


1. 1. Introdução

Perguntas:

1. O que é CD e qual sua relação com CI?

 Resposta: CD refere-se à **Entrega Contínua** (Continuous Delivery) ou à **Implantação Contínua** (Continuous Deployment). Ambos são a extensão natural e a fase final da **Integração Contínua (CI)**.

2. Quais são os benefícios da entrega contínua?

 Resposta:

A implementação da Entrega Contínua (e Implantação Contínua) oferece vários benefícios cruciais para o desenvolvimento e o negócio:

- **1. Redução de Riscos na Implantação:** Ao implantar alterações **pequenas e frequentes**, o processo se torna uma rotina de **baixo estresse**, em vez de um evento raro de alto risco. Se algo der errado, é fácil identificar o pequeno *commit* que causou o problema e reverter.
- **2. Feedback Rápido e Contínuo:** As equipes obtêm *feedback* do usuário final e métricas de produção **quase imediatamente** após a codificação. Isso permite correções rápidas (hotfixes) e ajustes do produto em tempo real.
- **3. Tempo de Lançamento no Mercado (Time to Market) Mais Rápido:** A automação elimina o tempo gasto em etapas manuais, como coordenação de *builds*, testes de regressão manuais e implantação. Isso permite que novas funcionalidades cheguem aos clientes em **dias ou horas**, em vez de semanas ou meses.
- **4. Melhor Qualidade do Código:** A dependência de testes automatizados abrangentes no pipeline de CI/CD força as equipes a escrever código mais limpo, bem coberto por testes e mais modular.
- **5. Maior Produtividade da Equipe:** Os desenvolvedores gastam menos tempo gerenciando fusões (merges), corrigindo erros de integração (graças ao CI) e gerenciando ambientes (graças ao CD), permitindo que se concentrem em **escrever código de valor**.

4. Para finalizar

- Qual é a principal diferença prática entre CI e CD?

A diferença prática reside no **objetivo final**. **CI (Integração Contínua)** foca em **testar e garantir** a qualidade do código na *branch* principal, gerando um artefato *pronto para uso*. **CD (Entrega/Implantação Contínua)** foca em **levar este artefato testado** automaticamente para os ambientes de teste final ou diretamente para a **produção**, tornando-o acessível aos usuários.

- O que aconteceria se o teste falhasse antes do deploy?

O *pipeline* seria **interrompido** (falha no *job* de test). Devido à condição **if: success()** no *job* de deploy, a **implantação não seria executada**. Isso impede que o código que está com falha seja liberado para o ambiente de produção (GitHub Pages), garantindo a estabilidade.

- Como a entrega contínua aumenta a confiança do time no processo?

A CD aumenta a confiança porque **transforma a implantação em uma rotina automatizada e de baixo risco**, em vez de um evento raro e estressante. O time confia no processo porque sabe que: 1) Apenas código **testado** (via CI) será implantado, 2) O processo é **repetível** e livre de erros humanos, e 3) Qualquer pequeno problema é fácil de reverter, pois as alterações enviadas são pequenas e frequentes.

Link do site publicado do GitHub Pages: <https://eduardo-silva-ti.github.io/ci-cd-teste/>