

# SILVERLIGHT

IFRN- Natal - Central

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Como o **Silverlight** é uma plataforma cliente, projetado para rodar dentro de um navegador, ele não foi projetado para ter acesso diretamente ao banco de dados. Como alternativa, os dados devem ser explicitamente exposto externamente do servidor, usando algo tal como um web service para fornecer um canal de ligação entre o banco de dados e a aplicação.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Há várias maneiras de extrair dados do servidor e consumi-los na aplicação. Entretanto, Windows Communication Foundation (WCF) é geralmente a tecnologia mais popular para criar serviços suportados pelo .NET Framework.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ A Microsoft criou uma nova tecnologia chamada WCF RIA Services (comumente referida como RIA Services), especificamente projetada para ser usada em aplicações Silverlight.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ RIA Services é uma camada no topo do WCF que fornece um padrão prescritivo e framework que você pode usar quando construir aplicações orientado-a-dados (data-driven) que consume dados de um servidor.

# Expondo Dados do Servidor: Usando WCF RIA Services

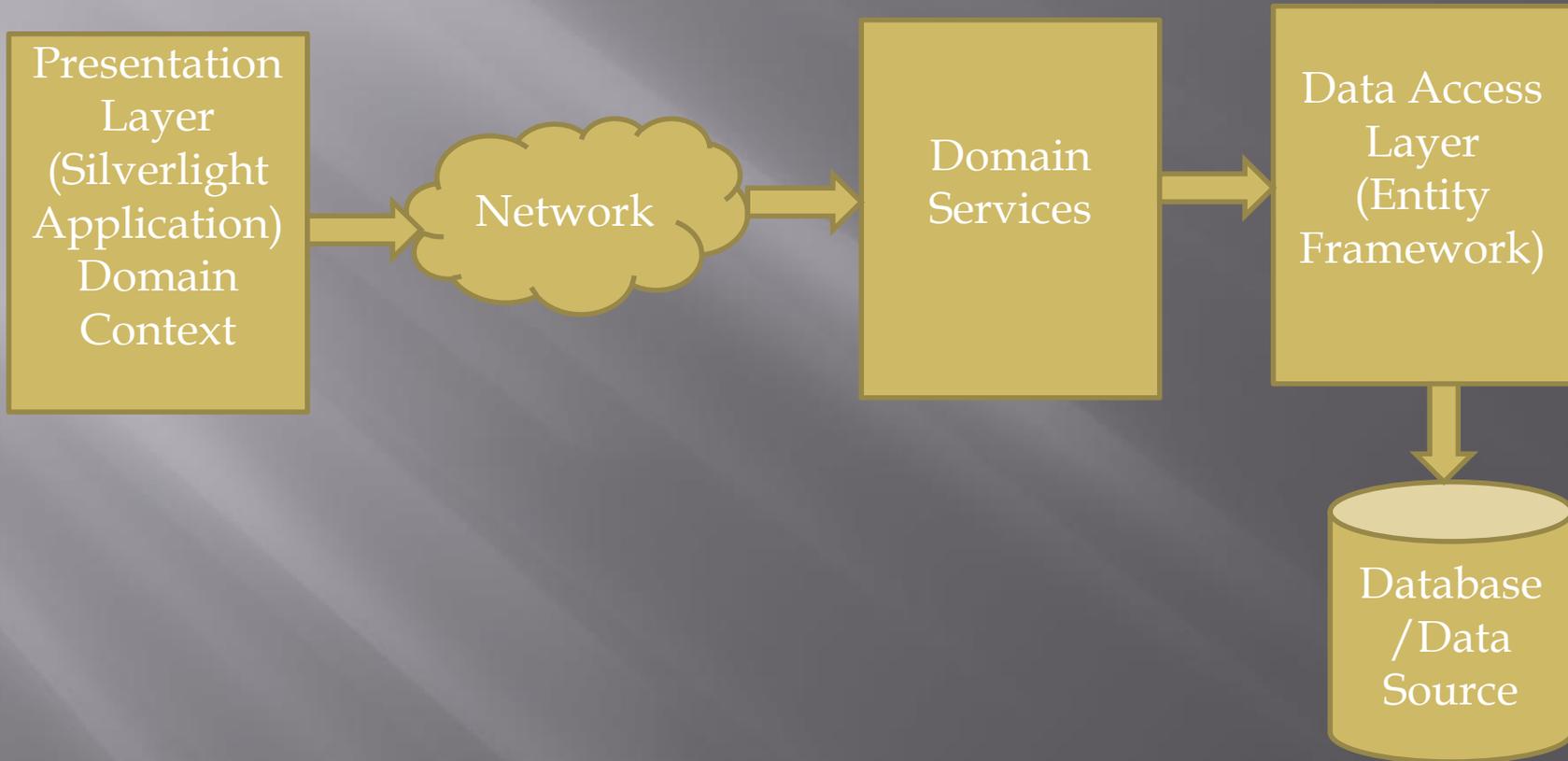
- ▣ Uma outra maneira de dizer é que RIA Services simplifica o processo de exposição de dados a partir de um servidor, consumindo em sua aplicação Silverlight, e compartilhando esses dados com a camada lógica de negócio – o qual é possível por fornecer um conjunto de padrões na criação do serviço, e fornece também um framework para suportá-los.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ O que é o WCF RIA Services?

Ria Services é em parte um padrão de design centrado nos dados para estruturar sua aplicação; é um framework que fornece avançado gerenciamento de dados, autenticação/autorização, e recursos de consultas. Ele pega carona em cima do WCF Services que fornece o mecanismo de comunicação entre a camada intermediária e a camada de apresentação (UI).

# Expondo Dados do Servidor: Usando WCF RIA Services



# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ O foco do padrão RIA Services é muito voltada para a camada intermediária (middle tier). Ele não está amarrado a uma tecnologia específica de acesso a dados, embora funcione bem com o Entity Framework e o LINQ to SQL.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Como funciona o Gerador de Código WCF RIA Services?
- ▣ RIA Services requer tanto o projeto cliente quanto o servidor dentro da mesma solução e ligados entre si. Um contém serviços que expõe dados (servidor), e o outro comunica-se com o serviço e consome os dados (cliente).

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Quando um serviço é escrito no projeto servidor seguindo os padrões RIA Services, o compilador RIA Services no Visual Studio gera o código correspondente no cliente, que simplifica o processo de comunicação com o serviço. Também no cliente são criadas as classes de dados proxy (referidas “entidades”) para que possam ser usadas pelo serviço, aplica atributos (validação) sobre as entidades e propriedades, e compartilha esse código entre servidor e cliente.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Vamos investigar o que acontece nos bastidores, quando você ligar os dois projetos através do RIA Services.
- ▣ Para isso, vamos criar uma solução usando o modelo “Silverlight Business Application”, e configurar esse projeto para usar o RIA Services. Teremos então, os dois projetos no solution explorer ligados com o RIA Services.

# Expondo Dados do Servidor: Usando WCF RIA Services

1. Crie um novo projeto no Visual Studio
2. Em Installed Templates, clique na pasta Silverlight
3. Clique no modelo Silverlight Business Application
4. Na janela New Silverlight Application
5. Marque a checkbox Enable WCF RIA Services
6. Veja nas propriedades do projeto Silverlight que o mesmo está ligado ao projeto Web através do RIA Services.
7. Quando você criar um serviço e compila, será criada uma pasta na aplicação cliente com o nome "Generated\_Code e nessa pasta será armazenada todas as classes de proxy para que o cliente possa manipular os dados.

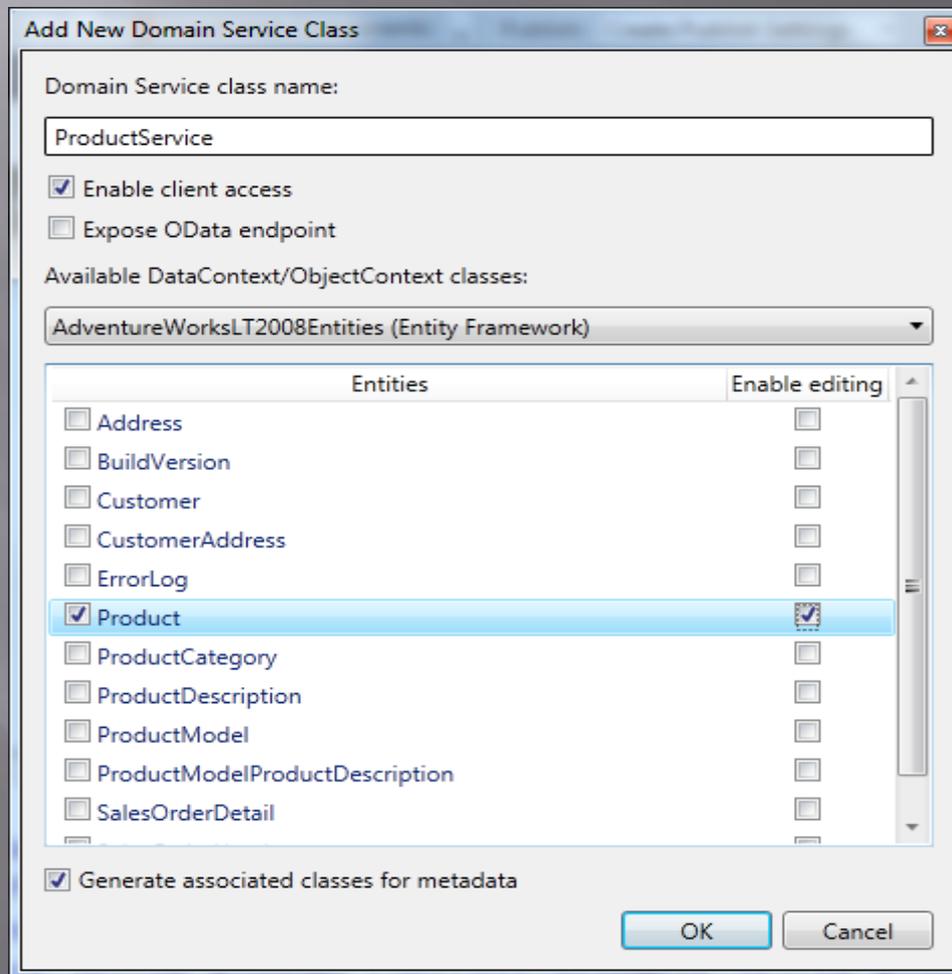
# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Criando um Modelo de Entidades
  1. Adicione um novo item ao seu projeto Web, selecione o item ADO.NET Entity Framework e chame-o de AdventureworksModel.edmx.
  2. Conecte-se ao banco de dados AdventureWorksLT2008 e selecione todas as tabelas
  3. Certifique-se de que as chaves estrangeiras estão marcadas.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Criando o Doman Services
  1. Nesse primeiro momento, vamos trabalhar com a entidade Product. O primeiro passo é adicionar um Domain Service ao projeto Web.
  2. Para isso adicione um novo item a pasta Services, usando o Domain Service Class (categoria Web) e dê o nome de ProductService.cs
  3. Conecte-se ao banco de dados AdventureWork

# Expondo Dados do Servidor: Usando WCF RIA Services



# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Vamos agora olhar cada um dos tipos de operações de domínio que você pode ter em um serviço de domínio.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Enable client access
- ▣ Isso irá decorar a classe de serviço de domínio com o atributo `EnableClientAccess`, que irá indicar para o gerador de código do RIA Services que deve gerar código no projeto Silverlight para habilitá-lo a acessar o servidor de domínio no cliente.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Exposing OData Endpoint
- ▣ Marcando esta check box, habilita o domain service a expor dados via protocolo Open Data. Odata é uma extensão para o protocolo ATOM para padronizar a criação de serviços de dados HTTP, semelhante ao REST, fornecendo as operações CRUD. Este é um novo protocolo que tem o apoio da Microsoft, e a Microsoft dá suporte para o OData a vários de seus produtos (tais como SharePoint e PowerPivot). Selecionando essa caixa de seleção, você permitirá que seu domain service possa ser consumido pelo produto que apoiam o protocolo OData.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Generating Associate Classes for Metadata
- ▣ Cria uma classe Metadata para cada entidade selecionada na lista Entities. Em resumo, eles permitem que você decore as propriedades de suas entidades com os atributos (tais como, validação de dados) sem que você modifique as próprias entidades.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Criando o ProductService Domain Service
- ▣ Para finalizar vamos criara o ProductService domain service, para isso siga os seguintes passos:

# Expondo Dados do Servidor: Usando WCF RIA Services

1. Selecione a entidade Product e o check box Enable Editing.
2. Assegure que a check box Enable Client está selecionada.
3. Assegure que a check box Generate Associated Classes for Metadata está selecionada.
4. Clique no botão OK.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Camada de apresentação
- ▣ Embora RIA Services funcione muito bem com os modelos Entity Framework e LINQ to SQL, felizmente, não está limitado a apenas a essas tecnologias. Vamos dar uma olhada em como implementar RIA Services que faz a transferência de dados entre o servidor e o cliente.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Criando a classe de modelo de apresentação
- ▣ O primeiro passo é criar uma classe padrão contendo apenas as propriedades que devem ser transferidas entre o cliente e o servidor. Atenção, você pode decorar essas classes com os atributos de validação, se assim desejar.
- ▣ Note-se que uma propriedade (ou mais, se necessário) deve ser decorada com o atributo-chave para indicar que o valor da propriedade identifica o objeto (caso contrário, você receberá um erro de compilação, porque o gerente de alterações no cliente não pode gerenciar as mudanças sem ele).

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Para esse exemplo, você vai criar um modelo de classe de apresentação chamada ProductPM, contendo apenas um subconjunto das propriedades de Product.

# Expondo Dados do Servidor: Usando WCF RIA Services

```
public partial class ProductPM
{
    [Key]
    [Editable(false)]
    public int ProductID { get; set; }

    public string Name { get; set; }
    public string ProductNumber { get; set; }
    public decimal ListPrice { get; set; }
    public DateTime ModifiedDate { get; set; }
}
```

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Usando o modelo de apresentação
- ▣ Agora precisamos criar um domain service para que o cliente possa recuperar uma coleção desse objetos, e atualizar/inserir/deletar objetos da coleção se desejar.

# Expondo Dados do Servidor: Usando WCF RIA Services

1. Adicione um novo item para o seu projeto (pasta Services), chame-o ProductPMService, usando o Domain Service Class, e selecione <empty domain service class> disponível na lista DataContexts/ObjectContexts.
2. Isso cria um empty domain service para o qual podemos adicionar operações. Observe que ele herda de DomainService.

# Expondo Dados do Servidor: Usando WCF RIA Services

```
[EnableClientAccess()]
```

```
public class ProductPMService : DomainService
```

```
{  
}
```

Agora precisamos adicionar a operação de consulta para o domain service. Como os dados são provenientes de uma consulta para o modelo Entity Framework, ainda podemos tirar proveito dos benefícios de retornar uma expressão IQueryable projetando dados do modelo Entity Framework para uma coleção de modelos de apresentação dos objetos, e retornando essa consulta.

# Expondo Dados do Servidor: Usando WCF RIA Services

```
[EnableClientAccess()]
public class ProductPMService : DomainService
{
    private AdventureWorksLT2008Entities context = new
AdventureWorksLT2008Entities();

    public IQueryable<ProductPM> GetProducts()
    {
        return from p in context.Product
            select new ProductPM()
            {
                ProductID = p.ProductID,
                Name = p.Name,
                ProductNumber = p.ProductNumber,
                ListPrice = p.ListPrice,
                ModifiedDate = p.ModifiedDate
            };
    }
}
```

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Quando executado, esta expressão preencherá e retornar uma coleção de nosso objeto de modelo de apresentação (ProductPM) para o cliente.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Você também pode implementar as operações de Insert/Update/delete.
- ▣ Você também pode implementar a inserção, atualização e exclusão, passando o objeto ProductPM como parâmetros, e manualmente atualizar o modelo Entity Framework com as mudanças.

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Particularmente quando se insere dados no servidor, você precisa retornar dados para o cliente (tal como a chave primaria, por exemplo). Você pode fazer isso usando o método Associado no objeto domain Service ChangeSet.

# Expondo Dados do Servidor: Usando WCF RIA Services

## ▣ Código para inserir produto usando (Associate)

```
public void InsertProduct(ProductPM productPM)
{
    Product product = new Product();
    product.Name = productPM.Name;
    product.ProductNumber = productPM.ProductNumber;
    product.ListPrice = productPM.ListPrice;
    product.ModifiedDate = DateTime.Now;
    context.Product.AddObject(product);
    context.SaveChanges();
    ChangeSet.Associate(productPM, product, UpdateProductPMKeys);
}
//Fazendo a associação entre as chaves primarias de ProductPM e Product
private void UpdateProductPMKeys(ProductPM productPM, Product product)
{
    productPM.ProductID = product.ProductID;
}
```

# Expondo Dados do Servidor: Usando WCF RIA Services

- ▣ Transação
- ▣ Muitas vezes você vai querer encapsular as mudanças que estão sendo submetidas ao banco de dados dentro de uma transação, especialmente quando se utiliza uma camada de acesso a dados que não seja o Entity Framework ou LINQ to SQL.
- ▣ Você pode fazê-lo em um domain service, substituindo o método *Submit* da classe base, encapsular o escopo de transação, como demonstradas aqui:

# Expondo Dados do Servidor: Usando WCF RIA Services

```
public override bool Submit(ChangeSet changeSet)
{
    bool result = false;

    TransactionOptions transOp = new TransactionOption();
    transOp.IsolationLevel = System.Transactions.IsolationLevel.ReadCommitted();

    using(TransactionScope transaction = new TransactionScope(
        TransactionScopeOption.Required, transOp))
    {
        result = base.Submit (changeSet);
        transaction.Complete();
    }
}
```