



Resenha

Projeto de Software

Gustavo Delfino Guimarães

1489062@sga.pucminas.br

Matrícula: 836079

Microserviços - Uma Definição deste novo termo arquitetônico

Martin Fowler

O artigo "Microserviços" vem para explicar este novo estilo de arquitetura que vem permeando os desenvolvedores e as grandes empresas. O microserviço vem pra disruptar com o "padrão" utilizado hoje onde se tem uma tela interativa, um banco de dados e um app que acessa e gerencia ao lado do servidor. Este padrão arquitetural vem tomando grande proporção devido a sua facilidade para manutenções sem ter que mexer em todo o sistema, mas apenas naquele serviço.

Não é possível dizer que existem regras para os microserviços como por exemplo existem no MVC, mas é possível enxergar características comuns entre programas que utilizam dessa arquitetura.

Componentização via serviços:

A arquitetura de microserviços se baseia na divisão do software em serviços independentes, proporcionando maior flexibilidade e escalabilidade. O sistema é construído a partir de componentes independentes, assim como ocorre no mundo físico. Segundo Martin Fowler, um componente é uma unidade de software substituível e atualizável de forma independente. Os serviços são executados fora do processo e comunicando-se via chamadas remotas (ex.: APIs web).

Uma das vantagens desta arquitetura é que em um sistema monolítico, qualquer alteração exige a reimplantação completa, já nos microserviços, mudanças em um serviço não afetam os demais, permitindo reimplantação isolada. Outra vantagem é que serviços possuem interfaces bem definidas, reduzindo o acoplamento entre componentes, diferente das bibliotecas, que dependem de documentação e disciplina para evitar violações de encapsulamento.

Organização em torno de capacidades empresariais:

Na abordagem de microserviços, tenta-se dividir um aplicativo grande em serviços organizados em torno da capacidade de negócios, onde cada serviço cobre toda a pilha tecnológica da área específica, possibilitando equipes multifuncionais que reúnem toda a habilidade necessária para o desenvolvimento e operação de um serviço.

Produtos não projetos:

Os defensores dos microserviços preferem um modelo de produto, onde a mesma equipe é responsável pelo software ao longo de sua vida útil. O conceito da Amazon de "você constrói, você executa" reflete essa abordagem, garantindo que os desenvolvedores acompanhem a operação do software em produção e interajam diretamente com os usuários. A visão de produto fortalece a conexão entre desenvolvimento e capacidades de negócio. Em vez de ver o software como um conjunto de funcionalidades a serem finalizadas, ele passa a ser um processo contínuo de melhoria para atender melhor os usuários.

Pontos de Extremidade Inteligentes e Pipes Burros

A Arquitetura de microserviços prioriza ponto de extremidades inteligentes e pipes burros, onde cada serviço mantém a própria lógica e opera como filtros Unix com o intuito de reduzir o acoplamento e melhora a escalabilidade.

Governança Descentralizada

Microserviços promovem governança descentralizada, permitindo que cada equipe escolha a melhor tecnologia para seu serviço, o artigo cita o exemplo de como usar Node.js para uma interface e C++ para processamento de alto desempenho. Em vez de padrões rígidos, as equipes criam ferramentas reutilizáveis, seguindo um modelo interno de código aberto, como a Netflix faz.

Gestão de Dados Descentralizada

Em microserviços, a descentralização ocorre tanto no modelo conceitual quanto no armazenamento de dados. Diferentes serviços podem ter visões distintas sobre a mesma entidade, um conceito alinhado ao Bounded Context do DDD. Além disso, cada serviço gerencia seu próprio banco de dados, utilizando a Persistência Poliglota, também evitam transações distribuídas devido ao alto acoplamento temporal, optando por consistência eventual e operações compensatórias para lidar com erros. Essa abordagem reflete práticas empresariais, onde inconsistências são toleradas para maior agilidade.

Automação de Infraestrutura

Testes automatizados e pipelines garantem implantações seguras e previsíveis, tornando a implantação de múltiplos serviços tão simples quanto a de um monólito. No entanto, o gerenciamento operacional de microserviços em produção apresenta desafios diferentes, exigindo maior automação para garantir estabilidade e escalabilidade. Ferramentas de nuvem foram primordiais para essas automações, como a AWS.

Projetar para o Fracasso

Em microserviços as falhas são inevitáveis, já que um serviço pode não estar disponível, porém o aplicativo continua funcionando já que os serviços são modularizados, diferente de um aplicativo monólito onde ele poderia parar por completo. Sendo assim, os sistemas desenvolvidos com essa arquitetura são projetados para lidar com falhas.

Design Evolucionário

Microserviços permitem maior flexibilidade na evolução do software, pois cada serviço pode ser modificado ou substituído sem afetar todo o sistema. O The Guardian, por exemplo, mantém um monólito central, mas adiciona novos recursos como microserviços temporários. O princípio fundamental é a substituíbilidade, garantindo modularidade e facilitando mudanças rápidas sem impactos negativos. A fragmentação dos serviços acelera o processo de lançamento, pois apenas os componentes modificados precisam ser atualizados. No entanto, exige atenção à compatibilidade entre serviços, preferindo-se projetá-los para tolerar mudanças, evitando versionamentos desnecessários.

Ao final o artigo questiona se os microserviços são o futuro e argumenta que não podem afirmar isso mas que é sim uma abordagem promissora para arquiteturas de software, embora a experiência inicial tenha sido positiva, ainda não conseguimos avaliar as consequências dessa arquitetura no longo prazo. O autor também traz outros pontos como

a dificuldade para definir os limites dos serviços, os obstáculos em refatorar em comparação com um monólito e o fato que uma má divisão dos serviços pode apenas deslocar de uma arquitetura monolítica para uma arquitetura de micros serviços.

O texto é muito bom e traz uma visão abrangente sobre o tema, apesar de tudo ainda é uma arquitetura que pode levar um tempo para ser padronizada e específica, mas tem um bom potencial de crescimento. Acredito que hoje muitas empresas usam um certo tipo de “microserviços”, que são monólitos gigantes que interagem entre si mas desfrutam de uma boa sinergia e entendimento. Acredito que este tema ainda será muito difundido e crescerá nos próximos anos, facilitando o desenvolvimento, manutenção e criação de código.