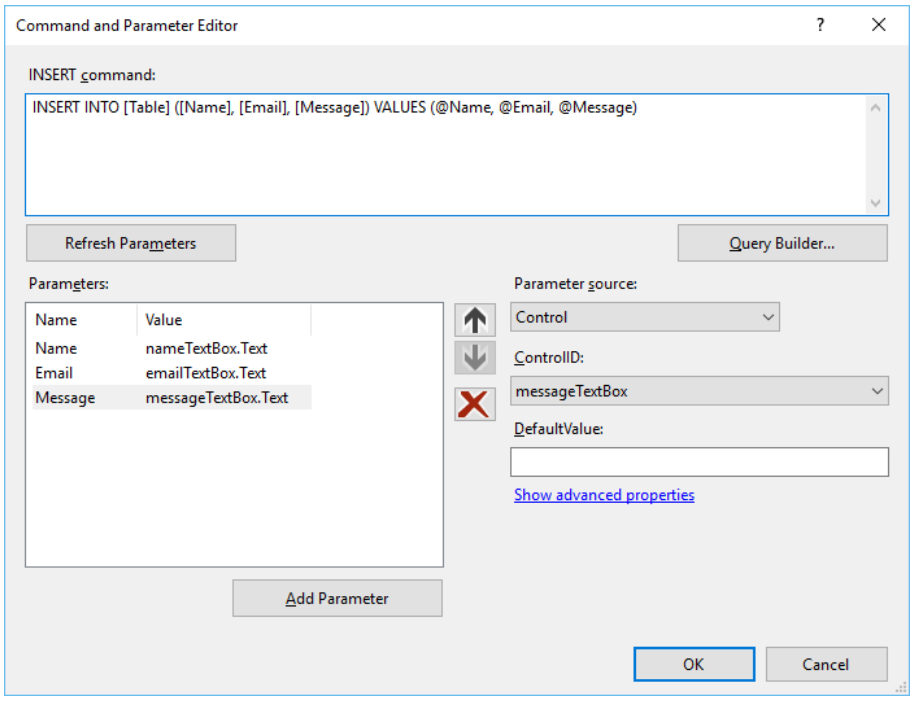


3º Passo: Configurar o Data *Source*. Clique no componente Data *Source*, junto da *Grid View*, vá na janela *Properties*, no *item Insert Query*, clique ao lado em (Query), irá abrir uma janela.



Configurar *Name, Email* e *Message* em *Parameter source* para "*Control*" e associar em *ControlID* o *TextBox* correspondente.





4º Passo: Inserindo o Código para o Botão. *Code behind* VB.NET no click do botão.

```
Protected Sub submitButton_Click(sender As Object, e As EventArgs) Handles submitButton.Click
    messageSqlDataSource.Insert()
    nameTextBox.Text = ""
    emailTextBox.Text = ""
    messageTextBox.Text = ""
    messageGridView.DataBind()
End Sub
```

5º Passo: Rodando a página

Digite um conteúdo nas caixas de texto e clique no botão enviar. O conteúdo será armazenado no banco de dados e mostrado na tabela *Grid View*.

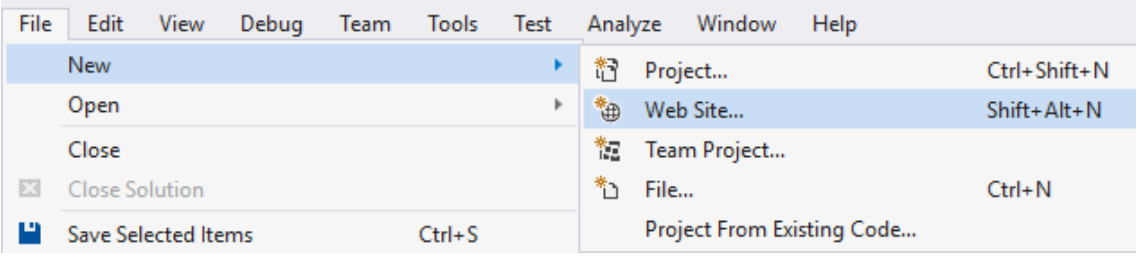
Agora é com o professor Ederson! Veja o que ele tem a dizer sobre esse assunto no material online!

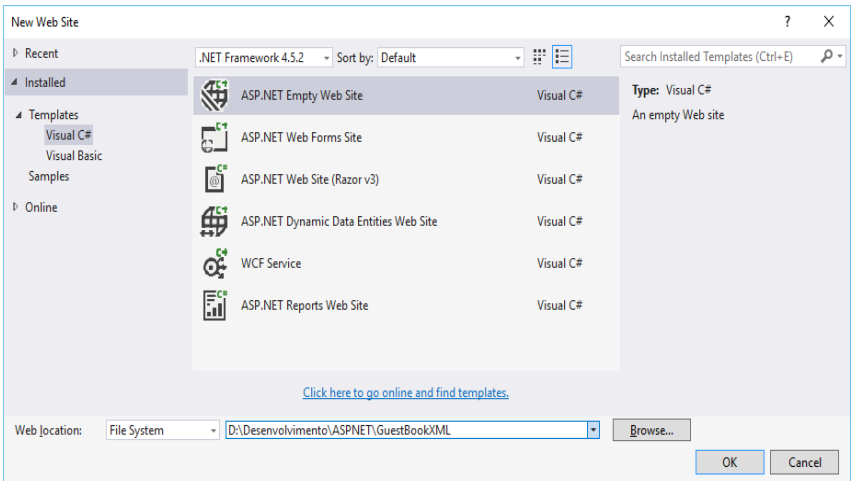
Tema 5 - ASP.NET com database XML

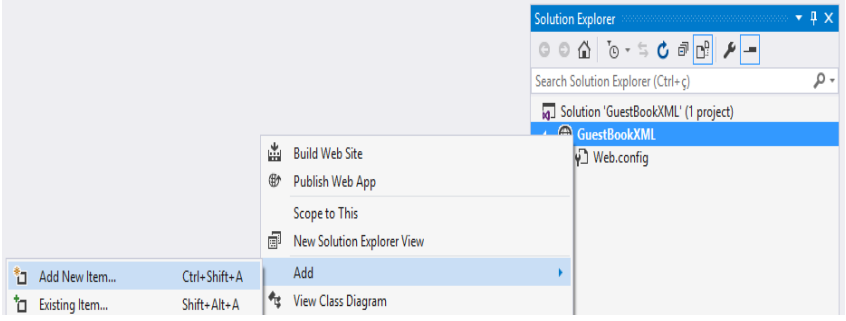
Por fim, vamos criar uma aplicação que acessa uma base de dados em XML.

1º Passo: Criando uma Página com ADO.NET e database XML.

Primeiramente criar um novo projeto e inserir um Web *Form*.

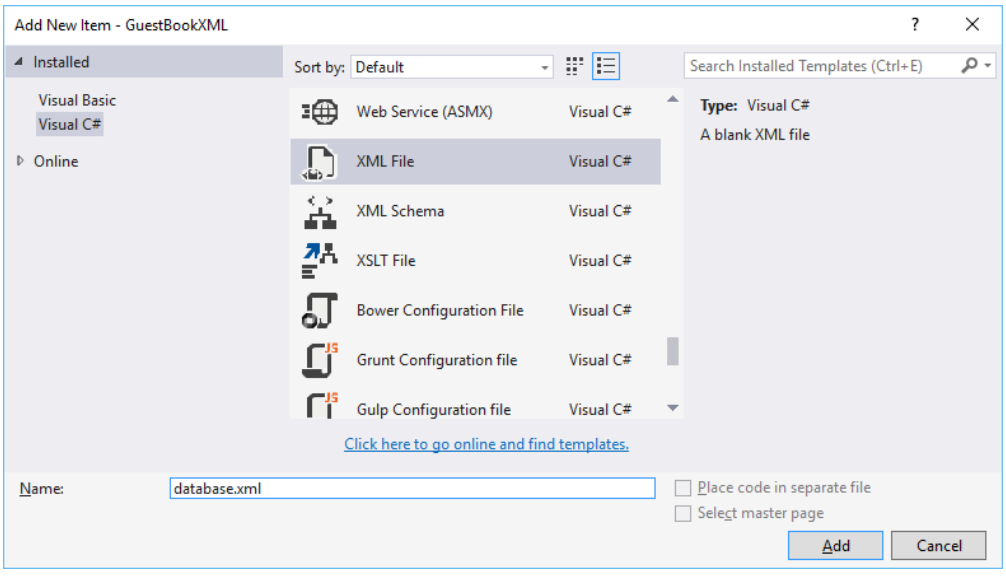






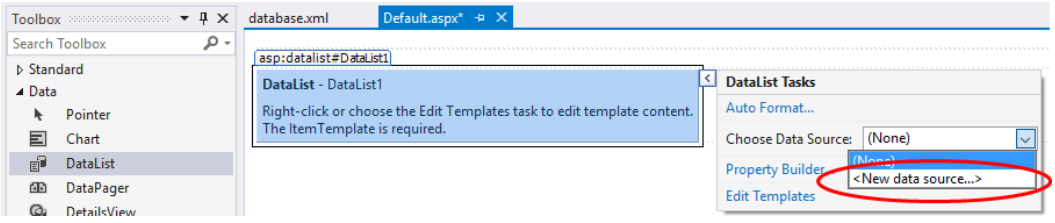
Agora vamos inserir um arquivo XML. Nomeá-lo, como neste exemplo, de database.xml.

Novamente Add → Add New Item.

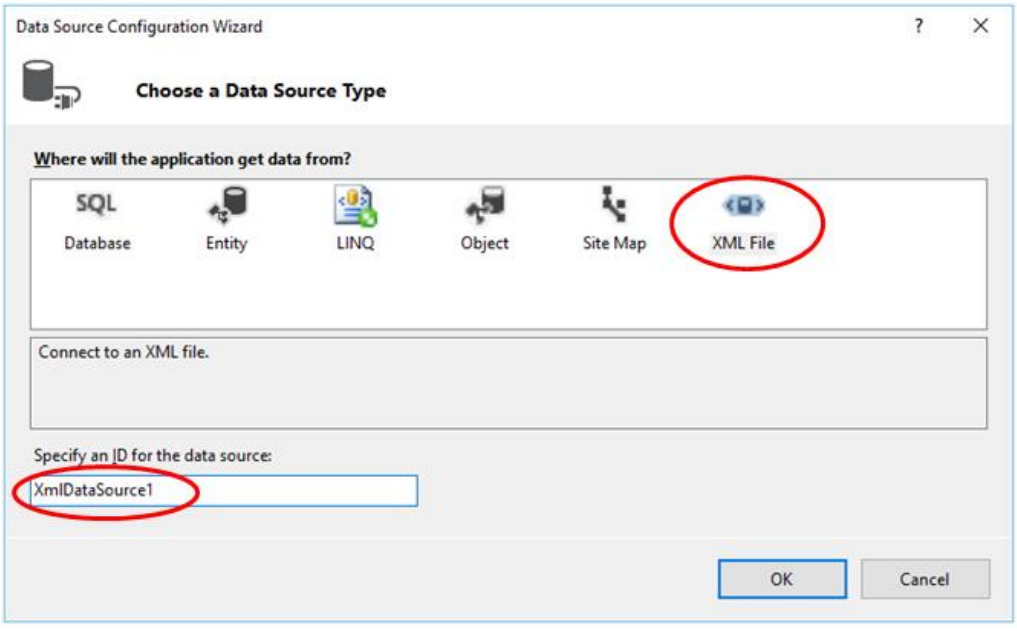


2º Passo: Inserindo um Componente que Acessa Dados

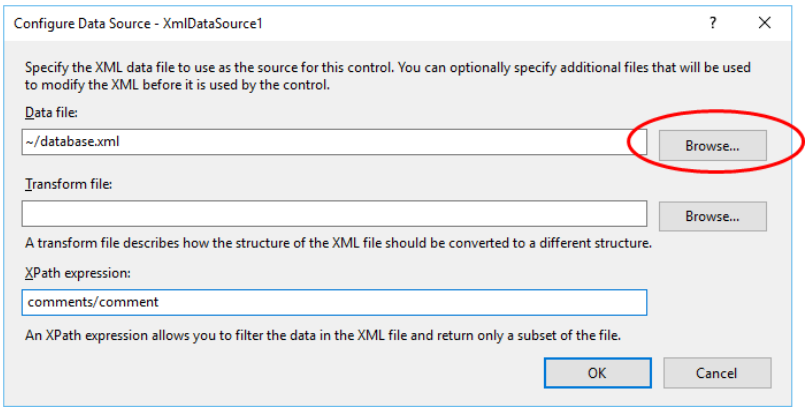
Inserir um componente *DataList* da seção Data do *Toolbox* no modo Design e configurar o Data *Source*.

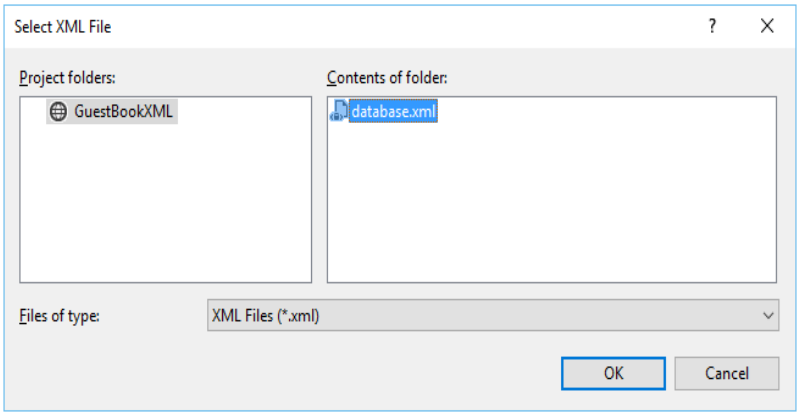


Criar um Data *Source* que aponta para o arquivo database.xml.



Em XPath *expression*, escreva "*comments/comment*", conforme abaixo. Mais para frente vamos entender o significado deste parâmetro ao criar o conteúdo do arquivo database.xml





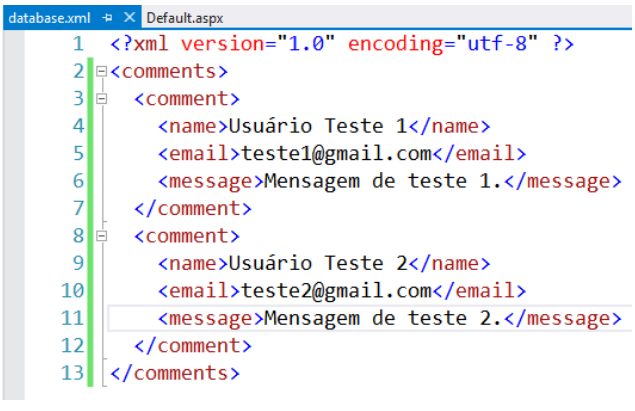
3º Passo: Codificação ASP.NET

Código ASP.NET com # usando XPath para acesso a dados do arquivo XML.

```
<asp:DataList ID="DataList1" runat="server" DataSourceID="XmlDataSource1">
        <ItemTemplate>
        <table>
            <tr><td>Message <%# XPath("message") %> </td> </tr>
            <tr><td>Posted by <b><%# XPath("name") %></b> (<%# XPath("email") %>)
            </td> </tr>
        </table>
    </ItemTemplate>
</asp:DataList>
```

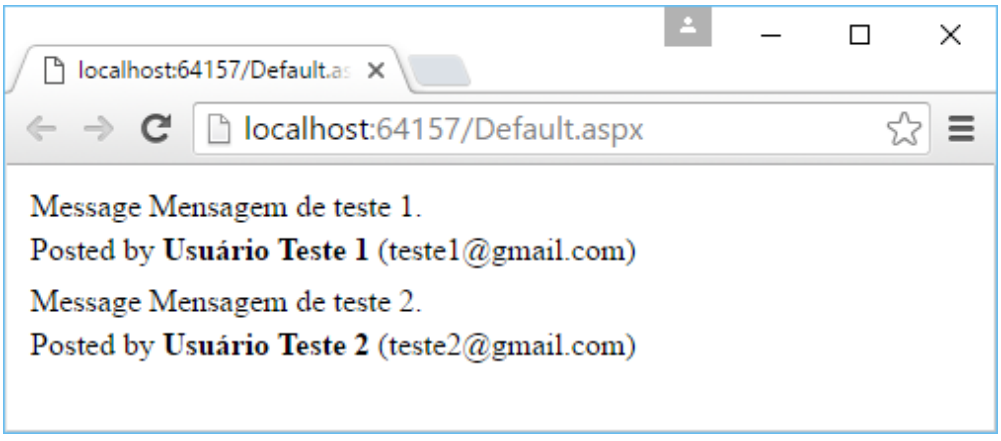
4º Passo: Código XML

Escrever o código abaixo no arquivo database.xml. Aqui podemos ver que temos uma *tag root* chamada *comments*, e as *tags* filhas chamadas *comment*, aonde irá constar a estrutura da tabela, tendo como atributos (ou colunas) as *tags name*, email e *message*.



5º Passo: Rodando a página no browser.

Os dados lidos do arquivo XML são mostrados na página WEB.



O que será que o professor André tem a ensinar sobre este conteúdo? Acesse o material online.

Na Prática

Para consolidar os conteúdos vistos nesta aula, vamos aprimorar a aplicação de formulário de contato (*guestbook*) desenvolvido no tema 4 desta aula, adicionando a possibilidade de realizar uma consulta no banco de dados. A consulta será feita pelo nome do usuário. Vamos lá?

1º Passo: Vamos abrir o projeto Formulário de Contato (*GuestBook*).

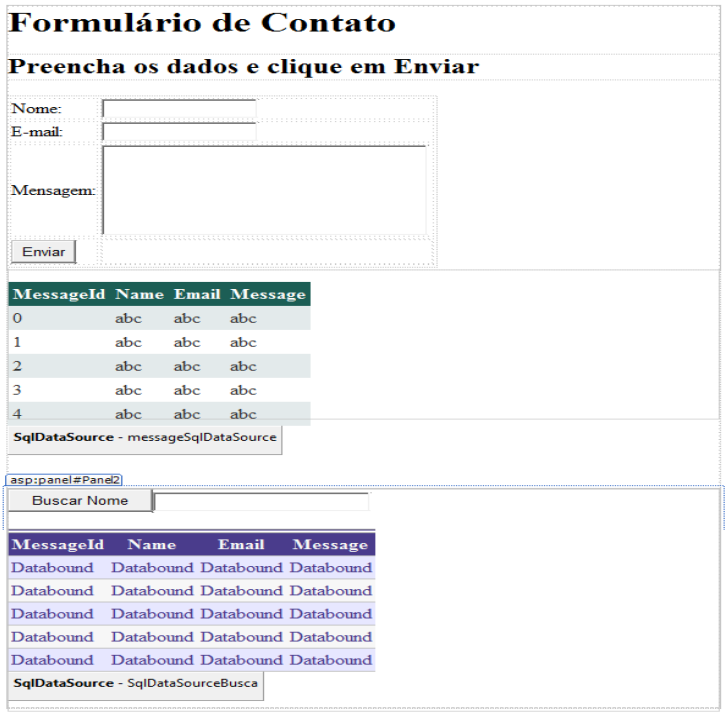
Inserir mais 3 componentes no modo Design:

- Botão: propriedade *Text* = Buscar Nome

- Caixa de Texto: ID = nomeTextBox

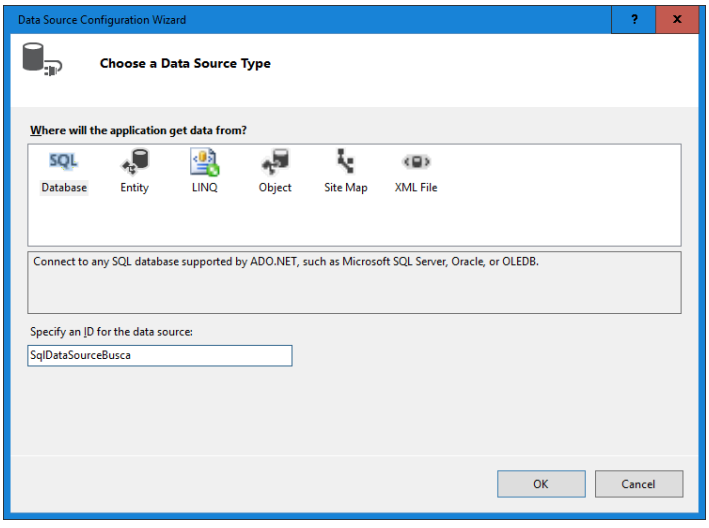
- *GridView*: ID = buscaGridView

O WEB *form* no modo Design deve ficar conforme a figura abaixo:



2º Passo: Configurar a base de dados para o buscaGridView que vai mostrar o resultado da busca.

Clicar na setinha para a direita no buscaGridView → Choose Data Source → New Data Source. Dê o nome de SqlDataSourceBusca.



As demais configurações são exatamente iguais às realizadas para o messageGridView, selecionando o mesmo banco de dados guestbook.mdf.

3º Passo: Configurar a *query* de busca

Clicar no componente SqlDataSourceBusca, ir na janela *Properties*, ir no item *SelectQuery* e clicar em *Query*.

No campo *SELECT command* inserir a *query*:

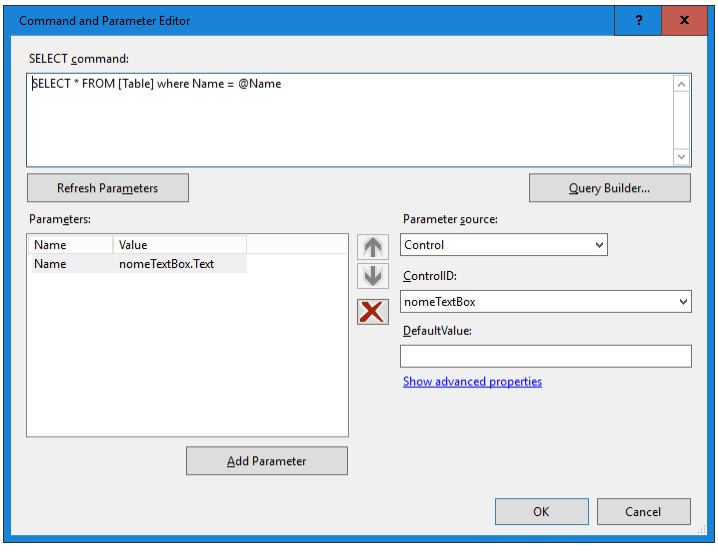
SELECT * FROM [Table] where Name = @Name

Então clicar no botão *Refresh Parameters.*

Clicar no parâmetro *Name*, ir em *Parameter source* e escolher "*Control*".

Ir em ControlID e escolher "nomeTextBox".

A tela deve ficar como a figura abaixo. Então clicar em OK.



4º Passo: Inserir código no botão Buscar Nome

Dar 2 cliques no botão Buscar Nome e inserir o código (*code behind* VB.NET):

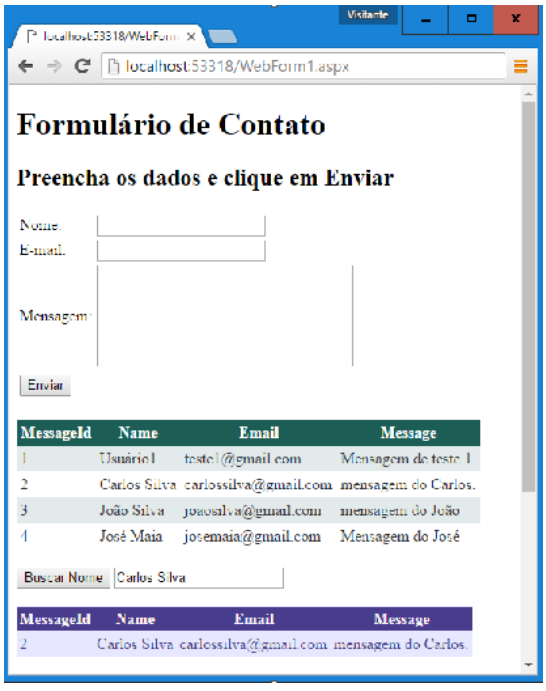
buscaGridView.DataBind()

O código do botão Buscar Nome fica assim:

```
Protected Sub buscarButton_Click(sender As Object, e As EventArgs) Handles buscarButton.Click
    buscaGridView.DataBind()
End Sub
```

5º Passo: Rodar a aplicação

No Visual Studio apertar no teclado a tecla F5 para rodar a aplicação, iniciando o servidor IIS *Express*, o servidor SQL *Server Express* e abrindo a página no browser. Ao digitar no campo de texto ao lado do botão de busca um dos nomes já armazenados no banco e clicar no botão Buscar Nome, será apresentado abaixo o resultado da busca, conforme a figura abaixo.



Síntese

Nesta aula vimos os conceitos de banco de dados e linguagem SQL. Conhecemos também a interface de acesso a bancos de dados ADO.NET, seus recursos básicos e desenvolvimento com Visual Studio. Por fim, criamos uma página ASP.NET que acessa um banco de dados SQL *Server Express* com *code behind* em Visual *Basic* .NET e também uma página ASP.NET que usa banco de dados em arquivo XML com *code behind* em C#.

O bom entendimento desta aula é fundamental, visto que trata dos recursos de interface visual baseada em *forms* para WEB com acesso a bancos de dados, utilizados na criação de sites da internet que rodam scripts no servidor e trabalham com bancos de dados diversos.

O professor Ederson faz uma síntese desses assuntos no material online, não perca!

Referências

DEITEL, P. J. Ajax, **Rich Internet Applications e desenvolvimento WEB para programadores**. São Paulo: Pearson, 2008.

WAHLIN, D. **XML e ASP.NET para desenvolvedores**. São Paulo: Pearson Education. 2003.

SHEPHERD, George. **Microsoft ASP.NET 3.5 Passo a Passo**. São Paulo: Bookman, 2009.