
Aprenda Domain-Driven Design

*Alinhando Arquitetura de Software e
Estratégia de Negócios*

Vlad Khononov

AMOSTRA



ALTA BOOKS

GRUPO EDITORIAL

Rio de Janeiro, 2024

Sumário

Apresentação	ix
Prefácio	xi
Introdução	xix
1. Analisando os Domínios de Negócios	3
O que É Domínio de Negócio?	3
O que É Subdomínio?	4
Exemplos de Análise de Domínio	14
Quem São os Especialistas de Domínio?	18
Conclusão	18
Exercícios	19
2. Descobrimo o Conhecimento de Domínio	21
Problemas de Negócio	21
A Descoberta de Conhecimento	22
Comunicação	22
O que é Linguagem Ubíqua?	24
Linguagem do Negócio	25
Modelo de Domínio de Negócio	27
Conclusão	31
Exercícios	32
3. Administrando a Complexidade do Domínio	33
Modelos Inconsistentes	33
O que É Contexto Delimitado?	35
Contextos Delimitados Versus Subdomínios	38
Limites	41

Contextos Delimitados na Prática	42
Conclusão	46
Exercícios	47
4. Integrando Contextos Delimitados.....	49
Cooperação	50
Cliente-Fornecedor	53
Caminhos Separados	56
Mapa de Contexto	57
Conclusão	59
Exercícios	60
5. Implementando uma Lógica de Negócio Simples.....	63
Script de Transação	63
Registro Ativo	70
Seja Pragmático	72
Conclusão	72
Exercícios	73
6. Lidando com a Lógica de Negócio Complexa.....	75
História	75
Modelo de Domínio	76
Conclusão	96
Exercícios	97
7. Modelando a Dimensão do Tempo.....	99
Event Sourcing	99
Modelo de Domínio Orientado a Eventos	108
Perguntas Mais Frequentes	113
Conclusão	116
Exercícios	117
8. Padrões de Arquitetura.....	119
Lógica de Negócio Versus Padrões de Arquitetura	119
Arquitetura em Camadas	120
Portas e Adaptadores	127
Segregação de Responsabilidade de Comando e Consulta	130
Escopo	136
Conclusão	137
Exercícios	138

9. Padrões de Comunicação	139
Tradução do Modelo	139
Integrando Agregados	145
Conclusão	155
Exercícios	156
10. Heurística de Design	159
Heurística	159
Contextos Delimitados	160
Padrões de Implementação da Lógica de Negócio	161
Padrões de Arquitetura	163
Estratégia de Teste	164
Árvore de Decisão do Design Tático	166
Conclusão	167
Exercícios	168
11. Desenvolvendo as Decisões de Design	169
Mudanças nos Domínios	169
Preocupações do Design Estratégico	172
Preocupações de Design Tático	173
Mudanças Organizacionais	178
Conhecimento de Domínio	179
Crescimento	180
Conclusão	182
Exercícios	183
12. EventStorming.....	185
O que é EventStorming?	185
Quem Deve Participar do EventStorming?	185
O que É Necessário para o EventStorming?	186
O Processo do EventStorming	187
Variantes	195
Quando Utilizar o EventStorming	195
Dicas de Realização	196
Conclusão	198
Exercícios	198
13. Domain-Driven Design na Prática.....	199
Análise Estratégica	200
Estratégia de Modernização	202

Domain-Driven Design Pragmático	208
Vendendo o Domain-Driven Design	209
Conclusão	211
Exercícios	212
14. Microserviços	215
O que é Serviço?	215
O que é Microserviço?	216
Domain-Driven Design e Limites dos Microserviços	223
Compactando as Interfaces Públicas dos Microserviços	227
Conclusão	229
Exercícios	230
15. Arquitetura Orientada a Eventos	231
Arquitetura Orientada a Eventos	231
Eventos	232
Projetando a Integração Orientada a Eventos	238
Conclusão	243
Exercícios	244
16. Malha de Dados	245
Modelo de Dados Analíticos Versus Modelo de Dados Transacionais	245
Plataformas de Gerenciamento de Dados Analíticos	250
Malha de Dados	255
Conclusão	260
Exercícios	261
17. Palavras Finais	263
Problema	263
Solução	264
Implementação	265
Leitura Adicional	265
Conclusão	267
18. Aplicando o DDD: Um Estudo de Caso	269
Cinco Contextos Delimitados	269
Discussão	279
Conclusão	283
Respostas dos Exercícios	285
Referências	293
Índice	295

Analizando os Domínios de Negócios

Se você for como eu, ama escrever códigos: resolver problemas complexos, criar soluções elegantes e construir novos mundos, cuidadosamente elaborando suas regras, estruturas e comportamentos. Acredito que é isso que tenha chamado a sua atenção para o domain-driven design (DDD): você quer ser melhor no seu ofício. No entanto, este capítulo nada tem a ver com escrita de código. Nele, você aprenderá como as empresas funcionam: por que elas existem, que objetivos elas buscam e quais são suas estratégias para atingir esses objetivos.

Quando ensino este material nas minhas aulas de domain-driven design, muitos alunos realmente me perguntam: “Precisamos aprender isso? Estamos desenvolvendo software, não administrando empresas.” A resposta é um contundente “sim”. Para projetar e construir uma solução eficiente, é preciso entender o problema. O problema, no nosso contexto, é o sistema de software que precisamos construir. Para entender o problema, é preciso entender o contexto em que ele existe — a estratégia de negócio da organização e que valor ela busca ganhar com a construção do software.

Neste capítulo, você aprenderá as ferramentas de domain-driven design necessárias para analisar o domínio de negócios de uma empresa e sua estrutura: seus subdomínios principais, genéricos e de suporte. Esse material é a base para projetar software. Nos capítulos restantes, aprenderá de que maneiras diferentes esses conceitos afetam o design do software.

O que É Domínio de Negócio?

O domínio de negócio define qual é a principal área de atividade de uma empresa. Em termos gerais, é o serviço que a empresa oferece a seus clientes. Por exemplo:

- A FedEx oferece o serviço de entrega.
- A Starbucks é mais conhecida pelo seu café.
- O Walmart é um dos estabelecimentos varejistas mais famosos.

Uma empresa pode operar em vários domínios de negócios. A Amazon, por exemplo, fornece varejo e serviços de computação em nuvem. A Uber é uma empresa de compartilhamento de viagens que também fornece entrega de alimentos e serviços de compartilhamento de bicicletas.

É importante notar que as empresas podem mudar seus domínios de negócios com frequência. Um exemplo aceito disso é a Nokia, que, ao longo dos anos, operou em campos tão diversos quanto processamento de madeira, manufatura de borracha, telecomunicações e comunicação móvel.

O que É Subdomínio?

Para atingir os objetivos e as metas do seu domínio de negócio, uma empresa precisa funcionar em múltiplos *subdomínios*. Um subdomínio é uma área de atividade comercial aprimorada. Todos os subdomínios de uma empresa formam seu domínio de negócio: o serviço que ela oferece aos seus clientes. Implementar um único subdomínio não é suficiente para que uma empresa tenha sucesso; é apenas um bloco de construção no sistema global. Os subdomínios têm que interagir uns com os outros para alcançar os objetivos da empresa em seu domínio de negócio. A Starbucks, por exemplo, pode até ser mais reconhecida pelo café, mas construir uma rede de cafeterias de sucesso exige muito mais do que apenas fazer um ótimo café. Também é necessário comprar ou alugar imóveis em locais eficazes, contratar pessoal e administrar finanças, entre outras atividades. Nenhum desses subdomínios isoladamente tornará uma empresa lucrativa. Todos juntos são necessários para que ela possa competir em seus domínios de negócio.

Tipos de Subdomínios

Assim como um sistema de software compreende vários componentes arquitetônicos — bancos de dados, aplicações de front-end, serviços de back-end e outros —, os subdomínios suportam diferentes valores estratégicos e comerciais. O domain-driven design se distingue entre três tipos de subdomínios: subdomínio principal, subdomínio genérico e subdomínio de suporte. Vejamos como eles são diferentes do ponto de vista estratégico de uma empresa.

Subdomínios principais

Um *subdomínio principal* é aquilo que a empresa faz de forma diferente de seus concorrentes. Isso pode envolver criar novos produtos e serviços ou reduzir os custos com a otimização de processos já existentes.

Vejamos a Uber como exemplo. Inicialmente, a empresa fornecia uma forma de transporte nova: o compartilhamento de viagens. Conforme a concorrência a alcançou, a Uber encontrou formas de otimizar e evoluir seu subdomínio principal: um exemplo é a redução de custos ao combinar usuários que vão na mesma direção.

Os subdomínios principais da Uber afetam seus resultados. Dessa forma, a empresa se diferencia da concorrência. Essa é a estratégia da empresa para proporcionar um melhor serviço para seus clientes e/ou maximizar sua lucratividade. Para manter uma vantagem competitiva, os subdomínios principais envolvem invenções, otimizações inteligentes, conhecimento de negócios ou outro tipo de propriedade intelectual.

Considere outro exemplo: o algoritmo de ranqueamento do Google Search. Até o momento em que escrevo este livro, a plataforma de publicidade do Google é responsável pela maior parte de seu lucro. Entretanto, o Google Ads não é um subdomínio, mas sim um domínio de negócio à parte com subdomínios próprios, entre eles um serviço de computação em nuvem (Google Cloud Platform), ferramentas de produtividade e colaboração (Google Workspaces) e outras áreas nas quais a Alphabet, empresa-mãe do Google, opera. Mas e o Google Search e seu algoritmo de ranqueamento? Embora a ferramenta de busca não seja um serviço pago, ela serve como a maior plataforma de exibição para o Google Ads. Sua habilidade de proporcionar excelentes resultados de busca é o que move o tráfego e, subsequentemente, é um componente importante da plataforma Ads. Se a empresa entregar resultados de busca inferiores devido a um bug no algoritmo ou um concorrente criar um serviço de busca ainda melhor, isso prejudicará o faturamento do negócio publicitário. Então, para o Google, o algoritmo de ranqueamento é um subdomínio principal.

Complexidade. Um subdomínio principal que seja simples de implementar pode fornecer apenas uma vantagem competitiva de curta duração. Portanto, os subdomínios principais são naturalmente complexos. Continuando com o exemplo da Uber, a empresa não só criou um novo mercado com o compartilhamento de viagens, mas também rompeu com uma arquitetura monolítica de décadas, a indústria do táxi, através do uso direcionado da tecnologia. Ao compreender seu domínio de negócio, a Uber conseguiu projetar um método de transporte mais confiável e transparente. É essencial que haja barreiras de entrada elevadas para o negócio principal de uma empresa; deve ser difícil para seus concorrentes copiar ou imitar a solução dela.

Fontes de vantagem competitiva. É importante notar que os subdomínios principais não são necessariamente técnicos. Nem todos os problemas de negócios são resolvidos com algoritmos ou outras soluções técnicas. A vantagem competitiva de uma empresa pode vir de várias fontes.

Considere, por exemplo, uma joalheria que vende seus produtos online. A loja virtual é importante, mas não é um subdomínio principal. O design das joias é. A empresa pode usar um modelo de loja virtual pronto para uso, mas não pode terceirizar o design de suas joias. O design é a razão pela qual os clientes compram os produtos da joalheria e se lembram da marca.

Como exemplo mais complexo, imagine uma empresa especializada em detectar fraudes *manualmente*. A empresa treina seus analistas para verificar documentos questio-

náveis e sinalizar casos de fraude em potencial. Você está construindo o sistema de software com o qual os analistas trabalham. Seria esse sistema um subdomínio principal? Não. O subdomínio principal é o trabalho que os analistas fazem. O sistema que você está construindo não tem nenhuma relação com a análise de fraude, ele apenas exibe os documentos e rastreia os comentários dos analistas.

Subdomínio Principal Versus Domínio Principal

Subdomínios principais também são chamados de domínios principais. Por exemplo, no livro de domain-driven design original, Eric Evans usa “subdomínio principal” e “domínio principal” como sinônimos. Embora o termo “domínio principal” seja usado com frequência, eu prefiro usar “subdomínio principal” por diversas razões. Em primeiro lugar, porque é um *subdomínio*, e prefiro evitar confusão com *domínios de negócios*. Em segundo lugar, você aprenderá no Capítulo 11 que não é incomum que os subdomínios evoluam com o passar do tempo e mudem de tipo. Por exemplo, o subdomínio principal pode se tornar um subdomínio genérico. Por isso, dizer que “um subdomínio *genérico* se tornou um subdomínio *principal*” é mais simples do que dizer “um *subdomínio* genérico se tornou um *domínio* principal”.

Subdomínios genéricos

Subdomínios genéricos são atividades comerciais que todas as empresas realizam da mesma forma. Como os subdomínios principais, os subdomínios genéricos geralmente são complexos e difíceis de implementar. No entanto, os subdomínios genéricos não oferecem nenhuma margem competitiva para a empresa. Não há necessidade de inovação ou otimização aqui: implementações testadas em combate estão amplamente disponíveis, e todas as empresas as utilizam.

Por exemplo, a maioria dos sistemas precisa autenticar e autorizar seus usuários. Em vez de criar um mecanismo de autenticação próprio, faz mais sentido usar uma solução já existente. Tal solução provavelmente é mais confiável e segura, pois já foi testada por várias outras empresas que têm as mesmas necessidades.

Voltando ao exemplo da joalheria que vende seus produtos online, o design de joias é um subdomínio principal, mas a loja virtual é um subdomínio genérico. Usar a mesma plataforma virtual de vendas — a mesma solução genérica — de seus concorrentes não impactaria a vantagem competitiva da joalheria.

Subdomínios de suporte

Como o nome sugere, *subdomínios de suporte* dão suporte aos negócios da empresa. No entanto, ao contrário dos subdomínios principais, os subdomínios de suporte não fornecem vantagem competitiva alguma.

Considere, por exemplo, uma empresa de publicidade online cujos subdomínios principais incluam a correspondência de anúncios com os visitantes, otimizar a eficiência dos anúncios e minimizar o custo do espaço publicitário. Porém, para ter sucesso nessas áreas, a empresa precisa catalogar seu material criativo. A forma como ela armazena e faz a indexação de seus materiais criativos físicos, como banners e páginas de entrada, não causa impacto nos lucros. Não há nada para inventar ou otimizar nessa área. Por outro lado, o catálogo criativo é essencial para implementar os sistemas de gestão e serviço publicitário da empresa. Isso faz com que a solução de catalogação de conteúdo seja um dos subdomínios de suporte da empresa.

A característica singular dos subdomínios de suporte é a complexidade da lógica de negócio da solução. Subdomínios de suporte são simples. Sua lógica de negócio se assemelha à maioria das telas de entrada de dados e operações ETL (extrair, transformar, carregar — em inglês: extract, transform, load), ou seja, as chamadas interfaces CRUD (criar, ler, atualizar e excluir — em inglês: create, read, update e delete). Essas áreas de atividade não proporcionam nenhuma vantagem competitiva para a empresa, portanto não exigem barreiras de entrada elevadas.

Comparando Subdomínios

Agora que já temos uma melhor compreensão sobre os três tipos de subdomínios comerciais, iremos explorar suas diferenças adicionais e ver como afetam as decisões de design de software estratégico.

Vantagem competitiva

Apenas os subdomínios principais proporcionam vantagem competitiva a uma empresa. Os subdomínios principais são a estratégia da empresa para se diferenciar da concorrência.

Os subdomínios genéricos, por definição, não podem ser uma fonte de vantagem competitiva. Eles são soluções genéricas — as mesmas utilizadas pela empresa e por sua concorrência.

Os subdomínios de suporte têm barreiras de entrada baixas e também não proporcionam vantagem competitiva. Geralmente, a empresa não se importaria se sua concorrência copiasse esses subdomínios de suporte, pois isso não afetaria a competitividade da empresa. Pelo contrário, estrategicamente a empresa prefere que seus subdomínios de suporte sejam genéricos, soluções prontas, que eliminam a necessidade de projetar e construir sua implementação. Você aprenderá detalhadamente sobre tais casos de

subdomínios de suporte que se tornam subdomínios genéricos, assim como outras possíveis permutações, no Capítulo 11. Um caso de estudo real desse cenário será descrito no Apêndice A.

Quanto mais complexos são os problemas que uma empresa consegue enfrentar, maior é o valor de negócio que pode fornecer. Problemas complexos não se resumem a entregar serviços a seus consumidores. Um problema complexo pode ser, por exemplo, tornar a empresa mais otimizada e eficiente. Por exemplo, proporcionar o mesmo nível de serviço da concorrência, mas a um custo operacional mais baixo, também é uma vantagem competitiva.

Complexidade

De uma perspectiva mais técnica, é importante identificar os subdomínios de uma organização, pois os diferentes tipos de subdomínio têm níveis de complexidade variados. Ao projetar software, temos que escolher ferramentas e técnicas que acomodem a complexidade das exigências do negócio. Portanto, identificar subdomínios é essencial para projetar uma solução de software de boa qualidade.

A lógica dos subdomínios de suporte é simples. Eles são operações básicas de ETL e interfaces CRUD, e a lógica do negócio é óbvia. Frequentemente, nada mais é do que validar inputs ou converter dados de uma estrutura em outra.

Os subdomínios genéricos são muito mais complicados. Deve haver uma boa razão para que os outros já tenham investido tempo e esforço para resolver esses problemas. Essas soluções não são nem simples nem triviais. Considere, por exemplo, os algoritmos de criptografia ou os mecanismos de autenticação.

Da perspectiva do conhecimento disponível, os subdomínios genéricos são “desconhecidos conhecidos”. Eles são aquilo que você sabe que não sabe. Além disso, é um conhecimento prontamente disponível. Você pode utilizar tanto as melhores práticas aceitas pelo setor ou, se necessário, contratar um consultor especializado na área para ajudar a projetar uma solução personalizada.

Os subdomínios principais são complexos. Eles devem ser tão difíceis para a concorrência copiar quanto possível — a lucratividade da empresa depende disso. É por isso que, estrategicamente, as empresas procuram resolver problemas complexos em seus subdomínios principais.

Às vezes, pode ser desafiador diferenciar subdomínios principais e de suporte. A complexidade é um princípio norteador útil. Pergunte a si mesmo se o subdomínio em questão pode se transformar em um negócio paralelo. Alguém pagaria apenas por ele? Se sim, ele é um subdomínio principal. Um raciocínio similar pode ser aplicado para diferenciar subdomínios de suporte e genéricos: seria mais simples e barato obter uma

implementação própria, em vez de integrar uma implementação externa? Se sim, estamos falando de um subdomínio de suporte.

Do ponto de vista mais técnico, é importante identificar os subdomínios cuja complexidade afetarão o design do software. Como discutido anteriormente, um subdomínio principal não é necessariamente relacionado ao software. Outro princípio norteador útil para identificar os subdomínios principais relacionados ao software é avaliar a complexidade da lógica de negócio que você terá que modelar e implementar em código. A lógica de negócio se assemelha às interfaces CRUD para a entrada de dados? Você tem que implementar algoritmos complexos ou processos comerciais orquestrados por regras de negócio complexas e invariantes? O primeiro caso é um sinal de subdomínio de suporte, enquanto o segundo é um típico caso de subdomínio principal.

O quadro na Figura 1-1 representa a inter-relação entre os três tipos de subdomínio em termos de diferenciação de negócios e complexidade da lógica de negócio. A interseção entre os subdomínios genéricos e de suporte é uma área indefinida (esquerda inferior): ela pode ir em ambas as direções. Se uma solução genérica para a funcionalidade de um subdomínio de suporte existe, o tipo de subdomínio resultante depende de ser mais simples e/ou mais barato integrar a solução genérica do que implementar essa funcionalidade do zero.

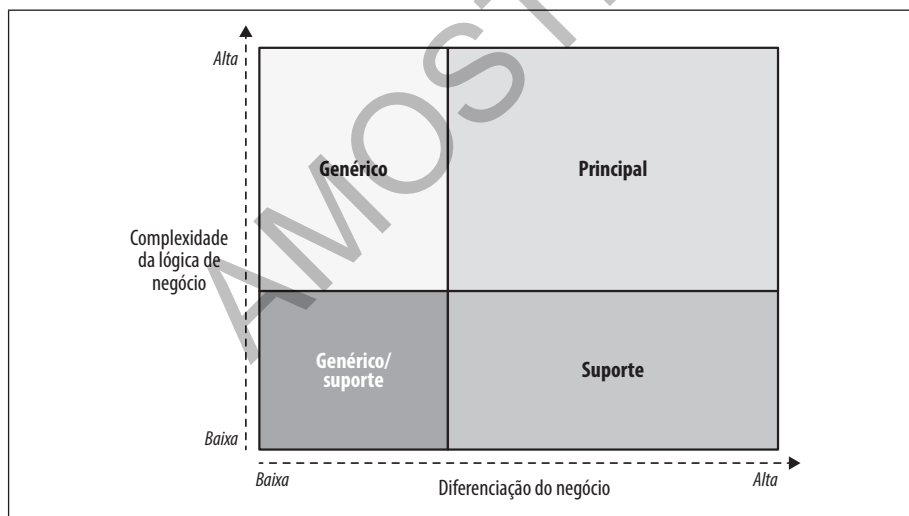


Figura 1-1. A diferenciação de negócios e a complexidade da lógica de negócio dos três tipos de subdomínio

Volatilidade

Como mencionado anteriormente, os subdomínios principais podem mudar com frequência. Se um problema pode ser resolvido na primeira tentativa, então provavelmente não é uma boa vantagem competitiva — a concorrência o alcançará rapidamente. Con-

sequestramente, as soluções para os subdomínios principais são emergentes. É preciso testar, refinar e otimizar implementações diferentes. Além disso, o trabalho nos subdomínios principais nunca acaba. As empresas continuamente inovam e evoluem esses subdomínios. As mudanças vêm na forma de novos recursos adicionados ou da otimização da funcionalidade existente. De qualquer forma, a evolução constante de seu subdomínio principal é essencial para que uma empresa fique à frente da concorrência.

Ao contrário dos subdomínios principais, os subdomínios de suporte não mudam com frequência. Eles não oferecem nenhuma vantagem competitiva, e, por isso, a evolução de um subdomínio de suporte oferece pouco valor de negócio se comparada ao mesmo esforço investido em um subdomínio principal.

Apesar de terem soluções existentes, os subdomínios genéricos podem mudar com o passar do tempo. Essas mudanças podem vir na forma de correções na segurança, correção de bugs ou soluções completamente novas para os problemas genéricos.

Estratégia da solução

Os subdomínios principais proporcionam à empresa a habilidade de competir com outros jogadores no setor. É uma responsabilidade crítica para o negócio, mas isso significa que os subdomínios genéricos e de suporte não são importantes? Claro que não. Todos os subdomínios são obrigatórios para que a empresa trabalhe em seu domínio de negócio. Os subdomínios são como blocos de construção básicos: se tiramos um bloco, toda a estrutura cai. Dito isso, podemos usar as propriedades inerentes dos diferentes tipos de domínio para escolher estratégias de implementação em que cada tipo de subdomínio seja implementado da maneira mais eficiente.

Os subdomínios principais precisam ser implementados internamente. Eles não podem ser comprados ou adotados; isso prejudicaria a noção de vantagem competitiva, pois os concorrentes da empresa poderiam fazer o mesmo.

Também seria imprudente terceirizar a implementação de um subdomínio principal. Ele é um investimento estratégico. Pegar atalhos com um subdomínio principal não é arriscado apenas no curto prazo, mas pode ter consequências fatais no longo prazo: por exemplo, bases de código de difícil manutenção que não dão suporte aos objetivos e às metas da empresa. O talento mais competente da empresa deve ser alocado para trabalhar em seus subdomínios principais. Além disso, a implementação de subdomínios principais internamente permite que a empresa faça mudanças e desenvolva a solução mais rapidamente, criando a vantagem competitiva em menos tempo.

Como as exigências dos subdomínios principais mudam continuamente e com frequência, a solução deve ser passível de manutenção e fácil de desenvolver. Assim, os subdomínios principais exigem a implementação das técnicas de engenharia mais avançadas.

Como os subdomínios genéricos são problemas difíceis, mas já resolvidos, é mais econômico adquirir um produto pronto para uso ou adotar uma solução de código aberto do que investir tempo e esforço para implementar um subdomínio genérico internamente.

A falta de vantagem competitiva faz com que evitar a implementação dos subdomínios de suporte internamente seja justificável. No entanto, diferente dos subdomínios genéricos, nenhuma solução pronta está disponível. Então, a empresa não tem outra escolha, exceto implementar os subdomínios ela mesma. Dito isso, a simplicidade da lógica de negócio e a infrequência de mudanças facilita a busca por atalhos.

Os subdomínios de suporte não exigem modelos de design elaborados ou outras técnicas de engenharia avançadas. Uma estrutura de desenvolvimento de aplicação rápida é suficiente para implementar a lógica de negócio sem introduzir complexidades acidentais.

Do ponto de vista dos funcionários, os subdomínios de suporte não exigem tanta aptidão técnica e são uma grande oportunidade para treinar talentos em ascensão. Guarde os engenheiros da sua equipe que tenham experiência em lidar com desafios complexos para os subdomínios principais. Por último, a simplicidade da lógica de negócios transforma os subdomínios de suporte em bons candidatos para a terceirização.

A Tabela 1-1 resume os aspectos nos quais os três tipos de subdomínio se diferenciam.

Tabela 1-1. As diferenças entre os três tipos de subdomínios

Tipo de subdomínio	Vantagem competitiva	Complexidade	Volatilidade	Implementação	Problema
Principal	Sim	Alta	Alta	Interna	Interessante
Genérico	Não	Alta	Baixa	Comprar/adotar	Resolvido
De suporte	Não	Baixa	Baixa	Interna/terceirizada	Óbvio

Identificando os Limites dos Subdomínios

Como você pode perceber, identificar subdomínios e seus tipos pode ajudar muito a tomar decisões de design diferentes ao construir soluções de software. Nos próximos capítulos, você aprenderá mais outras maneiras de utilizar os subdomínios para simplificar o processo de design do software. Mas como realmente identificar os subdomínios e seus limites?

Os subdomínios e seus tipos são identificados e definidos pela estratégia de negócios da empresa: seus domínios de negócio e como ela se diferencia para concorrer com outras empresas no mesmo ramo. Na grande maioria dos projetos de software, de um jeito ou de outro, os subdomínios “já estão lá”. Isso não significa, no entanto, que é sempre fácil e simples identificar seus limites. Se pedirmos a um CEO uma lista dos subdomínios de sua empresa, provavelmente ele não saberá sequer do que você está falando. Ele não conhece esse conceito. Portanto, você mesmo terá que analisar o domínio para identificar e categorizar os subdomínios em questão.

Um bom ponto de partida são os departamentos da empresa e outras unidades organizacionais. Por exemplo, uma loja varejista virtual pode incluir armazenamento, atendimento ao cliente, coleta, envio, controle de qualidade e departamentos de gestão de canais, entre outros. Essas, no entanto, são áreas de atividade relativamente não refinadas. Vejamos o exemplo do departamento de atendimento ao cliente. É razoável presumir que se trataria de um subdomínio de suporte, ou mesmo genérico, pois sua função geralmente é terceirizada para revendedores subcontratados. Mas essa informação seria o suficiente para tomarmos as decisões de design de software corretas?

Extraindo os subdomínios

Os subdomínios não refinados são um bom ponto de partida, mas os detalhes são o maior problema. Temos que nos certificar de que não estamos nos esquecendo das informações importantes escondidas nas complexidades da função de negócio.

Voltemos ao exemplo do departamento de atendimento ao cliente. Se investigarmos seu funcionamento interno, veremos que um departamento de atendimento ao consumidor normal tem componentes refinados, como um sistema de help desk, gestão e programação de turnos, sistema telefônico e assim por diante. Quando vistas como subdomínios individuais, essas atividades podem ter tipos diferentes: enquanto o help desk e o sistema telefônico são subdomínios genéricos, a gestão de turnos é um subdomínio de suporte. Uma empresa pode desenvolver um algoritmo engenhoso para encaminhar incidentes a agentes que obtiveram sucesso com casos similares no passado. O algoritmo de encaminhamento requer a análise dos casos recebidos e a identificação de semelhanças na experiência passada — ambas são tarefas não triviais. Uma vez que o algoritmo de encaminhamento permite que a empresa proporcione uma melhor experiência ao cliente do que seus concorrentes, o algoritmo de encaminhamento é um subdomínio principal. Esse exemplo é demonstrado na Figura 1-2.

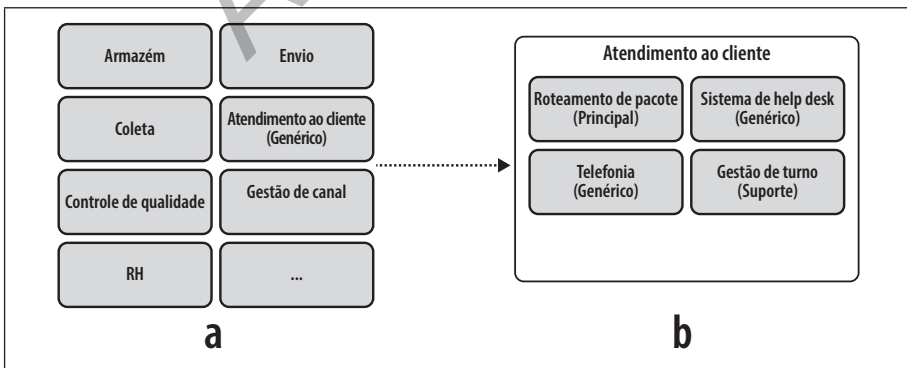


Figura 1-2. Analisando o funcionamento interno de um suposto domínio genérico para encontrar o subdomínio principal refinado, o subdomínio de suporte e dois subdomínios genéricos