



ASP .NET MVC - Criando uma aplicação básica - CRUD com dropdownlist

(Iniciante) -I



Neste artigo eu vou mostrar como criar uma aplicação [ASP .NET MVC](#) básica que realiza o **CRUD** em uma base de dados SQL Server usando o VS 2013 Express for Web e o Entity Framework 6.

Este artigo é uma revisão dos conceitos básicos envolvidos na criação de uma aplicação [ASP .NET MVC 5](#) usando o [Entity Framework 6.0](#) e o [SQL Server 2008 R2](#).

É um artigo para iniciantes é mostra como usar as tecnologias mais atuais da Microsoft para acessar dados e criar uma aplicação web dinâmica : [o Entity Framework e o ASP .NET MVC 5](#).

A linguagem usada pode ser tanto o Visual C# como o Visual Basic .NET. Neste artigo eu vou usar o VB .NET.

Você verá que o processo de criação de uma aplicação web usando MVC é relativamente simples usando essas ferramentas.

Vamos criar uma aplicação simples que gerencia os dados de Alunos, departamentos e assuntos e com ajuda do **Scaffolding** da ASP .NET MVC e do Entity Framework veremos como é fácil gerar as páginas, ou views, da aplicação e o código dos controladores de forma automática.

Como o artigo é dirigido a iniciantes eu vou descrever o processo passo a passo.

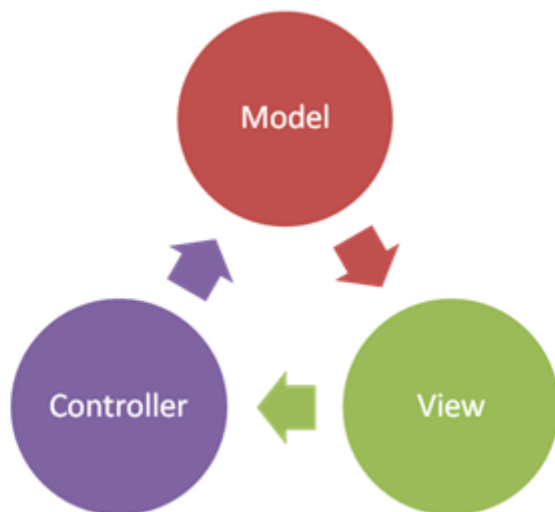
Então, acompanhe-me...

Recursos usados:

- [Microsoft Visual Studio 2013 Express for web](#)
- [SQL Server 2008 Express](#)

Conceitos bem básicos

O que é afinal MVC ?



A arquitetura/padrão MVC - (**Modelo Visualização Controle**) fornece uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados de uma aplicação. A arquitetura MVC não é nova e foi originalmente desenvolvida para mapear as tarefas tradicionais de entrada, processamento e saída para o modelo de interação com o usuário. Usando o padrão MVC fica fácil mapear esses conceitos no domínio de aplicações Web multicamadas.

Na arquitetura MVC o **modelo(Model)** representa os dados da aplicação e as regras do negócio que governam o acesso e a modificação dos dados. O modelo mantém o estado persistente do negócio e fornece ao controlador a capacidade de acessar as funcionalidades da aplicação encapsuladas pelo próprio modelo.

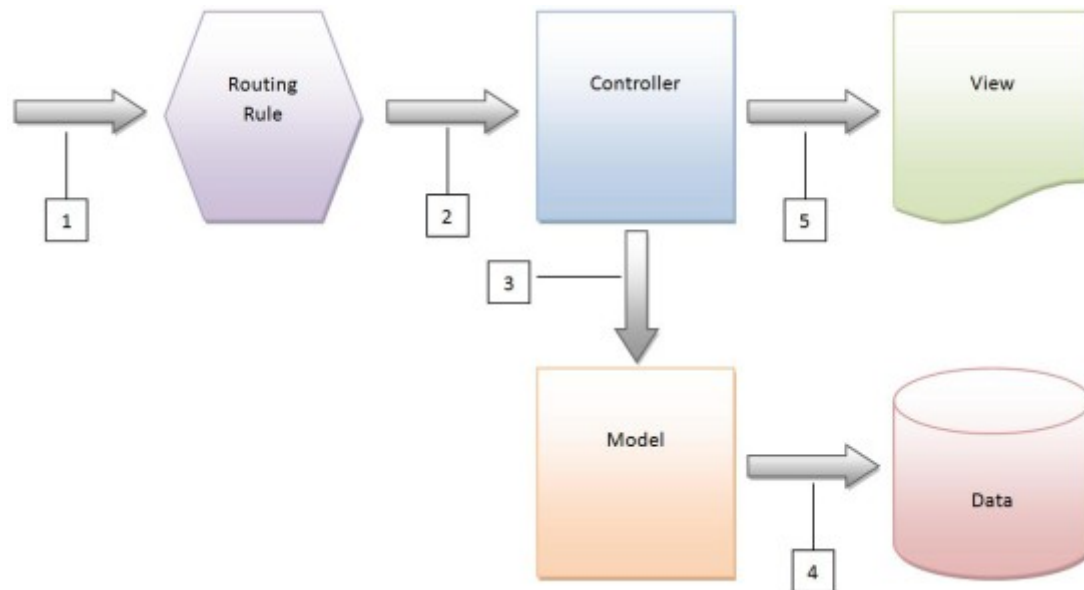
Um componente de **visualização(View)** *renderiza* o conteúdo de uma parte particular do modelo e encaminha para o controlador as ações do usuário; acessa também os dados do modelo via controlador e define como esses dados devem ser apresentados.

Um **controlador(Controller)** define o comportamento da aplicação, é ele que interpreta as ações do usuário e as mapeia para chamadas do modelo. Em um cliente de aplicações Web essas ações do usuário poderiam ser cliques de botões ou seleções de menus. As ações realizadas pelo modelo incluem ativar processos de negócio ou alterar o estado do modelo.

Resumindo:

1. **Models** - Classes que representam os dados da aplicação e que usa a lógica de validação para impor as regras de negócio para os dados;
2. **Views** - Arquivos Templates que sua aplicação usa para gerar resposta HTML dinâmica;
3. **Controller** - Classes que tratam as requisições do navegador, retornam os dados e então especificam templates views que retornam a resposta ao navegador para apresentação do resultado;

Como o MVC funciona ?

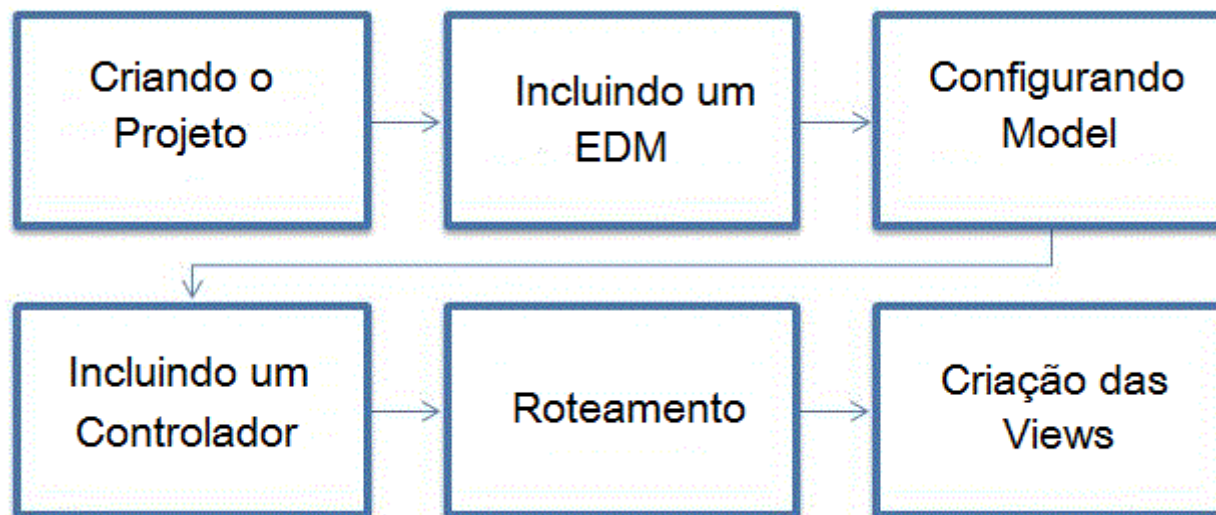


1. A requisição chega pelo navegador
2. As regras de roteamento são aplicadas
3. O controlador recebe a requisição e consulta o Model
4. O Model consulta os dados

5. O resultado é apresentado usando templates views

O processo de criação da aplicação MVC usando ASP .NET MVC 5

A figura abaixo mostra os passos que iremos seguir para criar a nossa aplicação usando a [ASP .NET MVC 5](#) e o [Entity Framework](#):

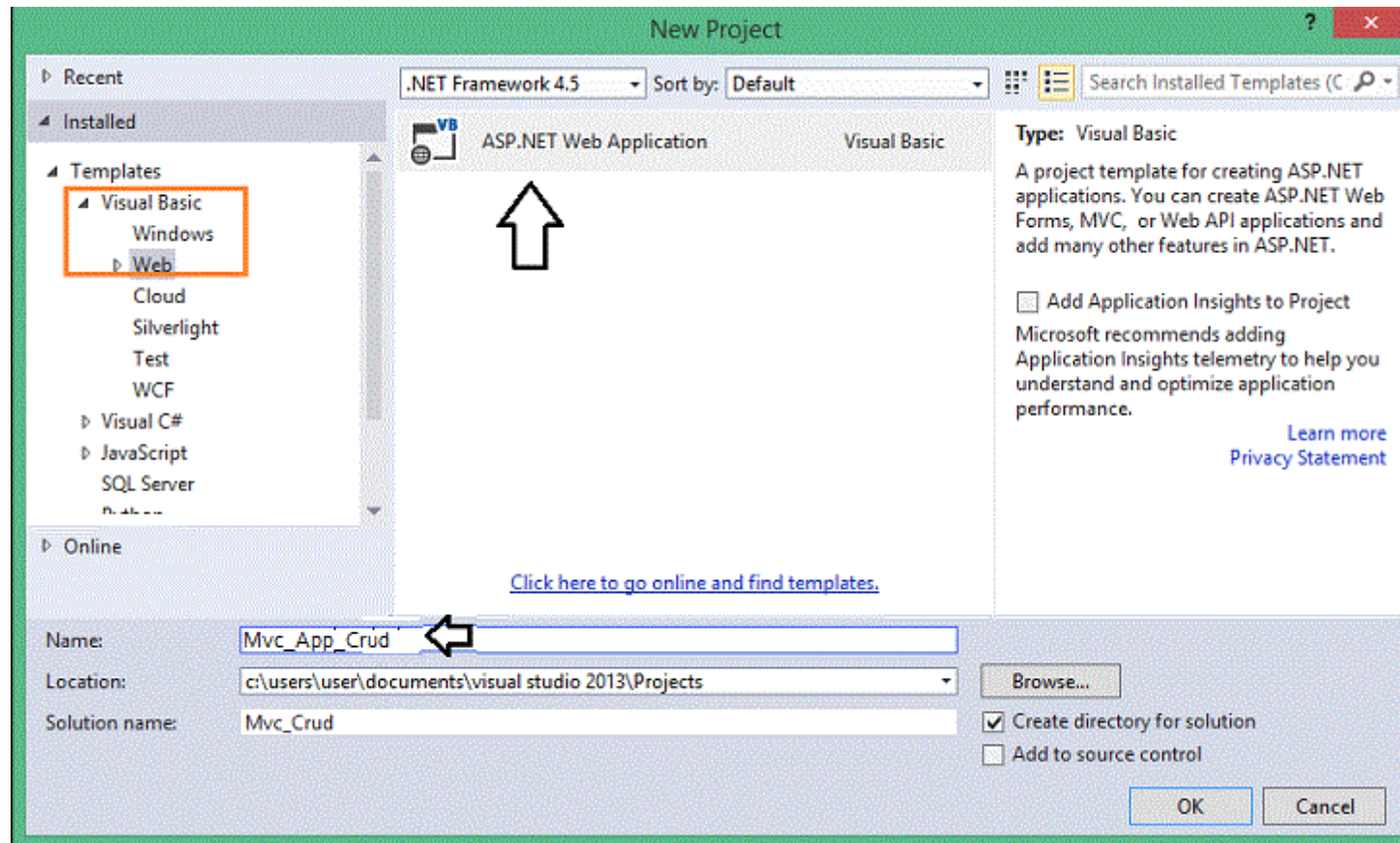


Criando o projeto no Visual Studio 2013 Express for web

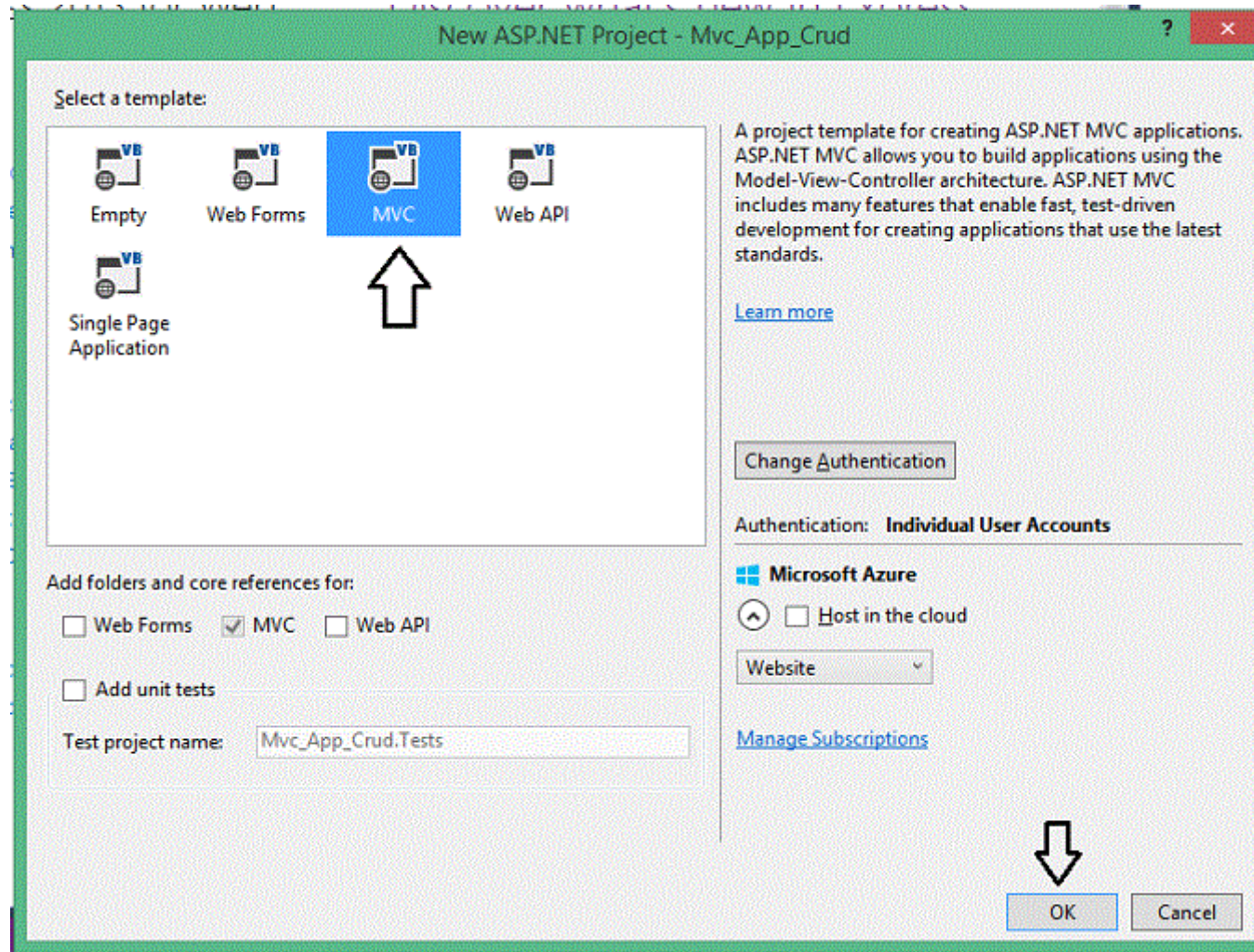
Abra o [VS 2013 Express for web](#) e clique em **New Project**;

Selecione a linguagem Visual Basic ou Visual C# (eu vou usar a linguagem VB.NET) e o template **ASP .NET Web Application**;

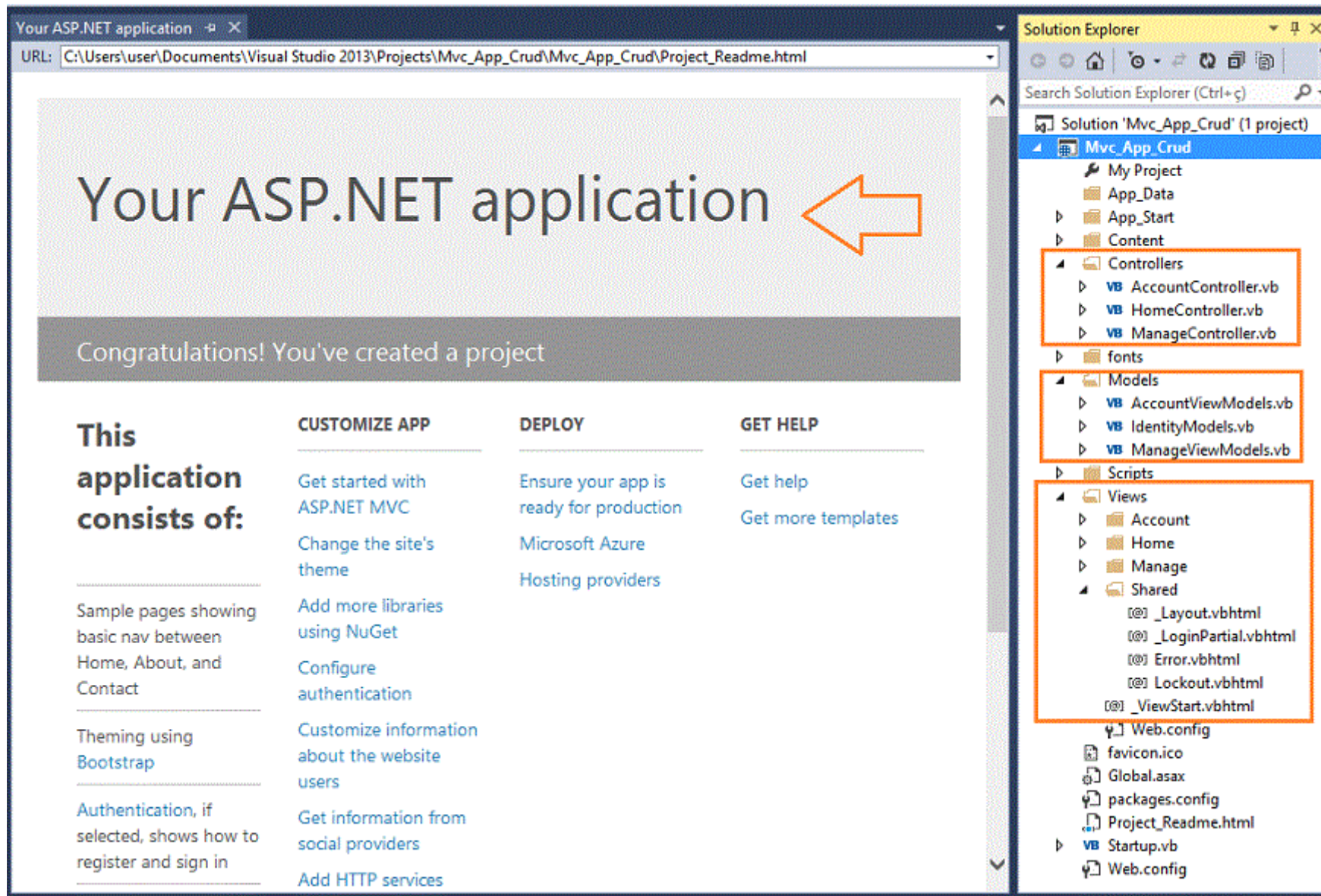
Informe o nome **Mvc_App_Crud** ou outro de sua preferência e clique no botão **OK**;



A seguir selecione o template **Empty**, marque o item **MVC** para criar as pastas e incluir as referências básicas ao projeto; finalmente clique no botão OK:



Ao final teremos uma solução criada com a seguinte estrutura exibida na janela **Solution Explorer**:



Notamos que o projeto já foi criado com uma estrutura contendo as seguintes pastas:

- **App_Start** - Contém os arquivos de inicialização da aplicação dentre eles o **RouteConfig** que define o roteamento padrão usado no projeto;
- **Controllers** - Contém os controladores usados no projeto;

- **Models** - Contém as classes com o modelo de dados usando no projeto padrão;
- **Views** - Contém os arquivos gerados para exibição nas respectivas pastas relacionadas com o nome do controlador. Os arquivos possuem a extensão **.vbhtml** e são arquivos **Razors** gerados para linguagem VB .NET.

Além disso temos o layout e o estilo da página gerados com os recursos do Twitter Bootstrap.

Podemos alterar alguns arquivos ajustando-os para a nossa necessidade. Eu vou alterar alguns dados dos arquivos [_Layout.vbhtml](#), [/Home/Index.vbhtml](#), [/Home/Contact.vbhtml](#) e [/Home/About.vbhtml](#) apenas para dar uma personalizada no texto do projeto. *(isso não é necessário para que a aplicação funcione é apenas uma customização que não é obrigatória)*

Definindo o Model - Incluindo um Entity Data Model (EDM) no projeto

Agora vamos definir nosso modelo de dados e vamos fazer isso incluindo um **Entity Data Model** na pasta **Models** do projeto.

Vamos usar como exemplo um pequeno banco de dados chamado **Escola** criado no SQL Server via [SQL Server Management Studio](#) contendo as tabelas : **Aluno**, **Departamento**, **Assunto**.

Abaixo vemos a estrutura de cada tabela:

Column Name	Data Type	Allow Nulls
AlunoID	int	<input type="checkbox"/>
AlunoNome	nvarchar(50)	<input checked="" type="checkbox"/>
DepartamentoID	int	<input type="checkbox"/>
AssuntoID	int	<input type="checkbox"/>
Aluno		

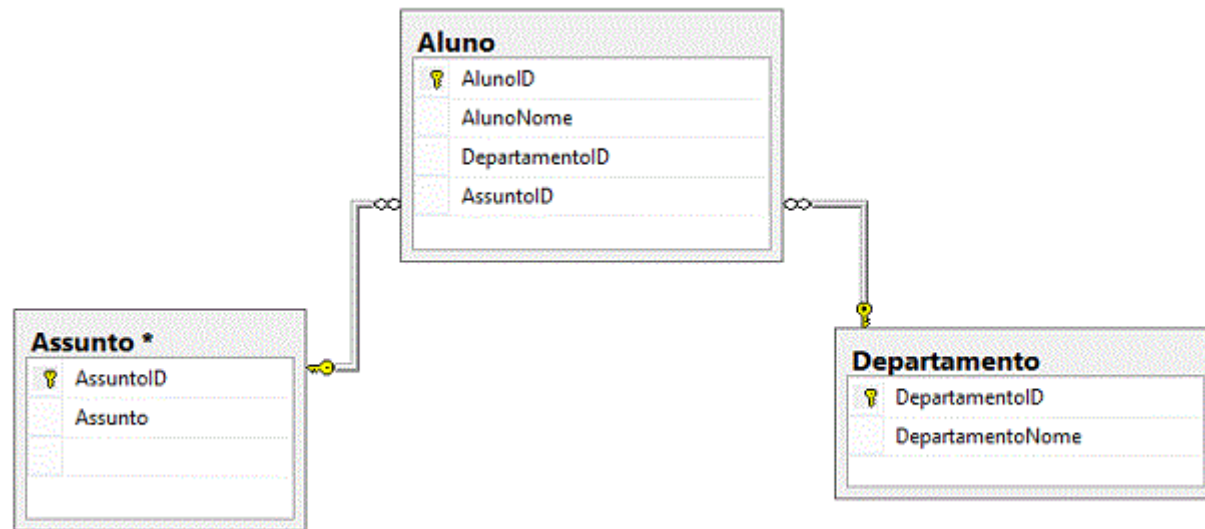
Column Name	Data Type	Allow Nulls
DepartamentoID	int	<input type="checkbox"/>
DepartamentoNome	nvarchar(50)	<input checked="" type="checkbox"/>
Departamento		

Column Name	Data Type	Allow Nulls
AssuntoID	int	<input type="checkbox"/>
Assunto	nvarchar(250)	<input checked="" type="checkbox"/>
Assunto		

Os campos **AlunoID**, **DepartamentoID** e **AssuntoID** são chaves primárias do tipo **Identity**.

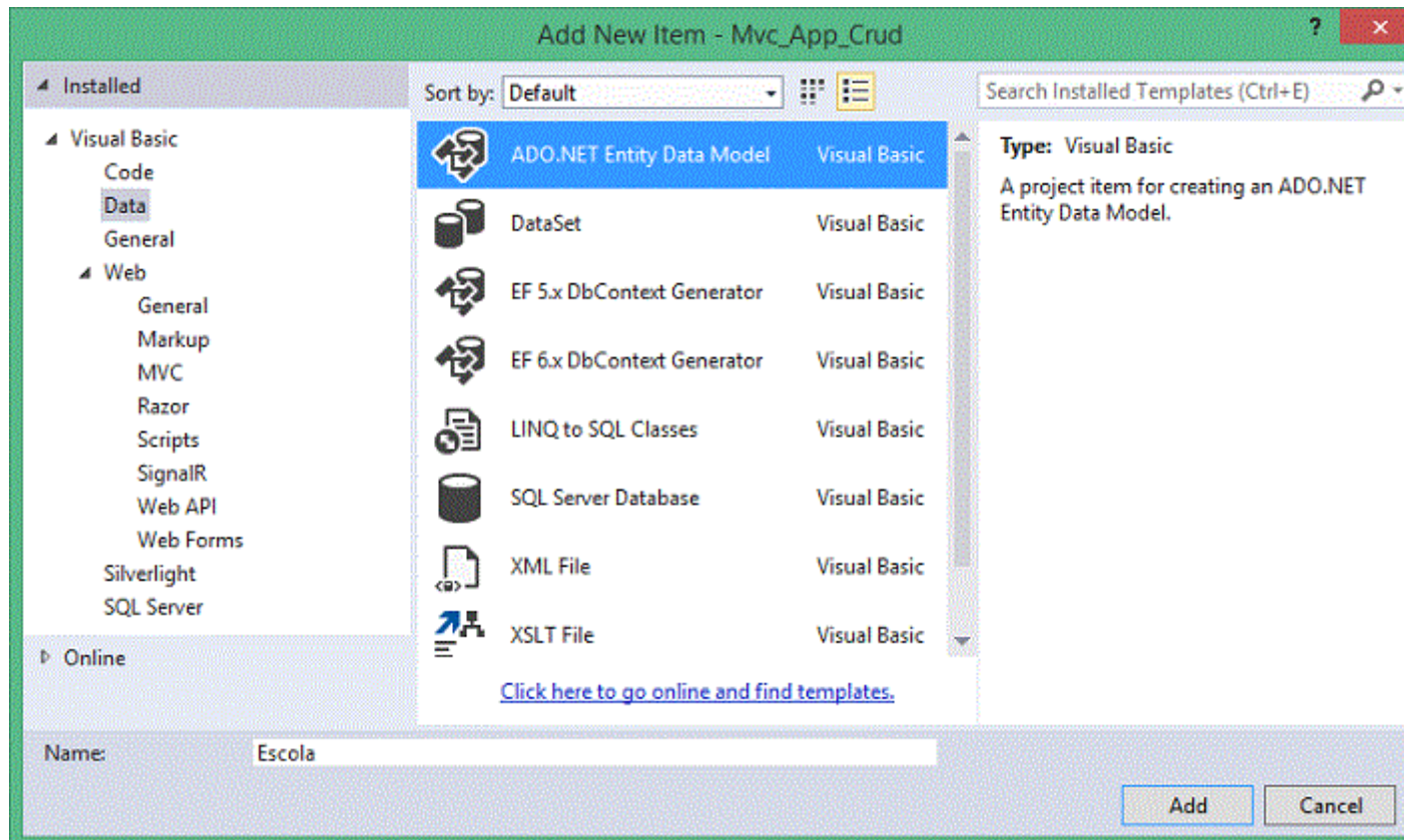
*Nota: Vamos incluir alguns dados na tabela **Departamento** e **Assunto** apenas para testes em nossa aplicação visto que iremos criar inicialmente o controlador para o **Aluno** e a na view para incluir um **Aluno** é necessário informar o departamento e o assunto.*

O relacionamento entre as tabelas e visto no diagrama abaixo:

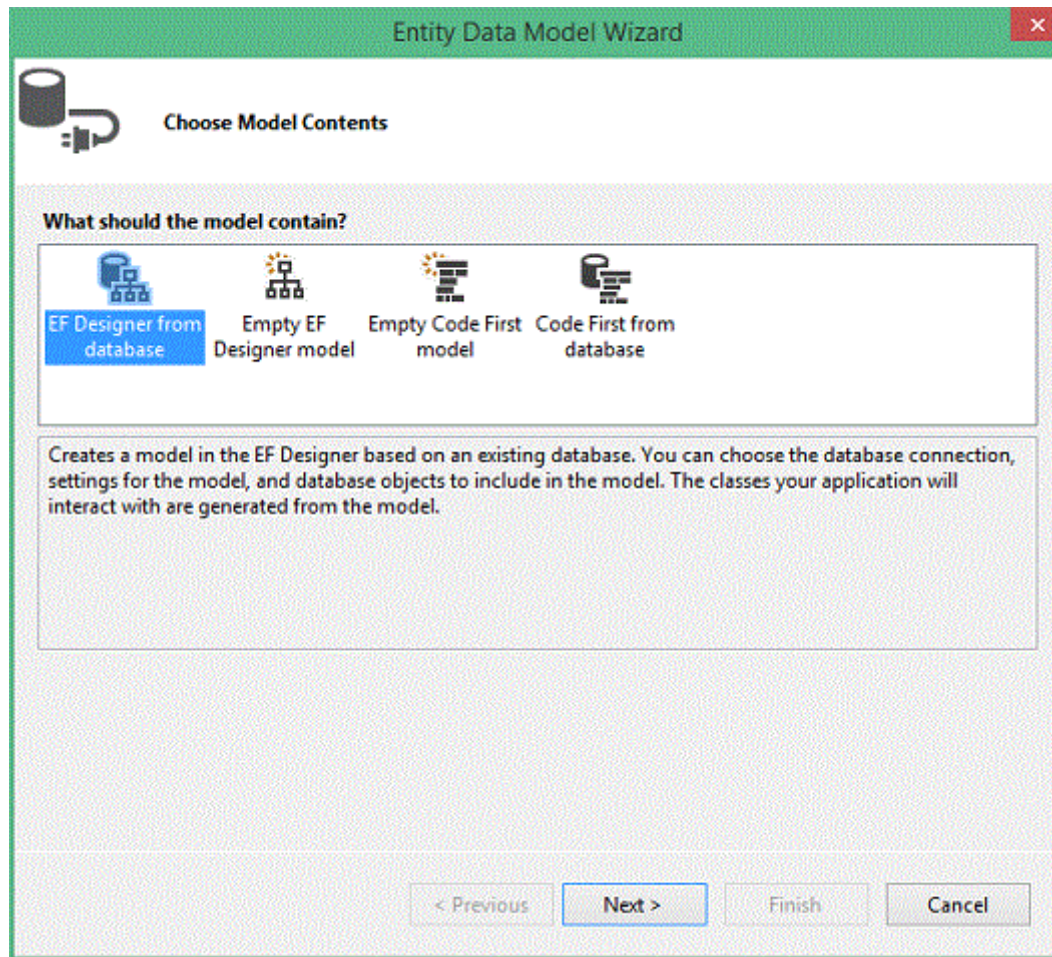


Para incluir um **Entity Data Model** clique com o botão direito sobre a pasta **Models** e a seguir em **Add -> New Item**;

A seguir clique em **Data** e selecione **ADO .NET Entity Data Model** e informe o nome Escola e clique no botão **Add**;



A seguir selecione a opção : **EF Designer from database** pois vamos gerar o **EDM** a partir do banco de dados que já existe:



Selecione o servidor SQL Server e o banco de dados **Escola** e clique no botão OK;

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:
.sqlexpress Refresh

Log on to the server

☒ Use Windows Authentication
☐ Use SQL Server Authentication

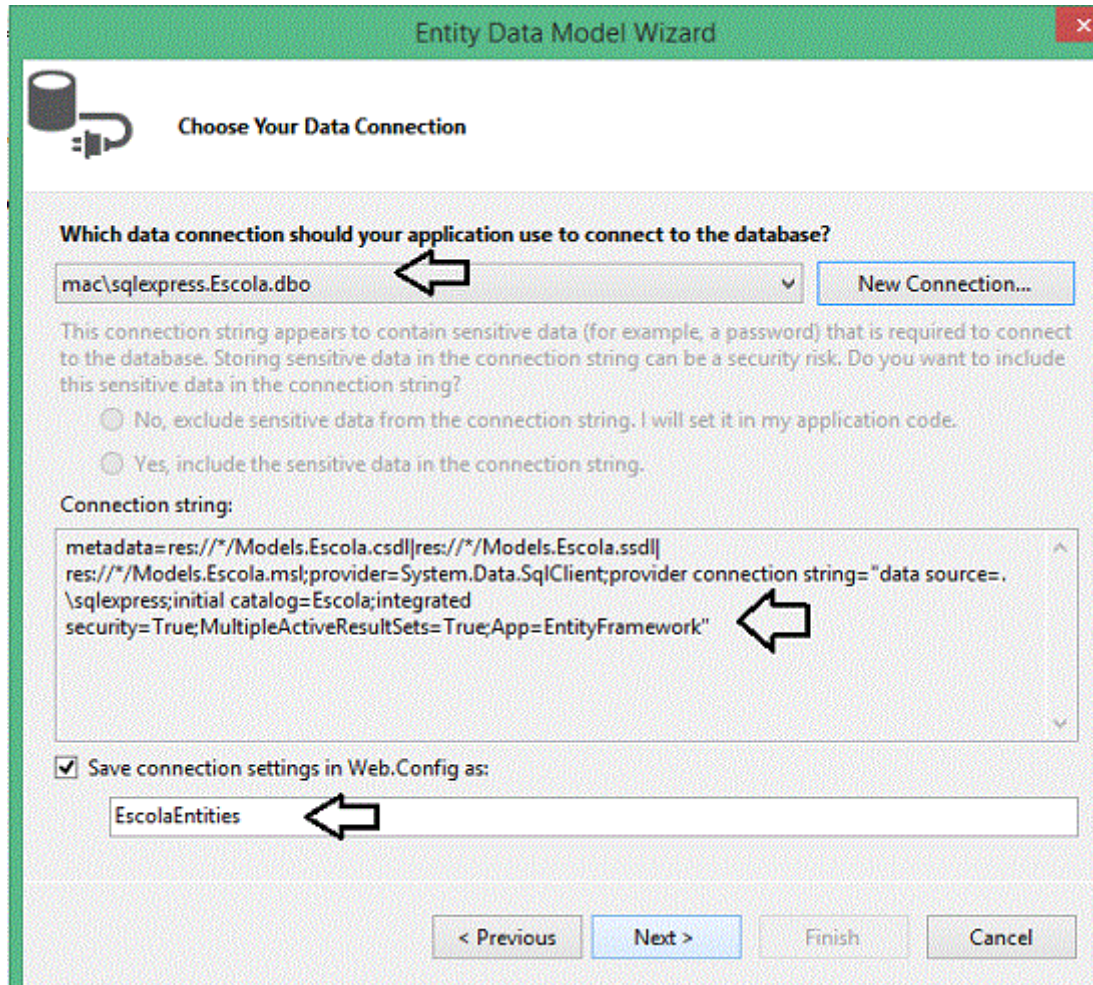
User name:
Password:
☐ Save my password

Connect to a database

☒ Select or enter a database name:
Escola
☐ Attach a database file:
Browse...
Logical name:
Advanced...

Test Connection OK Cancel

Aceite os valores gerados pelo assistente a partir do banco de dados selecionado e clique no botão **Next>**;



The image shows the 'Entity Data Model Wizard' window, specifically the 'Choose Your Data Connection' step. The window has a green title bar with the text 'Entity Data Model Wizard' and a close button. Below the title bar is a header area with a database icon and the text 'Choose Your Data Connection'. The main content area is titled 'Which data connection should your application use to connect to the database?'. It features a dropdown menu with the text 'mac\sqlexpress.Escola.dbo' and a 'New Connection...' button. Below this, there is a warning message: '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 the radio buttons is a text box labeled 'Connection string:' containing the following text: 'metadata=res://*/Models.Escola.csdl|res://*/Models.Escola.ssdl|res://*/Models.Escola.msl;provider=System.Data.SqlClient;provider connection string="data source=. \sqlexpress;initial catalog=Escola;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"'. Below the text box is a checkbox labeled 'Save connection settings in Web.Config as:' which is checked. Below the checkbox is a text box containing the text 'EscolaEntities'. At the bottom of the window are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'. There are three black arrows pointing to the dropdown menu, the connection string text box, and the 'EscolaEntities' text box.

Entity Data Model Wizard

Choose Your Data Connection

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

mac\sqlexpress.Escola.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:

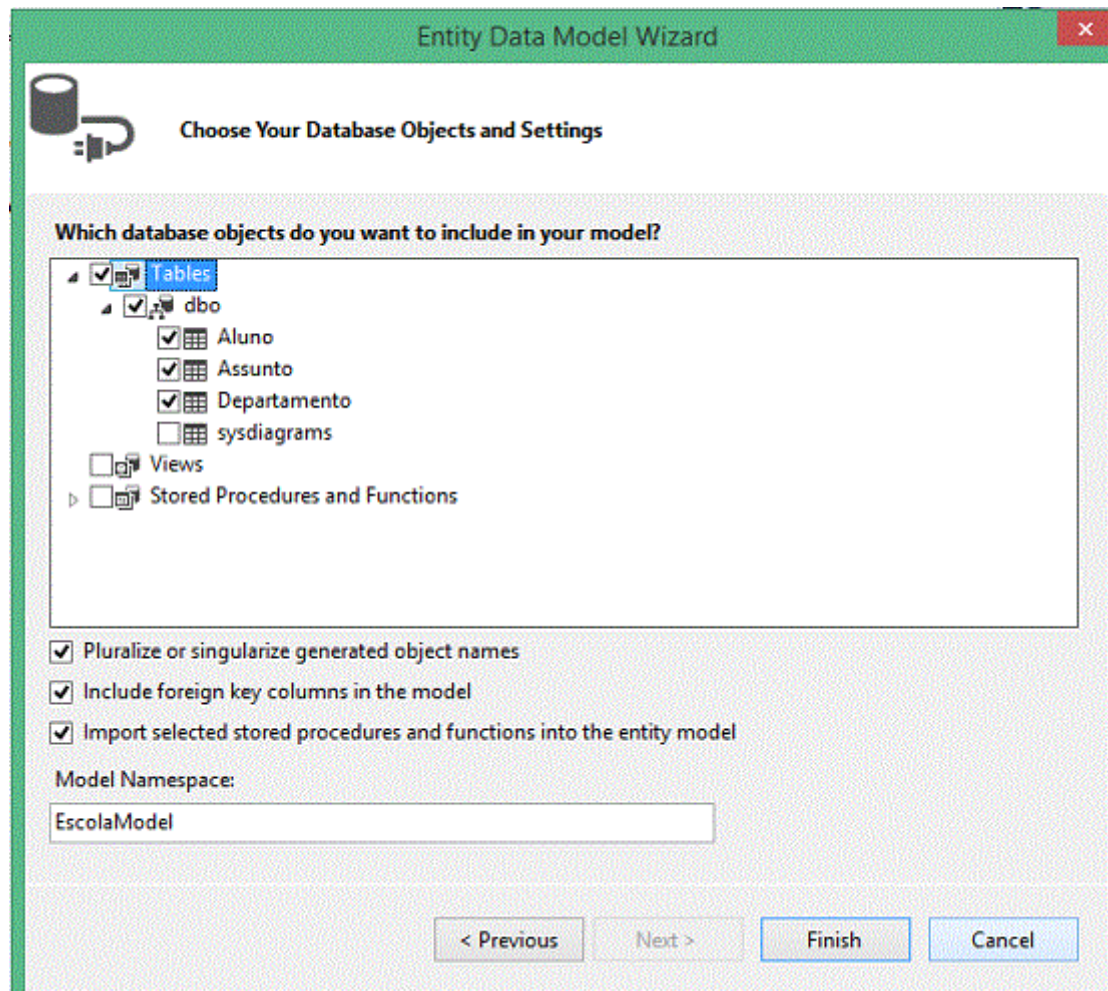
metadata=res://*/Models.Escola.csdl|res://*/Models.Escola.ssdl|res://*/Models.Escola.msl;provider=System.Data.SqlClient;provider connection string="data source=. \sqlexpress;initial catalog=Escola;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework"

☒ Save connection settings in Web.Config as:

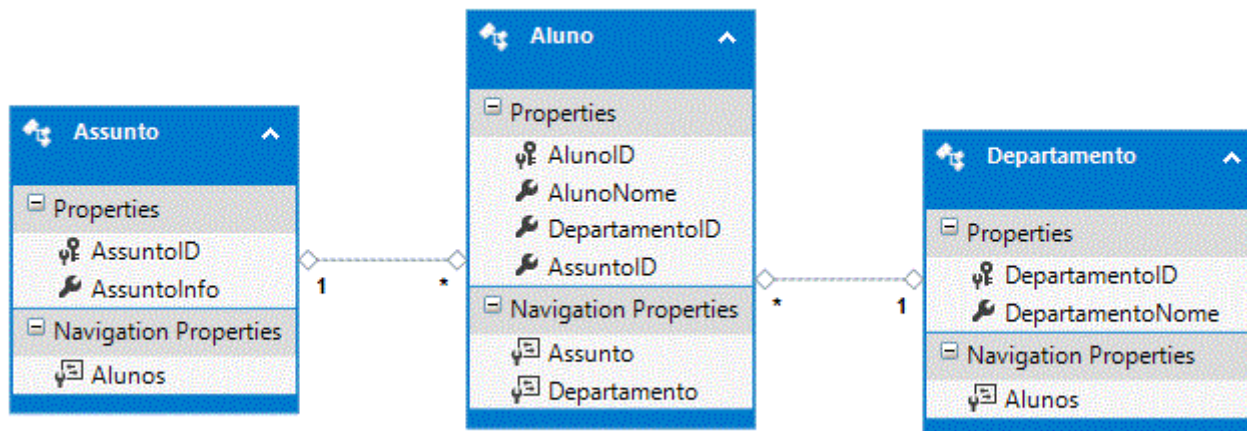
EscolaEntities

< Previous Next > Finish Cancel

Selecione as tabelas [Aluno](#), [Assunto](#) e [Departamento](#) e aceite o nome gerado e clique no botão **Finish**;



Será gerado o modelo de entidades conforme a figura a seguir:



Nota: Eu alterei o nome da propriedade Assunto para AssuntoInfo na entidade Assunto para evitar conflito de nome com o nome da entidade.

Para concluir compile o projeto e verifique se não há erros.

Na [segunda parte do artigo](#) vamos criar o controlador para a entidade Aluno e gerar as views para podermos realizar as operações básicas(CRUD) com alunos.

Falou-lhes, pois, Jesus outra vez, dizendo: Eu sou a luz do mundo; quem me segue não andar  em trevas, mas ter  a luz da vida.

[Jo o 8:12](#)

[Veja os Destaques e novidades do SUPER DVD Visual Basic \(sempre atualizado\) : clique e confira !](#)

Quer migrar para o VB .NET ?

- Veja mais sistemas completos para a plataforma .NET no [Super DVD .NET](#) , confira...
- [Curso B sico VB .NET - V deo Aulas](#)

Quer aprender C# ??

- Chegou o [Super DVD C#](#) com exclusivo material de suporte e vídeo aulas com curso básico sobre C#.
- [Curso C# Basico - Video Aulas](#)

Quer aprender os conceitos da Programação Orientada a objetos ?

- [Curso Fundamentos da Programação Orientada a Objetos com VB .NET](#) 

Quer aprender o gerar relatórios com o ReportViewer no VS 2013 ?

- [Curso - Gerando Relatórios com o ReportViewer no VS 2013 - Vídeo Aulas](#)

Quer aprender a criar aplicações Web Dinâmicas usando a ASP .NET MVC 5 ?

- [Curso ASP .NET MVC 5 - Vídeo Aulas](#)

Gostou ?  [Compartilhe no Facebook](#)  [Compartilhe no Twitter](#)

Referências:

- [Seção VB .NET do Site Macoratti.net](#)
- [Super DVD .NET - A sua porta de entrada na plataforma .NET](#)
- [Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#](#)
- [Seção C# do site Macoratti.net](#)
- [Super DVD C#](#)
- [Super DVD Visual Basic](#)
- [Curso Básico VB .NET - Vídeo Aulas](#)
- [Curso C# Básico - Vídeo Aulas](#)
- [macoratti - YouTube](#)
- [Macoratti.net | Facebook](#)

- [Jose C Macoratti \(@macoratti\)_| Twitter](#)
- [Padrões de Projeto - O modelo MVC - Macoratti.net](#)
- [ASP .NET - MVC - Introdução - Macoratti.net](#)
- [ASP .NET MVC 4 - Macoratti.net](#)
- [Seção ASP .NET MVC - Macoratti .NET](#)
- [C# - Validando dados com Data Annotations - Macoratti.net](#)
- [Entity Framework 4 - Usando Data Annotations - Macoratti ...](#)
- [Introdução ao Entity Framework - Macoratti.net](#)
- [C# - CRUD básico com Entity Framework - Macoratti.net](#)
- [Entity Framework 5 - Operações CRUD - Macoratti.net](#)

[José Carlos Macoratti](#)