

## **Sumário Didático da Aula gerado a partir da transcrição pelo LLM baseado na taxonomia de Bloom;**

### **\*Título da Aula:\* Introdução à Cultura DevOps**

### **\*Objetivos de Aprendizagem:\***

#### **1. \*Compreender o Conceito de DevOps:\***

- Justificativa: Entender que DevOps é mais do que ferramentas; é uma cultura que integra desenvolvimento e operações para acelerar a entrega de software. No mercado de trabalho, isso é crucial para as empresas que buscam eficiência e inovação contínua.

#### **2. \*Aprender Sobre Automação e Integração Contínua:\***

- Justificativa: Os alunos devem aprender a automatizar processos para reduzir o tempo de entrega e aumentar a confiabilidade. Isso é uma habilidade valorizada em ambientes empresariais que buscam melhorar a eficiência operacional.

#### **3. \*Entender os Princípios de Colaboração e Melhoria Contínua:\***

- Justificativa: Colaboração eficaz entre equipes de desenvolvimento e operações é essencial para o sucesso do DevOps, e a melhoria contínua garante que os processos se tornem cada vez mais eficientes e adaptáveis a mudanças.

#### **4. \*Familiarizar-se com Tecnologias Modernas:\***

- Justificativa: Conhecer tecnologias como Docker, Kubernetes, e infraestrutura como código é importante para entender a arquitetura de sistemas modernos. No mercado, isso prepara os alunos para trabalhar em projetos de larga escala e com tecnologias de ponta.

#### **5. \*Desenvolver Habilidades Práticas em DevOps:\***

- Justificativa: A prática de configurar pipelines de integração e entrega contínua prepara os alunos para desafios reais no mercado, onde a capacidade de implementar soluções eficazes e automatizadas é altamente demandada.

### **\*Principais Tópicos Cobertos:\***

#### **- \*Conceitos Gerais:\***

- Introdução a DevOps como cultura e metodologia.
- Benefícios do DevOps e sua importância no mercado.

#### **- \*Componentes:\***

- Princípios de automação, colaboração, medição e melhoria contínua.
- Ferramentas e tecnologias relevantes como Docker, Git, Jenkins, e infraestrutura como código.

#### **- \*Evolução e Contexto:\***

- Evolução das metodologias de desenvolvimento: Waterfall, Agile, e a introdução do DevOps.
- Comparação entre monolitos e micro serviços.

#### **- \*Exemplos Práticos:\***

- Dinâmica de criação de um "artefato analógico de software" para exemplificar a importância da automação e colaboração.

- Exposição dos benefícios de uma pipeline automatizada e da função do DevOps em um cenário prático.