

Autor(a): Isabella Luiza Dias dos Santos

Síntese do Artigo: “Microservices” - James Lewis e Martin Fowler

Em 2012, foi definido o termo “Arquitetura de Microserviços” pela primeira vez. Esse termo representa um tipo de arquitetura que muitos arquitetos de software estavam explorando naquele momento. Foram observadas características semelhantes quanto a forma de organizar e construir programas de computador, como a divisão do software em vários pequenos serviços com uma função específica cada. Cada um desses serviços pode ser desenvolvido, testado e colocado no ar de maneira independente, o que oferece mais flexibilidade, escalabilidade e simplicidade quanto a manutenção do mesmo.

Mas o que originou a arquitetura de microserviços? Os autores explicam detalhadamente a razão pelo surgimento desse novo tipo de estilo de arquitetura, comparando-a com a de estilo monolítico. No modelo monolítico, o programa de computador é construído como uma única peça, onde todas as funcionalidades estão agrupadas em um único programa. Há uma certa divisão nesse estilo, como a interface do usuário(frontend), que são as páginas vistas pelos usuários no navegador, o banco de dados, que é o local em que ficam armazenadas as informações e, o backend, que é onde se executa as regras do negócio e onde se recebe as requisições do sistema. Mas qual seria o problema em implementar esse modelo? Simplesmente, o fato de que, se houver a necessidade de alguma mudança no sistema, faz-se necessário a reconstrução e a reinstalação do sistema completo. Além disso, o crescimento da computação em nuvem tem tornado-o bastante limitante, uma vez que esse tipo de arquitetura não possibilita aproveitar os benefícios da nuvem, como atualizações independentes e maior resiliência entre componentes.

Nos seguintes parágrafos, são abordadas as características comuns da arquitetura de microserviços de maneira aprofundada. Começando pela componentização via serviços, que é um dos pilares da arquitetura de microserviços. Ela permite que o sistema seja dividido em unidades independentes que podem ser desenvolvidas e atualizadas separadamente. A comunicação por meio de chamadas remotas possibilita que os serviços funcionem fora do processo principal e, diferentemente das bibliotecas, que são vinculadas diretamente no código, essa abordagem garante flexibilidade e agilidade, eliminando a necessidade de reimplantar o sistema inteiro.

Em seguida, é apresentado uma outra característica dessa arquitetura, que é a organização em torno de recursos de negócios. Ao invés de dividir equipes segundo a tecnologia, cada um dos serviços é responsável por uma área, da interface ao banco de dados. Essa separação possibilita que as equipes trabalhem de forma independente, o que facilita as mudanças e mantém os limites bem definidos. Outra característica, é a de que o software construído é um produto e não um projeto, ou seja, uma equipe é responsável pelo software durante toda a vida útil do mesmo. Isso promove uma certa aproximação entre os

desenvolvedores e os usuários que utilizarão o produto, além de aumentar potenciais de negócio. Outra característica é a adoção do princípio de “endpoints inteligentes e pipes leves”, em que os serviços se comunicam por protocolos simples, mantendo o sistema desacoplado e coeso, uma abordagem que facilita a escalabilidade e reduz a sobrecarga da infraestrutura.

Mais uma característica comum dos microsserviços é a governança descentralizada, que permite às equipes escolherem as tecnologias mais adequadas para cada um dos serviços, ao invés de seguirem padrões rígidos impostos por algum cargo central. Com isso, as equipes assumem a responsabilidade por todo o ciclo de vida do produto, garantindo alinhamento e qualidade desde o início do desenvolvimento. A descentralização da gestão de dados, outra característica, resulta na ideia de que cada serviço tenha o seu próprio modelo de banco de dados, refletindo diferentes perspectivas sobre o projeto. Essa abordagem beneficia a autonomia dos serviços e permite o uso de tecnologias variadas, adaptadas às necessidades específicas de cada contexto. A automação de infraestrutura é outra característica dos microsserviços e está diretamente relacionada à prática de integração e entrega contínua, permitindo que os processos de testes e implantação ocorram de forma ágil e segura. Há o investimento em pipelines automatizados, devido ao avanço da computação em nuvem, o que torna a coordenação de múltiplos serviços mais confiável e eficiente. Essa abordagem reduz falhas humanas, acelera o processo de desenvolvimento e garante maior consciência na entrega do software.

O design para falhas e o design evolutivo são outras duas características comuns dos microsserviços. O design para falhas parte do reconhecimento de que erros podem ocorrer a qualquer momento, exigindo que o sistema esteja preparado para lidar com essas falhas de forma transparente. Já o design evolutivo se baseia no princípio de que as mudanças constantes no software são não apenas inevitáveis, mas desejáveis. Por isso, a divisão em serviços independentes é vista como uma estratégia que facilita a adaptação contínua e o desenvolvimento ágil do produto.

Diante das características analisadas ao longo deste artigo, os microsserviços revelam-se uma alternativa arquitetônica favorável para o desenvolvimento de sistemas modernos. Embora possam vir a se consolidar como uma abordagem permanente em projetos de software, sua adoção exige equipes qualificadas e um planejamento estratégico rigoroso para que os benefícios realmente se concretizem. Apenas com essa base sólida será possível promover uma transformação positiva na forma como construímos e evoluímos os sistemas.