Pesquisa sobre DevOps

MATHEUS PORTES OURIQUE

Sumário

[Introdução 3](#_Toc206408972)

# Introdução

O DevOps é uma metodologia voltada no conjunto de práticas no desenvolvimento de software que tem como objetivo oferecer soluções com mais rapidez e maior qualidade. Utiliza praticas, como ciclos de feedback contínuos, para entregar uma nova funcionalidade ou correção em um software que gere valor ao usuário, exigindo uma comunicação constante e de boa qualidade entre as equipes de desenvolvimento.

# Práticas

Integração e entregas contínuas

Uma das principais características da implementação do DevOps é o pipeline de integração e implementação continuas (CI/CD) com ciclos frequentes de feedback para identificar possíveis melhorias, possibilitando disponibilizar os apps ao cliente com maior frequência e averiguar essa qualidade com a menor interação humana possível. Essas práticas empregam automação e monitoramento ao longo do seu ciclo de vida, conseguindo identificar erros e problemas de maneira mais ágil. Quando combinadas, essas práticas são conhecidas como “pipeline CI/CD”, processo viabilizado pela colaboração ágil entre as equipes de desenvolvimento e operações.

Durante o ciclo de vida do software, o DevOps funciona com base em 5 estágios:

1. Planejamento: As equipes DevOps definem e idealizam o software que irão construir enquanto encontram os pontos da dor do cliente e como resolvê-los através do produto. Eles descrevem as capacidades e recursos do software.
2. Desenvolvimento: Na fase de desenvolvimento, as equipes DevOps realizam a codificação, revisão, integração e implantação em diferentes ambientes. As equipes se dedicam à inovação rápida de produtos de qualidade, mantendo a produtividade e a estabilidade.
3. Teste: São feitos testes para cada código para verificar se ele contém algum bug ou erro que possa reduzir a eficácia e o desempenho do produto final. Ele também ajuda a elevar a experiência do cliente.
4. Escalabilidade: Em seguida, as equipes DevOps implantam o software em um ambiente de produção de forma confiável e consistente. Esta fase também envolve a configuração e escalabilidade da infraestrutura fundacional totalmente controlada dos ambientes utilizados.
5. Operação: A fase de operação inclui monitoramento, manutenção, registro e software de solução de problemas em ambientes de produção. As equipes DevOps visam garantir a estabilidade do sistema, confiabilidade e alta disponibilidade, além de zero tempo de inatividade do software, reforçando ao mesmo tempo a governança e a segurança. Para isso, eles precisam de visibilidade completa do software e dos sistemas centrais, além de rica medição de desempenho, alteração acionável, coleta de feedback do usuário, e muito mais.

# Ferramentas

# As ferramentas utilizadas ajudam durante todo o ciclo de produção de software, sendo divididas em Ferramentas de automação e Ferramentas de Pipeline (CI/CD)

## Automação:

* Jenkins: é um servidor de automação de código aberto e gratuito que ajuda a automatizar processos de desenvolvimento de software, tais como construção, facilitação de CI/CD, implantação e testes.
* Líder em contentores de software, o Docker é utilizado por mais de 11 milhões de desenvolvedores em todo o mundo, o Docker ajuda os desenvolvedores a construir, empacotar e depois implantar os códigos com facilidade e rapidez através de recipientes com as dependências necessárias em vez de máquinas virtuais.

## Pipeline (CI/CD):

* CircleCI: Não importa o tamanho de sua equipe DevOps, esta ferramenta nativa da CI/CD CircleCI oferece confiabilidade e velocidade para seu processo de desenvolvimento de software. Você pode desenvolver e implantar rapidamente software de alto desempenho em sua infraestrutura ou na nuvem em escala. O CircleCI foi fundado em 2011, e agora funciona em torno de 35M+ construções por mês. É usado por empresas líderes como Spotify, PagerDuty, Ford Motor Company, Samsung e mais.
* Bamboo: Essa ferramenta une seus builds, releases e testes automatizados com um único fluxo de trabalho. Como ferramenta DevOps, ela ajuda você a criar planos de construção em vários estágios, estabelecer gatilhos, e alocar agentes para construções críticas, bem como implantações.

# Benefícios:

Um dos principais benefícios da utilização da metodologia DevOps é a redução de riscos ao ocorrerem alterações, permitindo que as equipes trabalhem juntas, identificando problemas de forma antecipada e a resolução de questões antes que afetem o produto final, refletindo diretamente na **qualidade das entregas.** Outro beneficio é de incorporar as mudanças no processo de desenvolvimento, mantendo a competitividade da empresa, assim, as equipes aprendem a cada nova implementação, ajustando o processo de forma mais iterativa. Isso aumenta a produtividade e fortalece a resiliência de empresas e times.

# Exemplos:

### 1. Integração Contínua (CI)

Durante o processo de desenvolvimento de software, os desenvolvedores mesclam suas alterações de código em um repositório compartilhado várias vezes ao dia. A cada mesclagem, testes automatizados são acionados para garantir que as alterações estejam prontas para implantação. Essa detecção antecipada de bugs ajuda a evitar problemas dispendiosos no futuro.

**Quanto mais rápido os bugs forem detectados, mais tempo e dinheiro serão economizados na sua resolução.** Isso ocorre porque bugs não detectados tendem a ter um efeito dominó, criando mais bugs e mais problemas quanto mais tempo sobrevivem.

A automação da integração contínua (CI) dispensa os desenvolvedores da necessidade de mesclar seus códigos e, em vez disso, extrai automaticamente as informações para eles, permitindo que os desenvolvedores permaneçam no fluxo criativo.

### 2. Integração Contínua (CD)

O lado CD da integração contínua é a entrega contínua, a implantação ou ambas. Tradicionalmente, os desenvolvedores precisariam executar testes em cada novo conjunto de códigos. **Com a automação, o sistema treinado executará os testes automaticamente ao receber novas alterações.**

Novamente, neste cenário, os desenvolvedores nunca precisarão interromper seu fluxo de trabalho, a menos que sejam notificados pelo sistema de que um bug foi detectado.

### 3. Testes de automação

Testes de software, em geral, são inerentemente melhores quando realizados por IA. Permitir que computadores realizem tarefas repetitivas de teste sem a necessidade de intervenção humana acelerará o processo e fornecerá melhores resultados. Você pode treinar seu sistema automatizado para:

* Mais rápido
* Mais preciso
* Eficiente
* Integrado com integração contínua e entrega contínua

Você pode automatizar testes em vários níveis, incluindo:

* Teste de unidade
* Teste de integração
* Teste de desempenho
* Teste funcional
* Teste de regressão

Em média, os desenvolvedores dedicam de 10% a 25% do seu tempo a testes. Transferir esse processo para a automação permite que o produto chegue ao mercado muito mais rapidamente.

### 4. Implantação de microsserviços

Microsserviços são uma das áreas em que a automação DevOps pode ser realmente útil. Por quê? **Porque o processo de aplicação aqui é dividido em serviços menores que podem ser trabalhados e gerenciados independentemente do todo.**

A tecnologia de contêiner é normalmente usada para serviços gerenciados por DevOps, permitindo que os desenvolvedores automatizem atualizações e implantações de vários microsserviços em diferentes ambientes.

Sem automação, uma parte significativa do dia do desenvolvedor seria gasta gerenciando esses processos.

# Conclusão:

Como vimos nos itens anteriores, o DevOps é uma ferramenta muito importante no desenvolvimento de softwares atualmente, sendo responsável pela consistência da qualidade do produto e por integrar diferentes times de desenvolvimento dentro de uma equipe, e sendo essencial na automatização de processos os quais seriam muito lentos se fossem realizados por um humano e ainda mitigando as chances de erros ou falhas que impactariam na qualidade geral do produto.

# Referências Bibliograficas:

**1. RED HAT.**  
O que é um engenheiro DevOps?. Red Hat, [s.d.]. Disponível em: <https://www.redhat.com/pt-br/topics/devops/devops-engineer#pr%C3%A1ticas-de-devops>. Acesso em: 18 ago. 2025.

**2. KINSTA.**  
Ferramentas DevOps: tudo que você precisa saber. Kinsta, [s.d.]. Disponível em: <https://kinsta.com/pt/blog/ferramentas-devops/#o-que-so-ferramentas-devops>. Acesso em: 18 ago. 2025.

**3. DUPLOCLOUD.**  
DevOps Examples. DuploCloud, [s.d.]. Disponível em: <https://duplocloud.com/devops-examples/>. Acesso em: 18 ago. 2025.