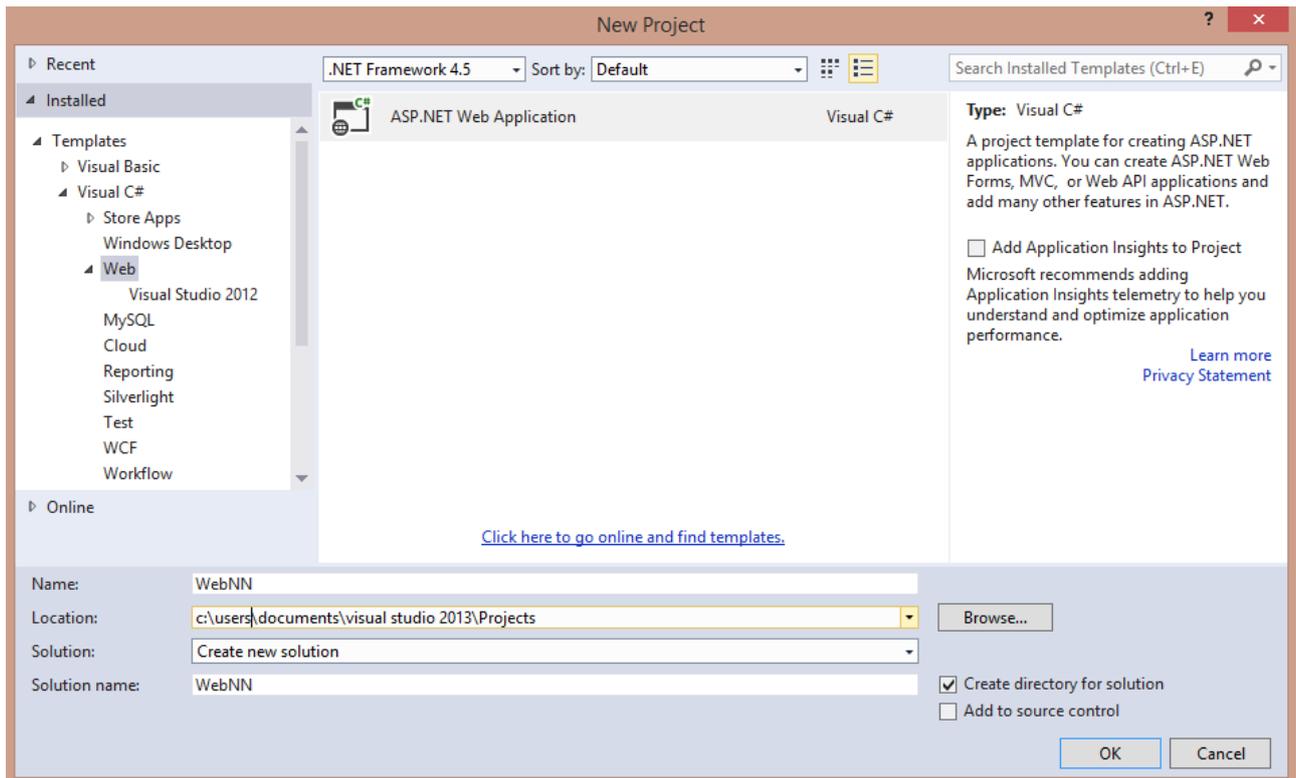


Associação N:N

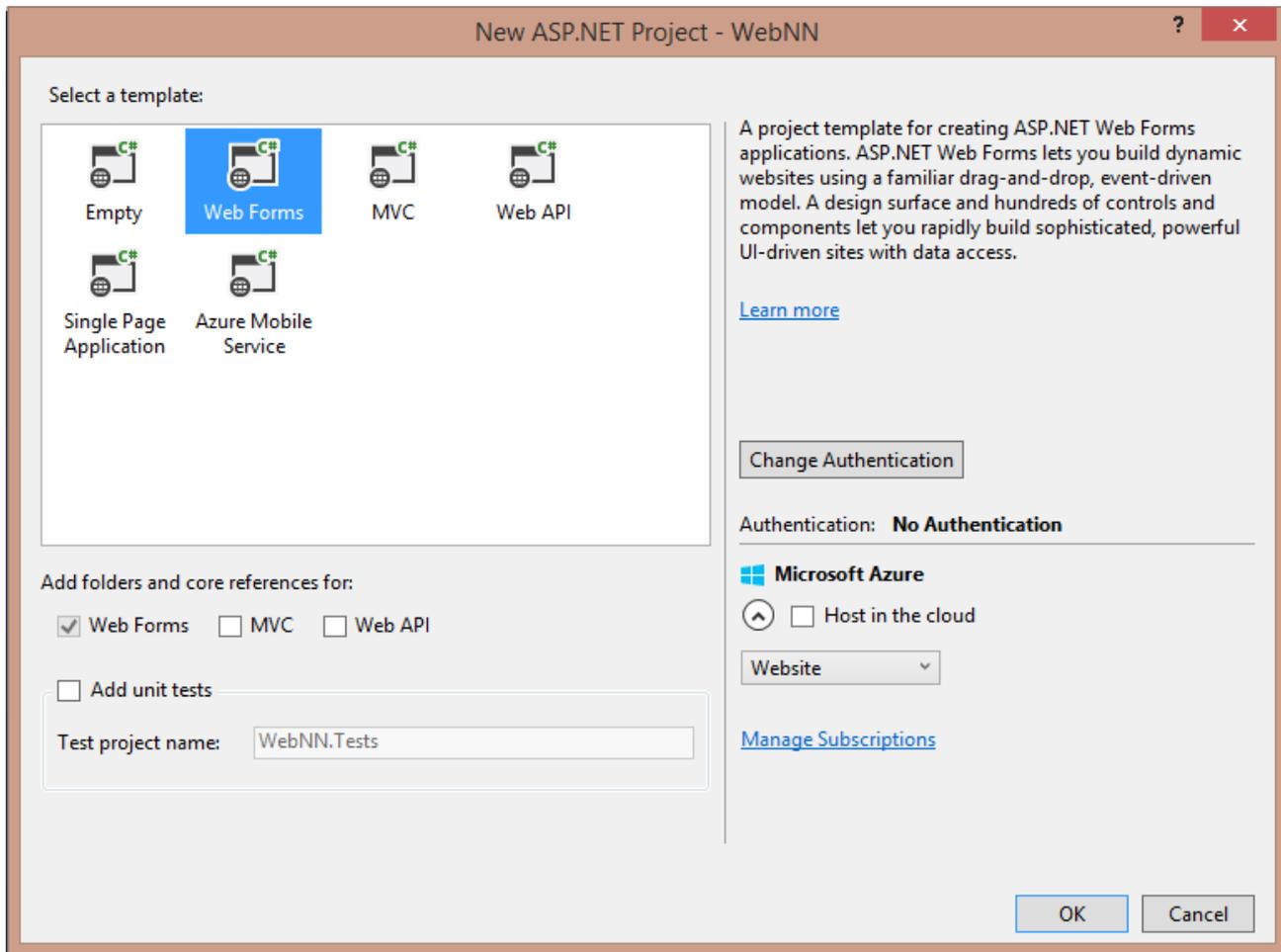
Abrir o Visual Studio

Criar um novo projeto (File > New Project)

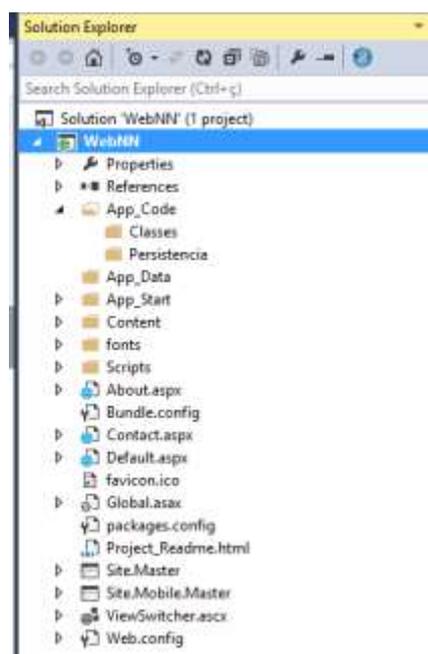
Selecione Visual C# > Web
.NET Framework 4.5 ou superior
Nome da Solução: WebNN
Clique em OK.



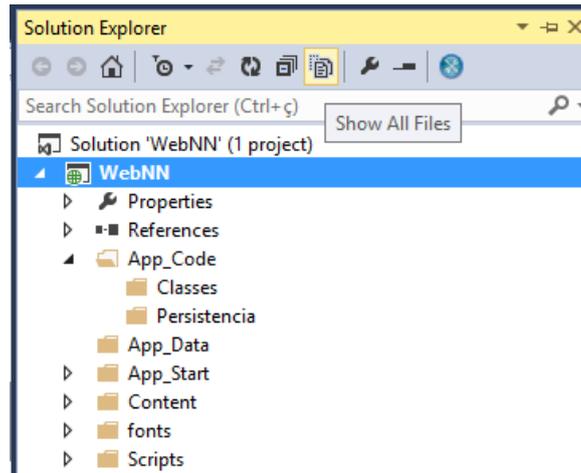
Selecione WebForms
Authentication: No Authentication
Clique em OK.



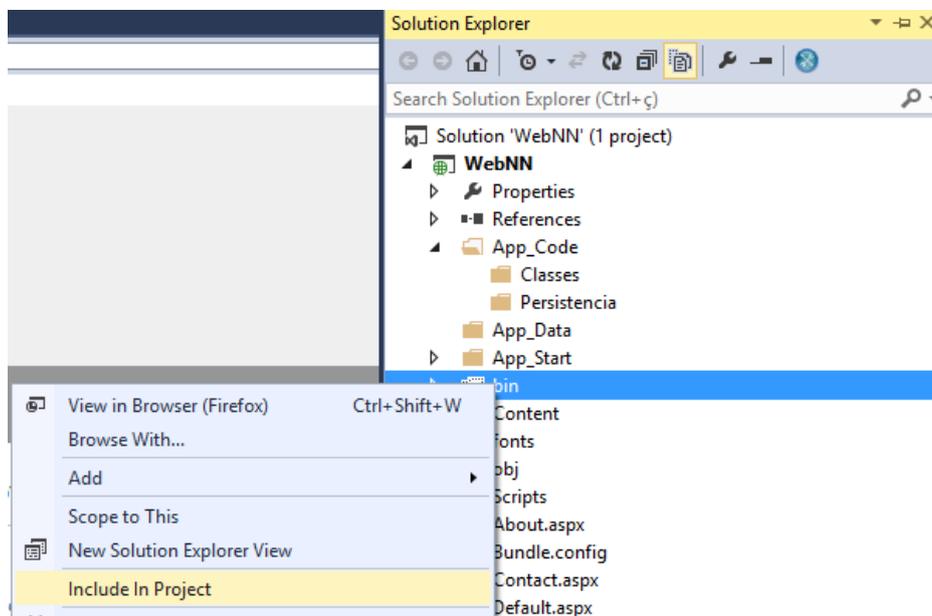
Na SolutionExplorer, adicione a pasta App_Code e dentro dela, as pastas Classes e Persistencia.



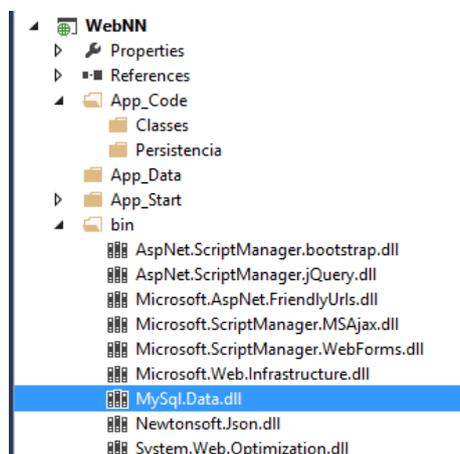
Na SolutionExplorer, clique no botão Show All Files



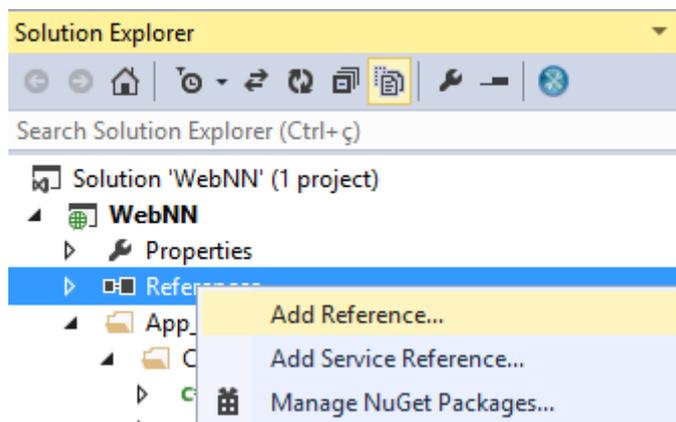
A pasta bin irá aparecer. Clique com o botão direito sobre ela e depois em Include in Project



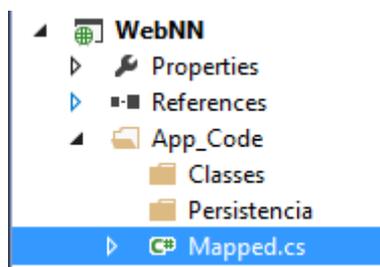
Dentro dessa pasta, adicione o arquivo MySQL.Data.DLL



Adicione a Referência do MySQL no projeto.
Botão direito no item References > Add Reference.
Selecione o arquivo MySQL.Data.DLL



Na pasta App_Code, adicione a classe Mapped.cs



Abra essa classe.
Renomeie o namespace dela para WebNN

namespace WebNN

Agora, iremos criar as classes para o projeto.

Dentro de App_Code > Classes, crie as classes Produto e Fornecedor

Classe Produto.cs

```
using System;
using System.Collections.Generic;
using System.Web;

namespace WebNN.App_Code.Classes
{
    public class Produto
    {
        public int Codigo { get; set; }
    }
}
```

```

        public string Nome { get; set; }
        public string Descricao { get; set; }
    }
}

```

Classe Fornecedor.cs

```

using System;
using System.Collections.Generic;
using System.Web;

namespace WebNN.App_Code.Classes
{
    public class Fornecedor
    {
        public int Codigo { get; set; }
        public string Nome { get; set; }
        public string CNPJ { get; set; }
    }
}

```

Após isso, precisamos definir o relacionamento (associação) entre as classes. Nesse exemplo, faremos uma associação N:N. Assim, um produto pode ter vários fornecedores e um fornecedor pode ter vários produtos.

Para isso, altere a classe Produto.cs

```

public class Produto
{
    public int Codigo { get; set; }
    public string Nome { get; set; }
    public string Descricao { get; set; }

    public List<Fornecedor> Fornecedores { get; set; }
}

```

Altere a classe Fornecedor.cs

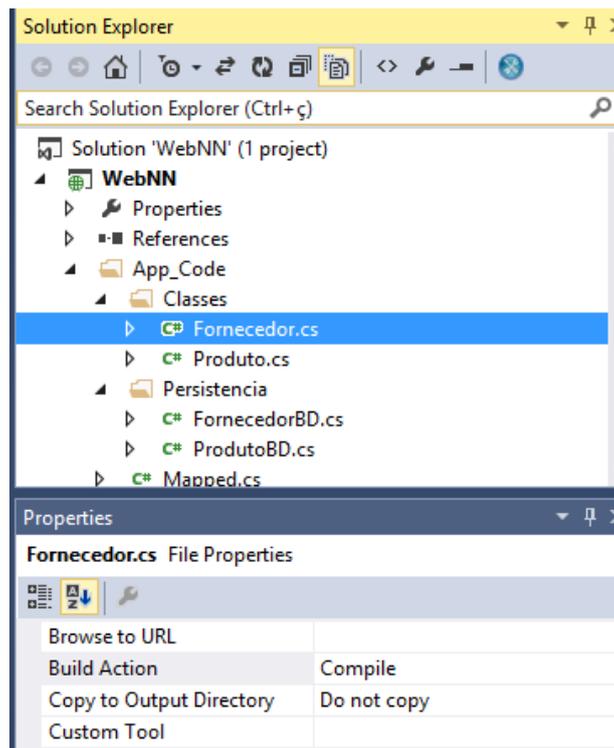
```

public class Fornecedor
{
    public int Codigo { get; set; }
    public string Nome { get; set; }
    public string CNPJ { get; set; }

    public List<Produto> Produtos { get; set; }
}

```

Lembre-se que cada classe criada deve estar com a propriedade BuildAction = Compile.



Agora, precisamos criar o Banco de dados.

Abra o MySQL Workbench e crie um novo schema chamado exemplonn

Numa Associação N:N, a regra é:

- Crie uma tabela para cada classe.
- Crie mais uma tabela para uma classe intermediária (que deve conter as FKs das outras classes).

Assim, crie a tabela tbl_produto

Table Name: Schema: **bdclasse01**
 Collation: Engine:
 Comments:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
pro_codigo	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
pro_nome	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
pro_descricao	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Crie a tabela tbl_fornecedor

Table Name: Schema: **bdclasse01**
 Collation: Engine:
 Comments:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
for_codigo	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
for_nome	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
for_cnpj	VARCHAR(18)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Crie a tabela tbl_produtofornecedor

Table Name: Schema: **exemplonn**
 Collation: Engine:
 Comments:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
pfo_codigo	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
pro_codigo	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
for_codigo	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Adicione alguns fornecedores e produtos diretamente no banco de dados.

for_codigo	for_nome	for_cnpj
1	Samsung	182638123
2	Apple	816238123
3	Sony	12312312
NULL	NULL	NULL

pro_codigo	pro_nome	pro_descricao
1	Celular	Celular de teste
2	TV	TV de teste
3	Relógio	Relógios
NULL	NULL	NULL

Adicione a string de conexão no arquivo Web.config

```
<configuration>
  <appSettings>
    <add key="strConexao" value="Database=exemplonn;Data
Source=localhost;User Id=root;Password=senha; pooling=false"/>
  </appSettings>
  ...

```

Observe a senha do seu banco de dados e altere na string de conexão.

Criando a classe que terá os métodos de acesso ao banco de dados relacionados a classe Produto

- Botão direito na pasta Persistencia
- Adicione Nova Classe
- Nome da classe: ProdutoBD.cs

Adicione o método SelectAll

```
public DataSet SelectAll()
{
    DataSet ds = new DataSet();

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    System.Data.IDataAdapter objDataAdapter;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command("SELECT * FROM tbl_produto ORDER BY pro_nome",
objConexao);
    objDataAdapter = Mapped.Adapter(objCommand);
    objDataAdapter.Fill(ds);

    objConexao.Close();

    objCommand.Dispose();
    objConexao.Dispose();

    return ds;
}
```

Lembre-se de adicionar o namespace para uso do DataSet.

```
using System.Data;
```

Criando a classe que terá os métodos de acesso ao banco de dados relacionados a classe Fornecedor

- Botão direito na pasta Persistencia
- Adicione Nova Classe
- Nome da classe: FornecedorBD.cs

Adicione o método SelectAll

```
public DataSet SelectAll()
{
    DataSet ds = new DataSet();

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    System.Data.IDataAdapter objDataAdapter;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command("SELECT * FROM tbl_fornecedor ORDER BY
for_nome", objConexao);
    objDataAdapter = Mapped.Adapter(objCommand);
    objDataAdapter.Fill(ds);

    objConexao.Close();

    objCommand.Dispose();
    objConexao.Dispose();

    return ds;
}
```

Lembre-se de adicionar o namespace para uso do DataSet.

using System.Data;

Na SolutionExplorer, no projeto WebNN, adicione uma pasta Paginas. Dentro dela, adicione a pasta Produtos.

Dentro da pasta Paginas > Produtos , adicione um WebForm, chamado Vincular.aspx.



Nesta página, adicione um método para carregar os Produtos e Fornecedores.

```
private void Carrega()
{
    //Carrega Produtos
    ProdutoBD produtobd = new ProdutoBD();
    DataSet produtods = produtobd.SelectAll();
    //vincula produtos ao dropdownlist
}
```

```

ddlProduto.DataSource = produtods.Tables[0].DefaultView;
ddlProduto.DataTextField = "pro_nome";
ddlProduto.DataValueField = "pro_codigo";
ddlProduto.DataBind();
ddlProduto.Items.Insert(0, "Selecione");

//Carrega Fornecedores
FornecedorBD fornecedorbd = new FornecedorBD();
DataSet fornecedords = fornecedorbd.SelectAll();
//Vincula fornecedores ao checkboxlist
cblFornecedor.DataSource = fornecedords.Tables[0].DefaultView;
cblFornecedor.DataTextField = "for_nome";
cblFornecedor.DataValueField = "for_codigo";
cblFornecedor.DataBind();

}

```

No Page_Load adicione o método criado anteriormente.

```

protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        Carrega();
    }
}

```

Execute a aplicação. Os produtos e fornecedores já podem ser selecionados.

Adicione um método em ProdutoBD para cadastrar as informações na tabela intermediária.

```

public bool Vincular(int idproduto, int idfornecedor)
{
    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    string sql = "INSERT INTO tbl_produtofornecedor(pro_codigo, for_codigo)
VALUES (?produto, ?fornecedor)";

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command(sql, objConexao);

    objCommand.Parameters.Add(Mapped.Parameter("?produto", idproduto));
    objCommand.Parameters.Add(Mapped.Parameter("?fornecedor", idfornecedor));

    objCommand.ExecuteNonQuery();
    objConexao.Close();
    objCommand.Dispose();
    objConexao.Dispose();

    return true;
}

```

Na página Vincular.aspx

Adicione um método para verificar se algum produto foi selecionado

```
private bool hasProduto()
{
    return (ddlProduto.SelectedItem.Text != "Selecione");
}
```

Adicione um método para verificar se algum fornecedor foi selecionado

```
private bool hasFornecedor()
{
    for (int i = 0; i < cblFornecedor.Items.Count; i++)
    {
        if (cblFornecedor.Items[i].Selected)
        {
            return true;
        }
    }
    return false;
}
```

Agora altere o click do botão Salvar

```
protected void btnSalvar_Click(object sender, EventArgs e)
{
    if (!hasProduto())
    {
        lblMensagem.Text = "Selecione um produto";
        return;
    }
    if (!hasFornecedor())
    {
        lblMensagem.Text = "Selecione um fornecedor";
        return;
    }

    int produto = Convert.ToInt32(ddlProduto.SelectedItem.Value);

    ProdutoBD bd = new ProdutoBD();
    for (int i = 0; i < cblFornecedor.Items.Count; i++)
    {
        if (cblFornecedor.Items[i].Selected)
        {
            int fornecedor = Convert.ToInt32(cblFornecedor.Items[i].Value);
            bd.Vincular(produto, fornecedor);
        }
    }
    lblMensagem.Text = "Produto vinculado com sucesso";

    //limpar ddl e cbl
}
```

Execute a aplicação.

Selecione um produto e depois alguns fornecedores para o produto. Em seguida, clique em Salvar.

Não foi contemplado:

- Limpar os campos depois de salvar
- Verificar se já existem fornecedores para o produto selecionado.

Mostrando os fornecedores de um produto.

Na classe ProdutoBD, adicione o método para Listar os fornecedores de um produto.

```
public DataSet GetFornecedores(int produto)
{
    DataSet ds = new DataSet();

    System.Data.IDbConnection objConexao;
    System.Data.IDbCommand objCommand;
    System.Data.IDataAdapter objDataAdapter;

    objConexao = Mapped.Connection();
    objCommand = Mapped.Command("SELECT * FROM tbl_produtofornecedor INNER JOIN
tbl_produto ON tbl_produto.pro_codigo=tbl_produtofornecedor.pro_codigo INNER JOIN
tbl_fornecedor ON tbl_fornecedor.for_codigo=tbl_produtofornecedor.for_codigo WHERE
tbl_produto.pro_codigo=?produto ORDER BY for_nome;", objConexao);

    objCommand.Parameters.Add(Mapped.Parameter("?produto", produto));

    objDataAdapter = Mapped.Adapter(objCommand);
    objDataAdapter.Fill(ds);

    objConexao.Close();

    objCommand.Dispose();
    objConexao.Dispose();

    return ds;
}
```

Na mesma página (Vincular.aspx), adicione um GridView (grvProdutos) abaixo do lblMensagem.

Selecione o ddlProduto e altere a propriedade AutoPostBack para True.

Em seguida, dê dois cliques no ddlProduto. O evento ddlProduto_SelectedIndexChanged será aberto.

```
protected void ddlProduto_SelectedIndexChanged(object sender, EventArgs e)
{
    if (!hasProduto())
    {
        lblMensagem.Text= "Selecione um produto";
        return;
    }

    int produto = Convert.ToInt32(ddlProduto.SelectedItem.Value);

    ProdutoBD bd = new ProdutoBD();
    DataSet ds = bd.GetFornecedores(produto);
}
```

```
if (ds.Tables[0].Rows.Count > 0)
{
    grvProdutos.DataSource = ds.Tables[0].DefaultView;
    grvProdutos.DataBind();
    grvProdutos.Visible = true;
    lblMensagem.Text = "";
}
else
{
    lblMensagem.Text = "Não existem fornecedores para este produto";
    grvProdutos.Visible = false;
}
}
```

Execute a aplicação.

Selecione um produto. Os fornecedores do produto devem aparecer no gridview.

Depois, formate melhor o gridview.