



Agile Testing



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



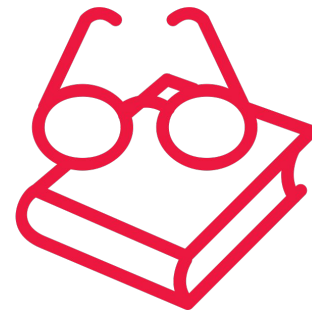
Oi!

Como estão?





O que vimos na aula passada?





O que vamos ver hoje?

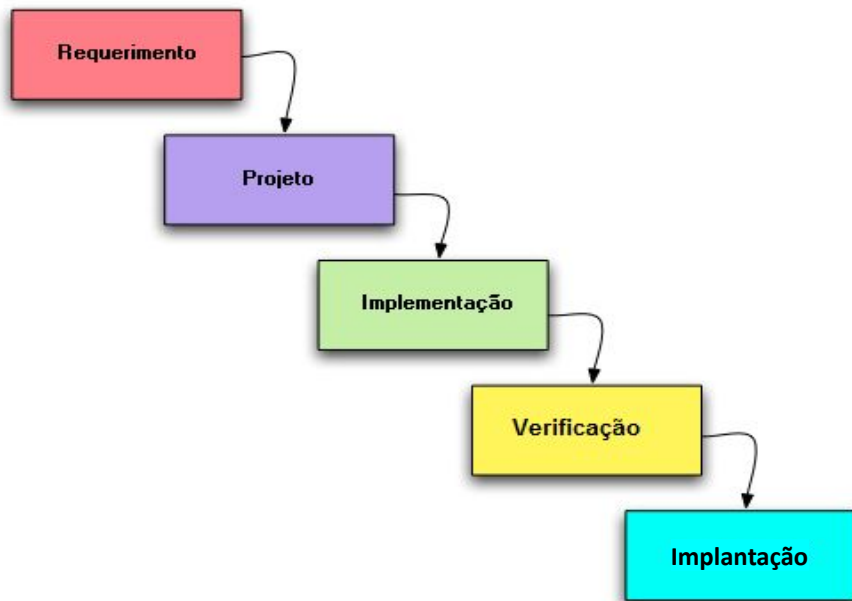
1. Agile Testing
2. UAT, QA TDD, caixa branca e caixa preta
3. DevOps



1 | Agile Testing



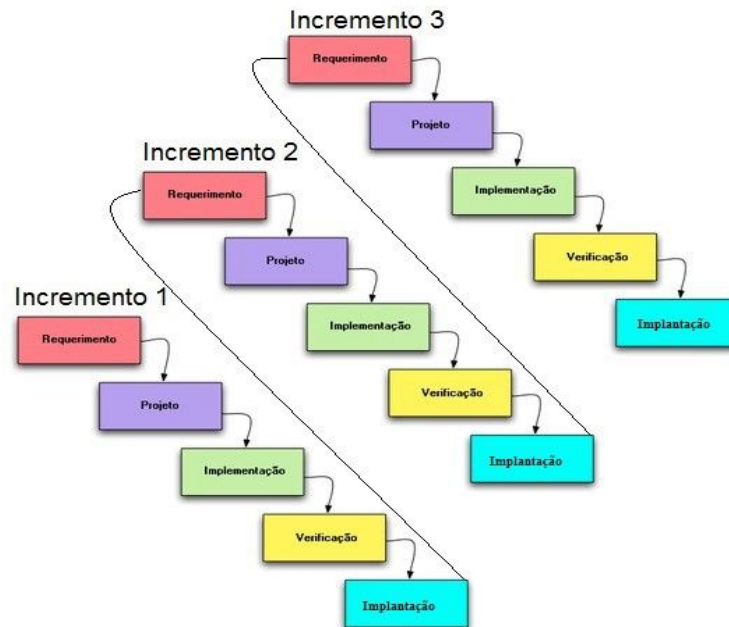
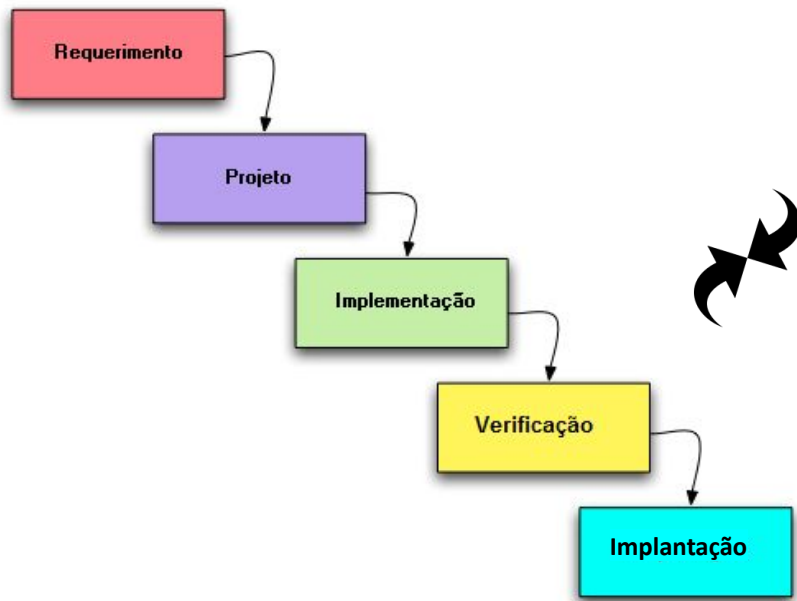
Ciclo de desenvolvimento e os testes?



Quando olhamos para o ciclo de desenvolvimento tradicional quais as principais mudanças na fase de testes?



Ciclo de desenvolvimento e os testes!

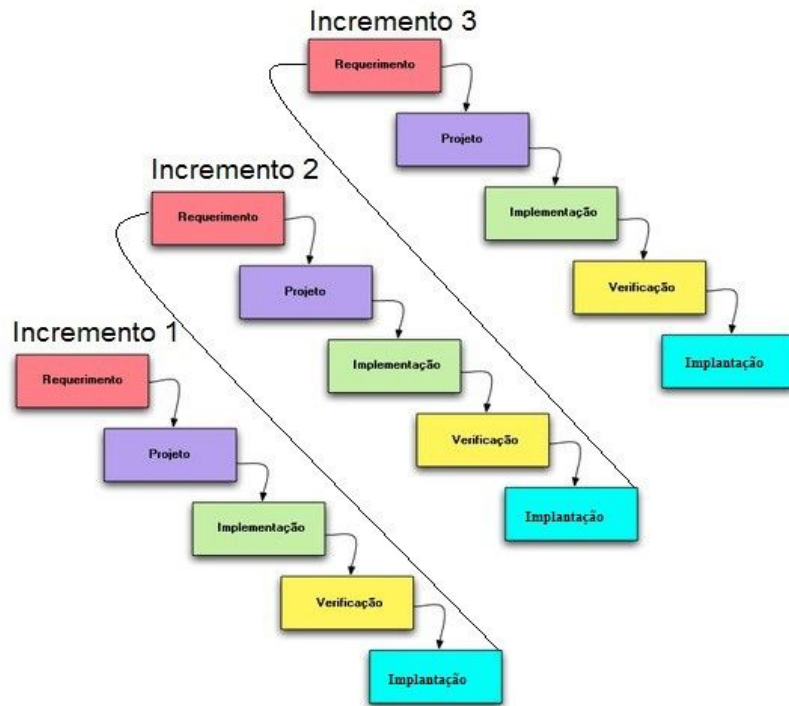




Ciclo de desenvolvimento?

Temos partes menores de código para testar, porém...

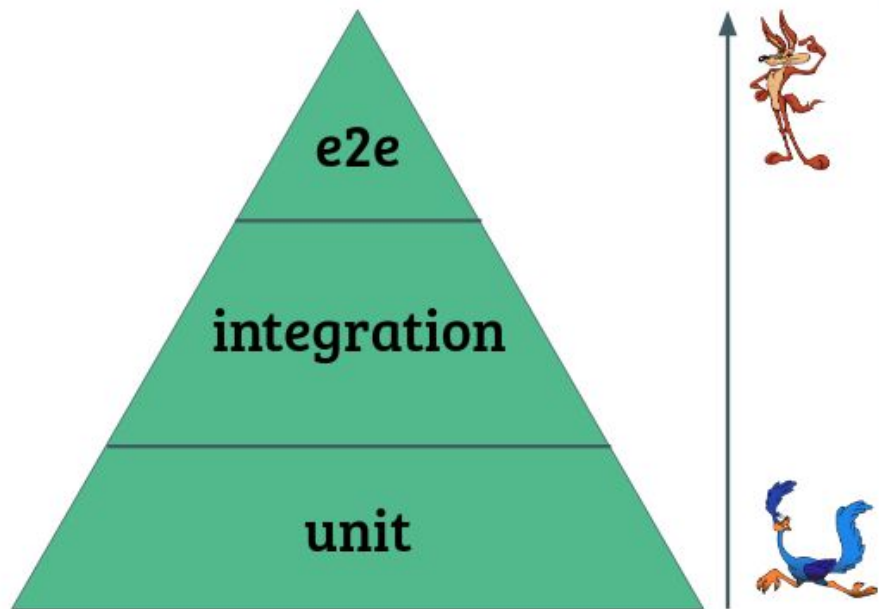
... precisamos de atenção contínua e pensar em como podemos ser mais eficientes nos testes.





Pirâmide de Testes: ponta a ponta

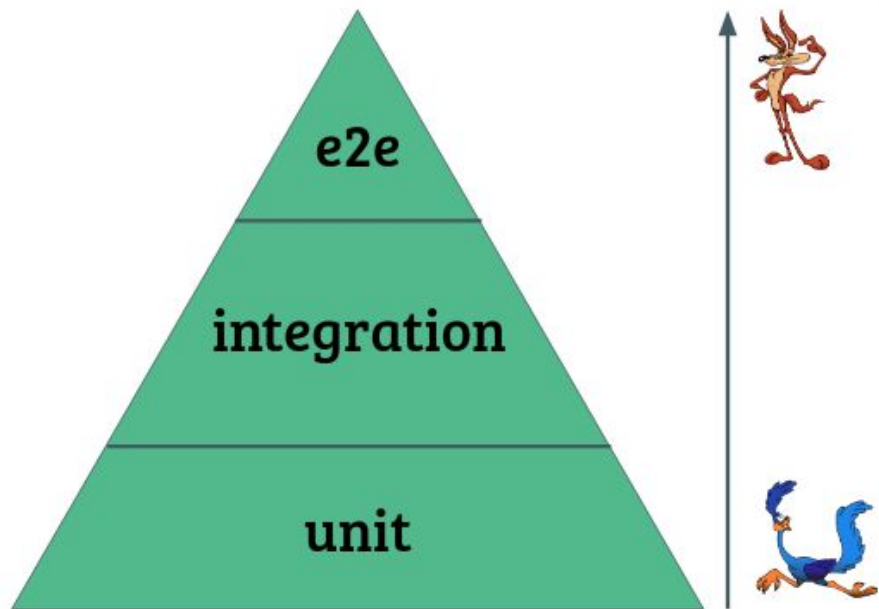
O objetivo é imitar o comportamento do usuário final nas nossas aplicações (seja ele uma pessoa, uma api, ou qualquer outro tipo de cliente).





Pirâmide de Testes: Integração

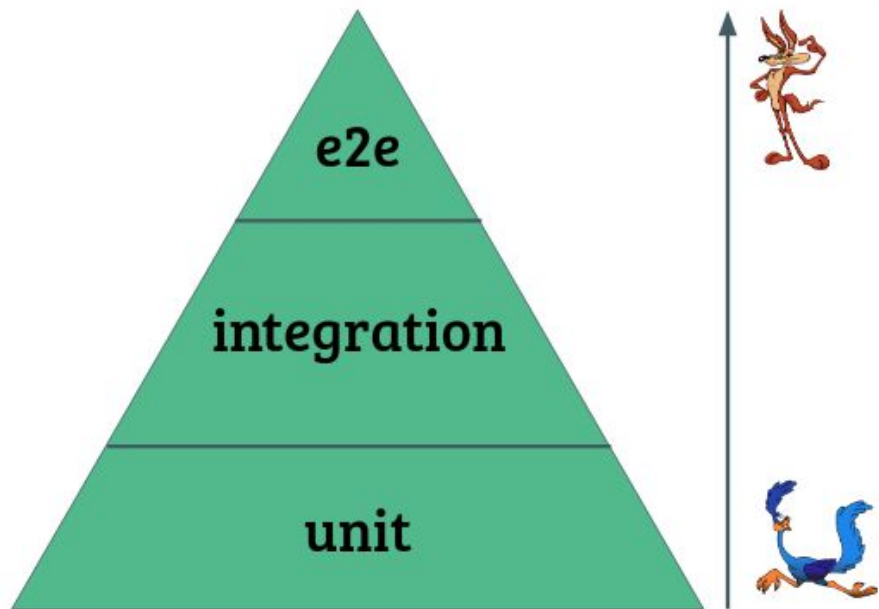
A ideia é verificar se um conjunto de unidades se comporta da maneira correta, só que de forma menos abrangente do que os testes de ponta a ponta.





Pirâmide de Testes: Unitários

Queremos verificar o funcionamento da menor unidade de código testável da nossa aplicação.

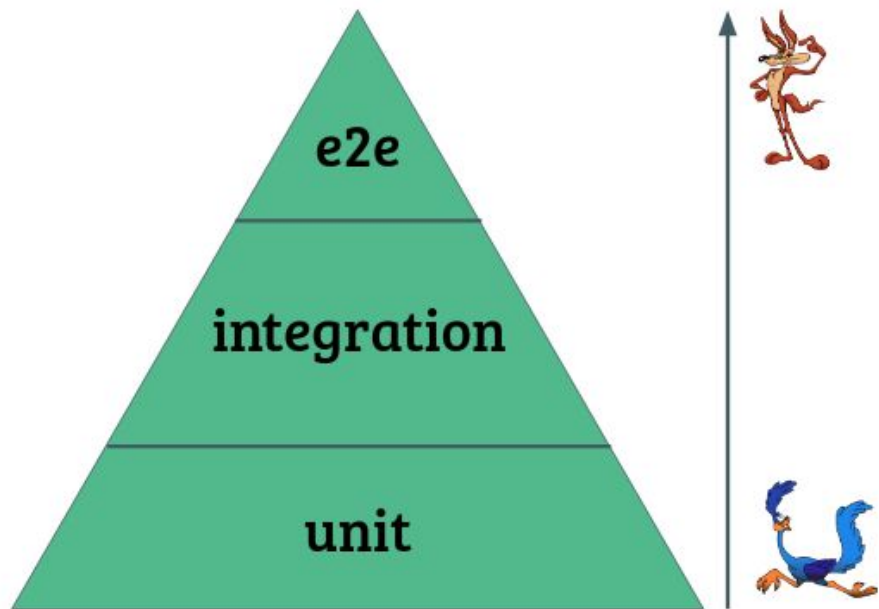




Pirâmide de Testes: unitários

Benefícios:

- Feedback instantâneo
- Reduz o custo do teste de regressão
- Reduz o tempo de depuração
- Permite que você implante com confiança



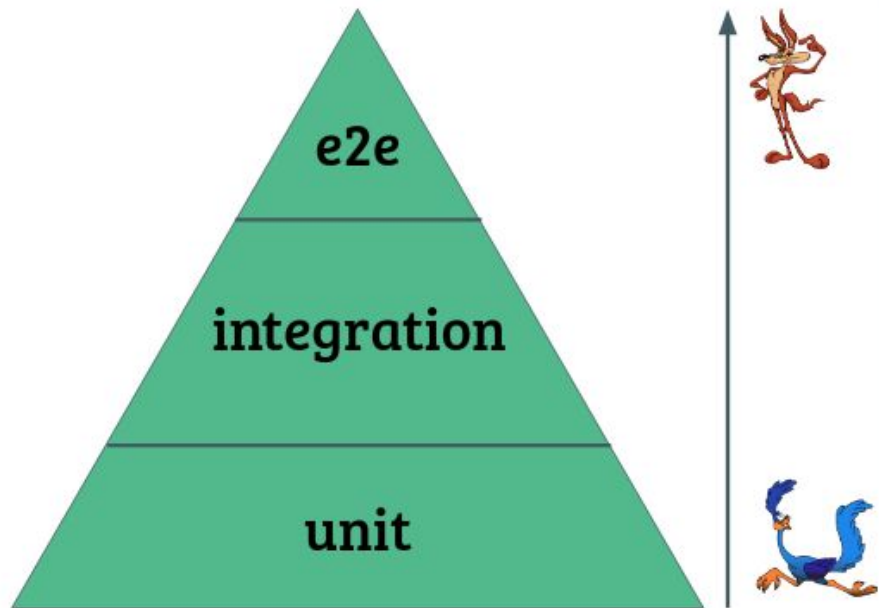


Pirâmide de Testes: Como usar?

Velocidade!

As camadas mais inferiores contêm testes mais rápidos, baratos e isolados ao comportamento da aplicação.

Na parte superior da pirâmide, teremos testes mais lentos, caros e integrados.





TDD

É uma prática onde...

Test Driven Development

... garantimos a eficácia dos recursos implementados em um código com os...

Desenvolvimento orientado a testes

... testes escritos antes dos próprios códigos.

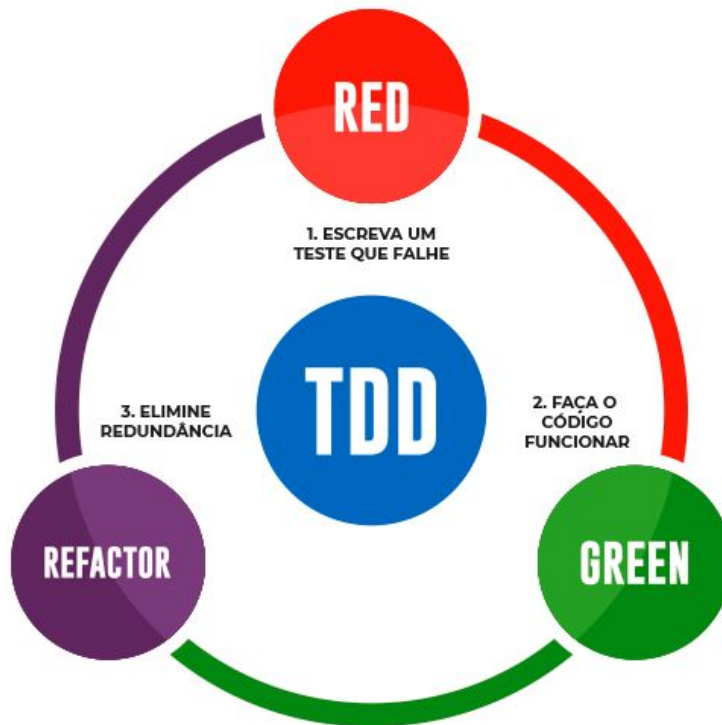




TDD

Test Driven Development

Desenvolvimento orientado a testes

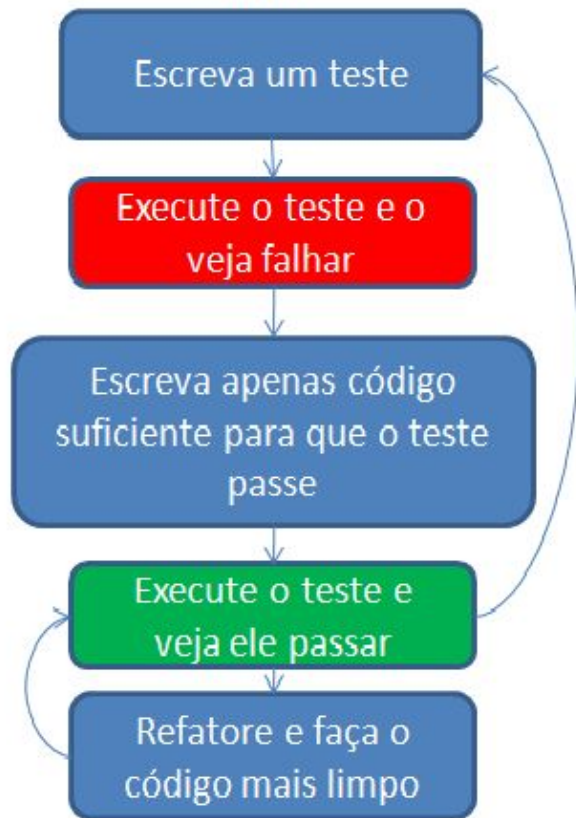




TDD

**Test Driven
Development**

**Desenvolvimento
orientado a testes**





Vamos debater?



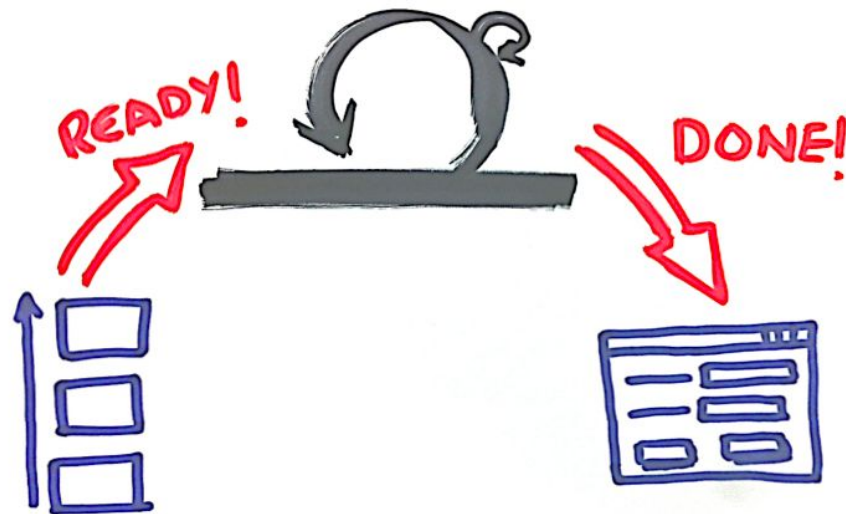
2

DOD, DOR e Critérios de Aceite



DoR - Definition of Ready

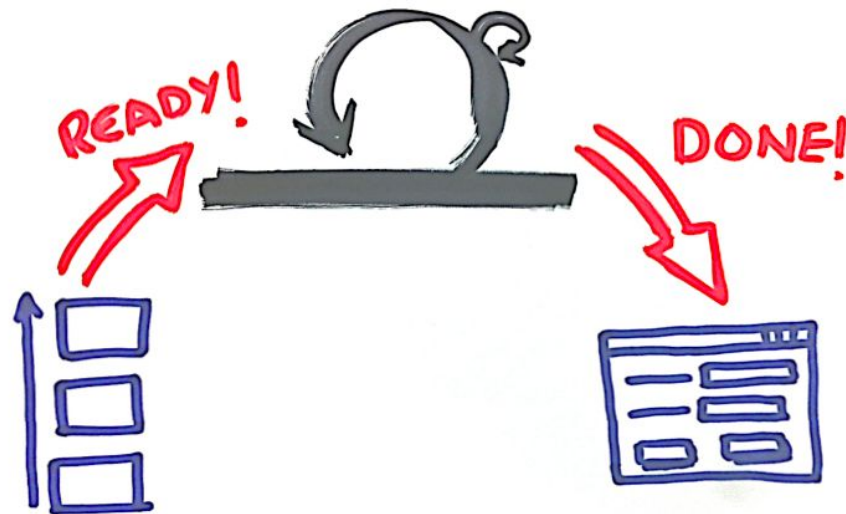
São **acordos de trabalho** entre o time de Desenvolvimento e o Product Owner, aplicados a **todas** as Histórias de Usuário. Estes, têm a intenção de que os **itens do Backlog** não cheguem para a reunião de planejamento com granularidade ruim, pouco ou nenhum detalhamento.





DoD - Definition of Done

É um acordo **genérico**, definido pelos membros do time Scrum (Desenvolvedores e Product Owner), **aplicável a todas as Histórias** de Usuário. Possui o intuito de que todos os membros do time tenham um entendimento compartilhado do que significa **"Done"**, para garantir a transparência.





Critérios de Aceite

Os critérios de aceite são **específicos** e diferentes para **cada História** de Usuário. São uma lista de itens que devem ser atendidos para que aquela história **seja disponibilizada** para o usuário.

User Story

Como <papel>
posso <função>
para <valor de negócio>

Critérios de Aceite

- ☒ critério para aceitação 1
- ☒ critério para aceitação 2
- ☒ critério para aceitação 3





Testes: quando usar o que?

Vamos ao Miro



3

DevOps



**Quem cuida do meu
código depois que
entreguei ele?**



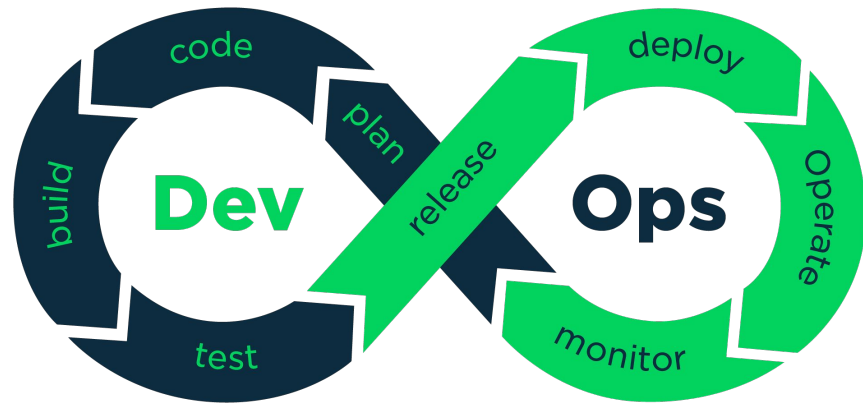


Como fazer devops?

CAMS:

(Culture, Automation, Measurement, Sharing)

- Cultura
- Automação
- Medição
- Compartilhamento





Como automatizar?

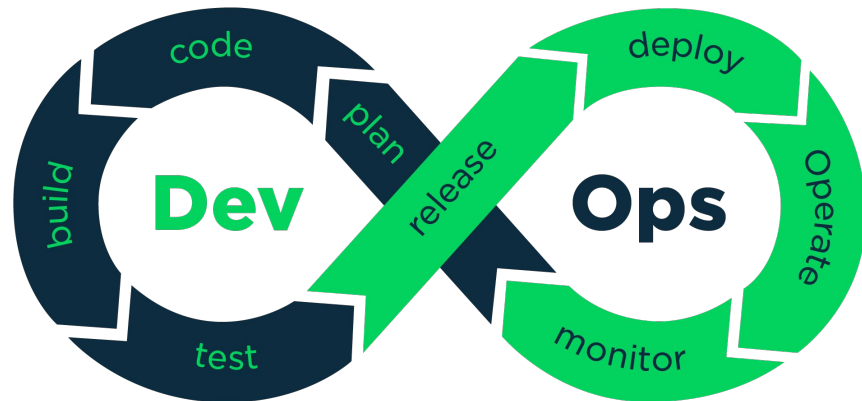
Versionamento e integração de código?

Testes?

Deploy?

Recuperação de desastre?

Monitoramento?





Como evoluir?

Retrospectivas com todas as partes

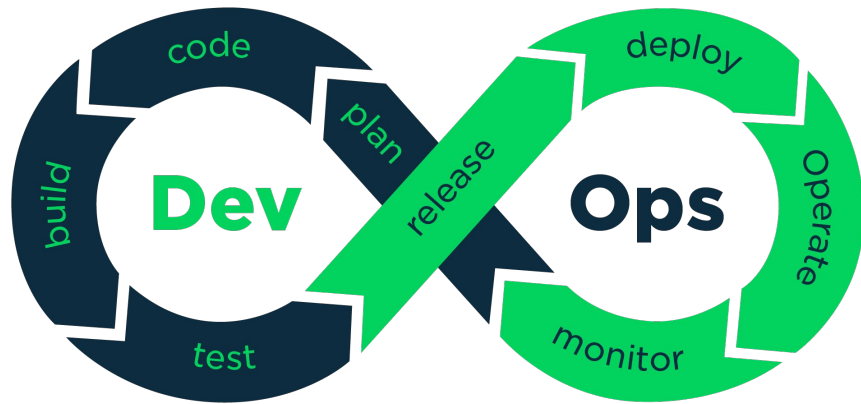
Novos DoDs

Code Review

Pair Programming

Refatorar Código

Completude de código





Devops - Mitos e Verdades

Desafio



Kahoot!

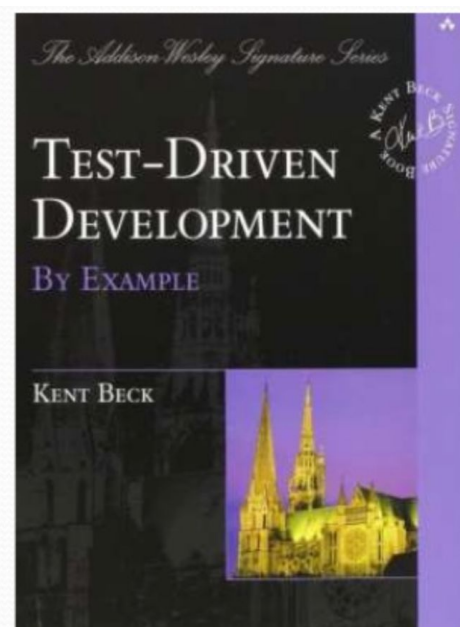
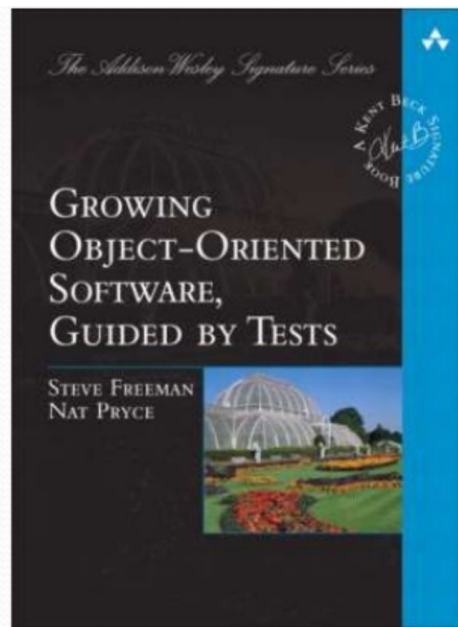


Como foi até aqui?





Bibliografia



**Fechamento e
Até Breve!**

DigitalHouse>
Coding School