

Nov
15**InterCon Arena 2014**
O maior congresso de criatividade+tecnologia do Brasil.

INSCREVA-SE JÁ

Faça sua busca

[DESIGN & UX](#) [FRONT-END](#) [DEV](#) [BD](#) [MOBILE](#) [MARKETING](#) [GESTÃO](#) [APIS](#) [SEARCH](#) [INFRA](#) [TECH](#) [E-COMMERCE](#) [ANALYTICS](#)**Você está aqui:** [Início](#) » [Framework](#) » [.NET](#) » Guia prático para o desenvolvimento de Aplicações C# em Camadas

Tweetar 0

8+1 2

Like Share 0

Oferecimento:
EV3O

.NET

Guia prático para o desenvolvimento de Aplicações C# em Camadas

07 de novembro de 2008 por [Carlos Olavo de Azevedo Camacho Junior](#)

III – Camada de Acesso a Dados

Essa camada também é normalmente chamada de DAL (Data Access Layer). Nessa camada vamos implementar os métodos de inserção, atualização, exclusão e listagem referentes a todas as tabelas existentes no nosso projeto.

Essa é uma tarefa simples, já que para criar cada classe usaremos os nomes dos campos da respectiva tabela existente no banco de dados.

Você se lembra do nosso desenho que representa as três camadas? Aqui está ele:



CURSOS RELACIONADOS



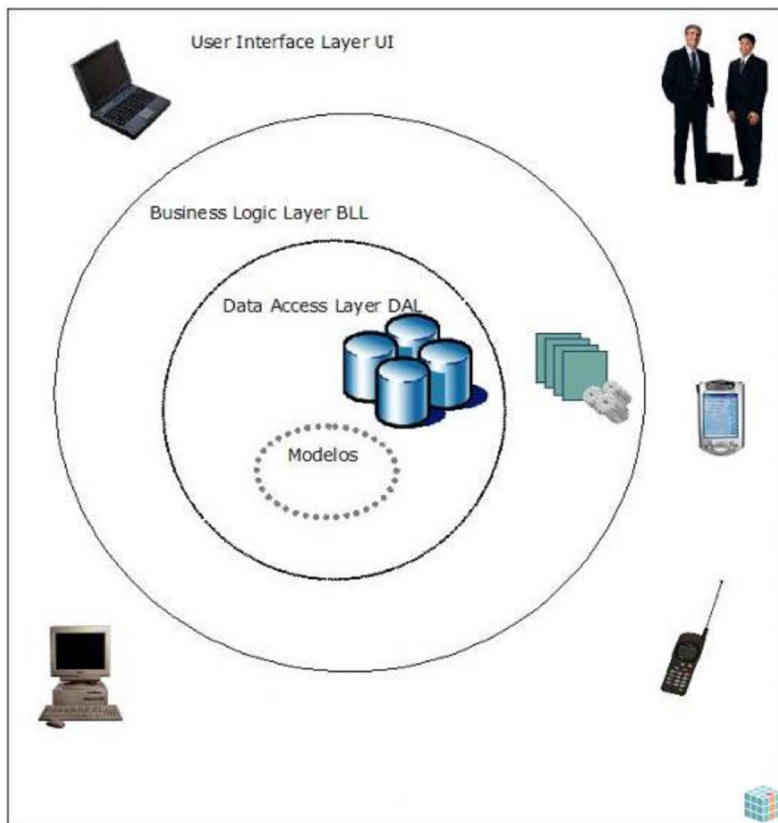
Integração .NET com ASP Clássico utilizando componente COM+

ÚLTIMAS NOTÍCIAS

NEWSLETTER

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR



Nós começamos o desenvolvimento da aplicação da camada mais interna até a mais externa. Dessa forma, iniciaremos a implementação com a camada de acesso a dados (DAL).

Dentro da camada DAL nós temos o projeto **Modelos**. Vamos criar esse projeto.

No projeto chamado **Modelos** iremos implementar as classes:

- ClientInformation.cs
- ProdutoInformation.cs e
- VendaInformation.cs

É necessário criar uma classe para cada tabela do nosso projeto.

A partir do menu Iniciar > Programas > Microsoft Visual Studio 2005, abra o **Microsoft Visual Studio 2005**.

Clique no Menu File > New > **Project...**

24/10 AS 04H00

LG ingressa no mercado de processadores para smartphones

24/10 AS 03H40

Smartphones modulares do Google devem ganhar loja

24/10 AS 02H30

Apple abre oficialmente sistema TestFlight para testes beta de apps



TODAS AS NOTÍCIAS »

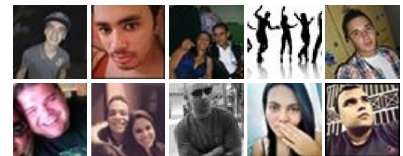
Find us on Facebook



iMasters

Like

102,437 people like iMasters.



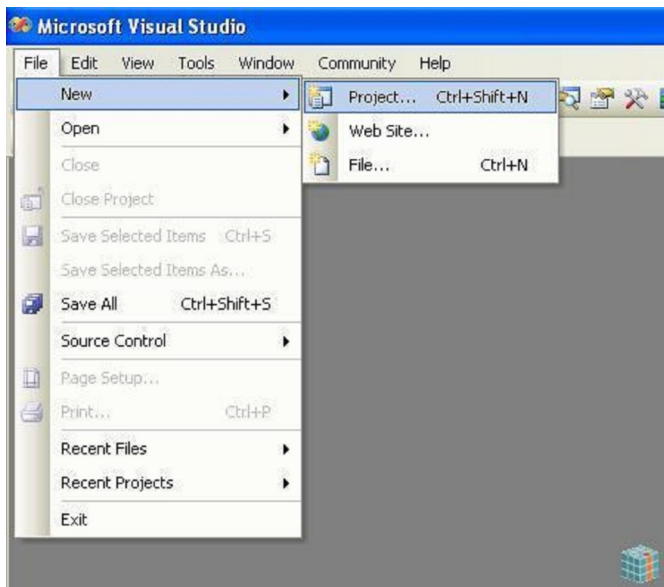
Facebook social plugin

NEWSLETTER

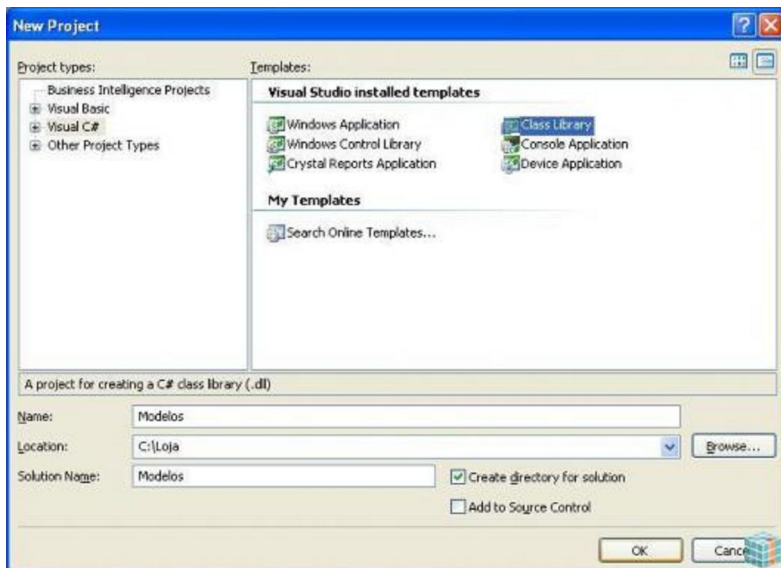
X

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR



Acompanhe as seleções da janela *New Project*:



No tipo de projeto selecione **Visual C#**;

No tipo de template selecione **Class Library**;

No Nome do projeto digite **Modelos**;

Na localização do projeto digite **C:\Loja**;

Deixe a opção "*Create directory for solution*" selecionada e clique em **Ok** para criar o projeto.

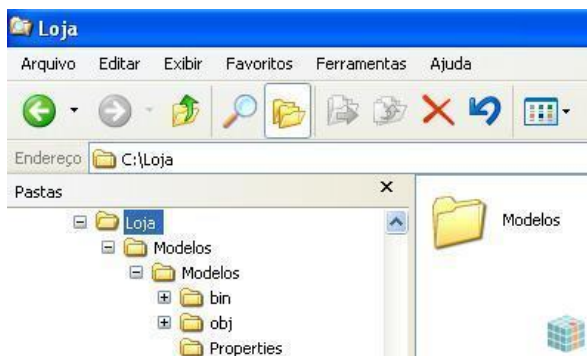
Abrindo o Windows Explorer, podemos ver que a pasta **Loja** foi criada no drive C:

Dentro da pasta Loja foi criada a pasta do projeto **Modelos**.

NEWSLETTER X

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR



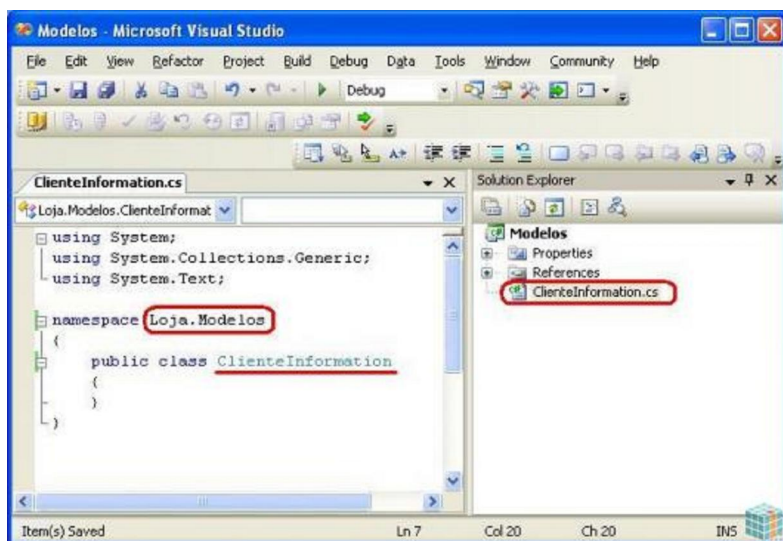
Ok, agora vamos voltar para o MS Visual Studio.

Do lado direito vemos a área do Solution Explorer.

- Clique com o botão direito sobre o arquivo Class1.cs e escolha **Rename** para renomear a classe para **ClienteInformation.cs**;

Ao renomear o arquivo da classe, perceba que o nome da classe muda automaticamente na janela de código (**public class ClienteInformation**).

- Na janela de código à esquerda, inclua o nome do nosso projeto (Loja) no namespace, de maneira que o namespace fique **Loja.Modelos** como na figura abaixo:



Vamos codificar a classe *ClienteInformation*. Para isso, copie e cole o código abaixo entre as chaves da classe *ClienteInformation*:

```
private int _codigo;
public int Codigo
{
    get { return _codigo; }
    set { _codigo = value; }
}

private string _nome;
public string Nome
{
    get { return _nome; }
    set { _nome = value; }
}

private string _email;
public string Email
{
    get { return _email; }
```

NEWSLETTER X

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR

```
        set { _email = value; }
    }

    private string _telefone;
    public string Telefone
    {
        get { return _telefone; }
        set { _telefone = value; }
    }
}
```

Na implementação da classe *ClienteInformation* vemos que estamos definindo campos e propriedades para cada campo da tabela de Clientes que criamos no banco de dados.

Agora a codificação da classe *ClienteInformation* está completa como na listagem a seguir:



```
using System;
using System.Collections.Generic;
using System.Text;
```

```
namespace Loja.Modelos
{
    public class ClienteInformation
    {
        private int _codigo;
        public int Codigo
        {
            get { return _codigo; }
            set { _codigo = value; }
        }

        private string _nome;
        public string Nome
        {
            get { return _nome; }
            set { _nome = value; }
        }

        private string _email;
        public string Email
        {
            get { return _email; }
            set { _email = value; }
        }

        private string _telefone;
        public string Telefone
        {
            get { return _telefone; }
            set { _telefone = value; }
        }
    }
}
```

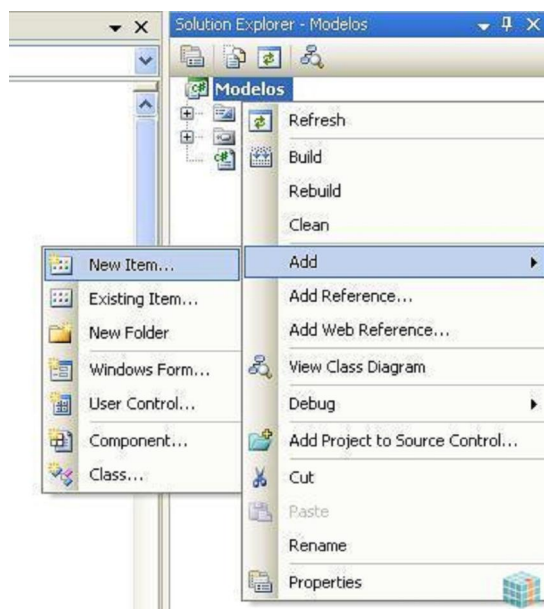
Vamos implementar a Classe *ProdutoInformation.cs*.

Clique com o botão direito sobre o projeto **Modelos** e então escolha **Add**. Em seguida escolha **New Item...** como na figura abaixo:

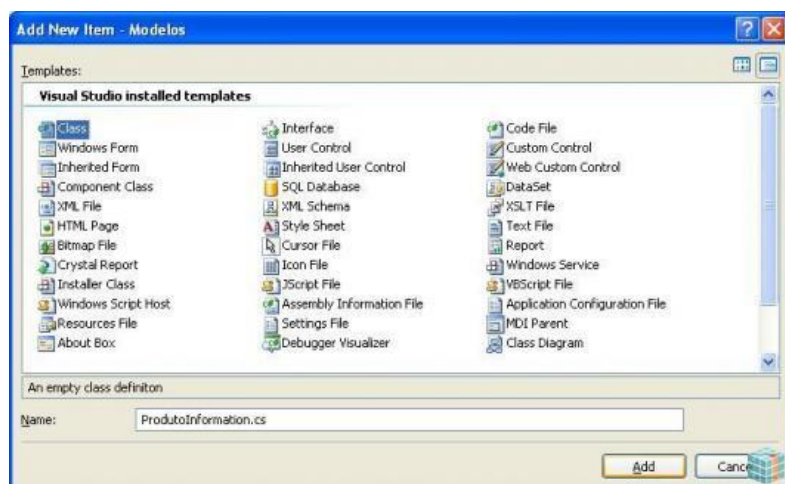
NEWSLETTER

Fique por dentro de todas as novidades, eventos, cursos e muito mais

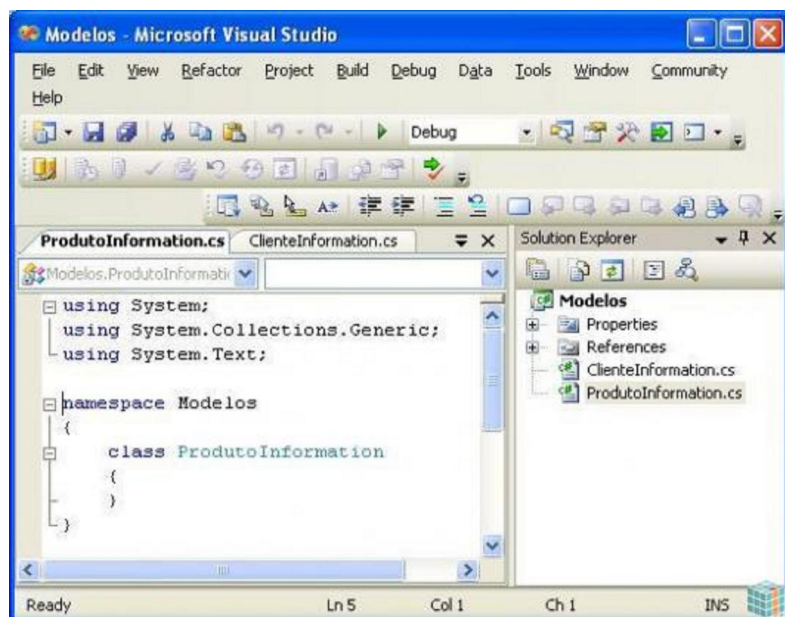
ENVIAR



Na janela Add New Item, escolha o template **Class** e digite **ProdutoInformation.cs** no campo nome. Clique em **Add** para adicionar a nova classe ao projeto Modelos.



A nova classe ProdutoInformation.cs agora já esta criada:



NEWSLETTER X

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR

Altere o namespace de modo que ele fique assim:

```
namespace Loja.Modelos
```

Ok. Agora copie e cole o código abaixo dentro da classe ProdutoInformation:

```
private int _codigo;
public intCodigo
{
    get { return _codigo; }
    set { _codigo = value; }
}

private string _nome;
public string Nome
{
    get { return _nome; }
    set { _nome = value; }
}

private decimal _preco;
public decimal Preco
{
    get { return _preco; }
    set { _preco = value; }
}

private int _estoque;
public int Estoque
{
    get { return _estoque; }
    set { _estoque = value; }
}
```



O código completo da nossa classe ProdutoInformation.cs ficará assim:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Loja.Modelos
{
    public class ProdutoInformation
    {
        private int _codigo;
        public intCodigo
        {
            get { return _codigo; }
            set { _codigo = value; }
        }

        private string _nome;
        public string Nome
        {
            get { return _nome; }
            set { _nome = value; }
        }

        private decimal _preco;
        public decimal Preco
        {
            get { return _preco; }
            set { _preco = value; }
        }

        private int _estoque;
        public int Estoque
        {
            get { return _estoque; }

```

NEWSLETTER

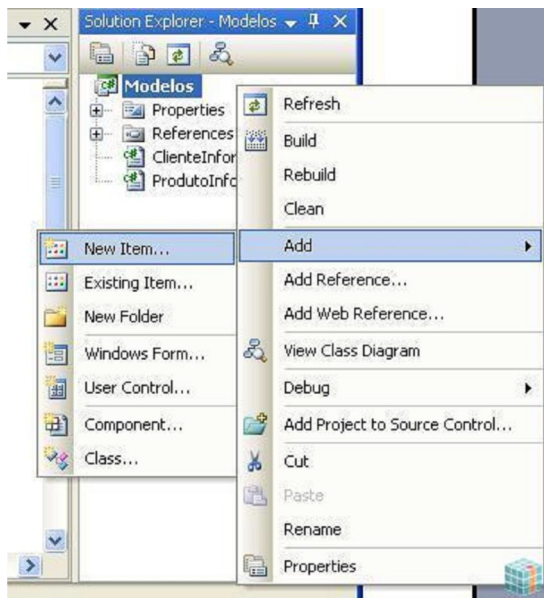
Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR


```
        set { _estoque = value; }  
    }  
}
```

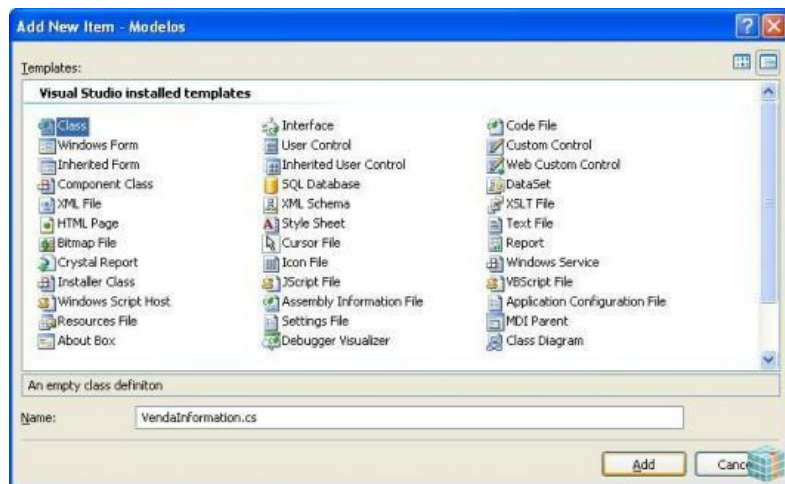
Muito bem! Já implementamos as classes do projeto Modelo para a tabela de Clientes e para a tabela de Produtos. Agora vamos fazer o mesmo para a tabela de Vendas:

Clique com o botão direito sobre o projeto **Modelos** e então escolha **Add**. Em seguida escolha **New Item...** como na figura abaixo:



[b]/[b]

Na janela Add New Item, escolha o template Class e digite **VendaInformation.cs** no campo nome. Clique em **Add** para adicionar a nova classe ao projeto Modelos.

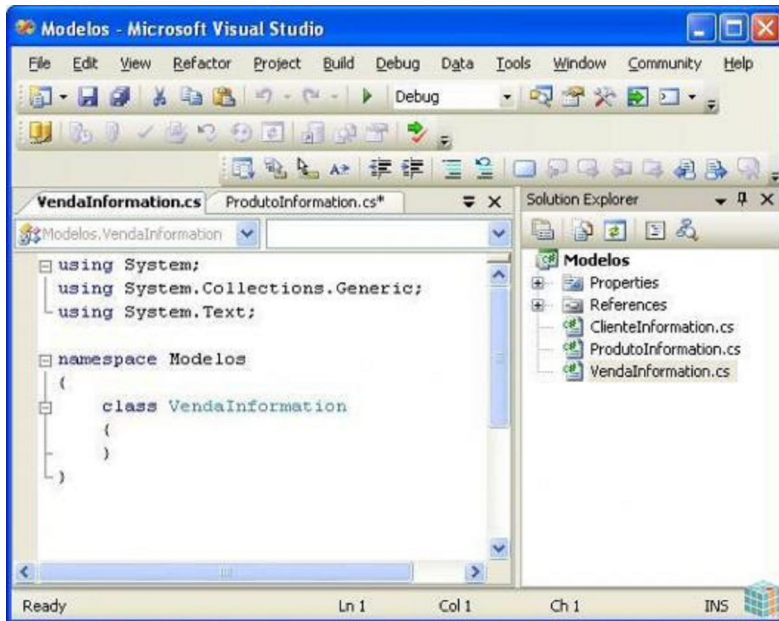


A nova classe VendaInformation.cs agora já esta criada:

NEWSLETTER X

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR



Altere o namespace de modo que ele fique assim:

```
namespace Loja.Modelos
```

Ok. Agora copie e cole o código abaixo dentro da classe *VendaInformation*:

```
private int _codigo;
public int Codigo
{
    get { return _codigo; }
    set { _codigo = value; }
}

private DateTime _data;
public DateTime Data
{
    get { return _data; }
    set { _data = value; }
}

private int _quantidade;
public int Quantidade
{
    get { return _quantidade; }
    set { _quantidade = value; }
}

private bool _faturado;
public bool Faturado
{
    get { return _faturado; }
    set { _faturado = value; }
}

private int _codigoCliente;
public int CodigoCliente
{
    get { return _codigoCliente; }
    set { _codigoCliente = value; }
}

private int _codigoProduto;
public int CodigoProduto
{

```

NEWSLETTER

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR

```
    get { return _codigoProduto; }  
    set { _codigoProduto = value; }  
}
```

```
private string _nomeCliente;  
public string NomeCliente  
{  
    get { return _nomeCliente; }  
    set { _nomeCliente = value; }  
}
```

O código completo da nossa classe VendaInformation.cs ficará assim:



```
using System;  
using System.Collections.Generic;  
using System.Text;
```

```
namespace Loja.Modelos  
{  
    public class VendaInformation  
    {  
        private int _codigo;  
        public int Codigo  
        {  
            get { return _codigo; }  
            set { _codigo = value; }  
        }  
  
        private DateTime _data;  
        public DateTime Data  
        {  
            get { return _data; }  
            set { _data = value; }  
        }  
  
        private int _quantidade;  
        public int Quantidade  
        {  
            get { return _quantidade; }  
            set { _quantidade = value; }  
        }  
  
        private bool _faturado;  
        public bool Faturado  
        {  
            get { return _faturado; }  
            set { _faturado = value; }  
        }  
  
        private int _codigoCliente;  
        public int CodigoCliente  
        {  
            get { return _codigoCliente; }  
            set { _codigoCliente = value; }  
        }  
  
        private int _codigoProduto;  
        public int CodigoProduto  
        {  
            get { return _codigoProduto; }  
            set { _codigoProduto = value; }  
        }  
  
        private string _nomeCliente;  
        public string NomeCliente  
        {
```

NEWSLETTER

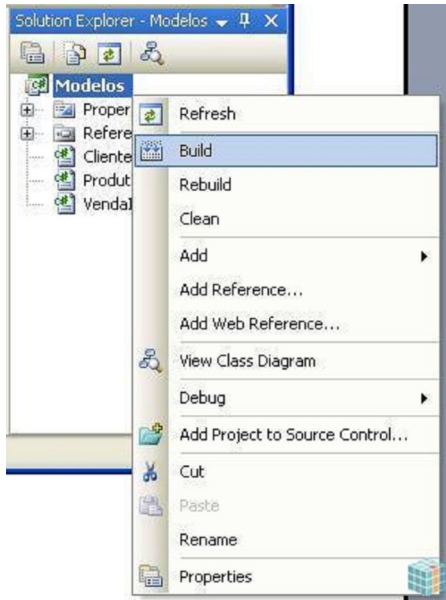
Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR

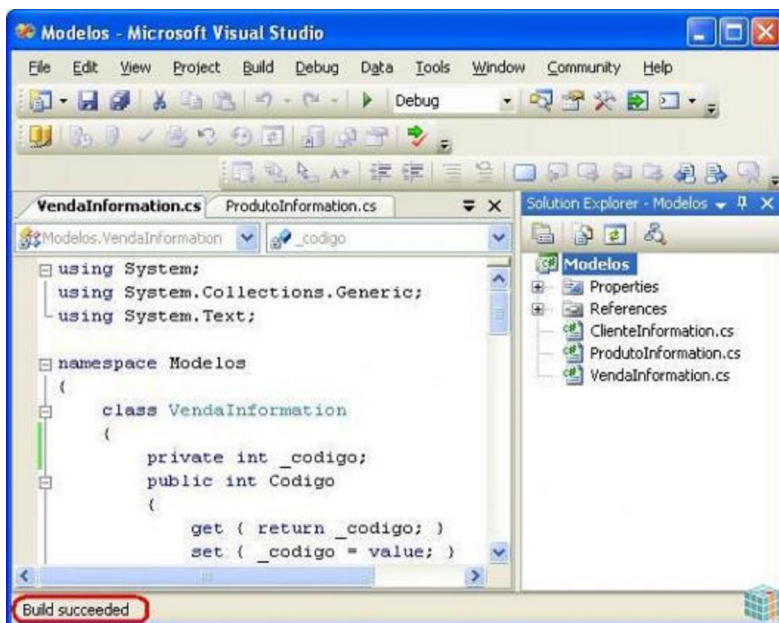
```
    get { return _nomeCliente; }  
    set { _nomeCliente = value; }  
}  
}
```

Agora que implementamos no projeto Modelos as classes referentes a todas as tabelas contempladas no nosso projeto, vamos compilar o projeto através da opção Build.

Para compilar o projeto vamos fazer o seguinte. No Solution Explorer, clique com o botão direito sobre o projeto **Modelos** e escolha a opção **Build**:



Se a compilação do nosso projeto for realizada com sucesso, aparecerá a mensagem *"Build succeeded"* na barra de status do Microsoft Visual Studio:

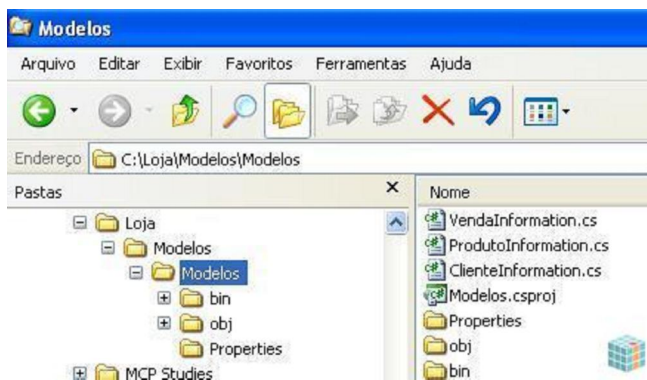


No Windows Explorer, observe que na pasta do projeto Modelos foi criado um arquivo para cada classe que implementamos.

NEWSLETTER X

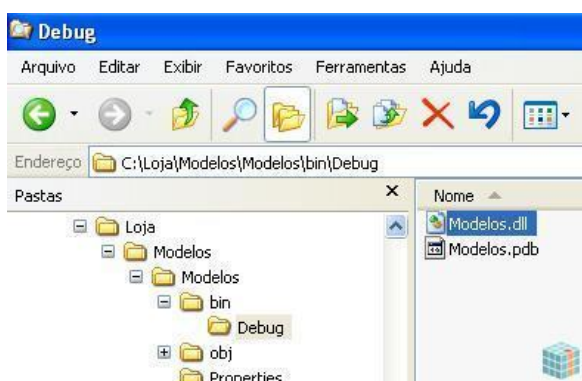
Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR



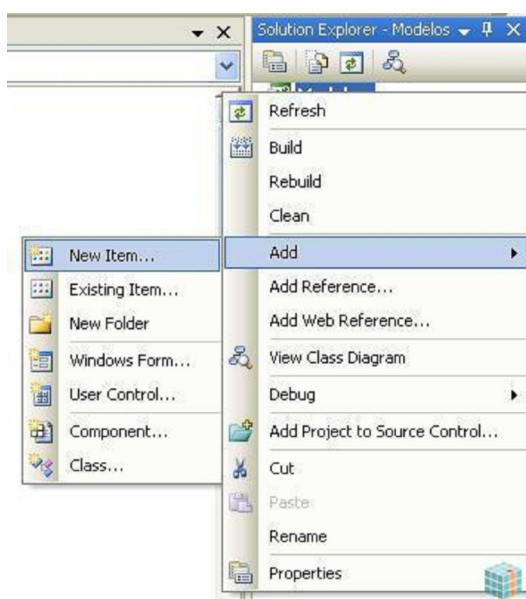
Quando compilamos o projeto com a opção Build, o MS Visual Studio criou o arquivo Modelos.dll.

O arquivo Modelos.dll contém toda a informação que implementamos nas classes desse projeto e é ele que será usado no próximo projeto a ser implementado, que é a Camada de Acesso a Dados ou DAL (Data Access Layer).



Após terminar o projeto Modelos, muitos programadores costumam criar um *Diagrama de Classes* para ter uma visão melhor da sua arte. Vamos criar o Diagrama de Classes?

De volta ao Visual Studio, clique com o botão direito sobre o projeto **Modelos** e escolha **Add > New Item...**

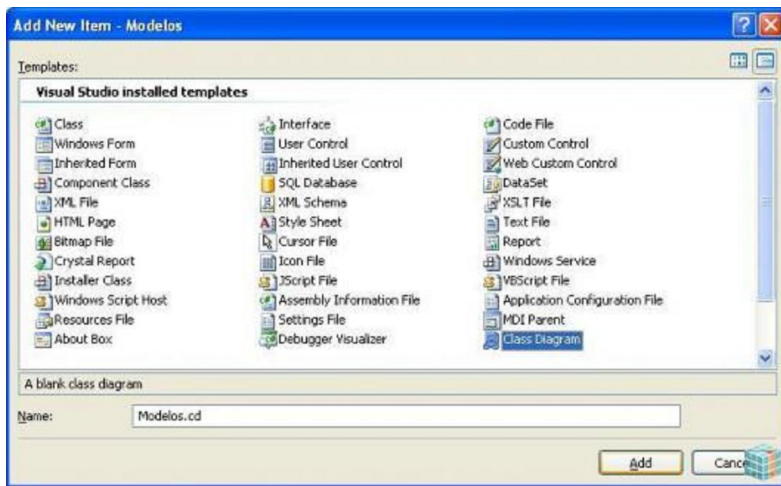


Na janela Add New Item, escolha o template **Class Diagram**. Digite **Modelos.cd** no nome e clique em **Add** para criar o nosso diagrama de classes como a seguir.

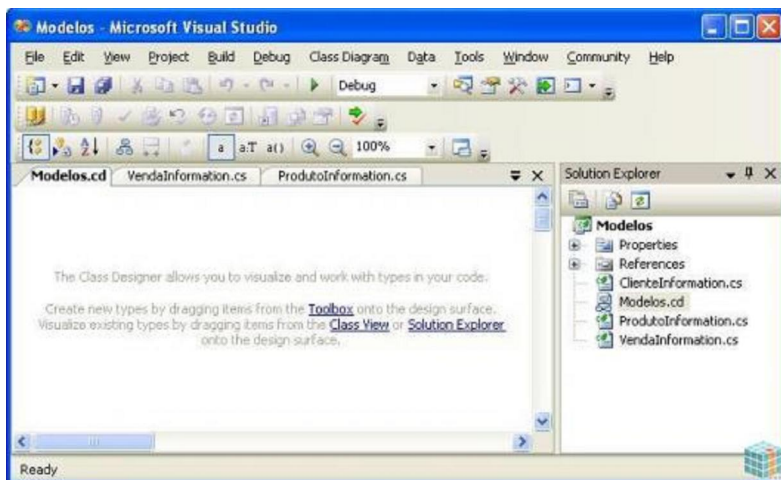
NEWSLETTER X

Fique por dentro de todas as novidades, eventos, cursos e muito mais

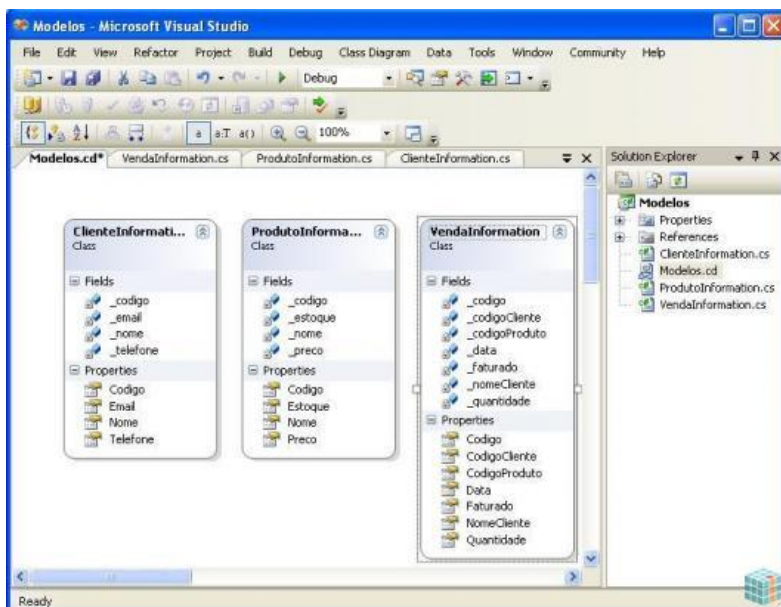
ENVIAR



O ambiente chamado *Class Designer* é exibido.



Para a criação do diagrama vamos arrastar as classes *ClienteInformation*, *ProdutoInformation* e *VendaInformation* do Solution Explorer para o Class Designer (Arquivo Modelos.cd) como a seguir:



Com o Diagrama de Classes fica fácil a distinção entre os campos e as propriedades que implementamos no nosso projeto.

NEWSLETTER X

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR

O asterisco ao lado do nome do diagrama **Modelos.cd*** indica que ele ainda não foi salvo. Digite **<Ctrl> + S** para salvar o arquivo.

Se olharmos para o nosso desenho das camadas, vamos verificar que o próximo passo é implementar as classes da camada DAL. Faremos isso no próximo artigo.

Até breve!



Artigos relacionados

Login com Fluent NHibernate, NHibernate 4.0 e PostgreSQL – Parte 03
.NET, Visual Basic

Login com Fluent NHibernate, NHibernate 4.0 e PostgreSQL – Parte 02
.NET, Visual Basic

Como usar ADO.NET no ASP.NET MVC
.NET, ASP

8 comentários

Comente!

2 comentários



Carlos Olavo de Azevedo Camacho Junior

é mestrando em Tecnologias da Inteligência e Design Digital pela PUCSP. Pós-graduado em Análise e Projeto de Sistemas pela UNIP onde também se formou em Bacharel em Ciência da Computação. Possui Licenciatura Plena em Matemática pelas Faculdades Oswaldo Cruz. MCP .Net, MCP SQL Server.

[Página do autor Email](#)

Leia os últimos artigos publicados por Carlos Olavo de Azevedo Camacho Junior

[Guia prático para o desenvolvimento de Aplicações C# em Camadas](#)

[Criando a infra-estrutura de Banco de dados](#)

[Guia prático para o desenvolvimento de Aplicações C# em Camadas](#)

23 COMENTÁRIOS

COMENTE TAMBÉM



Edimar Ramos

Parabens cara.. muito bom os artigos... muito bem detalhado... que venham os proximos...

Há 6 anos [Responder](#)



Rafael Rosa

Salve.

Boa idéia, exemplos bem detalhados, ótima iniciativa, só tenho duas “críticas”. Primeiro, acho que faltou uma explicação geral de como será implementado o sistema e quais os benefícios dessa abordagem no primeiro artigo. Dado que é um artigo “básico”, essas informações são importantes. Pelo menos indicações de artigos onde essas informações podem ser encontradas.

O segundo ponto é: você NÃO vai fazer todo o acesso usando stored procedures vai??? Sei que esse é a receita padrão da MS, inclusive cheguei a ler o manual de Patterns & Practices sobre isso, mas é uma abordagem antiga e que tem um monte de problemas, especialmente na parte que considero mais importante, a

NEWSLETTER

X

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR

manutenção. Espero que você o faça de outra maneira.

Abraço

Há 6 anos [Responder](#)

**Carlos Camacho Junior**

Amigo Edimar,

Muito obrigado pelo elogio!

Amigo Rafael,

Muito obrigado pelo elogio e também pelas críticas!

Aproveitando o seu comentário, vamos as boas notícias:

- Você tem razão quanto ao fato de não haver muitas explicações no decorrer do texto. Mas a idéia foi justamente partir para a parte prática de forma que o leitor veja como a construção de um projeto ocorre.

- Gostei bastante da sua questão sobre o uso de stored procedures.

Como você mesmo disse, isso é orientação de Patterns and Best Practices Microsoft e dessa forma eu não poderia deixar de abordar essa construção.

Por outro lado, entendo que existem times de desenvolvedores que não as utilizam e alguns se justificam explanando sobre facilidade de manutenção.

Pensando nisso acabei, neste projeto, implementando as duas formas. Na primeira todas as operações de gerenciamento do banco de dados via aplicativo são realizadas através de stored procedures. Na segunda forma, o gerenciamento é realizado sem utilizar nenhuma stored procedure.

Esse material ficou extenso e uma editora interessou-se em publicá-lo.

Por isso tenho prazer em informar-lhes que acabei de publicar o livro

Desenvolvimento em Camadas com C# .NET

Maiores informações podem ser obtidas em:

http://www.karate-do.com.br/camachojr/index.php?option=com_content&task=view&id=29&Itemid=26

Aproveito para agradecer o carinho e incentivo de todos os leitores do iMasters.

Boa leitura e sucesso!

Carlos Camacho.

Há 6 anos [Responder](#)

**Kledir Robenson Barkemeyer**

Gostei, estou aguardando a continuação :)

Há 6 anos [Responder](#)

**Junio Silva**

O artigo não continua?

Há 5 anos [Responder](#)

**Cleiton Wagner**

Bom dia Camacho...

Estou muito interessado em adquirir este livro.

Mas antes gostaria de saber qual a versão do .NET Framework e do VS utilizados no livro.

Já tive surpresas desagradáveis ao adquirir livros sobre .NET e ao abri-lo descobrir que o mesmo utilizava a versão 1.1 do framework (as vezes até versões beta).

Abraço

Há 5 anos [Responder](#)

**Carlos Olavo de Azevedo Camacho Junior**

Pessoal,

para quem já havia começado a implementar os artigos, informo que podem dar continuidade nos estudos através do link a seguir:

<http://www.linhadecodigo.com.br/Livro.aspx?id=3502>

Cleiton, o projeto foi desenvolvido com framework 2.0 (VS 2005), mas é compatível com o MS Visual Studio 2008.

Abraços.

Há 5 anos [Responder](#)

**Carlos Olavo de Azevedo Camacho Junior**

Anteriormente informei apenas o link do livro. :)

O link para a série completa de artigos é esse aqui:

<http://www.linhadecodigo.com.br/Colaborador.aspx?id=429>

Abraços e bons estudos!

Há 5 anos [Responder](#)

**Carlos Olavo de Azevedo Camacho Junior**

Para o amigo Rafael e demais interessados:

Na série de artigos implementei o CRUD (Inserção, Consulta, Atualização, Exclusão) sem a utilização de stored procedures. O uso de stored procedures foi abordado apenas no livro "Desenvolvimento em Camadas com C# .NET".

Abraços.

Há 5 anos [Responder](#)

Cleiton Wagner

E no livro foi implementado CRUD ou SP?

**NEWSLETTER**

X

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR



Abraço

Há 5 anos [Responder](#)**Carlos Olavo de Azevedo Camacho Junior**

Olá Cleiton,

No livro foram abordados os dois tipos de implementação.

Também foi abordada a criação de interfaces Web e há ainda um capítulo sobre segurança de dados (Criptografia).

Abraços.

Há 5 anos [Responder](#)

Bixo, fui na livraria Saraiva aqui em Brasilia-DF e comprei seu livro!

Tou louco pra chegar em casa e começar a estudar!

Há 5 anos [Responder](#)

Camacho, reparei que o livro é a copia fiel do seu tutorial aqui no IMasters!?

Há 5 anos [Responder](#)**Carlos Olavo de Azevedo Camacho Junior**

Ola Pacheco,

Na verdade o projeto do livro é uma ampliação do mini projeto exposto nos artigos.

No livro ha uso de stored procedures, implementação da camada de apresentacao Web, criptografia de dados para combater hackers, demonstração da compatibilidade entre linguagens na plataforma .Net e boas dicas sobre programação em geral.

Abraços.

Há 5 anos [Responder](#)

Ah..tah!

Estou satisfeito com seu livro Carlos.

Carlos, terminei de implementar a interface de Clientes.

Quando eu vou excluir um registro um erro aparece: Não foi Possível Excluir o Cliente 11. Sendo que 11 é o numero de telefone que eu coloquei quando Inclui esse registro e no código-fonte esta setado para aparecer o código do registro. Sendo assim....o registro não Exclui.

Ajude-me

Há 5 anos [Responder](#)

Olá Carlos Camacho

Comprei o livro para iniciar em C# e o assunto atraiu muito; Ao ler o livro posso afirmar ser o melhor livro que estou lendo, e olha que gastei uns 300 reais em outros livros, e são bons, mas se comparado ao "Desenvolvimento em camadas com C# .Net", precisariam serem re-escritos;

Camacho, vc está de parabéns e já aguardo novos livros com a mesma didática ou semelhante; Sou desenvolvedor há 15 anos, nos últimos anos trabalhei com C++, usando o C++ Builder 6, e agora quando resolvi migrar para .Net e escolhi C# e este livro... Não me arrependo, pois a didatica do livro é fantástica. Valeu Camacho.

Há 4 anos [Responder](#)O artigo é realmente muito bom.
onde posso encontrá-lo na íntegra?Há 4 anos [Responder](#)**Adilson**

Este artigo esta no livro "Desenvolvimento em camadas com C# .Net".

Muito bom!

Há 1 ano [Responder](#)

NEWSLETTER X

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR



Silmar

Fiquei feliz em ver um livro assim, e quando vi ele corri e comprei.
Sou assinante da devimedia da antiga linha do código e tenho gastado uma grana com livros para aprender mesmo o C#.
Porem não encontro um ensinamento que tenha a situação de procurar o ou seja pesquisar por um campo (código, telefone, nome) e ao termino da pesquisa trazer os dados e popular os dados necessários, mas só vejo com combobox e ou listbox.
Você não tem um exemplo para auxiliar-me ???
ou onde posso encontrar e ou ate mesmo comprar o livro se for preciso.
Comprei um livro a semana passada pela internet e achei que joguei dinheiro fora, assim como outros três. e isso me deixa muito chateado, porque tem muitos deferente de você que enche de detalhes tão técnico que não dão exemplos práticos e só teóricos e complica e acaba confundindo mais ainda
no entanto desde ja obrigado aee pela força que tem nos dados, em compartilhar seus conhecimento

Há 2 anos [Responder](#)



José Camargo

Camacho, seu livro é fantástico, muito detalhado, passo a passo, didática muito clara e organizada, continue sempre publicando livros com esses detalhes que faz a diferença, existem vários livros de camadas, mas o seu realmente superou as expectativas. Tomo como base seu livro para treinamento de meus estagiários, acredito que fixa muito rápido o conhecimento na cabeça desses futuros profissionais de desenvolvimento de sistemas .net.
Parabéns...

Há 2 anos [Responder](#)



ED

O livro está esgotado na livraria saraiva, gostaria de comprar.

Há 2 anos [Responder](#)



Adilson

Eu tenho este livro, gostei de sua didática, voce tem outros livros em c#, que quebre o paradigma que há na dificuldade de Programadores Pascal conseguir entender POO com mais facilidade?

Há 1 ano [Responder](#)



Alfredo Costa

Gosto imenso da explicação e aprendi coisas interessantes. Muito obrigado!

Há 11 meses [Responder](#)

QUAL A SUA OPINIÃO?

Escreva seu comentário aqui...

Conheça o Grupo Impacta
O maior e mais conectado centro de ensino em TI, Gestão e Design da América Latina

Grupo Educacional
IMPACTA
TECNOLOGIA
Ligue 11 3254 - 2200

Twitter

Siga o perfil do iMasters

LinkedIn

Cadastre-se no grupo iMasters

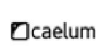
git

Cód

NEWSLETTER

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR



Copyright © 2014 Todos os direitos reservados

SOCIAL MEDIA

des,

eventos, cursos, conteúdos exclusivos e

[Sobre o iMasters](#)
[Política de Privacidade](#)
[Fale conosco](#)
[iMasters Expert \(english blog\)](#)

faleconosco@imasters.com.br

muito mais.

ENVIAR »



X

NEWSLETTER

Fique por dentro de todas as novidades, eventos, cursos e muito mais

ENVIAR