



Salvando seu eu do futuro:

Porque escrever testes?





Mas o que são?

De uma forma simplificada:

Testes são métodos com execução automatizada que validam diversos aspectos do código.



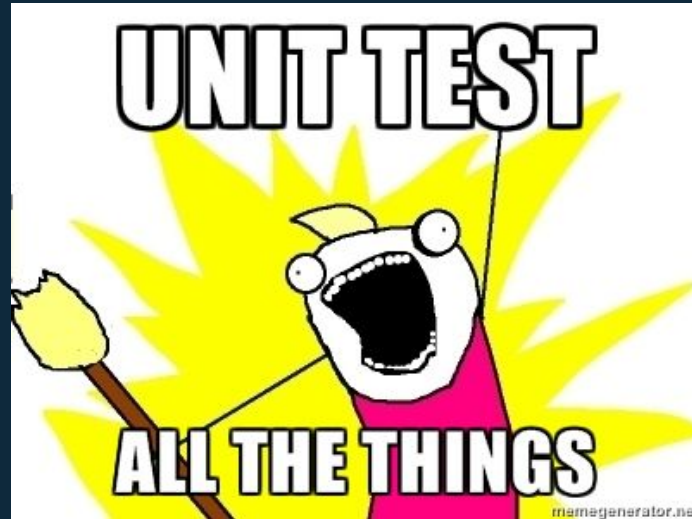


Quais os tipos?

- ◇ Unitários
- ◇ Integração
- ◇ Regressão
- ◇ Fumaça
- ◇ ...



Testes Unitários





Pra que servem?

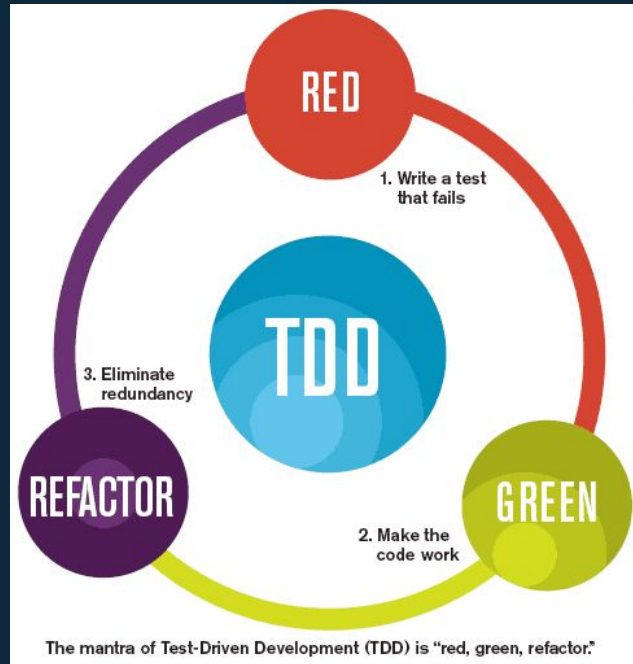
- ◇ Garantir funcionalidade da unidade
- ◇ Impedir que mudanças em uma unidade quebrem outra(s)
- ◇ Facilitar a manutenção
- ◇ Ajudar a indicar casos da quebra do padrão SOLID



O que evitam?



O processo:





Vantagens:

- Maior confiança na hora de realizar modificações
- Garantia de que outros trechos de código não foram afetados pelo novo código
- Ao implementar novas funcionalidades, garantir que todos os requisitos estão sendo cumpridos
- Ajudar a manter uma estrutura limpa e organizada
- Redução de custos com manutenção a longo prazo
- Indicar trechos que precisam de refatoração / melhorias
- Indicar de uma forma geral, a qualidade do código



Nem tudo são flores...





“Desvantagens”:

- Aumento no tempo de desenvolvimento
- Maior custo imediato
- Testes mal escritos não garantem funcionalidade, e geram mais custos
- Testes unitários sozinhos não garantem funcionalidades como integração





Encontrando o equilíbrio:

O que manter na cabeça:

- Um teste é melhor que nenhum, e onze são melhores que dez.
- Se tiver que escolher, implementar testes de trechos críticos
- ...





Encontrando o equilíbrio:

- A melhor prática é seguir o RGR, mas se não tiver tempo hábil para escrever o teste antes, a gente não julga ;)





Dúvidas?

