

Migrando sistemas monolíticos para microsserviços

Sam Newman

O'REILLY®
Novatec

Authorized portuguese translation of the english edition of Monolith to Microservices, ISBN 9781492047841
© 2020 Sam Newman. This translation is published and sold by permission of O'Reilly Media, Inc., the owner
of all rights to publish and sell the same.

Tradução em português autorizada da edição em inglês da obra Monolith to Microservices, ISBN 9781492047841
© 2020 Sam Newman. Esta tradução é publicada e vendida com a permissão da O'Reilly Media, Inc., detentora
de todos os direitos para publicação e venda desta obra.

© Novatec Editora Ltda. [2020].

Todos os direitos reservados e protegidos pela Lei 9.610 de 19/02/1998. É proibida a reprodução desta obra,
mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates

Tradução: Lúcia A. Kinoshita

Revisão gramatical: Tássia Carvalho

Editoração eletrônica: Carolina Kuwabata

ISBN: 978-65-86057-04-1

Histórico de impressões:

Março/2020 Primeira edição

Novatec Editora Ltda.

Rua Luís Antônio dos Santos 110

02460-000 – São Paulo, SP – Brasil

Tel.: +55 11 2959-6529

Email: novatec@novatec.com.br

Site: www.novatec.com.br

Twitter: twitter.com/novateceditora

Facebook: facebook.com/novatec

LinkedIn: linkedin.com/in/novatec

Sumário

Prefácio	9
Capítulo 1 ■ Apenas o suficiente sobre os microsserviços.....	14
O que são microsserviços?	14
Implantações independentes	15
Modelagem em torno de um domínio de negócio	15
Seja responsável pelos seus próprios dados.....	19
Quais as vantagens que os microsserviços podem trazer?	20
Quais os problemas criados pelos microsserviços?	20
Interfaces de usuário	22
Tecnologia	22
Tamanho	23
Responsabilidade	25
Arquitetura monolítica	27
Sistema monolítico de um só processo	27
Sistema monolítico distribuído	29
Sistemas caixa-preta de terceiros.....	30
Desafios dos sistemas monolíticos	30
Vantagens dos sistemas monolíticos.....	31
Sobre o acoplamento e a coesão	31
Coesão.....	33
Acoplamento.....	33
Apenas o suficiente sobre design orientado a domínios	45
Agregado	45
Contexto delimitado	47
Mapeando agregados e contextos delimitados aos microsserviços	48
Leituras complementares.....	48
Resumo.....	49
Capítulo 2 ■ Planejando uma migração	50
Compreendendo o objetivo	50
Três perguntas básicas.....	52
Por que você optaria pelos microsserviços?.....	53
Aumentar a autonomia das equipes	53

Reduzir o tempo para chegar ao mercado.....	54
Escalar para lidar com a carga, com um custo viável	55
Aumentar a robustez	56
Escalar o número de desenvolvedores	58
Incorporar novas tecnologias.....	59
Quando os microsserviços não seriam uma boa ideia?	61
Domínio nebuloso.....	62
Startups	63
Software instalado e administrado pelos clientes.....	64
Não ter um bom motivo!	65
Custo-benefício	65
Incluindo pessoas na jornada	67
Mudanças organizacionais.....	67
Estabelecimento de um senso de urgência	68
Criação da coalizão para direcionar a mudança	69
Desenvolvimento de uma visão e de uma estratégia.....	70
Comunicação da visão sobre as mudanças	70
Empoderamento dos funcionários para ações mais amplas.....	71
Conquista de vitórias rápidas	72
Consolidação dos ganhos e promoção de novas mudanças	73
Inclusão das novas abordagens na cultura.....	73
Importância da migração gradual.....	74
É o ambiente de produção que conta.....	75
Custo da mudança.....	75
Decisões reversíveis e irreversíveis.....	75
Pontos mais apropriados para um experimento	77
Então, por onde devemos começar?	78
Design orientado a domínios	78
Até que ponto devemos ir?.....	79
Event Storming	79
Usando um modelo de domínios para atribuição de prioridades.....	80
Um modelo combinado.....	82
Reorganizando as equipes.....	84
Modificando as estruturas.....	84
Não há uma solução única para todos.....	86
Fazendo uma mudança.....	88
Outras habilidades.....	90
Como você saberá que a transição está funcionando?	93
Ter pontos de verificação regulares	94
Critérios de avaliação quantitativos	94
Critérios de avaliação qualitativos.....	95
Evitando a falácia do custo irre recuperável	96
Esteja aberto a novas abordagens.....	96
Resumo.....	97

Capítulo 3 ■ Dividindo o sistema monolítico	99
Mudar ou não mudar o sistema monolítico?	99
Cortar, copiar ou reimplementar?	100
Refatorando o sistema monolítico	101
Padrões de migração	103
Padrão: aplicação strangler fig	103
Como o padrão funciona	104
Quando usar o padrão	105
Exemplo: proxy HTTP reverso	108
Dados?	111
Opções de proxy	111
Mudança de protocolos	115
Exemplo: FTP	119
Exemplo: interceptação de mensagens	120
Outros protocolos	123
Outros exemplos do padrão strangler fig	123
Mudança de comportamentos concomitantes à migração de funcionalidades	123
Padrão: composição de UI	124
Exemplo: composição de páginas	125
Exemplo: composição de widgets	126
Exemplo: Micro Frontends	130
Quando usar o padrão	131
Padrão: branch por abstração	131
Como o padrão funciona	132
Como método de fallback	139
Quando usar o padrão	140
Padrão: execução em paralelo	141
Exemplo: comparando os cálculos de preços de derivativos de crédito	142
Exemplo: geração de listas na Homegate	143
Técnicas de verificação	144
Usando Spies	145
Scientist do GitHub	146
Lançamento no escuro e versão canário	146
Quando usar o padrão	147
Padrão: colaborador decorador	147
Exemplo: programa de fidelidade	148
Quando usar o padrão	149
Padrão: captura de dados modificados	149
Exemplo: gerando cartões de fidelidade	150
Implementando a captura de dados modificados	150
Quando usar o padrão	154
Resumo	154

Capítulo 4 ■ Decompondo o banco de dados	155
Padrão: banco de dados compartilhado	155
Padrões para lidar com o problema	157
Quando usar o padrão	157
Mas isso não é possível!	157
Padrão: visão de banco de dados	159
Banco de dados como um contrato público	159
Visões a serem apresentadas	160
Limitações	162
Responsabilidade	162
Quando usar o padrão	163
Padrão: serviço de encapsulamento de banco de dados	163
Quando usar o padrão	165
Padrão: interface de banco de dados como serviço	166
Implementando uma engine de mapeamento	168
Comparação com as visões	168
Quando usar o padrão	169
Transferência de responsabilidades	169
Padrão: sistema monolítico expondo agregados	170
Padrão: mudança do responsável pelos dados	173
Sincronização de dados	174
Padrão: sincronização de dados na aplicação	176
Passo 1: Sincronizar grandes volumes de dados	177
Passo 2: Sincronizar na escrita, ler do esquema antigo	178
Passo 3: Sincronizar na escrita, ler do novo esquema	178
Quando usar esse padrão	179
Padrão: tracer write	181
Sincronização de dados	184
Exemplo: pedidos na Square	186
Quando usar o padrão	190
Separando o banco de dados	190
Separação física versus separação lógica de um banco de dados	191
Separar o banco de dados ou o código antes?	193
Separar o banco de dados antes	193
Separar o código antes	198
Separar o banco de dados e o código ao mesmo tempo	203
Então, o que devo separar antes?	204
Exemplos de separação de esquemas	204
Padrão: separação de tabelas	205
Quando usar o padrão	207
Padrão: transferência de relacionamentos de chave estrangeira para o código	207
Transferindo a junção	209
Consistência dos dados	210

Quando usar o padrão	213
Exemplo: dados estáticos compartilhados	213
Transações	223
Transações ACID	223
Continuar sendo ACID, mas sem atomicidade?	225
Commits de duas fases	226
Transações distribuídas – basta dizer não	229
Sagas	230
Modos de falha das sagas	232
Implementando as sagas	236
Sagas versus transações distribuídas	243
Resumo	244
Capítulo 5 ■ Dificuldades crescentes	245
Mais serviços, mais dificuldades	245
Responsabilidade em escala	247
Como esse problema pode se manifestar?	248
Quando esse problema poderia ocorrer?	249
Possíveis soluções	249
Mudanças que causam incompatibilidade	249
Como esse problema pode se manifestar?	250
Quando esse problema poderia ocorrer?	250
Possíveis soluções	251
Relatórios	255
Quando esse problema poderia ocorrer?	256
Possíveis soluções	256
Monitoração e resolução de problemas	257
Quando esses problemas poderão ocorrer?	258
Como esses problemas poderão ocorrer?	258
Possíveis soluções	259
Experiência do desenvolvedor local	263
Como esse problema pode se manifestar?	263
Quando esse problema poderia ocorrer?	264
Possíveis soluções	264
Administração de muitas tarefas	265
Como esse problema poderia se manifestar?	265
Quando esse problema poderia ocorrer?	266
Possíveis soluções	266
Testes fim a fim	267
Como esse problema pode se manifestar?	268
Quando esse problema poderia ocorrer?	268
Possíveis soluções	269

Otimização global versus local.....	271
Como esse problema pode se manifestar?	272
Quando esse problema poderia ocorrer?	272
Possíveis soluções	273
Robustez e resiliência.....	274
Como esse problema pode se manifestar?	275
Quando esse problema poderia ocorrer?	275
Possíveis soluções.....	275
Serviços órfãos	276
Como esse problema pode se manifestar?	276
Quando esse problema poderia ocorrer?	277
Possíveis soluções.....	277
Resumo.....	279
Capítulo 6 ■ Palavras finais	280
Apêndice A ■ Bibliografia.....	281
Apêndice B ■ Índice de padrões.....	284