



Instituto Infnet

Aula 4 - Arquitetura de Microsserviços e Mobile

Aula 4

- Como dividir a aplicação monolítica.

Ementa proposta Aula 4

- Leia o white paper do Postman sobre documentação de serviços;
- capítulo 2 de "API Testing And Development With Postman";



DDD no contexto de microsserviços

Domain-Driven Design (Erick Evans):

- Linguagem ubíqua;
- Agregar;
- Contexto delimitado.



DDD no contexto de microsserviços

Linguagem ubíqua:

"Linguagem ubíqua refere-se à ideia de que devemos nos esforçar para usar os mesmos termos em nosso código que os usuários usam. A ideia é que ter uma linguagem comum entre a equipe de entrega e as pessoas reais tornará mais fácil modelar o domínio do mundo real e também deve melhorar a comunicação."




DDD no contexto de microsserviços

Agregar:

“Em geral, porém, você deve pensar em um agregado como algo que tem estado, identidade, um ciclo de vida que será gerenciado como parte do sistema. Agregados geralmente se referem a conceitos do mundo real.”

“Então, um agregado deve ser gerenciado por um microsserviço, embora um único microsserviço possa possuir o gerenciamento de vários agregados.”

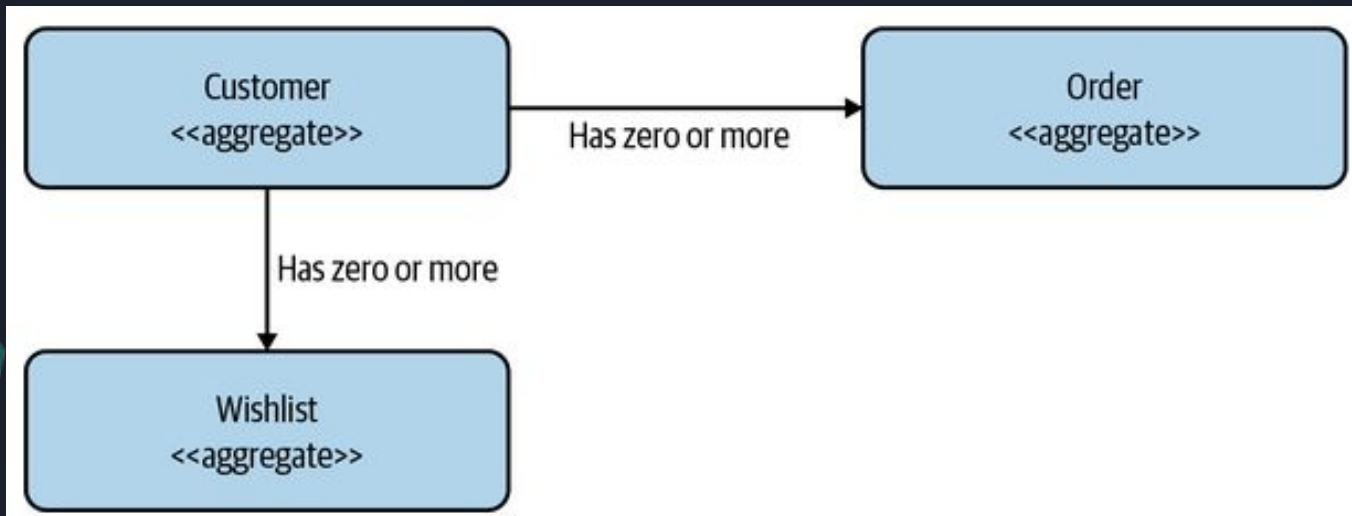


DDD no contexto de microsserviços

Agregar:

“O ponto-chave a entender aqui é que se uma parte externa solicitar uma transição de estado em um agregado, o agregado pode dizer não. O ideal é implementar seus agregados de tal forma que transições de estado ilegais sejam impossíveis.”

Ex.:

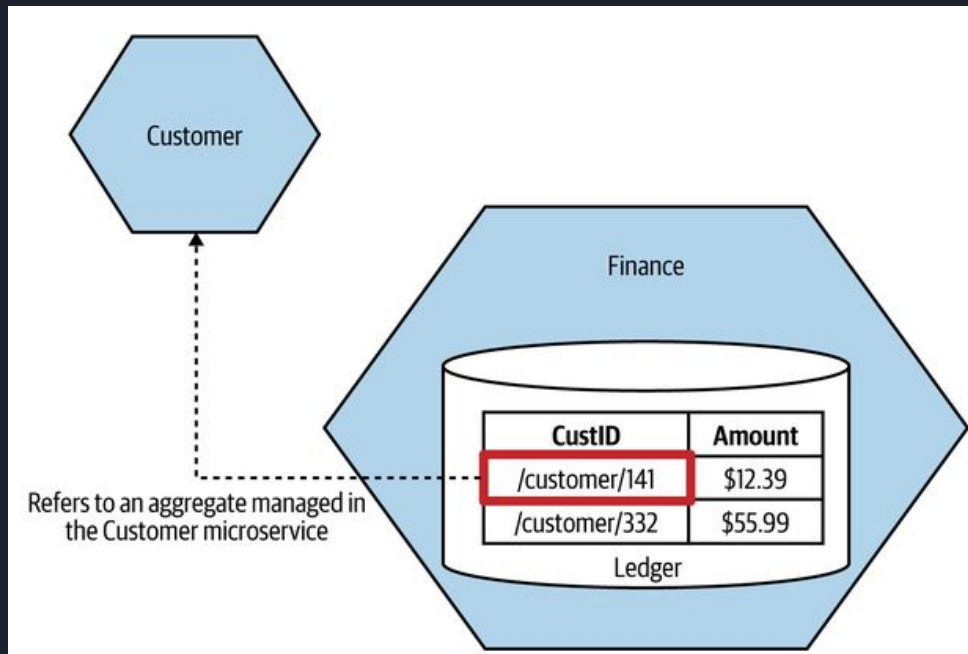


DDD no contexto de microsserviços

Agregar:

Explicitando o relacionamento de agregados em diferentes serviços.

ex.: REST



DDD no contexto de microsserviços

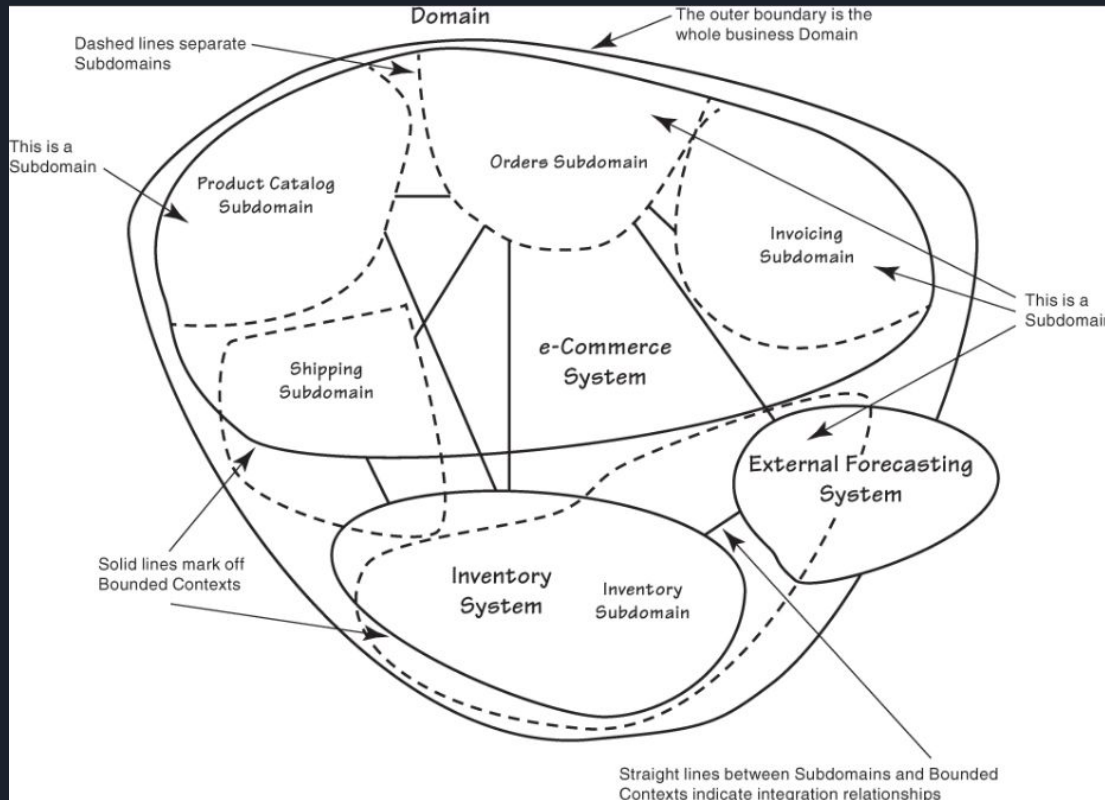
Contexto delimitado:

“Um contexto delimitado geralmente representa um limite organizacional maior.”



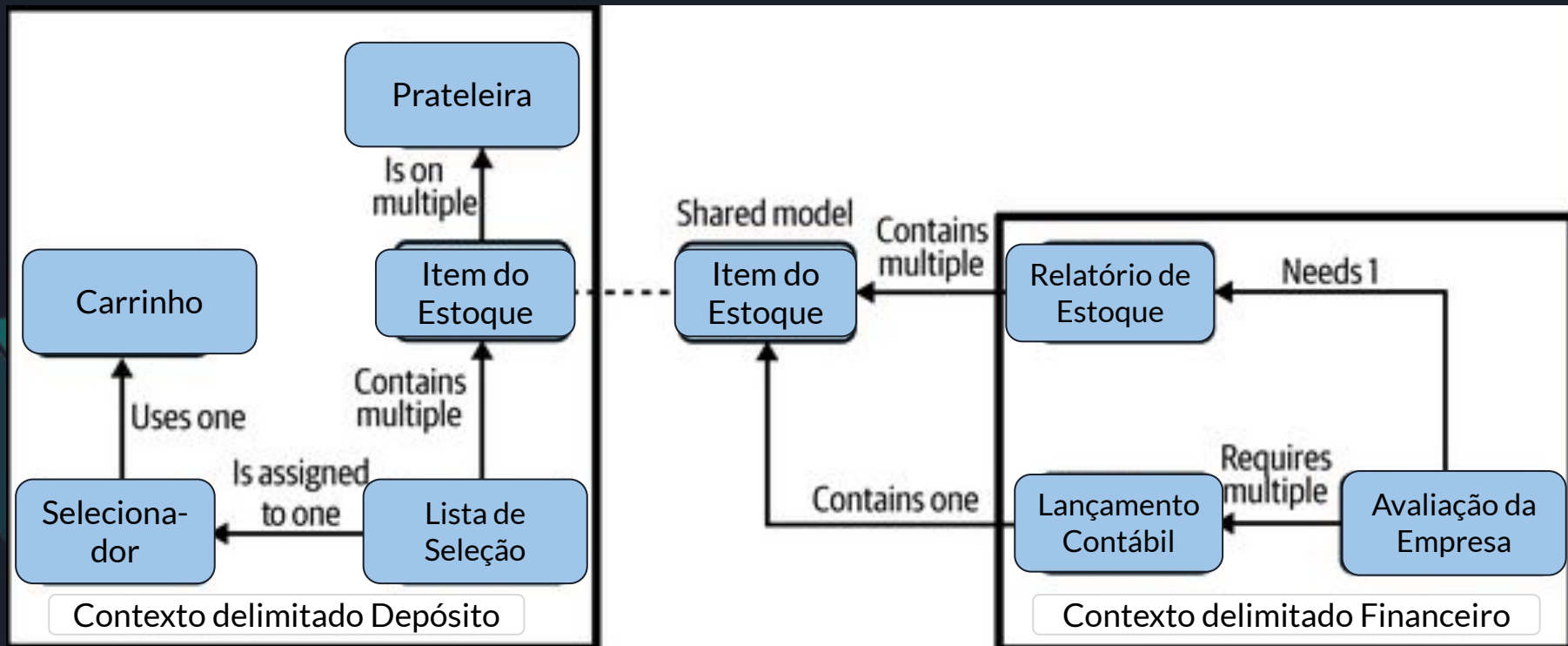
DDD no contexto de microsserviços

Contexto delimitado:



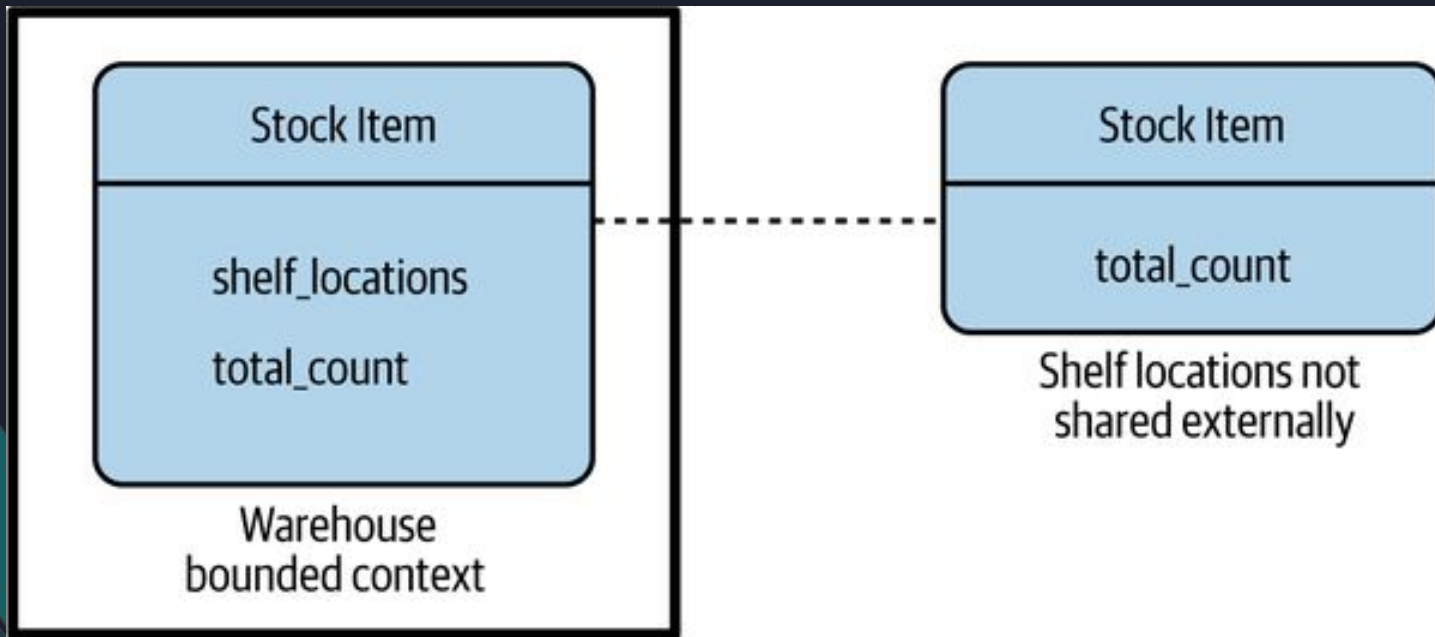
DDD no contexto de microsserviços

Contexto delimitado:



DDD no contexto de microsserviços

Modelos compartilhados:



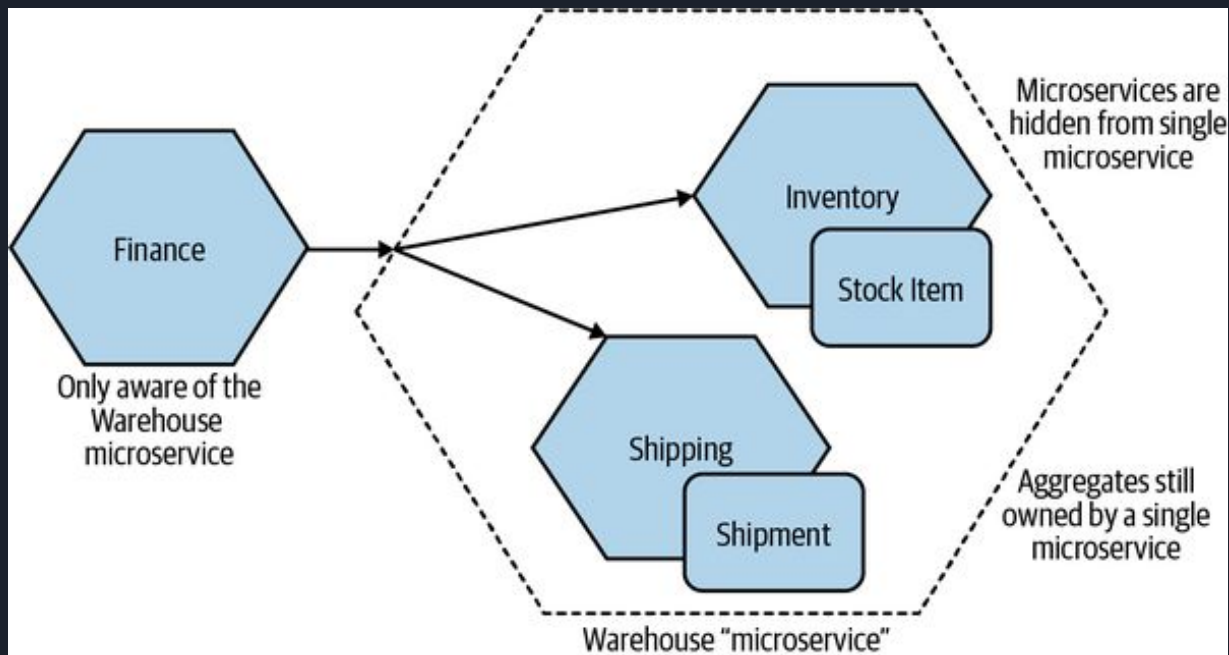
Mapeando Microserviços

“Ambos(Agregados e Contexto delimitados) podem, portanto, funcionar bem como limites de serviço. Ao começar, como já mencionei, você quer reduzir o número de serviços com os quais trabalha. Como resultado, você provavelmente deve mirar em serviços que abrangem contextos inteiros delimitados. Conforme você se firma e decide dividir esses serviços em serviços menores, precisa se lembrar de que os agregados em si não querem ser divididos — um microserviço pode gerenciar um ou mais agregados, mas não queremos que um agregado seja gerenciado por mais de um microserviço.”



Mapeando Microserviços

Separando por Contextos delimitados ou Agregados



Tempestade de eventos (Event storming)

Separando por Contextos delimitados ou Agregados

- Reunir todos os interessados;
- Participantes identificam os eventos de domínio;
- Participantes identificam os comandos que fazem esses eventos acontecerem;
- Destacar potenciais agregados;
- Agrupar agregados em contextos delimitados.

* Livro EventStorming por Alberto Brandolini (Leanpub)



Além do DDD

Alternativas ou complementos ao DDD:

- Verificar volatilidade;
- Dados;
- Tecnologia;
- Organizacional.



Dividindo um monolito

Tenha um **objetivo claro**:

- “Microsserviços não são o objetivo. Você não “ganha” por ter microsserviços. Adotar uma arquitetura de microsserviços deve ser uma decisão consciente, baseada em tomada de decisão racional.”
- “Microsserviços não são fáceis. Tente as coisas simples primeiro.”



Dividindo um monolito

Migração incremental:

- *"Se você fizer uma reescrita big-bang, a única coisa que você terá garantia é de um big bang."* – Martin Fowler;



Dividindo um monolito

Migração incremental:

- *"Se você fizer uma reescrita big-bang, a única coisa que você terá garantia é de um big bang."* – Martin Fowler;
- Migrar em pequenas etapas:
 - Pequenos erros;
 - Vitórias mais rápidas.



Dividindo um monolito

Migração incremental:

- *"Se você fizer uma reescrita big-bang, a única coisa que você terá garantia é de um big bang."* – Martin Fowler;
- Migrar em pequenas etapas:
 - Pequenos erros;
 - Vitórias mais rápidas.
- *"Você não vai perceber o verdadeiro horror, dor e sofrimento que uma arquitetura de microsserviços pode trazer até que esteja em produção."* -cap 3. Building Microservices Sam Newman.

Dividindo um monolito

O Monólito Raramente É o Inimigo:

- *"É comum que a arquitetura monolítica existente permaneça após uma mudança em direção a microsserviços, embora frequentemente em uma capacidade diminuída."*



Dividindo um monolito

Perigos da decomposição prematura:

- Domínio ainda pouco claros;
- Contextos pouco trabalhados.



Dividindo um monolito

O que dividir primeiro?


A resposta será de acordo com seu objetivo:

- Quer escalar o aplicativo?

A funcionalidade que atualmente restringe a capacidade do sistema de lidar com a carga estará no topo da lista;

- Quer melhorar o tempo de lançamento no mercado?

Observe a volatilidade do sistema para identificar as partes da funcionalidade que mudam com mais frequência e veja se elas funcionariam como microsserviços.



Dividindo um monolito

O que dividir primeiro?

A decisão sobre o que dividir acabará sendo um equilíbrio entre essas duas forças:

- a facilidade de extração
versus
- o benefício de extrair o microserviço em primeiro lugar.



Dividindo um monolito

O que dividir primeiro?

A decisão sobre o que dividir acabará sendo um equilíbrio entre essas duas forças:

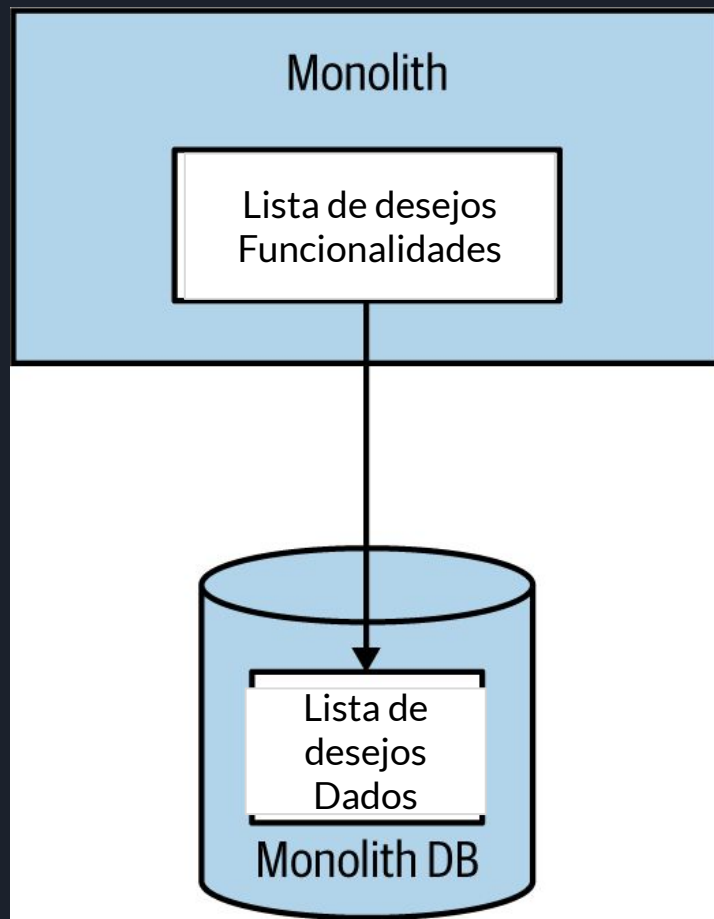
- a facilidade de extração (dê preferência a facilidade);
versus
- o benefício de extrair o microserviço em primeiro lugar.



Dividindo um monolito

Decomposição por camada

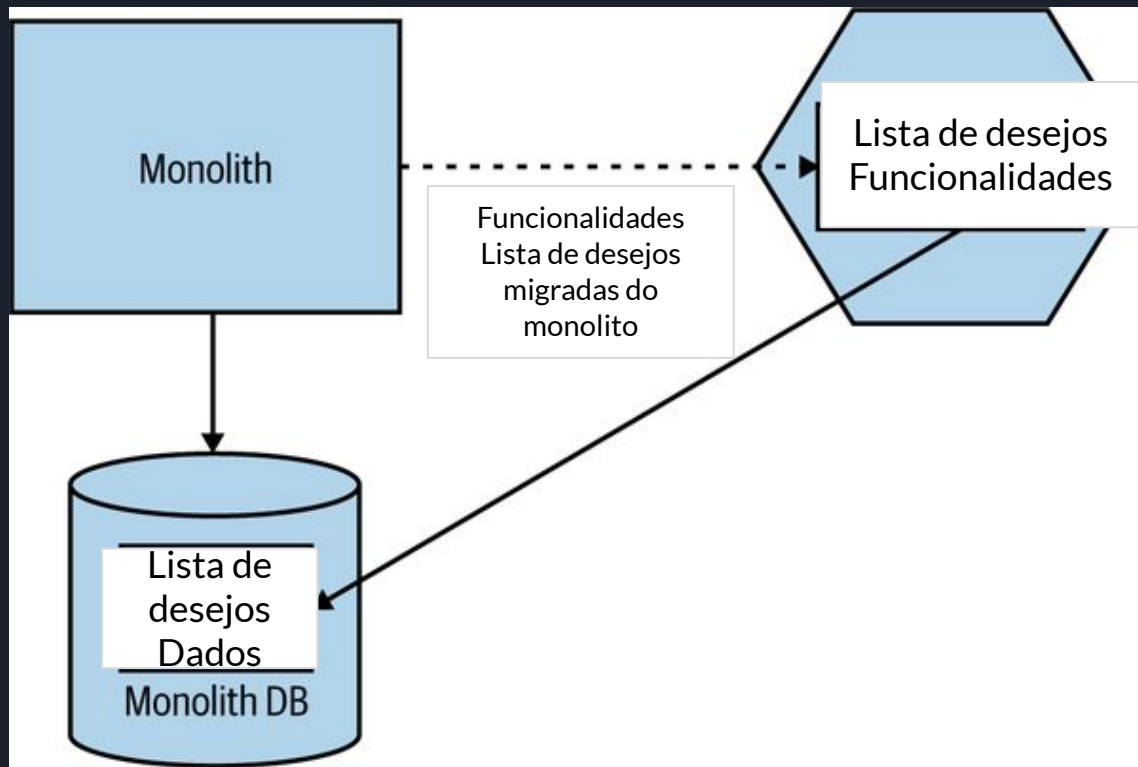
- Código primeiro;
versus
- Dados primeiro.



Dividindo um monolito

Decomposição por camada

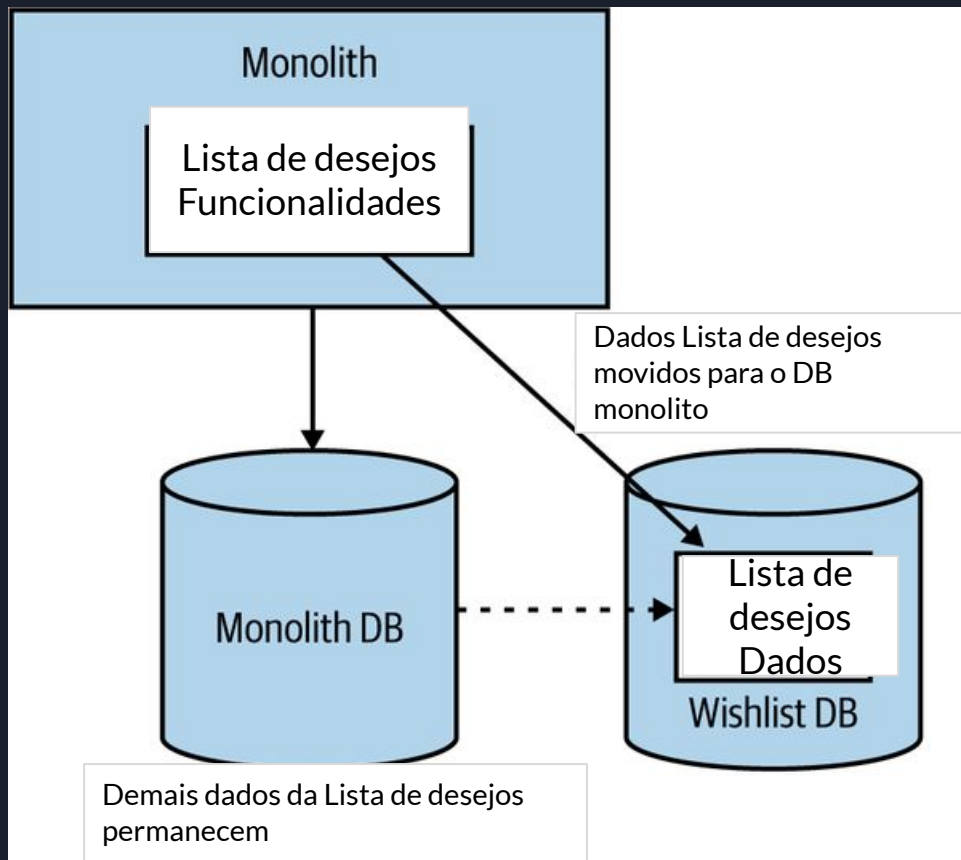
- Código primeiro:
 - * Verificar viabilidade dos dados também



Dividindo um monolito

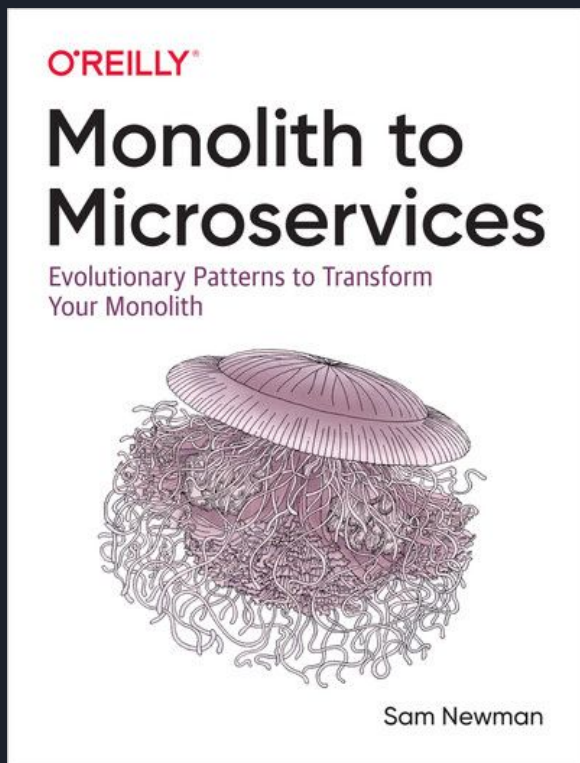
Decomposição por camada

- Dados primeiro:
 - * Indicados para quando há dúvida sobre a viabilidade



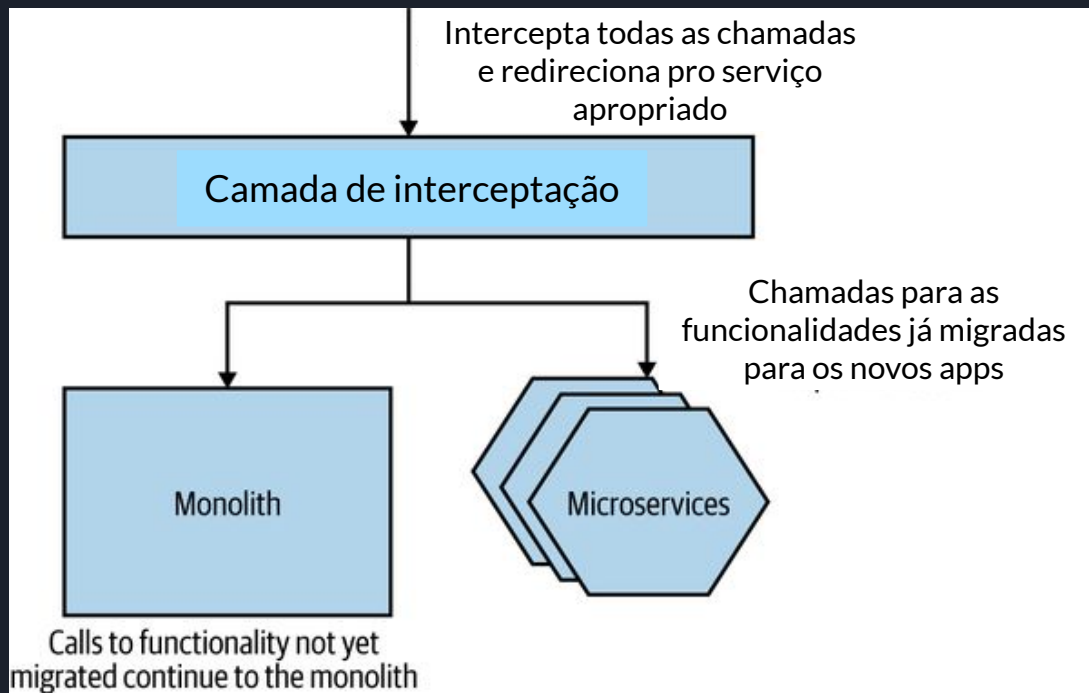
Padrões de decomposição

<https://learning.oreilly.com/library/view/monolith-to-microservices/9781492047834/>



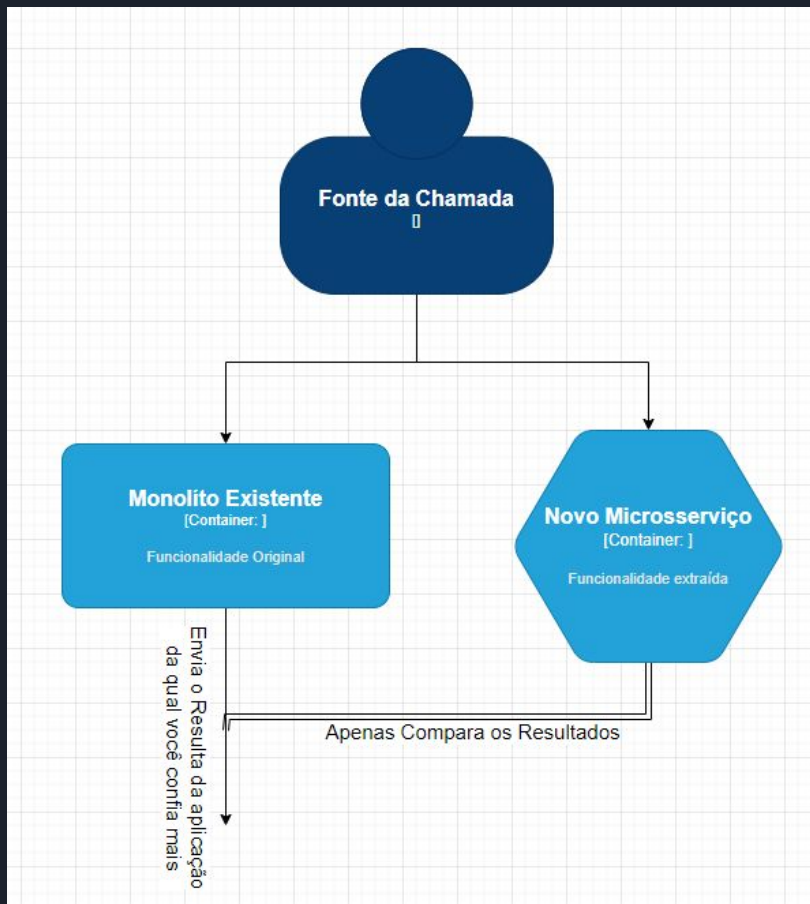
Padrões de decomposição

- Padrão Strangler Fig (Figueira-Estranguladora)
*Pode ser feito frequentemente sem fazer nenhuma alteração no monolítico existente



Padrões de decomposição

- Padrão Strangler Fig (Figueira-Estranguladora);
- Padrão Parallel Run (Execução Paralela);



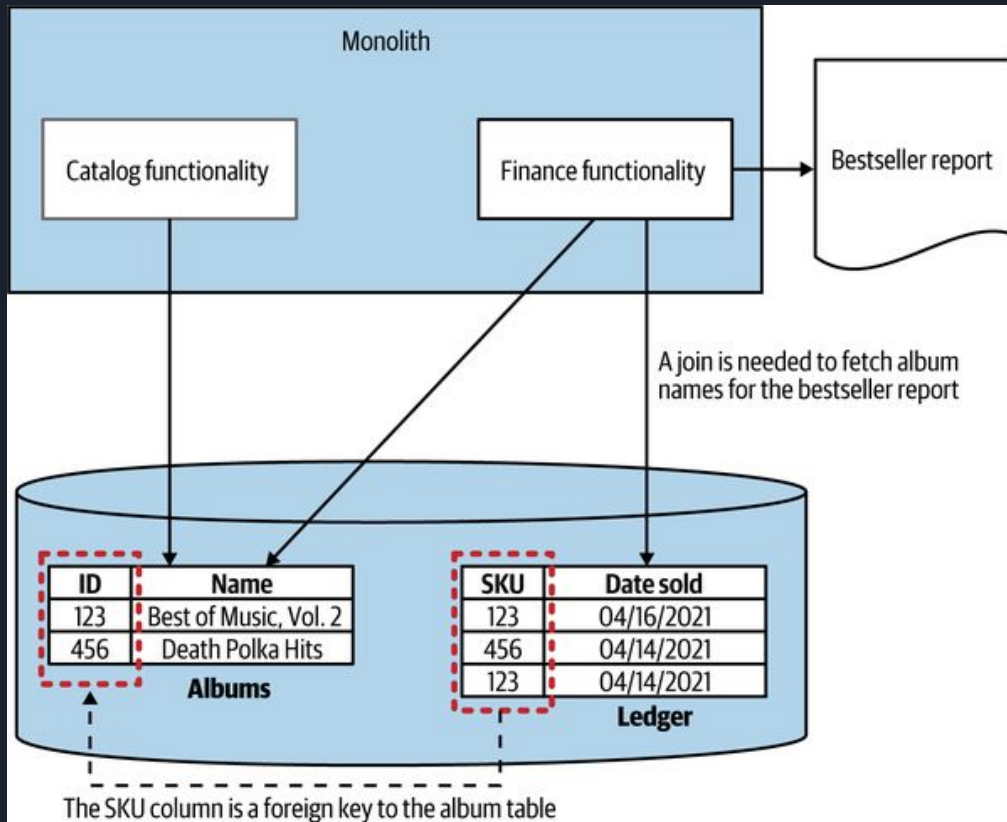
Padrões de decomposição

- Padrão Strangler Fig (Figueira-Estranguladora);
- Padrão Parallel Run (Execução Paralela);
- Feature Toggle (Alternância de Funcionalidade).



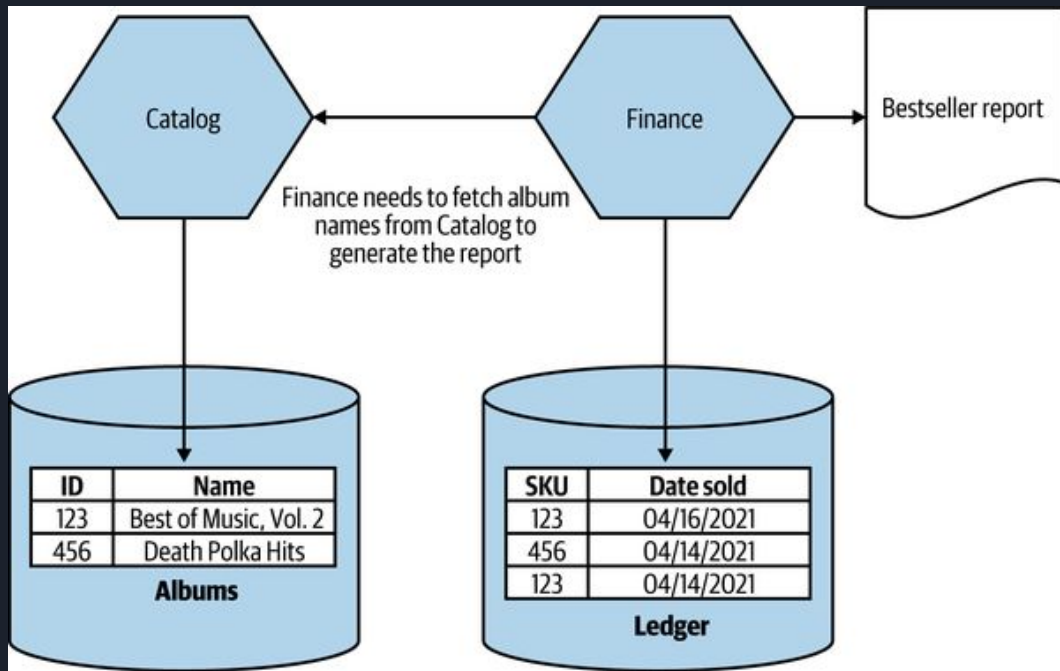
Preocupações ao decompor dados

- Desempenho;
- Integridade de dados;
- Transações;
- Ferramentas;
- Banco de dados de relatórios.



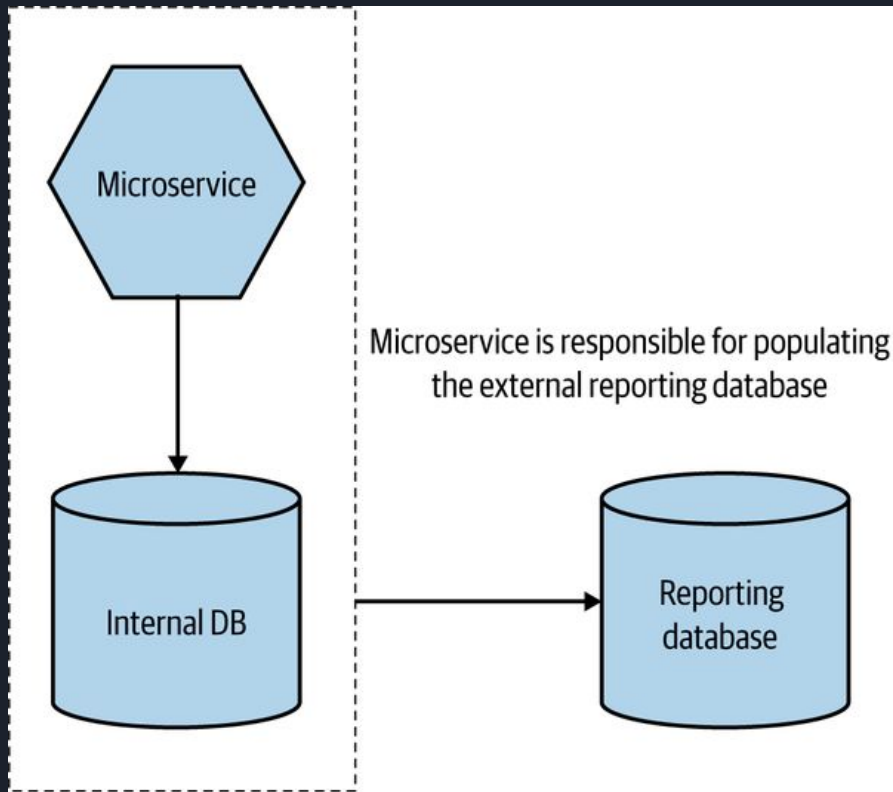
Preocupações ao decompor dados

- Desempenho;
- Integridade de dados;
- Transações;
- Ferramentas;
- Banco de dados de relatórios.



Preocupações ao decompor dados

- Banco de dados de relatórios.



Resumo dos Temas vistos





Instituto Infnet

Aula 5 - Arquitetura de Microsserviços e Mobile