

Artigo: Resenha do artigo Microservices

Autores do Artigo: James Lewis e Martin Fowler

Aluno: Pedro Henrique Pires Rodrigues

Matéria: Projeto de Software

Matrícula: 816373

O artigo em questão, como o próprio título evidencia, trata sobre a arquitetura de microservices. Essa é uma alternativa aos monólitos, da qual invés de se ter toda a lógica do back-end concentrada em uma única aplicação, se tem o código distribuído entre várias aplicações. O trabalho ainda diz que por ser algo recente, ainda não há uma definição muito formal do que se caracteriza um micro serviço, mas que vai buscar discorrer sobre o assunto por meio das características mais comuns entre eles.

O artigo começa apontando algumas vantagens enquanto explica seu funcionamento, afirmando que a arquitetura de microservices possibilita uma melhor divisão dos serviços em servidores de forma a dedicar mais processamento a aqueles que necessitam de mais recursos. Além de também possibilitam trabalhar com linguagens de programação diferentes aumentando a versatilidade

A pesquisa aponta que uma das principais vantagens dessa arquitetura é o uso de serviços como componentes, e reforça a positividade de tal dizendo que mais que uma biblioteca, um serviço quando separado de outro tem uma independência ainda mais significativa. Pois quando uma *lib* atualiza, ou mesmo uma parte qualquer do código fonte de uma aplicação, ela precisa ser inteiramente recompilada. Apesar de haverem diversos processos que os compiladores fazem atualmente para evitar retrabalho, o tempo de compilação ainda é uma espera necessária e considerável no desenvolvimento de qualquer software. Porém, quando usamos microsserviços, eles são independentes um dos outros, de forma que mesmo alterando o código de um deles, o outro não precisará passar por um processo de recompilação, a menos que seu código também seja alterado.

Ainda sim, há também o ponto negativo em relação à dificuldade em mudar uma responsabilidade de um componente para o outro. E além disso também a um aumento na burocracia na comunicação entre dois componentes, visto que esse é feito por meio de protocolos da web como

http, que exigem mais tempo e tratativas para serem interpretados pelo outro lado.

Outra vantagem dessa abordagem, é o fato de possibilitar uma melhor divisão de tarefas, podendo dividir em times de desenvolvimento que interagem entre si, invés de um grande time único. O artigo até fala sobre a possibilidade de se fazer isso em um monólito, o que tende a ser muito mais complicado por estarem muito próximos contextualmente falando. O que é diferente quando falamos de microsserviços, porque como já mencionado anteriormente, um serviço se torna independente do outro, de forma que uma modificação de um lado não necessariamente vai afetar o outro. Tornando factível uma separação de times, que apesar de se comunicarem, não precisam a todo momento ter plena noção do que cada um dos integrantes do outro time está desenvolvendo, por exemplo.

Um ponto a se observar é o quão simples se tornam as interfaces entre os componentes. Em um monolito, por exemplo, por muita das vezes é necessário fazer chamadas a diferentes camadas ou entidades na aplicação de forma a aumentar o acoplamento entre elas. Já em um micro serviço, tudo é feito entre aplicações por um simples protocolo web, o que dinamiza e facilita a comunicação entre as pontas.

A descentralização da camada de dados também é uma vantagem constantemente explorada pela arquitetura. Dessa forma é possível escolher de SGBDs diferentes para menor dependência de tecnologias específicas. Como também é completamente possível ter uma base de dados em um modelo que não seja nem mesmo relacional para lidar com parte desses dados. Essa é uma vantagem que também pode ser explorada por um monolito, mas utilizando um micro serviço tal abordagem é muito mais prática. Isso porque é possível ter uma aplicação server-side inteira para fazer toda a comunicação com um banco de dados específico, despreocupando todo o resto do projeto em ter que lidar com outra estrutura de banco de dados.

Apesar disso, outra consequência negativa do uso de micro serviços é a dependência da estabilidade do servidor para a comunicação entre dois serviços distintos. Ao menos essa é uma afirmação do artigo, da qual ainda quer ser verdadeira, pode se ver como uma outra forma de se analisar uma vantagem. Pois utilizando essa divisão de serviços em servidores diferentes, por mais que um deles venha a cair, os outros continuam funcionando, de modo a minimizar o dano sofrido pelo usuário em uma queda de algum servidor.

O artigo finaliza afirmando acreditar que esta é uma abordagem interessante para as aplicações do futuro, e que o mercado de tecnologia

atual parece ter uma tendência a se voltar para esse lado. A componentização oferecida por essa organização é de fato benéfica ao desenvolvimento de softwares, facilitando a paralisação entre os serviços tanto no sentido de hardware, quanto no sentido de recursos humanos. E sem dúvida, essa é uma abordagem que tende a crescer ainda mais.