

O que aprendemos?

Nesta aula, aprendemos que:

- Para usar integração contínua com sucesso, é preciso ter **testes automatizados**
 - Esses testes devem verificar a maior parte do código
 - TDD é uma metodologia que pode ajudar na criação dos testes
 - Os testes representam *feedback* sobre a saúde/qualidade do seu projeto
- O desenvolvedor deve rodar os testes *antes de enviar* as alterações para o repositório
 - Se for impossível executar todos os testes (por causa da infraestrutura ou demora da execução), é possível executar apenas os testes de unidade ou uma suíte de testes mais importantes (*smoke testes*)
- O *build* do projeto deve ser simples e totalmente automatizado
 - É boa prática o uso de uma ferramenta especializada para o *build* do software
 - Tente usar um "single command build" para executar o *build*
- A execução dos testes e *build* não deve se tornar uma razão para atrasar o *build*, para evitar:
 - *Otimizar build*
 - Tente respeitar o tempo de *10 minutos para o build* (alguns falam de 5 minutos)
 - Se o *build* for falhar, a ideia é que ele *falhe o mais rápido possível*