

**targettrust**  
treinamento e tecnologia



# Riguel Figueiró

---

Consultor de software na  
ThoughtWorks

Apaixonado por metodologias ágeis,  
integração contínua, qualidade e  
disseminação e compartilhamento  
de conhecimento!

· [riguel@adkosoftware.com.br](mailto:riguel@adkosoftware.com.br)

· [www.linkedin.com/in/riguel-figueiro/](https://www.linkedin.com/in/riguel-figueiro/)

· [github.com/riguelbf](https://github.com/riguelbf)





# **JAVA DEVELOPER**

## Imersão Completa



# Testes Automatizados

## TDD - Test Driven Development



O que é TDD?



“Uma prática de  
testes de software?”

“Design de  
códigos de testes?”

“Ou ambos??”

# TDD como prática de testes

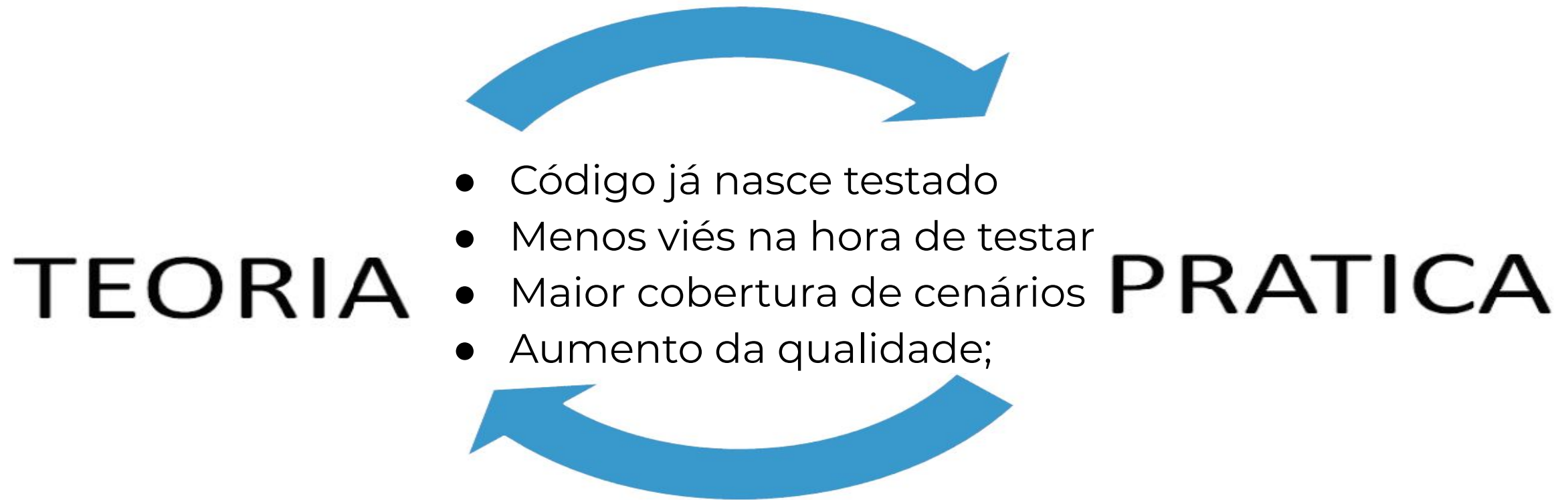
---

Quando o desenvolvedor pratica o TDD com o mero objetivo de aumentar a cobertura de testes o projeto.



# TDD como prática de testes

---





# TDD como prática de design

---

- O projeto de design de classes tem ganho
- Estrutura e padrões e projetos
- Autores com Martin Folwer, Kent Beck e Steve Freeman confirmam





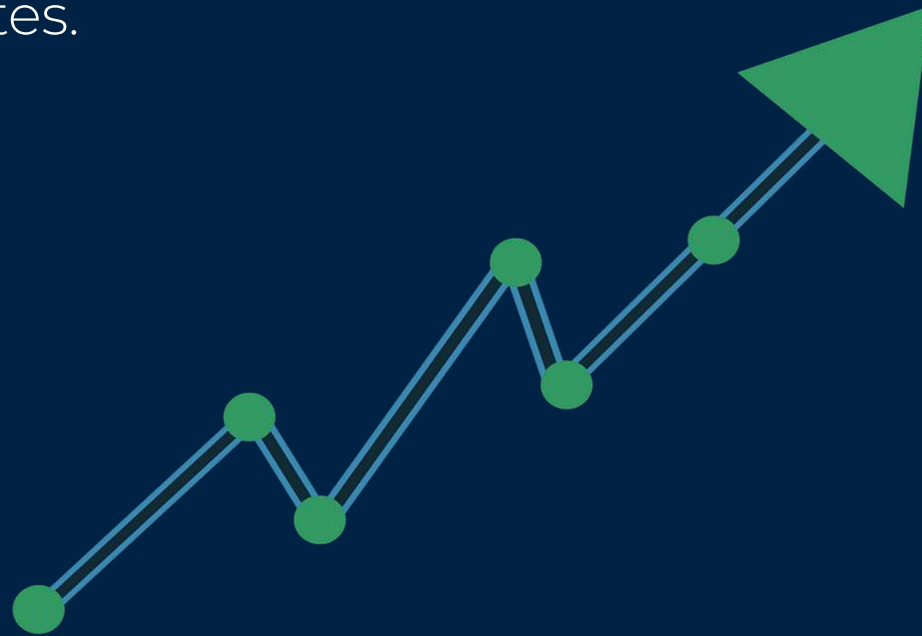
Mas é tão mágico assim?



# O que eu ganho com a prática?

---

- Qualidade na entrega;
- Diminuição da complexidade (esign de código);
- Velocidade na execução dos testes.

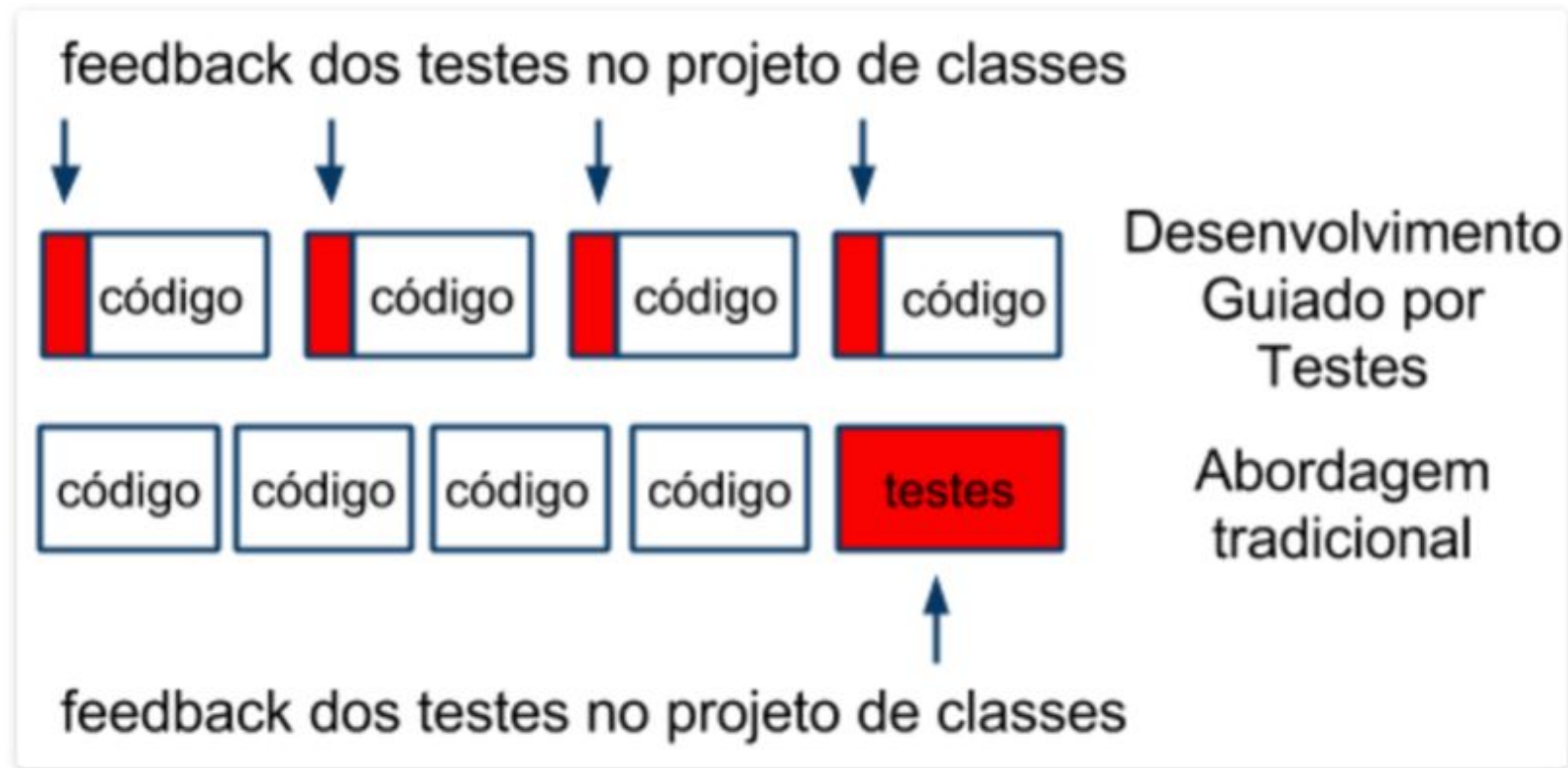


# Testes all time!!!

---

Devo praticar TDD 100% do tempo!

# Qual a diferença entre fazer TDD e escrever o teste depois?





Serei mais ou menos  
produtivo?



A grande resposta:



# Dicas para a vida!

---

- Se está difícil de testar, é por que provavelmente existe um problema de projeto ou de implementação no seu código.
- O segredo é perceber o mais rápido possível o momento em que escrever o teste passou a ser uma tarefa difícil.





# Isso quer dizer que...

---

- A busca pela testabilidade faz com que você busque projetos mais simples.
- Instanciar uma classe e fazer uso de comportamentos deve ser fácil.
- Classes não podem ser complicadas senão fica difícil testar.
- Maneira barata de validar seu projeto de classes.



# Pensando em exemplos...

---

- Um teste sempre tem 3 partes:

Cenário;

Ação;

Validação.



- Se tivermos grandes dificuldades para escrever nossos testes ao iniciar o cenário temos algo errado, logo, devemos reavaliar nosso design.



# E na prática, como ocorre?

---

- A mecânica da prática é simples: escreva um teste que falha, faça-o passar da maneira mais simples possível e, por fim, refatore o código. Esse ciclo é conhecido como:  
Ciclo **vermelho** - **verde** - **refatora**.



# Baby steps

---



# Baby steps - utilização

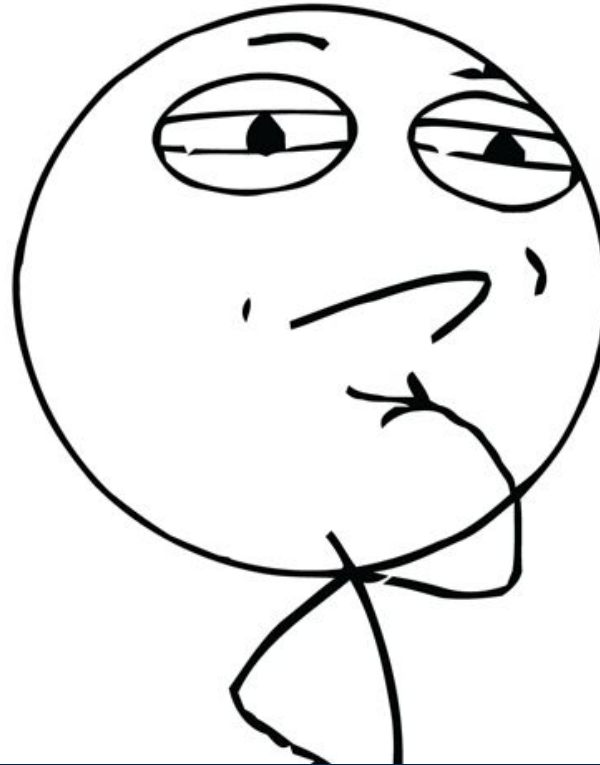
---

- Algo que devemos entender muito bem (ler mais sobre isso);
- Seja simples, mas não simplório demais e muito menos negligente;
- Avalie quando se deve utilizar;
- Utilize a experiência para ser um ganho com a prática e não o inverso.

# Desafio

---

**DESAFIO ACEITO?**



Desafio

---

LET'S GO





# Pergunta

---

Façam o máximo de perguntas que desejarem!!



**OBRIGADO PELA ATENÇÃO**



**APLAUSOS!**

GERADORMEMES.COM

