



Universidade Federal de Pernambuco

# Introdução a cultura DevOps

Iniciando com *microservices*

Augusto Lima (arl4)

Carlos Melo (chnm)

Pedro Neto (pjsn)

Rafael Jordão (rfpj)

## INTRODUÇÃO

As grandes empresas de desenvolvimento de software estão começando a exigir profissionais que sejam capazes de atuar tanto na área de desenvolvimento quanto na de operações. Esse novo profissional denominado DevOps exerce a função de automatizar e manter os processos da engenharia de software necessários para uma organização.

A disciplina de Seminário em SI 3 com foco em *microservices* ministrada pelo professor Vinicius Cardoso Garcia nos apresentou conteúdo teórico a respeito desse novo mundo DevOps e de todo ambiente, seja ele tecnológico ou cultural, ao seu redor. Ao término do período com aulas abordando o contexto desse novo profissional e de todo o aparato tecnológico que dar suporte para a realização das tarefas necessárias para criar um *pipeline* DevOps, a disciplina foi caminhando com uma abordagem mais prática para que os alunos fossem aprendendo a utilizar as ferramentas por meios de trilhas de atividades e adquirindo assim o conhecimento necessário para a realização do projeto final.

Portanto, o projeto final teve como objetivo realizar a migração de uma aplicação monolítica para um ambiente de *microservices*. Com isso, pretendeu-se melhorar o entendimento dos alunos a respeito das razões motivacionais para realizar essa migração aplicando todo conteúdo tecnológico visto na disciplina para construir um *pipeline* DevOps da aplicação *SOA-Monolith* disponibilizada pelo docente.

## RESULTADO E DISCUSSÃO

Para a realização do projeto final, nossa equipe subdividiu todo o processo em três fases, sendo elas: Aprender sobre as ferramentas e visão geral do projeto, realizar a comunicação entre os serviços e criar testes unitários e por fim, mas não menos importante, desenvolver a configuração como código dos projetos.

### FASE 1 - APRENDER SOBRE AS FERRAMENTAS E VISÃO GERAL DO PROJETO

A primeira fase constituiu-se na análise do código do projeto e no estudo de ferramentas necessárias para o desenvolvimento do mesmo, onde os membros da equipe ficaram responsáveis em aprender a utilizar e aplicar as ferramentas.

A equipe foi dividida em duas duplas, onde dois membros focaram o estudo nas ferramentas para realização de teste funcional e automatização de processos enquanto a outra dupla ficou responsável em realizar a migração do projeto para *microservices* e a correção do código da aplicação.

Nessa primeira fase aprendemos que o uso da ferramenta *Spring Initializr* facilita e otimiza a curva de tempo gasta para o *setup* inicial do projeto.

### FASE 2 – REALIZAR A COMUNICAÇÃO ENTRE OS SERVIÇOS E CRIAR TESTES UNITÁRIOS.

Nesta fase foram criados os testes unitários para o projeto *Cliente*, priorizando os métodos considerados críticos, como os quais que realizavam processos de criação de cliente, remoção do cliente, alteração de valores do cliente, etc. utilizando o *JUnit*, que é

uma biblioteca para testes automatizados para JAVA. A comunicação entre os serviços foi feita pelo uso de simples requisições *HTTP*, onde o endereço é o nome do serviço (*cliente* e *pedido*) configurado no *Docker Swarm*.

### FASE 3 – DESENVOLVER A CONFIGURAÇÃO COMO CÓDIGO DOS PROJETOS.

A última fase do projeto conteve a construção do *pipeline* em si, que foi feito usando a ferramenta *Jenkins* que está atualmente executando dentro de um container *Docker* que foi configurado em uma máquina virtual<sup>1</sup> do *google cloud*. Além disso, foi configurado um *Webhook* do *github* para execução do pipeline, ou seja, a cada commit novo na organização do projeto<sup>2</sup> no *github*, o *pipeline* é executado. A arquitetura inicial planejada foi a de utilizar o *ansible* para o deploy de cada projeto, porém durante os estudos tivemos o conhecimento que o *Ansible* ainda não possui suporte para o *Docker Service*, o qual usamos para replicar os containers usando o *Docker Swarm* que está configurado em duas máquinas virtuais no *google cloud*.

O pipeline segue os seguintes estágios:

- *Cliente Test*: Os testes unitários são executados dentro de um container temporário do *docker* criado utilizando a imagem pública do *maven*, garantindo assim a isolabilidade dos testes.
- *Cliente Package*: O empacotamento do projeto *Cliente* é feito utilizando o *docker*, transformando em uma imagem seguindo os passos presentes no *Dockerfile* situado no repositório.
- *Push Cliente Build*: Subida da imagem do *Cliente* para o *Docker-Hub*, utilizando o número da build do *Jenkins* para 'tagueamento' da imagem, bem como subindo também no lugar da *Tag latest*.
- *Deploy Cliente Image*: Com conexão direta ao *Host* do *Docker Swarm*, é feita a atualização do serviço, utilizando do comando *docker service update* que atualiza cada container por vez, seguindo as regras de *rolling* estabelecidas (apenas um update por vez com uma diferença de 60s para cada container).
- *Pedido Test*: Como no *Cliente Test* o mesmo é executado dentro de um container, porém o estágio conta com paralelismo pois os testes do projeto *Pedido* necessitam de que o serviço *Cliente* esteja ao seu alcance, ou seja, atualmente executando, para que seja possível a verificação do cliente.
- *Pedido Package, Push Pedido Build e Deploy Pedido Image*: Estágios idênticos aos mesmos de *Cliente*, porém para o projeto *Pedido*.

Apesar dos esforços da equipe, não conseguimos implementar a monitoração utilizando o *ElasticSearch* e o *Kibana*.

## CONCLUSÃO

A utilização da cultura DevOps, suas ferramentas e metodologias, amplificou melhorias no projeto desenvolvido e otimizou seus resultados e processos, dando um novo ar para a equipe que trabalhou com o DevOps.

Os membros do grupo adquiriram experiências de uma nova metodologia utilizada para realização de projetos e trabalhos em um ambiente ágil, conhecendo metodologias novas e se destacaram com o aprendizado, cada um com destaques em determinados pontos, onde alguns obtiveram uma melhoria em áreas como TDD e utilizações de ferramentas de testes automatizados pela máquina, e outros com a configuração das ferramentas para que todos os processos pudessem assim funcionar e dar procedimento ao projeto.

Por fim este trabalho nos proporcionou novas experiências em áreas de desenvolvimento, nos dando um novo leque de possíveis formas de melhoria de projetos e processos de trabalhos, e tornando cada membro do grupo um pouco mais adequado para trabalhar com desenvolvimento ágil e assim com DevOps.

## ARTEFATOS

<sup>1</sup> Endereço de acesso do jenkins: <http://35.194.135.177:8080>

<sup>2</sup> Endereço de acesso da organização: <https://github.com/WildCats-SOA>