

Domain Driven Design - baseado na peça de Eric
Evans

by
Daniel Cukier

ELENCO

Alexandre	<i>Líder técnico do time do Sistema de Reservas - Hollo</i>
Sérgio	<i>Consultor de Domain Driven Design - Daniel</i>
Roberto	<i>Responsável pelo serviço de Rotas - Saroka</i>
Narrador	<i>Pode ser o mesmo ator que faz o Sérgio - Daniel</i>
Cristiano	<i>Líder técnico do time de thank APIs - Ripoll</i>
Leandro	<i>Líder do time de Ordem de Trabalho - Soejima</i>
Patrícia	<i>Líder do time de Work FLOW - Lúcia</i>

Setting: São Paulo
Time: Março/2008

ATO I

Cena 1

[Reservas e Rotas - diálogo Sérgio, Alexandre e Roberto]

ALEXANDRE

Bom te encontrar, Sérgio! Eu sei que você já falou com o Roberto, do time de Rotas. Eu arrastei o Roberto comigo porque nossos sistemas são tão integrados, que eu pensei que faria mais sentido virmos juntos falar com você.

SÉRGIO

Bom te encontrar também, Alê! Para começar, você poderia me dar um mapa geral sobre o seu sistema de Reservas?

ALEXANDRE

Nós cuidamos de tudo relacionado com as cargas, o que é, onde está, e para onde está indo. O time de Reservas está no meio de quase todos os projetos da Super Entregas. Nós temos dez pessoas no nosso time, somos um dos maiores times. Nosso sistema não deveria parecer complicado, mas pelo fato de estarmos no centro de tudo, precisa ser sólido e nós gastamos bastante tempo fazendo testes automatizados. Escrevemos testes antes, na maior parte das vezes.

Quando agendamos uma Carga, chamamos o Serviço de Rotas e passamos todos os critérios do carregamento. O serviço responde com um Itinerário que satisfaça nossos requisitos. Esse é o departamento do Roberto. [Aponta para Roberto].

ROBERTO

Na verdade, trabalhamos junto com o time do Alexandre para definir a interface do Serviço de Rotas e o processo de conversão entre os dois sistemas.

SÉRGIO

O time de Reservas parece ser bem agressivo em relação a testes automatizados. Vocês tem um processo parecido?

ROBERTO

Nós temos alguns testes. [Movimento de se eximir] Não prestamos muita atenção sobre isso, porque temos revisões de design e revisões de código. Garantimos alta qualidade dessa maneira. Provavelmente metade dos nossos testes são do time de Reservas (em cima da nossa interface com eles).

ALEXANDRE

Nós temos uma boa especificação para a API do Serviço de Rotas e muitos testes para ela. O time de Rotas roda esses testes antes de nos passar uma nova versão do serviço. E nós rodamos os testes toda vez que fazemos checkin no código, claro!

ROBERTO

Sim, isso tem sido muito útil para nós. Nós nos certificamos de que todos os testes passam antes de lançar qualquer versão. Já encontramos alguns problemas usando isso, mas não muitos, porque somos muito cuidadosos. Sabemos que nosso sistema é crítico e pensamos bastante antes de fazer qualquer mudança.

Nós também escrevemos testes toda vez que encontramos algum bug. Acho que temos também alguns testes da estrutura básica de rede. Mas, em geral, preferimos fazer um sistema de alta-qualidade através de um design de qualidade ao invés de consertar quando acontece alguma coisa.

O sistema tem lá suas gambiarras, mas é elegante em alguns pontos. E funciona bem!

Cena 2

SÉRGIO

Então, parece que Reservas depende de Rotas

ALEXANDRE E ROBERTO

Sim.

SÉRGIO

Mas Rotas não depende de Reservas

ROBERTO

Exato. Reservas chama o Serviço de Rotas. O design de Rotas não tem referência a nenhum objeto em Reservas

SÉRGIO

Então você pode entregar versões sem se preocupar com Reservas?

ROBERTO

Ah, não. Quer dizer, sim, poderíamos algumas vezes lançar uma versão incremental sem eles, mas se for algo significativo, temos que coordenar nossas entregas. Poderíamos lançar uma funcionalidade nova sozinhos, mas ela não seria usada até que Reservas adaptasse a sua parte.

ALEXANDRE

E, claro, qualquer mudança na API precisaria ser sincronizada entre os dois

times.

Cena 3

SÉRGIO

Continuando, agora consigo ver como seu sistema de Reservas usa o sistema de Rotas. Você poderia me contar mais sobre outros times que provêm serviços para você?

ALEXANDRE

Na verdade não. Eu acho que somos o topo da cadeia alimentar!

SÉRGIO

Mas você disse que Reservas estava no centro das coisas. Com quais outros grupos você interage e por que?

ALEXANDRE

Bem, existem as APIs externas – tem um time que trabalha nelas. O líder do time de API é o Cristiano. Talvez você possa falar com ele. Eles usam o nosso sistema. De vez em quando, o time de API pede alguma nova funcionalidade para nós, para que eles possam oferecer uma interface para os clientes externos. Nós tentamos fazer tudo que eles precisam. Algumas vezes aperfeiçoamos nosso projeto para atendê-los, mas outras apenas mostramos como o nosso modelo funciona e como esse modelo pode resolver as questões deles.

Cena 4

SÉRGIO

Parece que as coisas estão bem tranquilas para você. Mas tenho certeza que você tem alguns problemas.

ALEXANDRE

Opa! Toneladas! Você poderia ser mais específico?

SÉRGIO

Qual você diria que é o problema mais cabeludo que você está passando hoje?

ROBERTO

Você tinha falado alguma coisa sobre uns bugs nas Ações de Itinerário?

ALEXANDRE

Bem, OK. Uma questão curiosa é o surgimento recente de alguns bugs nesses objetos das Ações de Itinerários. Eles eram confiáveis 6 meses atrás e, como eu disse, nós não mexemos muito nessa parte ultimamente.

SÉRGIO

Esse são bugs que você encontrou durante o desenvolvimento? Os testes pararam de funcionar?

ALEXANDRE

[*Engole seco, constrangido*] Não. Os clientes que encontraram os bugs. Eles passaram despercebidos pelos nossos testes.

Tivemos que alocar um desenvolvedor para fazer uma análise, escrever os casos de teste e então arrumar os bugs. Esse é um outro jeito de garantir que nossos testes estão cada vez melhor.

Só é estranho que tenhamos tantos clientes reclamando agora e não tínhamos isso no ano passado. Eles devem estar usando o sistema de algum jeito novo

NARRADOR

Bugs misteriosos num código estável é um típico sinal de fuga da delimitação de contextos.

Cena 5

NARRADOR

O próximo encontro é com Cristiano, o líder do time que desenvolve o “Pequena-Transportadora API”, que é usada para permitir que pequenas empresas de transporte integrem com a Super Entregas.

A maioria dos pedidos chegam através da interface web da própria Super Entregas, usada pelo pessoal de vendas para fazer pedidos de uma nova Carga, mas a empresa também transporta cargas de outras empresas de entregas, especialmente as pequenas.

Eles integram frequentemente com o sistema da “Super Entregas”, assim eles podem agendar para entregas para seus próprios clientes, injetando a carga no sistema da “Super Entregas”.

CRISTIANO

Somos um time pequeno. Nós tocamos o projeto de API desde que ele começou, há dois anos. Naquela época, nós fizemos algumas integrações personalizadas para alguns clientes grandes, mas tínhamos um limite de quantas dessas customizações conseguiríamos criar e manter.

O Marketing identificou uma oportunidade. Nós entregamos cargas para um monte de pequenas empresas de entrega, que não têm infra-estrutura própria. A idéia era prover um jeito leve e padrão de integrar os sistemas. Nós fizemos uma interface SOAP para o sistema de Reservas. SOAP não é muito rápido, mas é o suficiente para nossas necessidades. Só agora, recentemente,

estamos com tráfego grande o suficiente para precisar de um segundo servidor.

SÉRGIO

Então o seu sistema é essencialmente uma camada fina sobre Reservas e Roteamento?

CRISTIANO

Tem mais umas coisas, mas essencialmente é isso. Nosso primeiro release tinha só o básico do básico, mas há seis meses lançamos a versão 2.0 e ela já é bem completa. Na verdade, pelo menos um dos nossos velhos clientes está mudando para usar a API, porque acham que será mais fácil para eles. Nós ainda chamamos de “Pequena-Transportadora API”, mas pode ser usada por qualquer um que queira integrar.

Cena 6

SÉRGIO

O time de Reservas descobriu mais bugs nos últimos 6 meses, e sua nova versão está rodando há 6 meses. Será que tem alguma relação? Talvez o seu sistema esteja usando o sistema de Reservas de uma forma nova? Talvez vocês estejam mudando o sistema deles?

CRISTIANO

[Enfaticamente] Nós *não* estamos mudando o sistema deles. Nós nunca mudamos o código deles. Nós sempre tomamos bastante cuidado para fazer mudança apenas em nossos sistemas e só fazer chamadas ao código deles.

xxxx. No primeiro ano desse projeto, pedimos algumas mudanças no modelo de Itinerário e no Serviço de Rotas que tornariam as coisas muito mais fáceis para nós ou tornariam a interface para os nossos clientes externos. Por exemplo, nós queríamos quebrar o Itinerário em “paradas”, como uma alternativa a visão de “pernas” que eles usam nos sistemas internos deles. Isso parece muito mais natural e útil para nossos clientes. Então pedimos para eles incorporarem no módulo de Itinerário.

Bem, como time de Reservas nunca nos atendeu, nós desistimos da idéia. Nós tentamos oferecer a interface que queríamos para nosso cliente, mas a coisa ia ficar tão complicada que pensamos que seria inviável manter.

Então nos adequamos o máximo possível ao modelo do time de Reservas, com poucas melhorias, como derivar “paradas” de “pernas”. A API não é a ideal, mas é boa o suficiente. E dá prá usar, no fim.

[discussão]

NARRADOR

Ainda temos o mistério dos bugs nas Ações de Itinerário. O release de seis meses atrás deve ser só uma coincidência. Esses bugs inexplicáveis são um indício típico de problemas no relacionamento entre contextos, mas não parece que seja sobre esse relacionamento entre Reservas e API.

ATO II

NARRADOR

Vamos falar com Leandro, do time de Ordens de Serviço. Esse time escreve software para o departamento de operações.

NARRADOR

[*Explica o sistema de Ordens de Serviço*] O sistema de Ordens de Serviço pega as ações de itinerário para uma carga e as transforma em algo mais concreto: Instruções de Carregamento. Vejam o UML:

Cena 1

LEANDRO

Resumindo, nós pegamos o Itinerário para cada Carga. Nós acompanhamos a Carga, em cada passo. Nós olhamos no Itinerário para ver o que deve acontecer em seguida. Então, mandamos pedidos para as companhias portuárias para realmente fazer essas coisas. Descarregar essa carga daquele navio, carregar a carga naquele trem, etc.

SÉRGIO

Então, eu entendo que Eventos de Manuseamento são registros do que realmente acontece no porto?

LEANDRO

Isso. Somos responsáveis pela entrada dos Eventos de Manuseamento no sistema. Comparamos esses Eventos com as Instruções de Carga para ter certeza que tudo procedeu de acordo.

SÉRGIO

E vocês dependem do sistema de reservas.

LEANDRO

Sim, nós pegamos o Itinerário de lá e então usamos suas Ações para derivar Instruções de Carga mais concretas e detalhadas. Então passamos elas para o serviço de Ordem de Trabalho.

SÉRGIO

O serviço de Ordem de Trabalho é todo um subsistema a parte?

LEANDRO

Exato. O serviço é na verdade uma fachada para o sistema legado de Ordem de Trabalho. Nossa maior dor de cabeça é integrar com esse legado. Nós precisamos disso porque é o legado que tem realmente a funcionalidade de emitir ordens para as companhias portuárias. Com isso nós só temos que descobrir qual ordem emitir e então pedir para o legado fazer isso.

Então precisamos ter essa capacidade, mas o legado é realmente uma zona. Quando começamos nosso projeto há 4 anos, tentamos refatorar o código legado para torná-lo razoável, mas tivemos um monte de efeitos colaterais em várias partes do sistema. Então evitamos esse código como vampiro corre de alho. Não mudamos ele a não ser que seja extremamente necessário. Temos um código que traduz coisas nossas para o legado e coisas do legado para nosso sistema. A tradução está ficando cada vez mais complicada, mas achamos que vale a pena pagar o preço, por de outra forma não conseguiríamos fazer *nada*.

NARRADOR

Esse grupo gerencia 2 contextos. Identifique. Eles fizeram uma camada anti-corrupção. Onde está ela? Qual é a relação com Reservas? Para isso vamos fazer mais algumas perguntas.

SÉRGIO

Então, Leandro, existe uma interface de serviço que você usa para interagir com Reservas?

LEANDRO

Basicamente, a interface que usamos é o repositório. Nós apenas chamamos o repositório para obter os objetos de Itinerário.

SÉRGIO

E eles avisam você quando mudam o sistema de Itinerário?

LEANDRO

O projeto básico de Itinerário está bem estável no último ano, logo isso não é um problema. Nós tivemos umas confusões durante o desenvolvimento daquela parte, quando estávamos tentando fazer o Itinerário funcionar para os dois times. Eles nos pediram para fazer uma suite de testes que eles pudessem rodar e verificar que suas mudanças não impactariam na gente, mas nunca conseguimos fazer isso funcionar direito. Mas agora não temos mais problemas.

SÉRGIO

Qual é sua relação com o grupo de Reservas?

LEANDRO

Anos atrás, os dois grupos trabalhavam juntos. Quando “Super Entregas” começou a crescer, dividimos o grupo e eu me tornei líder técnico do time de Ordem de Trabalho. Alexandre se tornou líder do time de Reservas na mesma época.

SÉRGIO

E como vocês interagem agora?

LEANDRO

Nós levamos muito bem. Alexandre e eu sempre almoçamos às sextas para garantir que não tenha nada mal-resolvido entre os times.

SÉRGIO

Então os líderes do times têm boa comunicação. Como é a interação em decisões de design do dia-a-dia? Quero dizer, quando seu grupo precisa de uma mudança no código de Reservas, você tem que fazer uma requisição formal, ou isso é tocado informalmente entre os desenvolvedores? Isso vai diretamente de um desenvolvedor do seu time para outro do outro time, ou as pessoas do seu grupo precisam conversar com você antes e você passa isso adiante ou levanta a questão numa sessão de planejamento?

LEANDRO

Ah, entendo. Nós trabalhamos muito próximos do time de Reservas. Nós conhecemos o modelo de objetos, o código é bem fácil de entender. Os testes funcionam bem, o que torna fácil fazer qualquer mudança no sistema deles que nós precisamos. Nós temos que fazer umas mudanças lá de vez em quando.

SÉRGIO

Que tipo de mudança vocês fazem no código de Reservas?

LEANDRO

Por exemplo, uma das coisas que estamos brigando ultimamente é com a relação entre Ações e Instruções de Carga. São muito parecidos. Nem toda Ação é uma Instrução de Carga. Existem algumas informações no domínio de reservas que nós não precisamos, e outras coisas que precisamos que não estão no domínio de reservas. Então, ambas são necessárias. Nós estivemos clareando o modelo, e refatorando gradualmente para torná-los mais distintos. *[discussão: núcleo compartilhado e integração contínua]*

NARRADOR

Tô sentindo cheiro de problema aqui. Eles compartilham código, que poderia ser um “núcleo compartilhado”, mas o processo de integração na me parece

forte o suficiente para manter compartilhados os conceitos entre os dois times, nem os limites de cada “Contexto Delimitado” me parecem claros. O que é exatamente o processo de integração entre esse dois grupos? Vamos conferir.

Cena 2

SÉRGIO

Alexandre, você sabia que eles mudavam o seu código?

ALEXANDRE

Se eles precisarem de alguma mudança no nosso sistema eles simplesmente fazem. Por isso que a gente tem testes. Eles não precisam perguntar para nós. O escritório deles é no outro prédio, então fica mais difícil de perguntar. É melhor que eles façam por eles mesmos e não nos atrapalhem, uma vez que temos sempre muita coisa para fazer.

LEANDRO

Seria uma dor de cabeça se tivéssemos que ter a aprovação de alguém em Reservas. Nós podemos refatorar, rodar os testes para garantir que nada quebrou e então fazer commit do código. Nós sempre mandamos um email para que eles saibam que fizemos mudanças e eles olham nosso código se quiserem. Está tudo no Git.

NARRADOR

Aha! Estamos felizes em encontrar a *raiz* dos bugs nas Ações de Itinerários. Discussão: comparar o processo de Ordens de Trabalho com o padrão Núcleo Compartilhado. Como adequá-lo? Definindo um núcleo compartilhado de verdade

- Fronteiras de contexto explícitas para o núcleo
- Conceitos de Integração Contínua, não só o código
- Testes não garantem integração

Outras opções

- Unir os times
- Time de Ordem de Serviço virar Conformista ou Cliente/Fornecedor

Cena 3

NARRADOR

Vamos dar uma olhada nas outras relações.

LEANDRO

O único outro subsistema que nós usamos é o sistema de Workflow. Quando um evento que precisa de notificação ocorre, Flowmatic (que é a engine de Workflow) pega esse evento e descobre quem deve ser notificado e envia a notificação. É bem legal e fácil de configurar. O time da Paty cuida de todo trabalho do Flowmatic.

SÉRGIO

Sim, acho que eu devia falar com a Paty, mas até agora ela não respondeu minhas solicitações de reunião.

LEANDRO

Não meu surpreende. É difícil encontrar essas pessoas às vezes. Elas não estão oficialmente no departamento de desenvolvimento de software. Elas se reportam ao David. Mas Paty trabalha bem. Uma vez que você consiga a atenção deles, eles podem te ajudar.

Na verdade, eles estão cuidando de uma parte crítica do nosso próximo release. Quando uma carga é extraviada, eles não vão apenas notificar as pessoas. Eles devem chamar de volta o serviço de rotas, gravar um novo Itinerário em algum lugar e então enviar para alguém aprovar. E no fim eles anexam o Itinerário à Carga. Muito legal!

Cena 4

NARRADOR

É hora de falar com o time de WorkFlow

PATRÍCIA

Nosso time é pró em usar o Flowmatic. É um sistema completo de WorkFlow. Nós automatizamos um monte de tarefas administrativas do dia-a-dia. Você sabe. Coisas como encaminhamento de ordens de compra para aprovação. Mas imagino que você queira falar sobre o nosso envolvimento no software de acompanhamento de cargas.

SÉRGIO

Esse é meu interesse *principal*.

PATRÍCIA

Toda vez que alguém precisa notificar alguém, nós estamos envolvidos. Por exemplo, quando uma Carga chega, nós notificamos o cliente. Quando uma Carga é extraviada, notificamos o gerente de operações. Se o gerente não

pode responder, notificamos o subgerente no comando. O time de Ordem de Trabalho só chama a API que a gente disponibiliza e a gente faz o resto. Nós conhecemos o contato principal e o de backup e se eles devem ser contatados por email ou celular. Isso torna as coisas mais fáceis para os outros times.

SÉRGIO

Então é bem focado em notificações. Vocês também coordenam outros programas?

PATRÍCIA

Sim, fazemos isso, nos nossos fluxos de administração. Na verdade, a maioria dessas funcionalidades novas de acompanhamento de carga envolvem algum tipo de coordenação. Por exemplo, quando uma carga é extraviada, não vamos apenas notificar o gerente de operações, como fazemos agora. Vamos chamar o Serviço de Rotas e achar um novo Itinerário. Quando o Serviço de Rotas não consegue encontrar um Itinerário viável, então ele ativa um workflow para lidar com uma Carga não-roteável.

NARRADOR

Quantos contextos o time de workflow gerencia? Quais os modelos e suas relações? **Caminhos Separados**

SÉRGIO

Me parece um jeito legal de lidar com cargas extraviadas. Então, vocês têm uma API genérica que eles usam para notificar algo como extravio de carga ou quando surge uma carga não-roteável, ou vocês têm que estender sua API?

PATRÍCIA

Flowmatic é muito bom para conectar outros sistemas. Tem uma linguagem de script que usamos para chamar os sistemas deles. Tem uma fila onde eles inserem eventos. Nós pegamos esses eventos e iniciamos o workflow apropriado.

É assim que as funcionalidades atuais funcionam. Na verdade, eles pediram algumas funcionalidades novas que precisariam de muita mais integração. Nós ainda não trabalhamos muito nelas, mas já gastamos um bom tempo em reuniões e no telefone, trabalhando com o grupo de Ordem de Trabalho do Leandro, conversando sobre APIs. Os workflows podem ser bem simples, se conseguirmos descobrir como manipular os objetos deles.

SÉRGIO

Você parece desconfortável com essa situação.

PATRÍCIA

Eu não percebi, quando começamos com isso, que nós teríamos que aprender

como todo o resto funciona. Mas isso vai tomar algum tempo de um desenvolver dedicado período integral. Nosso quadro já está bem sobrecarregado com os fluxos administrativos. Não sei quando a gente vai conseguir fazer isso. Nenhum de nós entende o sistema deles o suficiente para tentar alguma coisa hoje.

NARRADOR

Expetativas irreais por parte do time de Ordens de Trabalho. Como consertar?

- Estabelecer uma relação de Cliente/Fornecedor de verdade
- Caminhos Separados?