

## **Trabalho Extra - Jenkins**

**Fabio Campidelli**

**Igor Sabarense**

**Rithie Natan**

### **Qual é o propósito do artigo ?**

O propósito do artigo é mostrar o que é a ferramenta Jenkins e como a integração contínua pode ser benéfica a um ambiente de desenvolvimento.

### **O que é o Jenkins ?**

Jenkins é um servidor de automação gratuito e de código aberto. Ajuda a automatizar as partes do desenvolvimento de software relacionadas à construção, teste e implantação. Facilitando assim a integração e entrega contínua.

### **O que se pode fazer com Jenkins ?**

Jenkins possibilita que você gere pacotes a cada intervalo predeterminado de tempo, a cada commit no código ou simplesmente quando desejar, ou seja, você define a maneira que melhor se adequa na sua empresa.

### **Por quê utilizar Jenkins e GitHub ?**

O GitHub é uma ferramenta de versionamento de código difundida na comunidade de desenvolvedores, ao utilizá-la em conjunto com o Jenkins, temos a segurança de nossas versões ( GitHub ) e a integração contínua ( Jenkins ) que permite que os deploys sejam feitos de acordo com a regra estabelecida. Exemplo : Ocorre um rebuild a cada commit feito na branch mestre do repositório .

### **Ele explica CI ? O que ele aborda?**

Relativamente. Ele dá uma ideia sobre o que é integração contínua ao explicar sobre o funcionamento do Jenkins e o uso de pipelines.

## **Explique de forma detalhada o que é Integração Contínua.**

Integração contínua (continuous integration) é uma prática de desenvolvimento de software que visa tornar a integração de código mais eficiente, através de builds e testes automatizados.

As metodologias ágeis propõem que o processo de desenvolvimento de software seja realizado com entregas mais frequentes.

Como consequência, o trabalho necessário para reunir, integrar e testar todo o código desenvolvido pela equipe em um repositório central também se tornou mais frequente.

Nesse contexto, a Integração Contínua surge para reduzir o trabalho manual necessário nesse processo de integração de código.

Além disso, com a automação de testes, o trabalho de correção de bugs é facilitado, garantindo que o software esteja funcionando após cada alteração.

Para adotar a Integração Contínua, é fundamental utilizar ferramentas de controle de versões, como o Gitlab ou Github.

Soluções como Jenkins e Hudson também são importantes para coordenar os builds, testes e integrações de código.