



Fabrício Cabral <fabriciofx@gmail.com>

[dotnetarchitects] Diferentes classes de dominio em diferentes namespaces20 messages

Priscila Mayumi Sato <mayumisatox@gmail.com>

Fri, Sep 14, 2012 at 3:04 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Olá, galerinha do mal. Para quem não sabe eu sou mó farsante, eu ando com feras mas eu mesmo não sei de nada. Mó noob.

É que eu me deparei com esse texto e queria saber o que vocês acham. Nele a pessoa possui um classe Pedido em diferentes namespaces, porem similares, e diferentes responsabilidades. Eu nunca tinha visto isso, achei bacana. O que eu fazia antes era diferente e dispensava ter classes de nomes iguais. Pois por exemplo em estoque eu não teria uma classe nova Pedido, eu só teria uma classe responsável pelo estoque e que ela teria uma lista de Pedidos, que é outra classe separada que nem sabe da existência de estoque.

~> o texto:

<http://ramaciotti.com/posts/2012-07-02-classes-e-dominios.html>

Então, eu viajei muito na maionese ou o cara que viajou o que?

#IF a thead for repetida me perdoem e a matem. é que eu fiz uma pesquisa rápida e não achei nada, mas eu posso ter pesquisado com os parametros errados ou não ter prestado atenção

—

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Fri, Sep 14, 2012 at 3:17 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Não li o texto, mas pelo que vc diz, é o que eu andei falando sobre contextos diferentes no mesmo domínio, que sugere modelos diferentes para aspectos diferentes... Pedido no contexto de vendas é uma coisa, no contexto de logística é, essencialmente a mesma coisa, mas o interesse dos diferentes contextos na entidade Pedido são diferentes também. Então você faz com que o Pedido do contexto de vendas não misture seu propósito com o contexto de Logística, apesar de serem essencialmente o mesmo pedido.

Atenciosamente,

Daniel Moreira Yokoyama.

@dmyoko

<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)

<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Rafael Ponte <rponter@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Fri, Sep 14, 2012 at 3:18 PM

Olá Priscila,

Na minha opinião o artigo está certo, seguir a abordagem proposta te ajudará a ter classes com responsabilidades bem definidas, relacionamentos e mapeamentos com seu framework ORM mais simples e até facilitará durante a escrita de testes automatizados.

O problema de ter uma entidade única e inflada para todos os domínios (ou contextos) é muito comum de acontecer quando se pensa apenas no relacionamento "Entidade-Tabela" (muito normal isso acontecer no mundo Java com Hibernate).

2012/9/14 Priscila Mayumi Sato <mayumisatox@gmail.com>
[Quoted text hidden]

—
Rafael Ponte
<http://www.triadworks.com.br>
[Quoted text hidden]

Cassiano Kuss <cassianokuss@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Fri, Sep 14, 2012 at 3:18 PM

Também gosto da ideia de modelar por contexto.
[Quoted text hidden]

Priscila Mayumi Sato <mayumisatox@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Fri, Sep 14, 2012 at 3:50 PM

Obrigada gente o/

Eu tinha ficado em dúvida quanto a isso :/
Achei que ter várias classes com o mesmo nome ficava...estranho.

Mas obrigada. E desculpe, @Yokoyama se eu bati numa tecla repetida >< nem leio todas as discussões do grupo, são muitas.

#busted agora fiquei com vergonha por uma dúvida tão noob
Obrigada
[Quoted text hidden]

—
Atenciosamente
Priscila Mayumi Sato
[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Fri, Sep 14, 2012 at 4:08 PM

Ah... eu não quis insinuar que vc devia ter prestado atenção no que eu ando falando. Eu mesmo acabo acompanhando poucas threads aqui. Eu só quis dizer, caso vc tenha lido eu falando disso na thread de VO, que o assunto era o mesmo. Mas o que eu falei lá, fiz questão de falar aqui pra vc se contextualizar caso não tivesse acompanhado.

E não precisa se sentir noobie...

Perto do tipo de coisa que de vez em quando rola aqui, sua dúvida chega a ser até pertinente.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Fernando Mondo <fernando.mondo@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Fri, Sep 14, 2012 at 4:14 PM

Eu achei bem pertinente, e na verdade nunca vi nenhum Dominio assim...
[Quoted text hidden]

Gerson Dias <gerson.dias.job@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Fri, Sep 14, 2012 at 4:27 PM

Fazer esse tipo de coisa fica fácil quando você tem subsistemas tão claros desta forma.... o problema é em um mesmo subsistema vc identificar os contextos (e oportunidades) de forma tão clara....

Um cenário em que usamos este tipo de coisas é para alguns relatórios que temos, que tem que ter uma visão maior que somente uma entidade (apesar de todas estarem debaixo da mesma raiz de agregação). Neste caso, por exemplo, o "Relatório de Pedidos" tem uma visão que pode vir de diversos lugares do sistema, permitindo que você faça esse tipo de coisa mesmo que você só lide com a parte de logística... Mas nisso estamos sendo simplórios, poderíamos usar em outras partes do sistema, mas as vezes é difícil identificar!

2012/9/14 Fernando Mondo <fernando.mondo@gmail.com>
[Quoted text hidden]

[Quoted text hidden]

Fabício Cabral <fabriciofx@gmail.com>
To: dotnetarchitects@googlegroups.com

Sat, Sep 15, 2012 at 1:57 AM

+1

2012/9/14 Fernando Mondo <fernando.mondo@gmail.com>

Eu achei bem pertinente, e na verdade nunca vi nenhum Dominio assim...
[Quoted text hidden]

--
-fx

Fabício Cabral <fabriciofx@gmail.com>
To: dotnetarchitects@googlegroups.com

Sat, Sep 15, 2012 at 2:00 AM

Perguntinha bem ingênua: há uma classe com o mesmo nome mas seus membros, de acordo com o seu contexto, será diferente. Também haverá um repositório para cada uma destas classes?

[]'s

2012/9/14 Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>
[Quoted text hidden]

--
-fx

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Sat, Sep 15, 2012 at 2:24 AM

Sim.

A regra nunca foi um repositório por classe, mas um repositório por raiz de agregação... ou até mais de um repositório por raiz de agregação, por exemplo, vc pode ter o repositório de clientes , e o repositório de clientesInadimplentes. São ambos da mesma agregação... mas o segundo trata de casos mais específicos que o primeiro.

Atenciosamente,

Daniel Moreira Yokoyama.
[@dmyoko](https://twitter.com/dmyoko)
<http://twitter.com/dmyoko>

Blogs:
Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]
[Quoted text hidden]

edmilson hora <edmilson_hora@yahoo.com.br>
Reply-To: dotnetarchitects@googlegroups.com
To: "dotnetarchitects@googlegroups.com" <dotnetarchitects@googlegroups.com>

Sat, Sep 15, 2012 at 10:07 AM

Como é que ficariam estas classes quanto a persitencia? Como seria modelado o Repositório ou Repositórios?

De: Fernando Mondo <fernando.mondo@gmail.com>

Para: dotnetarchitects@googlegroups.com

Enviadas: Sexta-feira, 14 de Setembro de 2012 16:14

Assunto: Re: [dotnetarchitects] Diferentes classes de dominio em diferentes namespaces

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Sat, Sep 15, 2012 at 10:43 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

A persistência não influencia no seu modelo. Se você usa RDBMS (que já terrível sem essa abordagem, não fica mais grotesco do que já é). Ou seja, você já tem que distribuir seus dados em tabelas naturalmente, você só vai ter que continuar fazendo isso (da forma como quiser).

Se quiser que o Pedido de Logística e o Pedido de Vendas sejam persistidos na mesma tabela, ou que haja uma tabela de genérica e outras contextualizadas, ou que o banco trate as duas coisas como totalmente diferentes, e o código controle esse binding sozinho... enfim... como eu disse, seu modelo de persistencia não precisa ser influenciado pelo seu modelo de domínio.

Atenciosamente,

Daniel Moreira Yokoyama.

@dmyoko

<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)

<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

edmilson hora <edmilson_hora@yahoo.com.br>

Sat, Sep 15, 2012 at 11:19 AM

Reply-To: dotnetarchitects@googlegroups.com

To: "dotnetarchitects@googlegroups.com" <dotnetarchitects@googlegroups.com>

Sinceramente, li o artigo e não concordo com as colocações do autor.

Tenho visto nos últimos anos vários artigos como este que mais confundem quem esta aprendendo do que informam sobre o que seria realmente uma Boa Prática. Pois, muita gente que esta começando se depara com isso e acha que deve seguir a indicação. Tomem cuidado!!.

1 -) "Apenas três departamentos da empresa foram (superficialmente) analisados e já ficou claro que essa classe **Pedido** extrapola o número de responsabilidades recomendado pelos *SOLID Principles*: uma."

De quais responsabilidades ele esta falando? O que esta sendo montada é a estrutura de dados

da classe, nenhuma responsabilidade esta sendo definida, o que é que esta classe esta fazendo a não ser armazenar as informações pertinentes a ela?

O Autor esta confundindo tudo. Quem sabe o que é SOLID, sabe que ele esta se referendo apenas ao "S - Principio da Responsabilidade Unica", a responsabilidade de uma classe não é definida pela quantidade de atributos nem mesmo pela quantidade de metodos, o cara ta induzindo o leitor a conceitos errados.

*2 -) "Isso acontece porque a classe **Pedido** como foi criada pertence a vários domínios."*

Pode até ser que dependendo do Negócio, seja necessário ter classes com o mesmo nome em "domínios" diferentes, mas em minha opinião não é o caso do exemplo do artigo. Eu não faria o que o autor indica pois não há ganho em nada, muito pelo contrário dificulta muita coisa.

=====

O que seria em minha opinião uma abordagem válida, seria passar O pedido ou melhor uma Abstração do mesmo com as suas informações aos serviços que tratariam as informações que competem:

Ou seja:

Classe Pedido

```
{  
  //Todos os atributos necessários.  
}
```

Classe EstoqueServicos

```
{  
  public void Processo(Pedido pedido)  
  {  
    // Aqui são processados os dados do pedido  
    //para ajustar Estoque.  
  }  
}
```

Classe EntregasServicos

```
{  
  public void Processo(Pedido pedido)  
  {  
    // Aqui são processados os dados do pedido  
    //para efetuar a entrega.  
  }  
}
```

Classe FinanceiroServicos

```
{  
  public void Processo(Pedido pedido)
```

```
{  
    // Aqui são processados os dados do pedido  
    //para efetuar a cobrança.  
}  
  
}
```

De: Fabrício Cabral <fabriciofx@gmail.com>

Para: dotnetarchitects@googlegroups.com

Enviadas: Sábado, 15 de Setembro de 2012 1:57

Assunto: Re: [dotnetarchitects] Diferentes classes de dominio em diferentes namespaces

[Quoted text hidden]

[Quoted text hidden]

Winston Pacheco Junior <winston.pacheco@gmail.com>

Sat, Sep 15, 2012 at 3:12 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Cara, eu acho que você tá entendendo isso errado.

Bounded Contexts existem e eu acho que é uma coisa legal quando o projeto é grande, tem vários times e "negócios", cada contexto vai ter a sua visão do dominio e não vai influenciar a visão do dominio do outro contexto. Com isso é possível dividir trabalho, responsabilidade, manutenibilidade e ajuda o dominio a comunicar melhor o que ele representa. Pra mim é simples assim.

Seus exemplos, na minha opinião, não tem nada a ver com o que a pessoa quis mostrar no post.

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Sat, Sep 15, 2012 at 3:51 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Edmilson, na boa... misturar tudo quanto é informação (relevante ou não) num único modelo e espalhar responsabilidades distintas em serviços... isso, sim, é pura programação procedural.

DDD e a idéia de modelos contextuais são uma forma de simplificar o modelo distribuindo complexidades inerentes a contextos separados para que tais complexidades não prejudiquem a comunicação e manutenção do modelo em pontos não relativos.

Atenciosamente,

Daniel Moreira Yokoyama.

@dmyoko

<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)

<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

edmilson hora <edmilson_hora@yahoo.com.br>

Sat, Sep 15, 2012 at 5:21 PM

Reply-To: dotnetarchitects@googlegroups.com

To: "dotnetarchitects@googlegroups.com" <dotnetarchitects@googlegroups.com>

@Daniel,

utilizar o conceito de serviços é programação procedural?

Você leu o artigo?

Você notou que o autor confunde conceitos ?

No meu entendimento, simplesmente separar atributos da classe Pedido em namespaces diferentes, não facilita nada e não agrega valor.

De: Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Para: dotnetarchitects@googlegroups.com

Enviadas: Sábado, 15 de Setembro de 2012 15:51

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Sat, Sep 15, 2012 at 5:44 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Edmilson, sua crítica tá muito apoiada no modelo de exemplo, que é falha de todos nós quando queremos ilustrar algo. O modelo é meramente ilustrativo. Quando alguém tentar te explicar algo através de exemplos, use os exemplos somente pra ilustrar o que ele está dizendo, não pra criticar o modelo de exemplo como sugestão de solução.

Eu realmente vejo separações claras entre contextos do pedido. Mesmo quando a separação não envolve comportamento, o estado também faz parte da estrutura do objeto, até por que, volto a dizer, a questão aqui é principalmente sobre o contexto do modelo (e não OOP em si).

O conceito de serviços sempre foi procedural. DDD não é apegado a purismos, é um modelo dirigido pelo domínio, não pelo paradigma, ou pela linguagem, ou pela plataforma. No entanto, mesmo sem apêgo ao paradigma de objetos, e ainda que se defina bons motivos para usar abordagens procedurais (com domain services, factories, etc), a idéia não é distribuir tudo em serviços, afinal, DDD se trata de Domain Driven... não Procedure Driven.

Atenciosamente,

Daniel Moreira Yokoyama.

@dmyoko

<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)

<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Priscila Mayumi Sato <mayumisatox@gmail.com>

Sat, Sep 15, 2012 at 6:08 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Bem, eu vou comer pipoca enquanto leio a discussão. Obrigada a todos.

Eu fazia algo parecido como o Edmilson, não igual. Continuem explicando seus pontos de vista, está me ajudando e muito :*

[Quoted text hidden]

[Quoted text hidden]

Rafael Mueller <muel182@gmail.com>

Wed, Sep 19, 2012 at 9:42 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

No projeto que trabalho atualmente, nós temos essa divisão, contudo não usamos namespaces, usamos projetos e solutions separadas.

Usando o modelo do post, nós temos uma classe Pedido em um projeto estoque, financeiro, entregas... e por projeto, eu me refiro a um conjunto de projetos, o domínio, projetos webs, banco de dados, infra, tudo separado.

Para comunicação entre esses serviços/bounded contexts, usamos nservicesbus e em alguns poucos casos onde request/response é necessário, temos uma interface http exposta.

O projeto não é pequeno, e eu particularmente gosto dessa divisão, permite trabalhar em vários projetos menores e independentes.

Grato,

—

Rafael Mueller

2012/9/14 Fernando Mondo <fernando.mondo@gmail.com>

Eu achei bem pertinente, e na verdade nunca vi nenhum Dominio assim...

[Quoted text hidden]

[Quoted text hidden]