Aluno: Lucas Terra Cunha

RA: 322123117

O que é CD e qual sua relação com CI?

Resposta: CD é a prática de automatizar a entrega e implantação de software em ambientes de produção ou homologação. Sua relação com CI é que o CD é a continuação natural do processo: enquanto o CI garante que o código está funcionando corretamente através de testes automatizados, o CD pega esse código validado e o implanta automaticamente, completando o ciclo de desenvolvimento até a disponibilização para os usuários.

Quais são os benefícios da entrega contínua?

- Redução do tempo entre desenvolvimento e disponibilização para usuários
- Diminuição de erros humanos no processo de deploy
- Feedback mais rápido sobre problemas em produção
- Maior frequência de releases com menor risco
- Padronização do processo de implantação
- Possibilidade de reverter mudanças rapidamente
- Aumento da produtividade da equipe

Qual é a principal diferença prática entre CI e CD?

Resposta: A principal diferença prática é que o CI (Integração Contínua) foca em validar e testar o código automaticamente a cada mudança, garantindo que ele funciona corretamente e se integra bem com o resto do sistema. Já o CD (Entrega/Implantação Contínua) vai além: ele pega esse código validado e o coloca automaticamente em produção ou ambiente de homologação, tornando-o disponível para os usuários finais. Em resumo: CI verifica se o código está correto, CD coloca o código correto em funcionamento.

O que aconteceria se o teste falhasse antes do deploy?

Resposta: Se o teste falhasse antes do deploy (especialmente quando usamos a configuração condicional com needs: test e if: success()), o job de deploy não seria executado. O workflow pararia na etapa de testes, sinalizaria a falha no GitHub Actions, e nenhuma alteração seria publicada no ambiente de produção.

Isso protege a aplicação de receber código com problemas, garantindo que apenas código testado e aprovado seja implantado.

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

Resposta: A entrega contínua aumenta a confiança do time porque:

Padronização: Todo deploy segue o mesmo processo automatizado, eliminando variações e erros humanos

Rastreabilidade: Cada mudança é registrada e pode ser auditada

Testes automáticos: Garante que apenas código validado vai para produção

Reversibilidade: Facilita o rollback em caso de problemas

Frequência: Deploys menores e mais frequentes são menos arriscados que grandes releases

Feedback rápido: Problemas são identificados rapidamente em produção

Transparência: Todos veem o mesmo processo e resultados no GitHub Actions

Isso cria um ambiente onde o time se sente seguro para fazer mudanças, sabendo que há uma rede de segurança automatizada.