

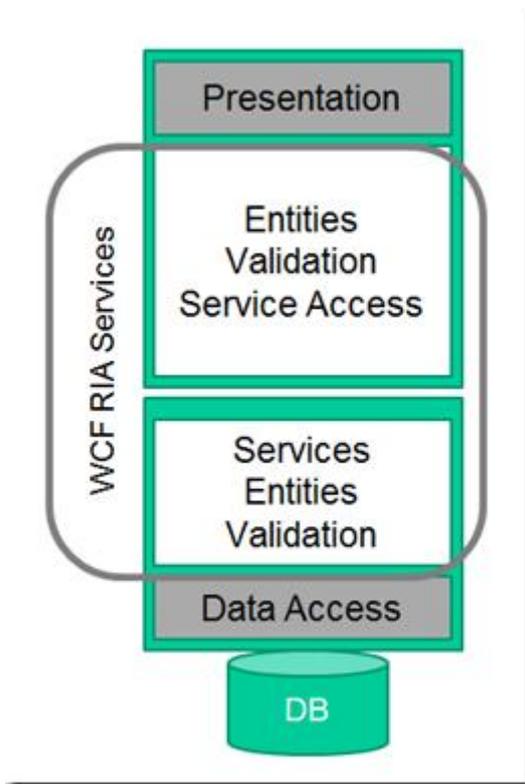
WCF RIA Services Part 1

Introdução

Para construir aplicativos de negócios comerciais em Silverlight (e outras tecnologias de cliente), você tem que trabalhar com dados. Esses dados geralmente não são residentes na máquina do cliente, normalmente são distribuídos entre muitos clientes e são armazenados e operados pelos serviços de back-end. Na realidade, o que você faz é enviar e puxar os dados do cliente para o back-end e invocar as operações no servidor do aplicativo cliente do Silverlight. Seria ótimo se a arquitetura de empacotamento e envio e recuperação desses dados fossem automatizadas para você, permitindo que você se concentre apenas sobre os dados que você precisa, as regras que envolvem a manipulação de dados, e como apresentá-lo no cliente.

Isso é exatamente o WCF RIA Services faz por você. O WCF RIA Services é uma nova parte do NET. Silverlight 4 que lhe permite criar rapidamente aplicações N-Tier cliente Silverlight sem a necessidade de você se preocupar com, Lógica e acesso a dados, nem nos serviços para obter dados dentro e fora do seu aplicativo cliente.

O RIA Services cria para você um conjunto de código do servidor, e disponibiliza esse código no cliente. No lado do servidor, o WCF RIA Services ajuda você a definir os seus serviços, as entidades do domínio e da lógica de negócio. No lado do cliente, o WCF RIA Services gera o código correspondente das classes que lhe permitem facilmente chamar os serviços, dando suporte a validação e outros códigos que podem ser compartilhado entre o cliente e o serviço. O diagrama abaixo mostra em que parte da arquitetura o WCF RIA Services trabalha.



Nesta parte 1, vou fazer uma rápida apresentação dos princípios básicos do WCF RIA Services.

O aplicativo que vou usar como exemplo é um gerenciador de tarefas. Imagine que você precisa inserir tarefas para as pessoas, controlar o tempo que gastam com essas tarefas, associar as tarefas aos clientes e projetos e assim por diante.

Vamos ver os seguintes conceitos-chave no WCF RIA Services:

- **Projeto de Links** - A ligação entre um projeto de cliente do Silverlight e um aplicativo da Web ou servidor de biblioteca de classes. O projeto define o servidor ou referências a serviços de domínio, as entidades, metadados e código compartilhado definido no lado do servidor. Como resultado da ligação, o código apropriado é gerado no projeto do cliente em tempo de compilação.
- **Domain Services** – Este é o cerne da construção WCF RIA Services. Um serviço de domínio define as operações apoiadas no lado do servidor. Estes são geralmente focada principalmente em operações de CRUD na entidade. Estas operações são expostas automaticamente via WCF e podem ser chamadas no cliente através do código gerado.
- **Entities** – você define os tipos entity no servidor, e uma definição do lado do cliente e a entidade é gerada para uso no lado do cliente. Você pode adicionar validação e outros tipos de metadados para a definição da entidade que irá afectar o código gerado no lado do cliente, permitindo-lhe manter apenas uma definição de servidor. Os tipos de entidade no cliente e servidor são usados para serializar e desserializar

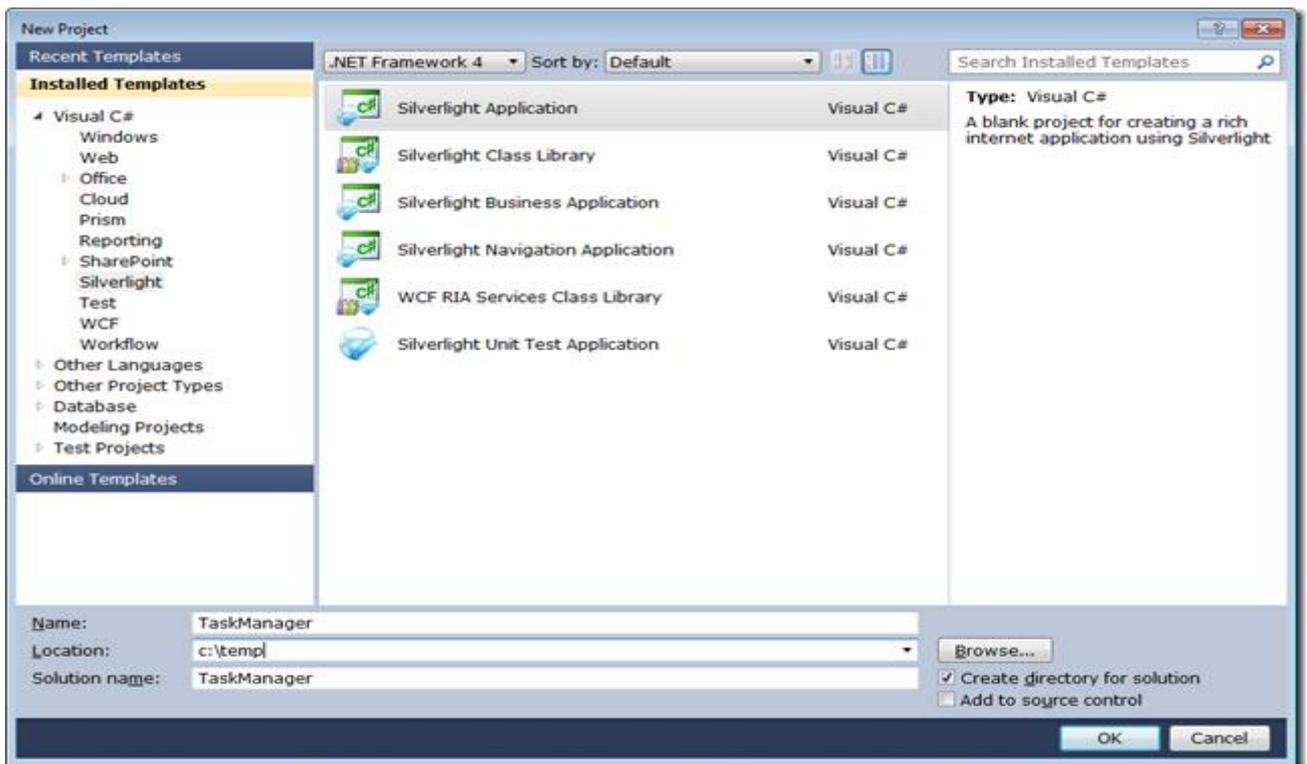
os dados através do serviço WCF. Os tipos de entidade pode ser criada com o Entity Framework, LINQ to SQL, ou pode ser objetos que você mesmo definiu.

- **Domain Context** – Esta é a contraparte do lado do cliente para o serviço de domínio. É o código gerado no lado do cliente que lhe dá acesso fácil a funcionalidade que reside no lado do servidor. Internamente, ele contém um proxy WCF que faz as chamadas de serviço, e que também gerencia a criação de consultas que são executadas no lado do servidor, o controle de alterações para as entidades sendo manipulado no cliente e muito mais.
- **DomainDataSource** – Este é um objeto fonte de dados que pode ser usado para executar consultas e apresentar mudanças para o lado do servidor. Ele fica associado com o contexto de domínio e faz chamadas através desse contexto de domínio em seu nome para entidades para realizar as consultas e atualizações. Faz também a ligação direta de dados XAML.

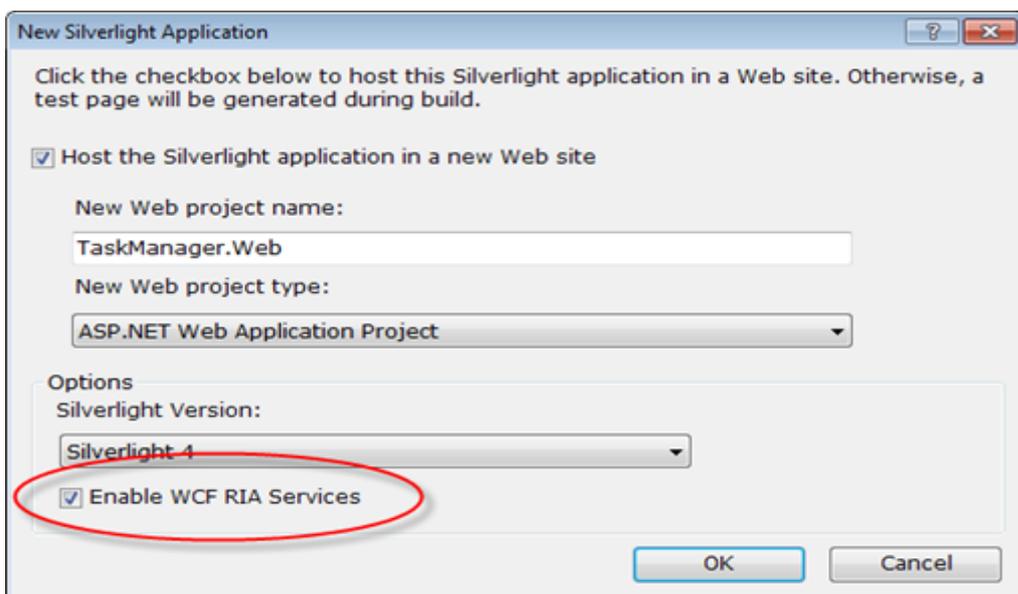
Passo 1: Criar uma aplicação Silverlight

A aplicação que vou construir nesta série é baseada em um simples banco de dados que armazena informações de tarefas e entidade relacionada. Para acompanhar os passos, você terá de executar o script TaskManager.sql para criar o esquema de banco de dados e preenchê-lo com uma amostra de algumas entidades. Depois de ter o banco de dados, você está pronto para começar a criar o aplicativo. Você também precisa ter o Silverlight 4 instalado e o Visual Studio 2010, e instalar WCF RIA Services também.

Crie um novo projeto de aplicativo do Silverlight chamada TaskManager. Você pode usar WCF RIA Services com qualquer dos tipos de projeto do Silverlight, incluindo o modelo de negócios de aplicativos Silverlight que é adicionado pelo WCF RIA Services em si. Mas para começar de forma simples, eu vou usar o modelo de aplicativo Silverlight.



Após clicar em OK na janela de novo projeto, será apresentada a janela normal para criar um projeto de hospedagem de aplicativos web. O que você pode ver que há uma opção para ativar o WCF RIA Services. Verifique se caixa para criar o vínculo entre o projeto do cliente e do projeto do servidor para que o WCF RIA Services, estar marcada. Pois ela gerar o código de cliente apropriado com base nos serviços de domínio e entidades que você irá criar

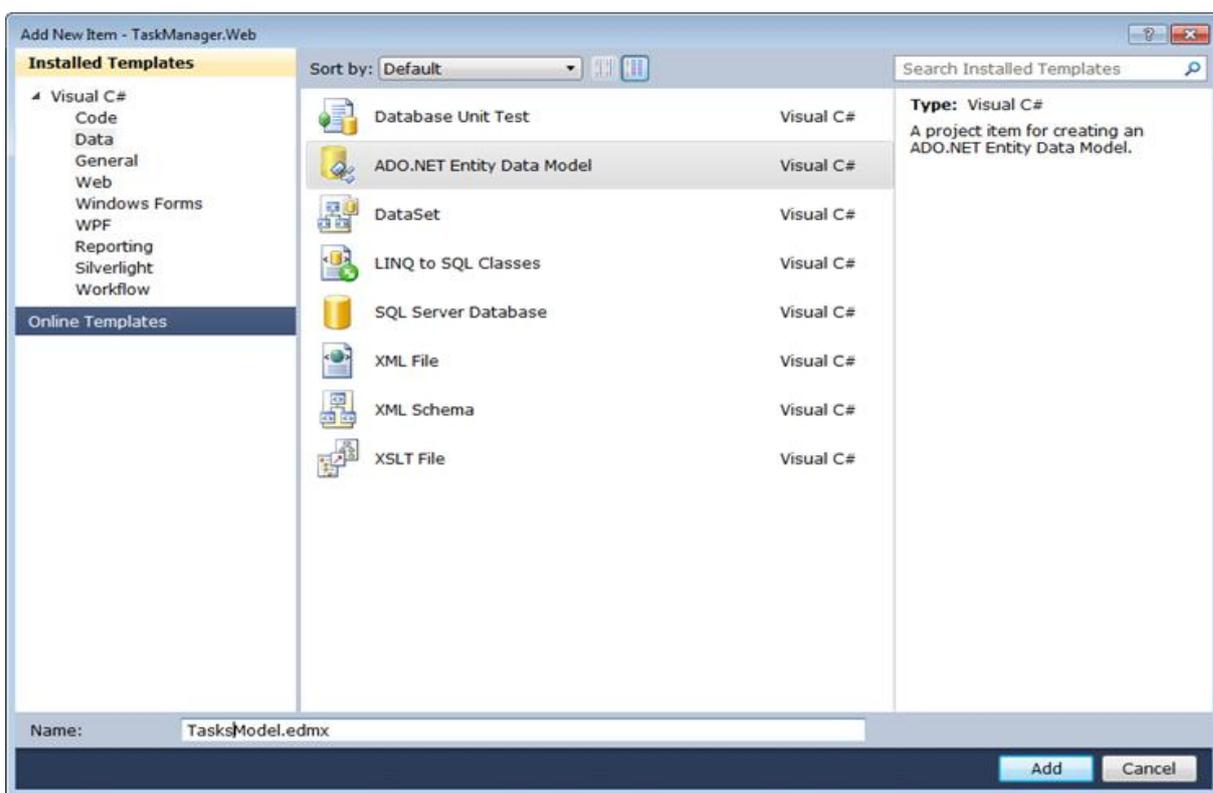


Clique em OK e os projetos será criada.

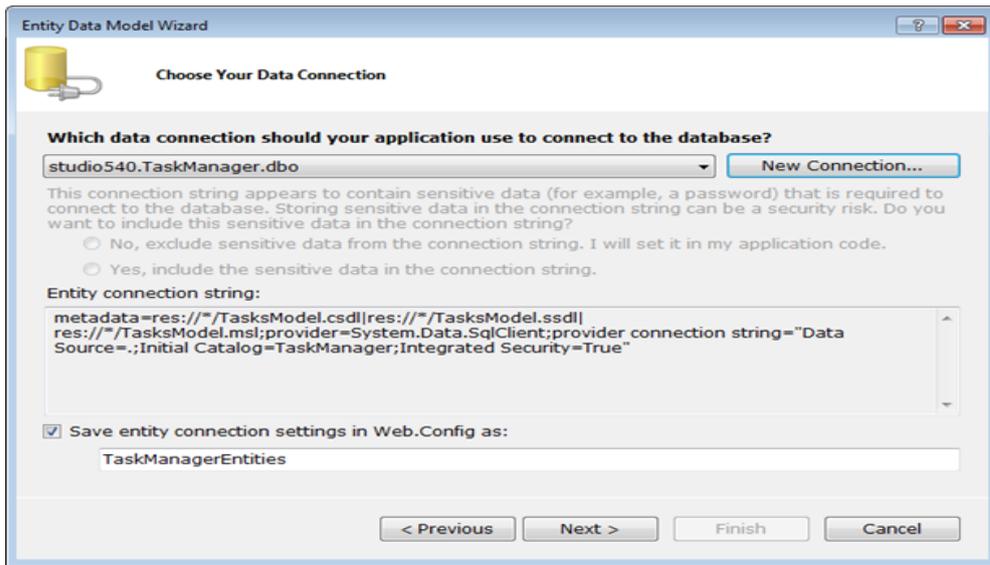
Step 2: Criar o Domínio Modelo de Entidades

O WCF RIA Services faz para você a movimentação de dados entre o cliente e o servidor. Você pode utilizar o Entity Framework, o LINQ to SQL, etc. E você pode criar seus próprios serviços de domínio se você quiser trabalhar com POCOS. Para este artigo, você irá usar Entity Framework.

Clique com o botão direito do mouse sobre o projeto GerenciadorDeTarefa.Web no Solution Explorer e selecione Add> New Item, escolha a categoria de dados, à esquerda, e selecione ADO.NET Entity Data Model, e nomeá-la TasksModel.edmx.

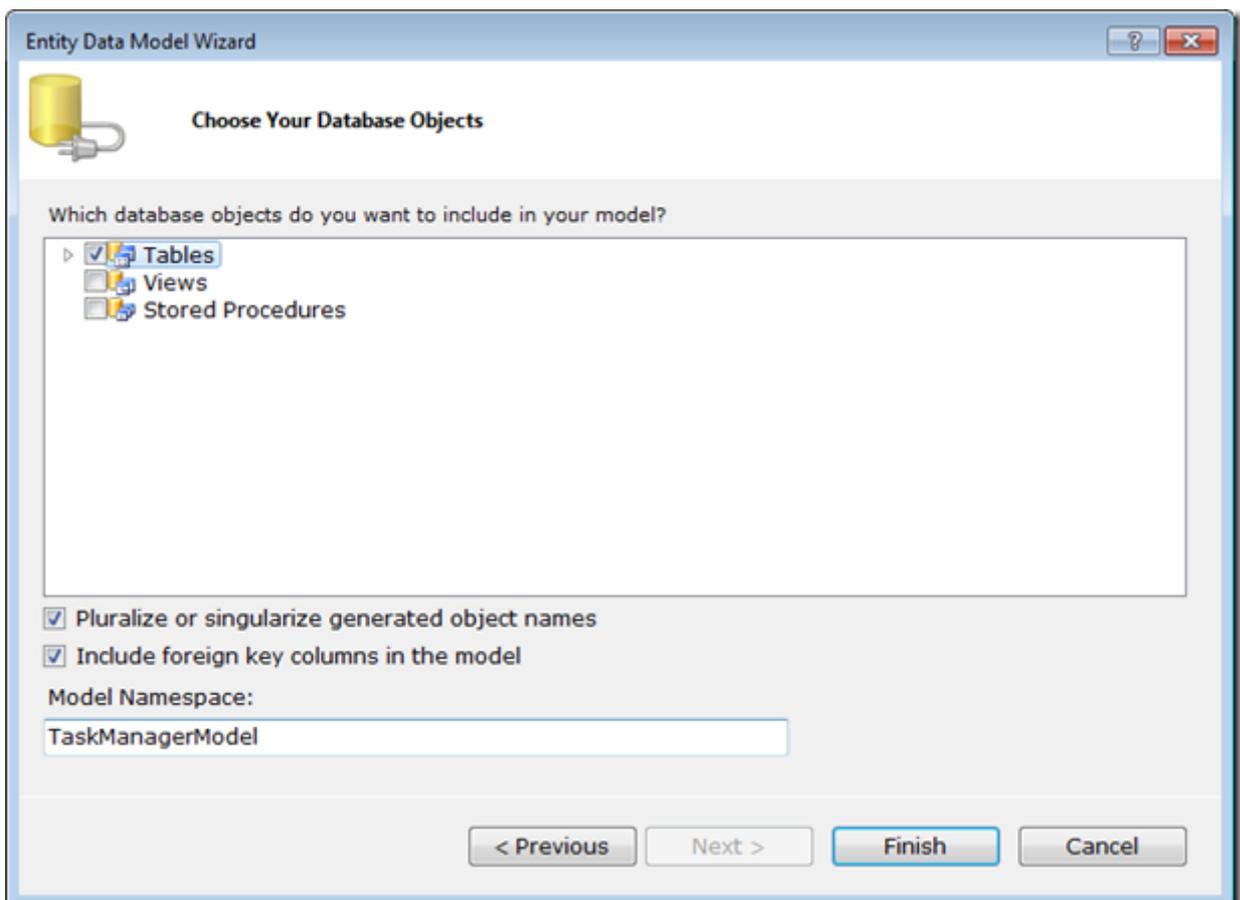


Clique em Adicionar e você verá o Entity Data Model Wizard. Na primeira etapa, deixe a opção de gerar a partir do banco de dados selecionado e clique em Avançar. Criar uma nova conexão para o banco de dados TaskManager que você criou com o exemplo de script de código SQL no início.



O nome da conexão padrão deverá ser para TaskManagerEntities. Clique em Avançar.

Na próxima etapa, Choose Your Database Objects, marque a caixa para para selecionar todas as tabelas e clique em Concluir.

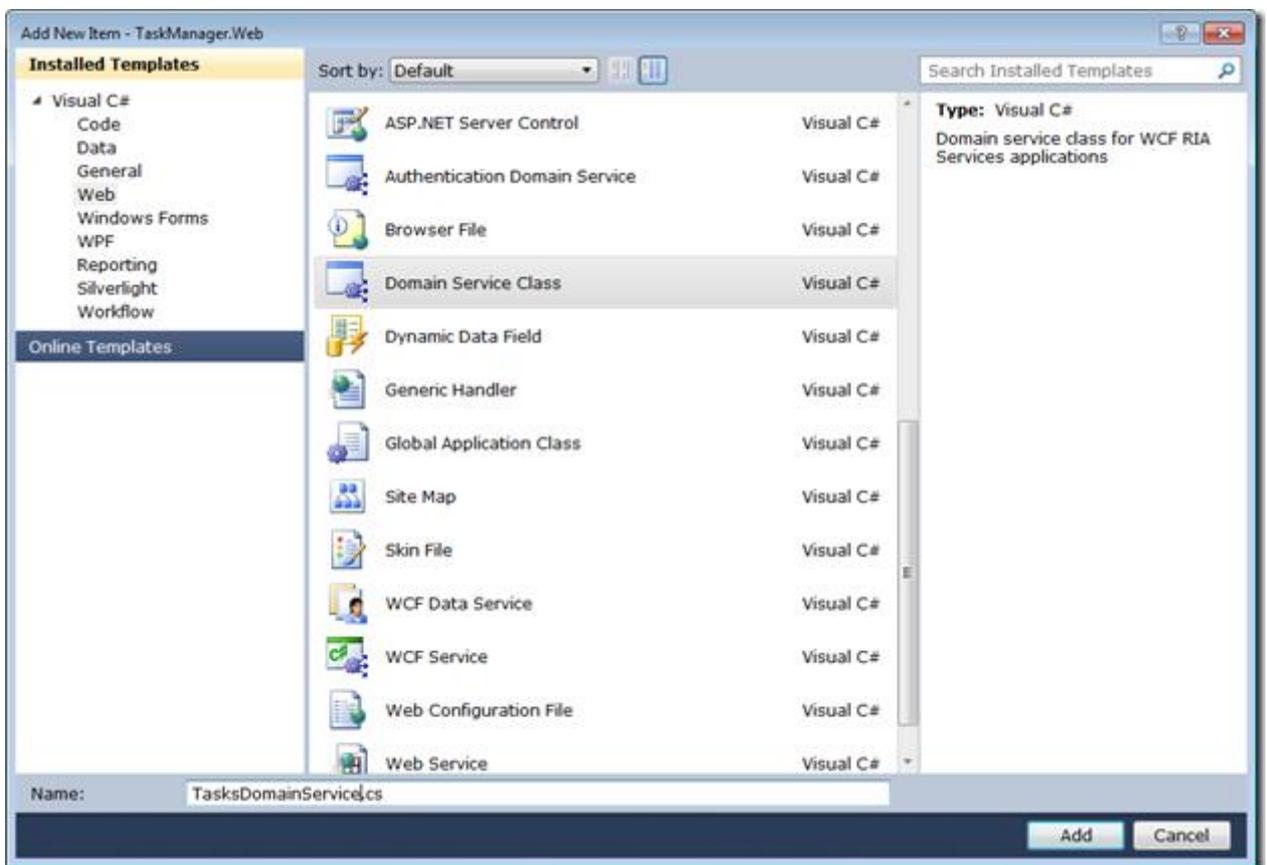


Você tem agora um Entity Framework modelo de dados para todas as tabelas no banco de dados.

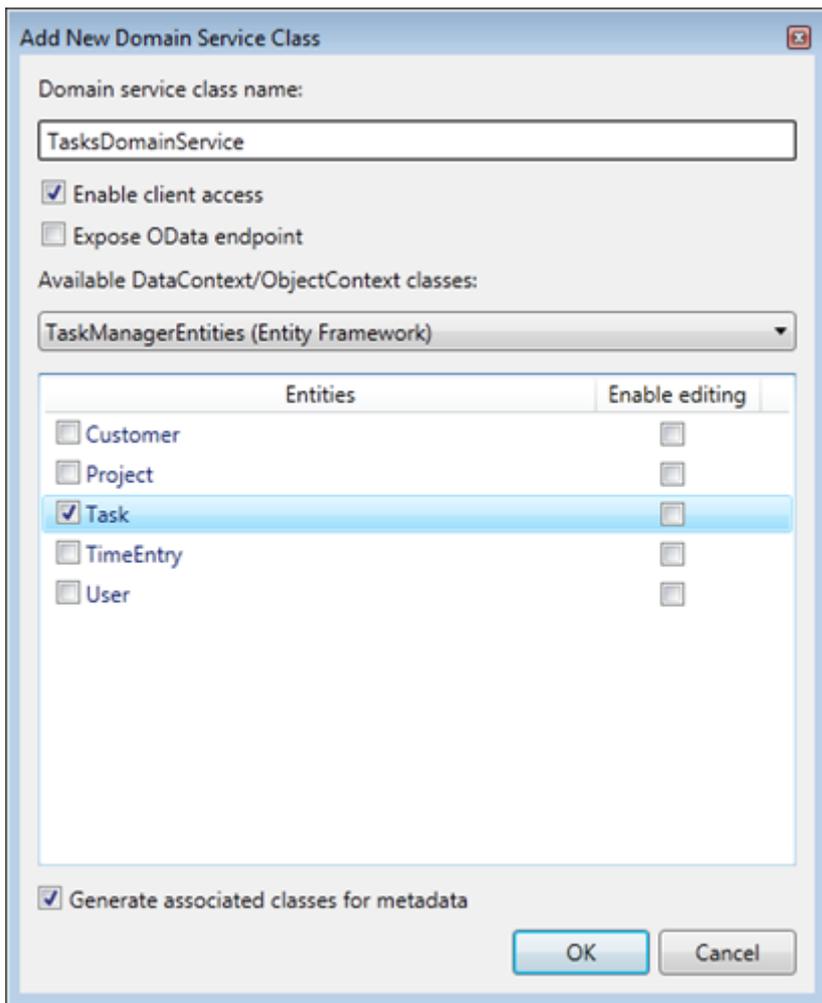
Crie a solução para se certificar de que tudo está bem. Se você esquecer de construir após a adição de seu modelo de dados entidade, ele não vai aparecer ao criar o serviço de domínio na próxima etapa. Ou seja, der um **build** na solução.

Step 3: Definir um Domain Service

Até agora, a única coisa específica para WCF RIA Services você tem feito foi marcar a caixa para criar o link entre o cliente e o projeto de servidor. Agora você vai definir um serviço de domínio. Clique como botão direito do mouse sobre o projeto TaskManager.Web e selecione Adicionar> Novo Item. Selecione a categoria Web e selecione **Domain Service Class** e chame-o de TasksDomainService classe.



Em seguida, será apresentada uma caixa de diálogo que permite selecionar um número de opções para o seu serviço, incluindo a opção mais importante que é a entidades que você irá expor.



Para este exemplo, você só vai estar trabalhando com as tarefas, é só selecionar que tipo de entidade e deixar o resto dos padrões selecionados. Estarei entrando em mais detalhes sobre algumas das outras opções e o que elas fazem em outros laboratórios. Clique em OK e seu serviço de domínio será criado.

Sem os comentários que foram inseridos, a classe resultante parece com esta:

```
[EnableClientAccess () ]  
public class TasksDomainService :  
LinqToEntitiesDomainService<TaskManagerEntities>  
{  
    public IQueryable<Task> GetTasks ()  
    {  
        return this.ObjectContext.Tasks ;  
    }  
}
```

A classe `LinqToEntitiesDomainService <T>` oferece a ligação entre um modelo de dados entidade e os serviços que serão expostos pelo WCF RIA Services. O atributo `EnableClientAccess` é o que faz com que o código do cliente a ser gerado em tempo de compilação seja vinculado ao projeto do cliente. De lá, você adiciona os métodos para o seu serviço de domínio para executar operações CRUD sobre as entidades e,

possivelmente, apenas expor as operações que podem ser chamadas a partir do cliente também.

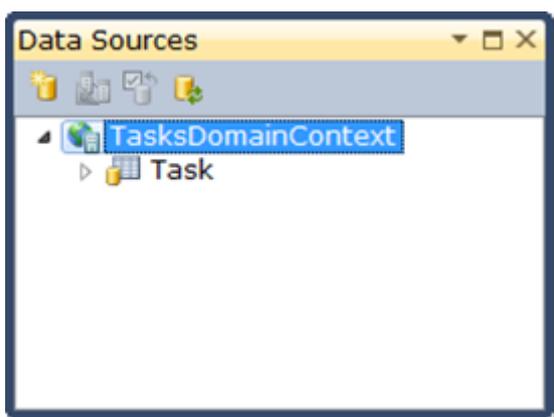
Veja que o método gerado usa convenção para indicar que se trata de um método de consulta com base no tipo de retorno de IQueryable <T> e o nome do método. Ele apenas delega a uma instância do ObjectContext modelo de entidade de dados que é criado como um membro da classe base e usá-lo para retornar todas as tarefas. Se você desejar, você pode modificar a consulta para ordenar, filtrar ou paginar os resultados.

Reconstrua a solução para se certificar de que tudo está bem e gerar o código do lado do cliente. Se você mostrar todos os arquivos no projeto TaskManager, você verá uma pasta com um arquivo Generated_Code TaskManager.Web.g.cs nele. Este contém o código gerado..

Step 4: Recuperando Dados com DomainDataSource

Para esta etapa, você vai aproveitar os recursos de arrastar e soltar do WCF RIA Services usando o DomainDataSource diretamente na interface do usuário. Em um laboratório posterior discutiremos o padrão MVVM, porque esta não é a melhor arquitetura para aplicações complexas.

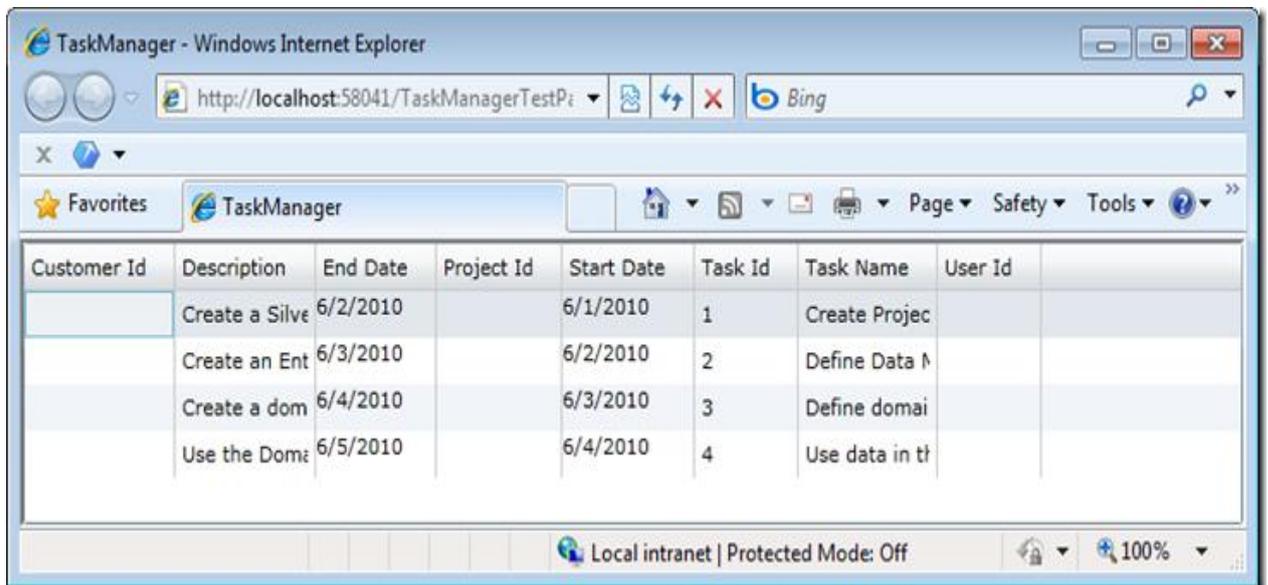
Na MainPage.xaml do projeto GerenciadorDeTarefas. Abrir a janela Data Sources (menu Dados> Show Data Sources). Você deve ver as entidades para os tipos retornados pelo serviço de domínio listado na janela Data Sources.



Arraste e solte o tipo de entidade de tarefas para o MainPage.xaml no designer. Um DataGrid será gerada com colunas adequadas para cada uma das propriedades do tipo de entidade e um DomainDataSource também serão declarados e ligado ao método GetTasksQuery na classe TasksDomainContext. Você pode ir no XAML e remover todas as propriedades de dimensionamento e posicionamento do DataGrid (altura, largura,

HorizontalAlignment, VerticalAlignment, e Margem) para que o DataGrid preencha a grade toda.

Criar e executar o aplicativo. Você deve ver algo como o seguinte.



Resumo

Neste laboratório, foram abordos os conceitos básicos por trás de um aplicativo WCF

Services RIA . Você criou um Entity Data Model, um RIA Services DomainService, e

consumiu o serviço no cliente utilizando o DomainDataSource e o

TasksDomainContext gerado