

Slide 01

Introdução

Nossa apresentação será sobre as definições de conceitos básicos dentro de Integração e Entrega Contínua, explicando seu funcionamento, sua importância e por fim, sua exemplificação com um simples projeto.

Slide 02

Integração e Entrega Contínua - Definição

São práticas de desenvolvimento de software que visam automatizar e melhorar a eficiência do processo de desenvolvimento, teste e entrega de software. Onde:

- Integração Contínua

É a prática de integrar o código desenvolvido por diferentes membros de uma equipe de forma frequente (geralmente várias vezes ao dia). O objetivo é garantir que, sempre que uma nova mudança for feita no código, ela seja testada e verificada automaticamente, para detectar problemas de forma precoce.

- Entrega Contínua

É a extensão natural da Integração Contínua. Ela garante que, após o código ser integrado e testado com sucesso, ele esteja sempre em um estado "pronto para produção". O código pode ser automaticamente implementado em um ambiente de produção, ou estar em um estado pronto para ser implementado manualmente.

Benefícios da Integração Contínua

Redução de erros, colaboração eficiente e menos conflitos de código.

Benefícios da Entrega Contínua

Entrega rápida de novas funcionalidades, feedback constante e confiança no código em produção.

Slide 03

Dev Ops

DevOps - Definição

É uma abordagem de desenvolvimento de software que une desenvolvimento (Dev) e operações (Ops).

DevOps - Função

Possuem como função integrar ferramentas de automação, gerenciam infraestrutura como código, monitoram continuamente e aplicam práticas ágeis para entrega rápida e eficiente de software.

DevOps - Importância

Aumenta a velocidade de entrega de software, reduz falhas em produção, melhora a colaboração entre equipes e permite uma resposta ágil às mudanças no mercado.

DevOps - Benefícios

Essa cultura promove a integração entre equipes, automação de processos, e monitoramento contínuo para entregar software de forma rápida, eficiente e confiável.

Slide 04

Controle de Versão - Definição

O controle de versão é um sistema que permite acompanhar, gerenciar e controlar as mudanças feitas no código ou em qualquer outro tipo de arquivo ao longo do tempo.

Controle de Versão - Importância

Ele facilita o trabalho em equipe, garantindo que múltiplos desenvolvedores possam colaborar no mesmo projeto sem conflitos e com histórico rastreável das modificações.

Controle de Versão - Exemplo de Ferramenta

O Git é amplamente utilizado e serve como base para plataformas como o GitHub.

Slide 05

CI e CD no GitHub

CI - Como Funciona

Automatiza o processo de teste e integração do código logo após os commits, garantindo que novas mudanças funcionem bem com o código existente.

CD - Como funciona

Automatiza o processo de deploy, permitindo que novas versões do código sejam entregues ao ambiente de produção com rapidez e segurança. O código que passa nos testes de CI pode ser automaticamente preparado ou enviado para deploy.

Slide 06

Testes Automatizados - Definição

São scripts ou programas que verificam automaticamente o funcionamento correto de uma aplicação, sem a necessidade de intervenção humana durante a execução.

Testes Automatizados - Funções

Eles são projetados para garantir que o software funcione como esperado em diversas situações, testando desde pequenas unidades de código (funções ou métodos) até grandes integrações de sistemas.

Slide 07

Esteira CI/CD no GitHub - Exemplo Prático

É um conjunto de automações que orquestra as etapas necessárias para garantir a qualidade e a entrega contínua de software, desde a integração do código até o deploy em produção.

Esteira CI/CD no GitHub - Como Funciona

Integração Contínua (CI)

Sempre que uma mudança é feita no repositório (através de um commit ou pull request), a esteira de CI é acionada.

O código é automaticamente testado e construído (build), verificando se a nova alteração não quebrou nada no projeto.

Caso os testes passem, o código pode ser integrado ao código principal.

Entrega Contínua (CD)

Após passar pelos testes de CI, a etapa de CD pode ser acionada.

Entrega Contínua (Continuous Delivery): O código é empacotado e fica pronto para ser enviado a um ambiente de produção (ou staging). A entrega pode ser manual.

Deploy Contínuo (Continuous Deployment): O código é automaticamente implementado no ambiente de produção, sem a necessidade de intervenção humana, caso todos os testes tenham sido bem-sucedidos.

Mostrar esteira criada no GitHub

<https://github.com/exemplosdeaula/Biblioteca.git>