

MAC0499 - Trabalho de Formatura Supervisionado

Proposta de trabalho

Thiago Cunha Ferreira

23 de maio de 2021

Esta é a proposta para o meu Trabalho de Conclusão de Curso, que tem como tema de estudo **o processo de transição entre sistemas monolíticos para microsserviços**, mais especificamente relacionado às dificuldades encontradas durante esse processo.

Introdução

A arquitetura de microsserviços é uma abordagem para desenvolvimento de aplicações e sistemas complexos que vem se popularizando desde os anos 2010. Ela surgiu como um conjunto de práticas e princípios de desenvolvimento de *software* baseados em conceitos que estiveram sendo explorados há um certo tempo na indústria (como "*Service-oriented architecture*", "*Domain-driven design*", dentre outros) e, através de uma grande quantidade de trabalhos exploratórios na academia e, principalmente, a partir da adoção precoce dessa arquitetura por empresas, algumas das quais hoje são gigantes do mercado (como Netflix, Amazon e Spotify), a ideia de **microsserviços** se desenvolveu e se popularizou a ponto de ser uma das mais novas tendências no jeito de se estruturar aplicações.

Para situar o que representa exatamente a arquitetura de microsserviços, é interessante primeiro conhecer como a maioria das aplicações foram, e ainda são, estruturadas: através do que se chama "sistemas monolíticos". A arquitetura utilizada nesses sistemas envolve manter **a aplicação como uma unidade única**, onde partes distintas do código, com responsabilidades distintas, estão agrupadas sobre a mesma base de código e altamente interligadas por chamadas de funções. Além disso, por ser uma única unidade, o processo de **deploy** e **distribuição de carga** do sistema deve também ser feitas de modo unitário. Principalmente para aplicações naturalmente mais complexas, essas características trazem diversos prejuízos e desvantagens a longo prazo nos processos de desenvolvimento e manutenção, na eficiência no uso de recursos e na produtividade de novos desenvolvedores, mas é uma estrutura relativamente comum devido a sua **relativa simplicidade em diversos aspectos**, como na implantação e desenvolvimento inicial, na realização de testes, no processo de *deploy* e no aumento na escala da aplicação (ainda que isso seja feito de forma

relativamente ineficiente).

Sabendo disso, podemos perceber o padrão de microsserviços como o completo oposto da arquitetura anterior em quase todos os sentidos. Apesar de não haver uma definição unificada do que faz parte da arquitetura, há um certo número de elementos em comum: ela é formada por um conjunto de microsserviços, **unidades autônomas** (com *deploys* individuais e geralmente com bases de códigos separadas) e **suficientemente pequenas** que representam a lógica de funcionamento completa de uma **pequeno domínio da aplicação**, que **interagem entre si** (seja de forma coordenada entre serviços ou orquestrada por um serviço central) através de protocolos leves de comunicação em rede, assim formando a aplicação final. Muitos outros conceitos e práticas atualmente populares no mercado, como o *deploy* de sistemas na nuvem, a adoção de metodologias ágeis e o uso de *pipelines* de *Continuous Delivery* e *Continuous Integration*, **não** fazem parte necessariamente da arquitetura de microsserviços, mas se integram relativamente bem com ela.

Existem diversos benefícios associados à arquitetura de microsserviços que surgem especialmente de sua natureza distribuída e orientada a serviços, como a maior facilidade de desenvolvimento e *deploy* de um componente (microsserviço), melhor ajuste da aplicação à estrutura da organização, mais autonomia para os times, maior facilidade de introdução de novas tecnologias, dentre muitos outros. Assim, não é de se estranhar que muitas empresas e times que trabalham no desenvolvimento e manutenção de aplicações monolíticas desejem realizar a transição dos seus sistemas para essa arquitetura, principalmente quando elas se expandiram tanto e de tal forma que tornaram a sua manutenção um processo complexo, rígido e, em geral, desagradável (e a sua eficiência, baixa). Porém, o que muitas vezes se esquece é que essa não é uma solução mágica. Proceder com essa transição não é algo nada simples e, se não for feita adequadamente, pode trazer mais prejuízos que benefícios. E mesmo assim, sempre haverá uma série de desvantagens e complicações trazidas para a aplicação e aos times responsáveis por ela, a maioria vinda de um **um aumento geral na complexidade**.

Não é de hoje que se conhece o quanto a complexidade geral de uma aplicação aumenta ao transitamos de um funcionamento local (numa máquina só) para um sistema distribuído. Sendo microsserviços um exemplo desse tipo de sistema, eles também apresentam essas complicações. Juntando isso ao fato deles estarem estruturados não ao redor de componentes lógicos, mas sim ao redor de domínios abstratos da aplicação, temos que um grande número de restrições e imposições devem ser feitas sobre como a aplicação é estruturada, desenvolvida e testada para que obtenha as vantagens prometidas pela arquitetura, também aumentando a complexidade. Além disso, em casos onde se planeja trocar a arquitetura de uma aplicação monolítica para microsserviços, há complicações adicionais no processo de quebra e reestruturação do sistema. Considerando tudo isso e mais o fato de, por ser uma proposta razoavelmente nova, haver menor disponibilidade de conhecimento técnico, ferramental e arquitetural por parte dos times responsáveis pelas aplicações,

muitas implementações de sistemas em microsserviços acabam sendo feitas de forma não ideal, perdendo assim parte ou quase todas as vantagens oferecidas pela arquitetura e ainda mantendo a sua complexidade.

Por fim, conclui-se que é necessário haver, por parte de times que pensam em fazer a migração de um sistema monolítico para microsserviços, consideração por dois pontos: o primeiro são as vantagens e desvantagens de realizar essa transição e se fazer isso é o melhor caminho ou não (é necessário também considerar o ambiente específico ao redor de cada aplicação). O segundo seriam os possíveis *pitfalls* e problemas que possam surgir durante o processo, caso se decida por continuar, como forma de evitar complicações futuras que possam impactar na eficácia da aplicação final. Porém, devido ao *hype* ao redor do assunto, é comum que esses pontos sejam poucos explorados e que a transição seja considerada como um "passo natural" da aplicação, o que pode gerar problemas futuros. Este trabalho está preocupado justamente com explorar esses dois pontos.

Objetivo

Este Trabalho de Conclusão de Curso tem por objetivo explorar as dificuldades e complicações existentes no processo de transição de sistemas monolíticos para sistemas estruturados em microsserviços. Isso envolve tanto as complicações que surgem pela própria natureza complexa da arquitetura quanto pelo processo não trivial de desestruturar a aplicação (ou parte dela) e depois adequá-la às recomendações da arquitetura. Os tipos de dificuldades que podem ser encontradas nesse processo são diversas, indo desde a divisão dos domínios da aplicação até a coleta de métricas para avaliação de desempenho do sistema.

Espera-se que com este trabalho, um leitor que faz parte de um time que pretenda realizar essa transição em sua aplicação possa ficar mais ciente do contexto ao redor da arquitetura de microsserviços, das motivações e dificuldades associadas ao processo e, por fim, considerar se essa mudança é realmente algo necessário ou até mesmo vantajoso para o futuro de sua aplicação. Caso decida prosseguir com a mudança, espera-se que o leitor tenha algumas noções sobre o assunto que poderão auxiliá-lo no processo de transição.

Passo-a-passo

A realização do trabalho irá se basear principalmente na exploração de tópicos destacados em estudos de casos, trabalhos acadêmicos sobre aspectos específicos da arquitetura e, caso necessário, materiais sobre tecnologias específicas. Ela será ordenada da seguinte forma:

- **Passo 1:** Conhecer mais sobre o cenário envolvendo microsserviços (e sua relação com sistemas monolíticos)

Inicialmente, é recomendável ter, além de uma visão técnica dos componentes associados ao tópico "microserviços", um conhecimento geral da área que a cerca, o que envolve sua história, sua atual posição e papel no mercado e como é vista por ela e pela academia. O primeiro passo está relacionado a isso, servindo como uma introdução ao assunto.

- **Passo 2:** Estudar sobre detalhes mais técnicos da arquitetura

A proposta do trabalho está associada ao processo de transição de sistemas monolíticos para microserviços e suas dificuldades, então é essencial que, para se poder discutir com base os possíveis empecilhos ao processo, conheça-se sobre os aspectos técnicos que compõem a arquitetura de microserviços e a monolítica, bem como procedimentos recomendados para essa transição.

- **Passo 3:** Anotar pontos e tópicos de interesse presentes em materiais de pesquisa

Após ter uma visão ampla do tópico com os dois passos anteriores, é hora de coletar informações de forma concreta. Nessa etapa, utiliza-se tanto dos trabalhos e materiais coletados durante os passos 1 e 2 quanto dos itens relacionados a eles (referências) para coletar pontos interessantes para serem incluídos ao trabalho a partir de notas resumidas. Os materiais envolvidos aqui são de diversos tipos, incluindo livros técnicos, estudos de casos, *papers* e vídeos de conferências, *posts* em *blogs*, etc.

- **Passo 4:** Escrever introdução da monografia

Essa etapa, que pode ser feita simultaneamente com o passo 3, envolve pegar os conhecimentos obtidos principalmente dos passos 1 e 2 para desenvolver a introdução da monografia.

- **Passo 5:** Unir anotações por temas similares, escolher quais serão inclusas no trabalho e desenvolver pontos na monografia

Planeja-se que o trabalho consista de um conjunto estruturado de tópicos associados à área de microserviços (e possivelmente áreas associadas) que representem um empecilho ou dificuldade extra no processo de transição. Essa etapa cuidará de pegar as anotações feitas no passo 3, uni-las a partir da similaridade de temas entre elas, decidir quais pontos serão discutidos no trabalho e então desenvolver os tópicos a partir das notas unificadas, dos conhecimentos obtidos nos passos 1 e 2 e, caso necessário, utilizando materiais complementares.

- **Passo 6:** Escrever conclusão da monografia

Por fim, conclui-se o trabalho recapitulando os pontos desenvolvidos anteriormente, fazendo menção ao *hype* envolvendo o assunto, à necessidade de se entender as razões por trás da migração para microserviços e do possível abandono da ideia e a como deve-se atentar para os possíveis *pitfalls* desse processo.

Cronograma aproximado

		Meses								
		A	M	J	J	A	S	O	N	D
Passos	1	X								
	2	X	X	X						
	3			X	X					
	4				X	X				
	5					X	X	X	X	
	6								X	X

Tabela 1: Tempo estimado para as tarefas. Colunas representam meses (de abril a dezembro). Linhas representam os passos especificados na seção anterior (passo 1 a 6). As marcações em X indicam que se planeja realizar as atividades da linha correspondente durante o mês da coluna correspondente.