UNIVERSIDADE FEEVALE

JONAS RAFAEL COLLING

TRANSFORMAÇÃO DE MÓDULOS DE UM *E-COMMERCE* DE ARQUITETURA MONOLÍTICA EM MICROSSERVIÇOS

(Título Provisório)

Anteprojeto de Trabalho de Conclusão

Novo Hamburgo

2021

JONAS RAFAEL COLLING

TRANSFORMAÇÃO DE MÓDULOS DE UM *E-COMMERCE* DE ARQUITETURA MONOLÍTICA EM MICROSSERVIÇOS

(Título Provisório)

Anteprojeto de Trabalho de Conclusão de Curso, apresentado como requisito parcial

à obtenção do grau de Bacharel em

Ciência da Computação pela

Universidade Feevale

Orientador: Juliano Varella de Carvalho

Novo Hamburgo

2021

RESUMO

Softwares desenvolvidos até poucos anos atrás eram em sua totalidade construídos no formato tradicional, com arquitetura monolítica, sendo compostos por um único serviço que respondia por todas funcionalidades da aplicação. Normalmente, para trazer novos negócios, incrementar receitas, evoluir o produto ou apenas acompanhar o mercado, as empresas evoluem seus sistemas constantemente. Esta evolução tende a trazer uma complexidade enorme para as aplicações. Tal complexidade acarreta em muitos problemas para a equipe que mantém o produto e o sistema acaba se tornando caro para as empresas. A saída para esta problemática tem sido, em muitos casos, a transformação da arquitetura do sistema. A estratégia arquitetural utilizada na construção dos sistemas tem sido peça fundamental no ambiente tecnológico. Uma arquitetura bem pensada contribui para a manutenção da aplicação, realizando entregas mais rápidas, além da utilização otimizada de recursos computacionais. Sendo assim, este trabalho tem como objetivo transformar módulos de um *e-commerce* já existente com arquitetura monolítica em uma arquitetura microsserviços, trazendo características das arquiteturas, além de apresentar os desafios e soluções para esse tipo de transformação.

Palavras-chave: Microsserviços. Arquitetura de software. Transformação de arquitetura. Evolução de sistemas legados. Arquitetura monolítica.

SUMÁRIO

MOTIVAÇÃO ...........................................................................................................................5

OBJETIVOS ..............................................................................................................................8

METODOLOGIA ......................................................................................................................9

CRONOGRAMA ....................................................................................................................11

BIBLIOGRAFIA ....................................................................................................................13

**MOTIVAÇÃO**

Com a evolução do mercado e das pessoas, as necessidades sistêmicas são modificadas a cada dia, os sistemas precisam ser construídos com mais rapidez e se tornam cada vez maiores e mais complexos. Para a construção dos sistemas, a engenharia de software precisa desenvolver novas técnicas para enfrentar os desafios e fornecer o software mais completo (SOMMERVILLE, 2019).

Definir a arquitetura do software é um dos itens de maior importância no planejamento de um sistema. Na sua evolução eles podem rapidamente se tornar muito complexos e caros na manutenção (SOMMERVILLE, 2019).

A arquitetura monolítica foi uma abordagem tradicional no desenvolvimento dos sistemas, usado em grande escala em empresas como Amazon e Ebay (LAURETIS, 2019, tradução nossa). Segundo SARITA; SEBASTIAN (2017, tradução nossa) a arquitetura monolítica é uma maneira fácil e comum de desenvolver e implantar um aplicativo de software, como uma unidade única de código, tendo todas funcionalidades reunidas. Essa vantagem normalmente é válida para pequenas ou médias aplicações, onde existe um pequeno time de tecnologia durante sua evolução.

Os sistemas legados monolitos normalmente estão sempre crescendo em tamanho e complexidade, tornando sistemas monstruosos após algum tempo de desenvolvimento e as desvantagens desta arquitetura acabam sendo maiores que as vantagens (KAZANAVICIUS; MAZEIKA, 2019, tradução nossa).

De acordo com SARITA; SEBASTIAN (2017, tradução nossa), aplicações monolíticas têm grandes limitações:

* O forte acoplamento entre os módulos interromperá um aplicativo de estar pronto para SaaS (Software como Serviço), onde houver a necessidade de publicar as mudanças várias vezes ao dia. Atualizar um único módulo garante a necessidade de construir e implantar todo o aplicativo novamente;
* Difícil de alcançar escalabilidade para recursos conflitantes para diferentes módulos dentro de um aplicativo;
* Difícil de adotar novas linguagens e estruturas;
* Entrega e *deploy* contínuo.

Por outro lado, a arquitetura em microsserviços procura resolver alguns problemas e limitações da arquitetura monolito (PRASANDY *et al*., 2020, tradução nossa). Nessa abordagem, o software consiste em pequenos serviços independentes, que se comunicam usando APIs (*Application Programming Interface*) bem definidas. Como são executados de forma independente, cada serviço pode ser atualizado, implantado e escalado para atender a demanda de funções específicas de uma aplicação.

Essa forma de desenvolvimento em serviços independentes traz alguns benefícios para a manutenção dos sistemas, como por exemplo: agilidade, pois os microsserviços podem ser atualizados de forma independente, sem a necessidade de reiniciar o aplicativo completamente; a falha de um serviço afeta apenas aquele módulo e seus consumidores; cada microsserviço pode ser escalável de forma separada, de acordo com a necessidade da aplicação (KAZANAVICIUS; MAZEIKA, 2019, tradução nossa).

De acordo com KAZANAVICIUS; MAZEIKA (2019, tradução nossa), a transformação do sistema monolito deveria ser modernizado para microsserviços quando:

* Monolito torna-se muito grande ou complexo para manter ou ampliar;
* Modularidade e descentralização são importantes aspectos;
* Preferência por benefícios de longo prazo em comparação aos de curto prazo.

O *e-commerce* a ser utilizado na transformação utiliza arquitetura monolito. O sistema foi implementado a partir de uma plataforma de *e-commerce* de mercado. Plataforma de comércio eletrônico, que contempla venda de produtos para pessoas físicas e empresas. Sem customizações, a plataforma já entrega serviços para gerir o *e-commerce*, porém normalmente cada empresa tem suas peculiaridades no modelo de negócio. Pensando nisso, é possível customizar todas funcionalidades, ou até mesmo a adição de novas.

O sistema foi lançado no ano de 2014. Naquele momento apenas um *e-commerce* era hospedado nesta aplicação e uma equipe composta de oito pessoas atuava na evolução dele. Atualmente, existem oito *e-commerces* e três aplicativos *mobile* rodando nesta estrutura. Para sustentar este ambiente tecnológico, mais de sessenta pessoas estão distribuídas em oito *squads*.

O projeto cresce desde o lançamento. Funcionalidades foram adicionadas e customizadas, integrações com plataformas externas foram criadas, *APIs* desenvolvidas e o aumento de complexidade é cada vez maior.

Mesmo sem customizações, os módulos deste *e-commerce* estão altamente acoplados, com uma forte dependência entre eles. Esta característica gera uma dependência não só no sistema, mas também entre as *squads*. As modificações se tornam complicadas, pois o desenvolvedor não impacta apenas a funcionalidade de sua responsabilidade, as demais *squads* têm grandes chances de serem afetadas. Esse nível de dependência somado a toda complexidade do sistema dificulta a entrega de software contínuo, tornando a implantação e desenvolvimento lentos e dolorosos (SARITA; SEBASTIAN, 2017, tradução nossa).

Este trabalho propõe a transformação de módulos do sistema *e-commerce* de arquitetura monolítica para uma arquitetura moderna em microsserviços. O trabalho, através da utilização de boas práticas da arquitetura microsserviço, tem por objetivo indicar melhores maneiras na realização desse tipo de transformação, levando em conta que a evolução do sistema atual não para durante a execução. Além disso, também se direciona para a resposta da seguinte questão de pesquisa: Com a aplicação da nova arquitetura, é possível diminuir a complexidade e o acoplamento entre *squads* nas entregas de novas funcionalidades no *e-commerce* já existente?

**OBJETIVOS**

**Objetivo geral:**

Transformar módulos de um *e-commerce* já existente com arquitetura monolítica em uma arquitetura microsserviços, a fim de apresentar desafios e soluções para esse tipo de transformação.

**Objetivos específicos:**

* Discutir vantagens e desvantagens de sistemas monolíticos e sistemas microsserviços.
* Investigar módulos e funcionalidades do sistema monolítico já existente.
* Definir quais módulos serão implementados na transformação.
* Detalhar a ordem da transformação e as entregas dos módulos que serão extraídos para microsserviços.
* Definir estratégia da transformação, levando em conta que a evolução do sistema não para nem mesmo durante a migração.
* Detalhar módulos transformados.
* Indicar boas práticas no processo de transformação;
* Validar se a arquitetura proposta diminui a complexidade e o acoplamento entre os *squads*

METODOLOGIA

Para a construção do conhecimento na realização de uma pesquisa acadêmica são realizados técnicas e procedimentos, com propósito de comprovar a validade e utilidade das informações. Tais técnicas e procedimentos são denominadas de Metodologia (PRODANOV; FREITAS, 2013). Segundo PRODANOV; FREITAS (2013), metodologia é uma disciplina que consiste em estudar, compreender e avaliar os vários métodos disponíveis para a realização de uma pesquisa acadêmica.

A metodologia utilizada para desenvolvimento deste projeto será *Design Science Research* (DSR). Esta metodologia é composta por seis etapas principais que permitem avaliar a eficácia dos resultados.

Segundo JUNIOR *et al.* (2017), baseado em PEFFERS (2007), as etapas são descritas abaixo juntamente com a aplicação da transformação de módulos de um *e-commerce* já existente com arquitetura monolítica em uma arquitetura microsserviços:

1. **Identificação do problema e motivação:** buscar na literatura principais estratégias na transformação de sistemas de arquitetura monolito para microsserviços. Como motores de busca serão utilizados o *Web of Science*, ACM e IEEExplorer, que reúnem artigos de várias origens de dados;
2. **Definição dos objetivos para a solução:** definir quais módulos serão implementados na transformação, juntamente com a ordem de execução. Nesta etapa também serão definidas e detalhadas estratégias da transformação;
3. **Projeto e desenvolvimento:** desenvolver a transformação dos módulos escolhidos do *e-commerce* já existente para a arquitetura microsserviço, sem afetar as funcionalidades de cada módulo;
4. **Demonstração:** executar o *e-commerce* com os módulos extraídos na nova arquitetura, mostrando no detalhe o funcionamento das partes transformadas e afetadas;
5. **Avaliação:** aplicar uma pesquisa direcionada para pessoas envolvidas no sistema atual: que atuam na evolução e que operam a ferramenta. Será avaliado se a nova arquitetura contribui para entregas mais rápidas e a diminuição do acoplamento das *squads* envolvidas;
6. **Comunicação:** realizar a apresentação final do TCC.

CRONOGRAMA

Trabalho de Conclusão I

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Etapa | Meses | | | |
| Ago | Set | Out | Nov |
| Anteprojeto |  |  |  |  |
| Buscar na literatura principais estratégias na transformação de sistemas monolíticos em microsserviços |  |  |  |  |
| Estudar principais características dos dois tipos de arquitetura |  |  |  |  |
| Analisar módulos do sistema *e-commerce* já existente |  |  |  |  |
| Definir quais módulos serão transformados para a nova arquitetura |  |  |  |  |
| Elaborar TC 1 |  |  |  |  |

Trabalho de Conclusão II

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Etapa | Meses | | | |
| Mar | Abr | Mai | Jun |
| Definir tecnologias utilizadas no desenvolvimento |  |  |  |  |
| Definir microsserviços a serem criados |  |  |  |  |
| Desenvolver a transformação |  |  |  |  |
| Validar a transformação dos módulos |  |  |  |  |
| Elaborar TC 2 |  |  |  |  |

BIBLIOGRAFIA

HASSAN, Sara; ALI, Nour; BAHSOON, Rami, ***Microservice Ambients****: An Architectural Meta-modelling Approach for Microservice Granularity.* 2017.

JUNIOR, Vanderlei *et al.* ***Design Science Research Methodology As Methodological Strategy for Technological Research***. *Revista Espacios*, 2017. vol. 38, p. 25.

KAZANAVICIUS, Justas; MAZEIKA, ***Dalius. Migrating Legacy Software to Microservices Architecture***. *Open Conference of Electrical, Electronic and Information Sciences*, 2019.

LAURETIS, Lorenzo De, ***From Monolithic Architecture to Microservices Architecture***. IEEE *International Symposium on Software Reliability Engineering Workshops*, 2019.

MUNIZ, Antonio, **Jornada DevOps:** unindo cultura ágil, Lean e tecnologia para entrega de software de qualidade, 2019. vol. 1.

PRASANDY, Teguh *et al*., ***Migrating Application from Monolith to Microservices***. IEEE *International Conference on Information Management and Technology*, 2020.

PRESSMAN, Roger, **Engenharia de Software**.Uma Abordagem Profissional, 2016. 8º ed.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do trabalho científico:** métodos e técnicas da pesquisa e do trabalho acadêmico. 2º ed. Novo Hamburgo: Feevale, 2013.

SARITA; SEBASTIAN, Sunil, ***Transform Monolith into Microsservices using Docker***. IEEE *International Conference on Computing, Communication, Control and Automation*, 2017.

SOMMERVILLE, Ian, **Engenharia de Software**. Editora Pearson, 2019. 10º ed.