UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

JOÃO VICTOR GOULART DE ALMEIDA

MODELO DE REFATORAÇÃO DE ARQUITETURA MONOLÍTICA PARA UMA ARQUITETURA BASEADA EM MICROSSERVIÇOS

DISSERTAÇÃO

CORNÉLIO PROCÓPIO
2018

RESUMO

ALMEIDA, João Victor Goulart. **Modelo de Refatoração de Arquitetura Monolítica para uma Arquitetura Baseada em Microsserviços**. . 9 f. Dissertação – , Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

Com o tempo aplicações corporativas tendem a crescer muito se tornando grandes monólitos, os famosos sistemas legados. Com isso, foi idealizado algumas soluções para resolver esse problema, uma delas é refatorar o sistema em uma arquitetura baseada em microsserviços. Porém, existem diversos desafios complexos no meio do caminho, este trabalho propõe um modelo de refatoração que tem como objetivo amenizar e evitar ao máximo possíveis problemas para que se consiga atingir objetivo final, que é quebrar esse monólito gigante em pequenos microsserviços.

Palavras-chave: Monólito. Microsserviços. Refatoração

ABSTRACT

ALMEIDA, João Victor Goulart. **Monolithic Architecture Refactoring Model for a Microservice Based Architecture.** . 9 f. Undergraduate thesis – Software Engineering Graduate Program, Federal University of Technology - Paraná. Cornélio Procópio, 2018.

Over time corporate applications tend to grow greatly becoming large monoliths, the famous legacy systems. With this, some solutions have been devised to solve this problem, one of them is to refactor the system in a microservice-based architecture. However, there are several complex challenges in the middle of the way, this work proposes a refactoring model that aims to minimize and avoid possible problems in order to achieve the final goal, which is to break this giant monolith into small microservices.

Keywords: Monolith. Microservices. Refactoring.

SUMÁRIO

1 INTRODUÇÃO

Grande parte das soluções desenvolvidas por organizações se transformam em sistemas monolíticos. Quando isso ocorre a manutenção se torna problemática, por conta do seu tamanho enorme e regras de negócios complexas. A solução para este problema é refatorar o sistema em uma arquitetura baseada em microsserviços, porém, essa é uma jornada bem complicada.

As empresas que optam por essa solução estão buscando tornar a aplicação escalável e facilitar possíveis mudanças. Com a solução desfragmentada em microsserviços tornará possível equipes trabalharem de forma paralela em partes diferentes do sistema, sem ter dependência um do outro. De forma geral, a empresa busca diminuir seu custo de manutenção e aumentar a escalabilidade.

Mas quando decidido seguir essa jornada é encontrado diversos desafios de arquiteturais para conseguir decompor o monólito em um ecossistema de microsserviços. De acordo com uma pesquisa de 2015 feito pela NGINX, 68% estão usando ou estudando a possiblidade de utilizar microsserviços. Em 2 anos esse número cresceu para 80%, em uma pesquisa feita pela LeanIX. A arquitetura de microsserviços já se tornou um assunto recorrente no dia a dia de desenvolvimento, e com isso essas organizações tentam implementar não somente em seus novos projetos mas também em suas soluções "legados"/monólitos.

1.1 PROPOSTA

A proposta desse trabalho(ABNTEX, 2009) é apresentar um modelo de refatoração de um sistema monólito para uma arquitetura baseada em microsserviços, detalhando todos processos necessários e possíveis caminhos a se tomar quando estiver enfrentando essa jornada. Para que possa servir como referência para desenvolvedores e engenheiros terem um contexto de como deve ser feito para minimizar ao máximo possíveis problemas durante essa fase de refatoração.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo desse trabalho é definir um modelo de refatoração de sistemas legados para uma arquitetura de microsserviços.

1.2.2 Objetivos Específicos

Os objetivos específicos consistem em:

- Pegar como base um sistema legado
- Refatorar ele passo a passo e documentar os desafios encontrados.
- Comparar o modelo executado com diferentes métodos de refatoração existentes.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 REVISÃO BIBLIOGRÁFICA

Existem alguns artigos descrevendo desafios de conseguir quebrar um monólito em microsserviços, mostrando suas dificuldades e cuidados a serem tomados. No Google Next '18 foi apresentado como que a sede Britânica deles tratam sobre o assunto, apresentando possíveis soluções utilizando seus produtos, mas nenhum define um modelo a ser seguido e nem mostra um cenário real dessa refatoração.

2.2 ARQUITETURA MONOLÍTICA

Linguagens de programação oferecem formas de modularizar a aplicação. Contudo, essas soluções são projetadas para um único pacote executável monolítico, onde toda modularização é empacotada e executada na mesma máquina. Nesse pacote são compartilhados os mesmos recursos de máquina, memória e banco de dados.

Uma arquitetura monolítica, onde todos recursos são compartilhados, pode ser representado pela figura abaixo.

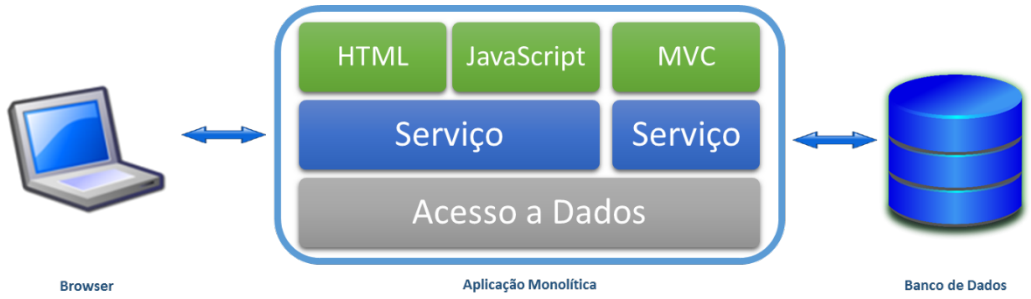

Figura 1 – Representação de uma arquitetura monolítica

Com o passar do tempo, a solução vai crescendo e tornando-se mais complexo, utilizando ainda mais recursos. E para manutenção tempos alguns desafios como:

- Aumento no tamanho e complexidade ao longo do tempo: A solução se torna tão complexa que toda mudança é cara e lenta, forçando o desenvolvedor a navegar em milhões de linhas de código.
- Alto acoplamento: A manutenção em um sistema monólito afeta a aplicação como um todo, as funções e serviços estão entrelaçados, de forma que uma alteração pode causar comportamentos inesperados no sistema.
- Escalabilidade baixa: Se somente uma funcionalidade do sistema exige um processamento mais alto, será necessário escalar a aplicação inteira para consumir mais recurso, trazendo assim custos maiores para a empresa.

2.3 ARQUITETURA BASEADA EM MICROSSERVIÇOS

O termo **microsserviços** foi discutido pela primeira vez em uma conferência de arquitetos de software em 2011 para demonstrar um estilo de arquitetura de sistemas. A filosofia por de trás desse conceito é "fazer somente uma coisa e fazer bem feito", que basicamente é focar os serviços em realizar uma única função.

Essa arquitetura é utilizada com objetivo de construir um conjunto de pequenos serviços, cada um auto suficiente, tendo sua própria e única funcionalidade. Cada serviço construído possui sua regra de negócio específico, e são independentes um do outro.

Dito isso, a arquitetura baseada em microsserviços tem a capacidade de ultrapassar obstáculos impostos por uma arquitetura monolítica, como:

- Diminuição nos custos de manutenção e evolução dos serviços: A manutenção não será aplicada no sistema inteiro e sim na funcionalidade, é um escopo menor, menos tempo e mais fácil, logo, é mais barato. E cada serviço não irá crescer indefinidamente conforme o crescimento do sistema.

- Baixo acoplamento: A manutenção em um serviço não afeta a funcionalidade de outro. E caso uma mudança seja feita em algum serviço, colocar essa alteração em ambiente de produção é muito mais fácil, pois, estará atualizando um serviço somente e não o sistema como um todo.

- Escalabilidade: A implantação e replicação de microsserviços é feita por máquinas virtuais e container de maneira independente. Isso torna o sistema muito mais dinâmico e flexível.

3 METÓDOS E MATERIAIS

3.1 FERRAMENTAS

Será realizado um estudo de caso com objetivo de formalizar um modelo com as melhores soluções se atingir a glória da refatoração de um sistema legado em uma arquitetura baseada em microsserviços. Será utilizada como fontes artigos e conteúdos diversos existentes para formalização do modelo e, por fim, será conduzido um experimento utilizando um software legado real de mercado como modelo de testes.

3.2 DESENVOLVIMENTO

O desenvolvimento consistirá em utilizar um software monólito real, feito em JAVA voltada ao cenário WEB, e conseguir refatorar esse software em uma arquitetura de microsserviços. Por fim, será definido o modelo rafatoração, independente da linguagem, baseado na experiência real.

REFERÊNCIAS

ABNTEX. **Absurdas normas para TEX**. 2009. Disponível em: <http://sourceforge.net/apps/mediawiki/abntex/index.php>. Acesso em: 8 de novembro de 2009. Citado na página 4.

CTAN. **The comprehensive TEX archive network**. 2009. Disponível em: <http://www.ctan.org>. Acesso em: 8 de novembro de 2009. Nenhuma citação no texto.

JABREF. **JabRef reference manager**. 2009. Disponível em: <http://jabref.sourceforge.net>. Acesso em: 8 de novembro de 2009. Nenhuma citação no texto.

MENDELEY. **Mendeley:** academic software for research papers. 2009. Disponível em: <http://www.mendeley.com>. Acesso em: 8 de novembro de 2009. Nenhuma citação no texto.

MIKTEX. **The MiKTEX project**. 2009. Disponível em: <http://www.miktex.org>. Acesso em: 8 de novembro de 2009. Nenhuma citação no texto.

TEX-BR. **Comunidade TEX-Br**. 2009. Disponível em: <http://www.tex-br.org/index.php>. Acesso em: 8 de novembro de 2009. Nenhuma citação no texto.

TEXNICCENTER. **TEXnicCenter:** the center of your LATEX universe. 2009. Disponível em: <http://www.texniccenter.org>. Acesso em: 8 de novembro de 2009. Nenhuma citação no texto.

WIKIBOOKS. **LATEX**. 2009. Disponível em: <http://en.wikibooks.org/wiki/LaTeX>. Acesso em: 8 de novembro de 2009. Nenhuma citação no texto.