

at_S19_A1_SL6_backend

Roteiro de Atividade Prática

Nome: _____ Turma: _____

Título da atividade: Automatizando fluxos – Integração contínua e deploy contínuo (CI/CD)

1. Introdução

A integração contínua (CI) e o deploy contínuo (CD) são práticas fundamentais para o desenvolvimento moderno de software, que visam reduzir riscos, automatizar processos e garantir entregas frequentes e confiáveis.

1.1 Integração contínua (CI): processo de integrar código em um repositório compartilhado regularmente, seguido de testes automatizados para verificar a qualidade.

1.2 Deploy contínuo (CD): automação do processo de entrega de código em ambientes de produção, reduzindo intervenções manuais.

2. Ferramentas de CI/CD

2.1 **Jenkins:** uma das ferramentas mais populares para automação de CI/CD, com suporte a pipelines personalizados.

2.2 **GitHub Actions:** solução integrada ao GitHub, ideal para projetos hospedados na plataforma.

2.3 **GitLab CI/CD:** uma alternativa poderosa, com integração ao GitLab.

3. Vantagens da automação

- 3.1 Redução de erros humanos.
- 3.2 Aceleração do ciclo de desenvolvimento.
- 3.3 Feedback rápido sobre a qualidade do código.

4. Exemplo prático

Imaginem que vocês trabalham em uma equipe que desenvolve APIs para um serviço de streaming. O objetivo é configurar um pipeline de CI/CD que:

- 4.1 Execute testes automatizados a cada push no repositório.
- 4.2 Faça o build da aplicação e crie artefatos (ex.: imagens Docker).
- 4.3 Automatize o deploy para um ambiente de teste.

5. Passo a passo

- 5.1 Escolham uma ferramenta de CI/CD, como Jenkins ou GitHub Actions.
- 5.2 Configurem o pipeline por etapas:
 - 5.2.1 Build – Compilem o código e criem artefatos.
 - 5.2.2 Teste – Executem os testes automatizados.
 - 5.2.3 Deploy – Implementem a aplicação no ambiente de teste.

- 5.3 Adicionem notificações para alertar a equipe sobre falhas no processo.

Situação-problema

Vocês receberam uma mensagem no Slack:

“Estamos enfrentando problemas com erros não detectados antes do deploy para produção. Precisamos de uma solução para automatizar os testes e os deploys, reduzindo falhas. Podem sugerir um plano detalhado?”

Situação fictícia produzida pela SEDUC-SP.

Perguntas dissertativas

1. Quais são os benefícios de implementar CI/CD em um projeto?
2. Como um pipeline de CI/CD pode ajudar a detectar erros antes do deploy?
3. Que fatores vocês consideram ao escolher uma ferramenta de CI/CD?
4. Como vocês estruturariam um pipeline para evitar erros em produção?
