



**Trabalho 8**  
**Documento de Artigos selecionados**  
**Primeira etapa**

**Automated Testing in DevOps**

**Arthur do Prado Labaki – 11821BCC017**  
**arthur.labaki@ufu.br**

## SUMÁRIO

INTRODUÇÃO .....	3
TRABALHOS RELACIONADOS .....	3
Docker .....	3
Mesos Apache .....	3
Gebish .....	4
Jenkins.....	4
DESCRIÇÃO DA PESQUISA.....	4
RESULTADOS ALCANÇADOS .....	5
CONCLUSÃO.....	5

## **INTRODUÇÃO**

A área de TI no setor de saúde é complexa, especialmente quando se está trabalhando em um sistema de adjudicação de sinistros (claims adjudication). Atualmente, uma grande parte da entrega de TI, nessa área, é feita em lançamentos mensais, trimestrais ou mais, devido à natureza complexa dos sistemas de saúde. Em virtude desse problema, surgiu a ideia de desenvolver uma nova plataforma de administração de saúde, usando ferramentas e softwares avançados. O objetivo primário é o suporte para lançamentos diários. Além disso, o projeto deve seguir um modelo de Integração Contínua e Entrega Contínua (CI/CD).

Para o programa, a fase de teste não será separada, ela será realizada em todo o ciclo de desenvolvimento. Os testes são feitos em um ambiente de nuvem, sendo totalmente automatizados em construção e implementação, além de testes de regressão 24 horas por dia. Com isso cria-se uma pirâmide de testes bem desenvolvidos, sem o uso de ambientes testes estáticos.

Para atingir os objetivos de negócios, a equipe de TI teve que redefinir completamente o software atual, usando técnicas de engenharia, processos e diferentes habilidades. Os principais objetivos de negócios são inovar e modernizar a tecnologia, aumentar a velocidade de lançamentos frequentes no mercado, reduzir metade dos custos de tecnologia e, por fim, melhorar a experiência do cliente.

## **TRABALHOS RELACIONADOS**

Os trabalhos relacionados com o artigo são:

### **Docker**

O Docker é uma plataforma de código aberto, desenvolvido na linguagem Go, que possui alto desempenho e é desenvolvida diretamente no Google. Ele agrupa partes de softwares de um sistema de arquivo completo e que abrange todos os recursos necessários para a sua execução, os containers, que são ambientes isolados.

### **Mesos Apache**

Apache mesos é um projeto de código aberto para gerenciar cluster de computadores, que são conjuntos de computadores conectados que funcionam juntos para que possam ser vistos como um único sistema.

## **Gebish**

Geb é uma solução de automação de navegador. Usado para scripts, scraping e automação em geral.

## **Jenkins**

Jenkins é um servidor de automação de código aberto, ele fornece centenas de pug-ins para ajudar a construção, implementação e automação de projetos.

## **DESCRIÇÃO DA PESQUISA**

O artigo mostra como transformou sua equipe de Controle de Qualidade em uma equipe de Engenharia de qualidade, executando mais de 1,5 milhão de testes por dia, de maneira automática, usando DevOps. O DevOps (Development e Operations) é um termo criado para descrever um conjunto de práticas para integração entre as equipes de desenvolvimento de softwares, operações e de apoio envolvidas (como controle de qualidade) e a adoção de processos automatizados para produção rápida e segura de aplicações e serviços.

É detalhado no artigo como os testes de pirâmide diferem do modelo cascata legado, além de demonstrar as tecnologias e ferramentas avançadas disponíveis no mercado que ajudaram a criar o ambiente de teste em tempo real, na nuvem. O modelo de entrega escolhido foi o Desenvolvimento Orientado ao Teste de Aceitação, tendo como critério que a estrutura de teste automatizado deve suportar CI/CD e outros artifícios de teste. CI/CD se refere as praticas de integração continua e entrega continua, para garantir uma entrega frequente de aplicações aos clientes.

Os membros da equipe de Garantia de Qualidade (QA) e os desenvolvedores não são obrigados a executar um teste manualmente, pois a ferramenta de integração continua implementa os testes automatizados, usando Gebish para criar a maioria dos scripts de automação, suportando os requisitos de entrega rápida e Jenkins para mesclar o código mais recente, construindo o software e implementando nos ambientes transitórios.

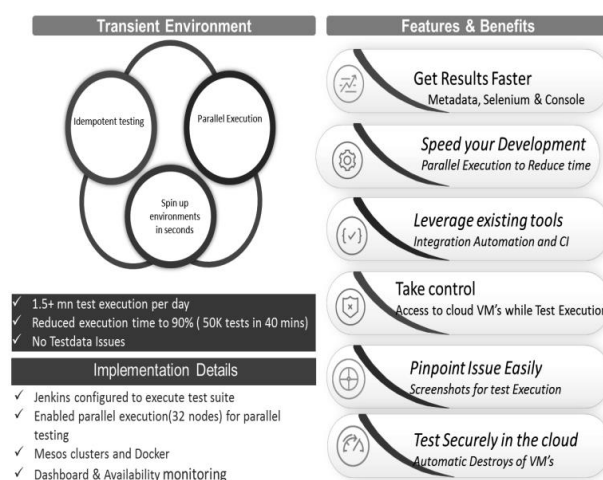
Para ser possível que essa automação seja rápida e barata, fez-se necessário utilizar os ambientes transitórios, que são ambientes em tempo real para testar os softwares. Para fins de execução de teste, Docker e Mesos criam o ambiente e executam todos os testes conforme necessário. Um ambiente transitório da oportunidade de projetar testes idempotente (o resultado de uma requisição realizada com sucesso é independente do número de vezes que é executada) e repetíveis, certificando-se de que cada teste gere os mesmos resultados.

## RESULTADOS ALCANÇADOS

A equipe de projeto pode alcançar uma automação em tempo real e de até 100% da penetração de teste.

Com os custos de infraestrutura em nuvem ser mais barata que os servidores físicos, e equipes de softwares capazes de criar um ambiente em tempo real, auxilia e flexibiliza a utilização de pilha de tecnologia mais recente, fornecendo a execução mais rápida. Com essa implementação, é possível manter os custos da QA do projeto em menos de 10% e a penetração da automação em 95%.

A figura abaixo mostra a diferença de custo da QA e os benefícios da entrega do teste automatizado em DevOps.



## CONCLUSÃO

Foi possível executar milhares de testes em minutos para cada construção. Esta técnica reduz potencialmente o tempo desperdiçado resultante da execução de testes manuais em ambientes físicos, eliminando toda a fase de testes em um ciclo de desenvolvimento de software em escala ágil. O feedback do teste é dado ao desenvolvedor no mesmo ciclo de construção, sem ter q implementar o código para o ambiente de desenvolvimento.

Os testes são executados automaticamente por uma ferramenta de integração continua. Se as compilações estão acontecendo com vezes por dia, o servidor de CI pode executar uma suíte de regressão de qualquer tamanho com vezes por dia, e promover o código para maior ambientes automaticamente.

Assim, os objetivos de negócios são atendidos pela equipe de desenvolvimento de software, que são inovação e modernização da tecnologia, velocidade de lançamento no mercado é aumentada, o custo é reduzido pela metade e a experiencia do usuário é melhor.