Universidade Católica de Minas Gerais

Projeto de Software
2024

Prof. João Paulo Carneiro Aramuni
Aluno: Jhonatan Gutemberg Rosa Ferreira

Resenha do artigo:

Microserviços

James Lewis, Martin Fowler

Os microsserviços representam um estilo arquitetônico que está ganhando popularidade na construção de aplicativos corporativos. Ao contrário da abordagem monolítica tradicional, onde todo o sistema é desenvolvido e implantado como uma única unidade, os microsserviços dividem o aplicativo em pequenos serviços independentes, cada um rodando em seu próprio processo e se comunicando por meio de APIs leves, como HTTP. Esses serviços são organizados em torno de funcionalidades de negócios e podem ser escritos em diferentes linguagens e usar diversos bancos de dados.Com essa versatilidade buscar proporcional maior flexibilidade, escalabilidade e agilidade no desenvolvimento.

Apesar de nem todas as características apresentem as mesmas qualidades, a maioria compartilha elementos centrais. Os autores, baseiam suas descrições em experiências próprias e observações de outras equipes. O seu objetivo não é estabelecer uma norma fixa, mas sim refletir as práticas comuns nesse campo emergente.

Desde o início da indústria de software, há o desejo de construir sistemas compondo partes, como no mundo físico. Nos microsserviços, componentes são divididos em serviços independentes, que se comunicam externamente, ao contrário das bibliotecas tradicionais, ligadas ao processo. A principal vantagem dos serviços é a implantação independente, permitindo que mudanças em um serviço não afetem todo o sistema. Essa abordagem traz desafios, como o custo maior das chamadas remotas e a complexidade de gerenciamento de APIs mais granulares.

A divisão de um aplicativo em microsserviços geralmente se organiza em torno de capacidades de negócios, ao invés de camadas tecnológicas (UI, lógica do servidor, banco de dados). Essa abordagem permite que equipes multifuncionais, com habilidades diversas, sejam responsáveis por um serviço completo, desde a interface de usuário até o banco de dados. Isso difere com o modelo tradicional, onde equipes separadas por tecnologia enfrentam dificuldades de coordenação, resultando em processos mais lentos e complexos.

Os grandes aplicativos monolíticos podem ser modularizados em torno de capacidades de negócios, mas essa prática não é comum. O principal desafio é tentar abranger vários contextos de negócio, o monólito dificulta a compreensão dos limites modulares, exigindo grande disciplina para manter a separação. A arquitetura de microsserviços, exigir uma divisão mais explícita e clara entre os componentes, facilita a manutenção de fronteiras entre as equipes e serviços, tornando a modularização mais eficaz.

A abordagem tradicional de desenvolvimento de software vê os projetos como tarefas com prazo para entrega, onde o produto final é entregue a uma equipe de manutenção e os desenvolvedores são liberados. Já no modelo de microsserviços, a mentalidade de "você

constrói, você executa", inspirada pela Amazon, defende que a equipe de desenvolvimento permaneça responsável pelo software durante todo o seu ciclo de vida. Isso aproxima os desenvolvedores do comportamento real do software em produção e dos seus usuários, promovendo melhorias contínuas com base nas necessidades do negócio.

A governança centralizada frequentemente resulta na padronização em plataformas de tecnologia únicas, o que pode ser restritivo, já que nem todo problema se encaixa em uma única solução. Embora aplicativos monolíticos possam utilizar diferentes linguagens em certa medida, essa flexibilidade é limitada. Ao dividir um monólito em microsserviços, as equipes ganham liberdade para escolher a melhor tecnologia para cada componente. Essa flexibilidade permite que os desenvolvedores adotem tecnologias que se alinhem melhor às necessidades específicas de cada parte do sistema.

Segundo os autores as equipes ao invés de seguirem regras rígidas, elas buscam criar ferramentas úteis baseadas em suas experiências, que podem ser compartilhadas com outros desenvolvedores. Essa filosofia é exemplificada pela Netflix, que promove o compartilhamento de código testado em batalha como bibliotecas, permitindo que outros resolvam problemas semelhantes de maneira eficaz, enquanto ainda possibilita a escolha de alternativas quando necessário.

Nos microsserviços, a descentralização do gerenciamento de dados leva a modelos conceituais diferentes entre sistemas, causando problemas de integração. Em vez de um único banco de dados como nos monólitos, cada serviço pode gerenciar seu próprio banco, utilizando a Persistência Poliglota, o que permite adaptação a necessidades específicas.

As técnicas de automação de infraestrutura avançaram significativamente com a evolução da nuvem, especialmente a AWS, facilitando a criação e operação de microsserviços. Equipes experientes em Entrega e Integração Contínua utilizam extensivamente automação, realizando testes automatizados e promovendo software de forma eficiente através de pipelines. Embora a implantação de monólitos seja simples após a automação, o gerenciamento de microsserviços em produção apresenta desafios operacionais distintos, mesmo quando a implantação se torna entediante.

Os praticantes de microsserviços valorizam a decomposição de serviços como uma maneira de controlar mudanças em aplicativos, permitindo atualizações frequentes e bem gerenciadas. A chave para essa abordagem é a capacidade de substituir e atualizar componentes independentemente, incentivando a ideia de que serviços podem ser descartados em vez de evoluídos ao longo do tempo.

Um exemplo é o site do The Guardian, que começou como um monólito e agora utiliza microsserviços para adicionar recursos temporários. Essa estratégia facilita a construção e remoção rápida de funcionalidades, especialmente para eventos específicos. A modularidade, centrada na ideia de agrupar partes que mudam juntas, permite lançamentos mais ágeis, embora exija atenção para garantir que alterações em um serviço não quebrem seus consumidores, minimizando a necessidade de versionamento.

Os desafios para a definição da arquitetura incluem a definição de limites entre componentes e a complexidade das interações entre eles. Equipes menos habilidosas podem

enfrentar dificuldades semelhantes com ambas as arquiteturas. O autor sugere iniciar com um monólito modular e considerar a transição para microsserviços quando necessário,as decisões arquitetônicas devem ser baseadas nas informações disponíveis e nas habilidades da equipe.

Por fim, a arquitetura de microsserviços representa uma evolução significativa na forma como os aplicativos corporativos são projetados e gerenciados. Ao promover a independência dos serviços, a flexibilidade na escolha de tecnologias e a responsabilidade contínua das equipes de desenvolvimento, os microsserviços oferecem uma abordagem mais ágil e adaptável às demandas do mercado. No entanto, essa estrutura também traz desafios, como a complexidade na integração e gerenciamento. Assim, a transição de um monólito para microsserviços deve ser considerada com cuidado, levando em conta as habilidades da equipe e as necessidades específicas do negócio, visando sempre a otimização e a eficiência no desenvolvimento de software.