

## **Parte 1 – Pesquisa teórica**

### **Os Princípios do DevOps e Suas Diferenças em Relação ao Modelo Tradicional**

O modelo tradicional de desenvolvimento de software caracteriza-se pela separação rígida entre as equipes de desenvolvimento e de operações. Os profissionais de desenvolvimento concentram-se na criação de novas funcionalidades e na aceleração dos ciclos de entrega, enquanto a equipe de operações prioriza a estabilidade, a manutenção e a confiabilidade dos sistemas em produção. Essa divisão, embora tenha sido eficaz em determinados contextos históricos, tende a gerar conflitos de interesse, atrasos no processo de entrega e falhas de comunicação entre as áreas. Nesse cenário, surge o DevOps como uma proposta de integração que visa superar essas limitações por meio da colaboração e da automação.

#### **O que é DevOps e quais problemas ele resolve?**

O DevOps pode ser definido como um conjunto de princípios, práticas e ferramentas voltados à integração entre desenvolvimento e operações em um fluxo de trabalho contínuo. Seu objetivo fundamental consiste em reduzir barreiras organizacionais, promover maior sinergia entre equipes e automatizar etapas críticas do ciclo de vida do software. Entre os principais problemas mitigados pelo DevOps destacam-se a fragmentação comunicacional entre áreas distintas, a morosidade dos ciclos de entrega, a elevada ocorrência de erros em ambiente de produção e a ausência de feedback rápido quanto à qualidade das alterações implementadas. Dessa forma, o DevOps contribui para processos mais previsíveis, ágeis e confiáveis, representando uma evolução em relação ao modelo tradicional.

#### **O que são CI (Integração Contínua) e CD (Entrega Contínua)?**

A Integração Contínua (Continuous Integration – CI) consiste na prática de incorporar, de maneira frequente, as alterações realizadas no código a um repositório central, acompanhado da execução automatizada de testes. Esse processo permite identificar precocemente falhas de compatibilidade e defeitos funcionais, evitando a acumulação de erros ao longo do projeto. A Entrega Contínua (Continuous Delivery – CD) constitui a extensão natural da Integração Contínua, assegurando que o código validado avance por pipelines automatizados de construção, verificação e disponibilização. Em níveis mais avançados, esse processo pode culminar na Implantação Contínua (Continuous Deployment), em que a publicação em produção ocorre de forma inteiramente automatizada. Em conjunto, tais práticas ampliam a confiabilidade, a previsibilidade e a celeridade do processo de entrega de software.

#### **Três ferramentas DevOps bastante utilizadas**

No âmbito das ferramentas mais difundidas na adoção de práticas DevOps, destaca-se o Docker, que possibilita a criação de containers capazes de empacotar aplicações juntamente com suas dependências, assegurando consistência e portabilidade entre ambientes distintos. O Kubernetes constitui outro recurso amplamente empregado, responsável pela orquestração de containers e pela oferta de funcionalidades de escalabilidade, monitoramento e gerenciamento de aplicações distribuídas. Por fim, o GitHub Actions apresenta-se como uma solução integrada ao repositório GitHub, destinada à automação de pipelines de integração e entrega contínuas, o que simplifica a adoção de práticas modernas de desenvolvimento.

### **Conclusão**

O DevOps configura-se como uma mudança paradigmática em relação ao modelo tradicional baseado na separação entre desenvolvimento e operações. Por meio da integração entre equipes, da automação de processos e da implementação de práticas como a integração contínua e a entrega contínua, possibilita-se a entrega de software de forma mais ágil, eficiente e segura. Diferentemente do modelo clássico, em que as áreas atuavam de forma isolada e com objetivos frequentemente divergentes, o DevOps propõe uma cultura de colaboração, de responsabilidade compartilhada e de inovação contínua. Nesse sentido, constitui um elemento essencial para organizações que almejam competitividade em um ambiente tecnológico em constante transformação.

### **Parte 2 – Execução prática**

[https://github.com/mucefe/Faculdade/tree/77f501f03a69dae6cca4496675679c4e273d2792/src/Terceiro\\_Ano/Atividades%20EaD/Segundo%20Semestre/Atividades%20EaD/3%20-%20Dev-Ops](https://github.com/mucefe/Faculdade/tree/77f501f03a69dae6cca4496675679c4e273d2792/src/Terceiro_Ano/Atividades%20EaD/Segundo%20Semestre/Atividades%20EaD/3%20-%20Dev-Ops)