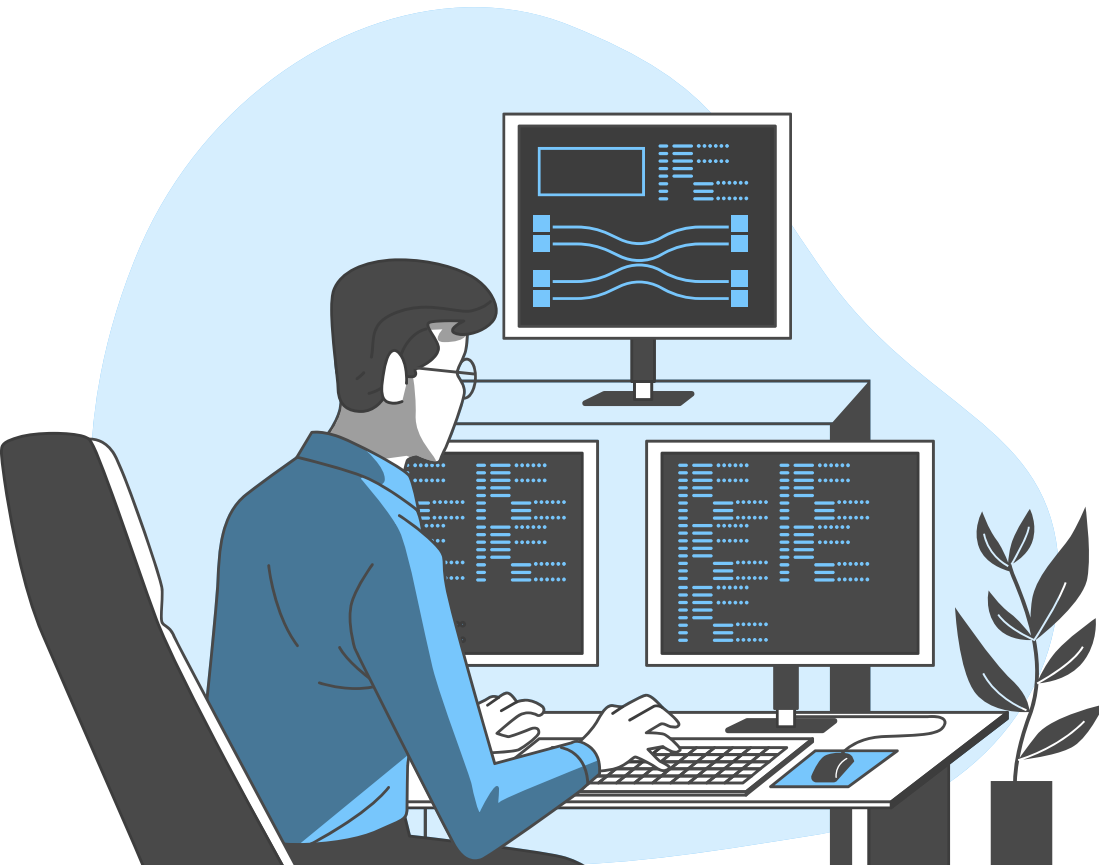
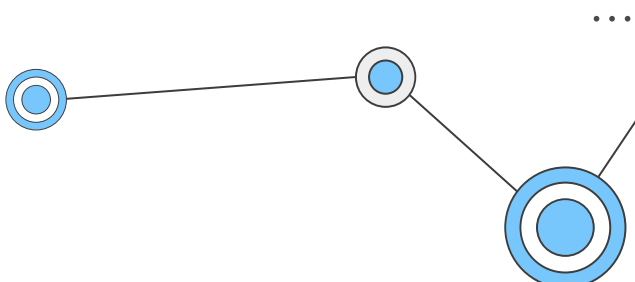


Qualidade de Software - TSW-006/2021-3

Prof.: Kechi Hirama



...



On the Effectiveness of Unit Tests in Test-driven Development

...

Augusto Calado Bueno

Autores

Ayse Tosun

Universidade Técnica
de Istambul

Natalia Juristo

Universidade Politécnica
de Madri

Burak Turhan

Universidade de
Brunel, Londres

Muzamil Ahmed

Universidade de Oulu



Agenda

01

Introdução do Artigo e
Objetivos

02

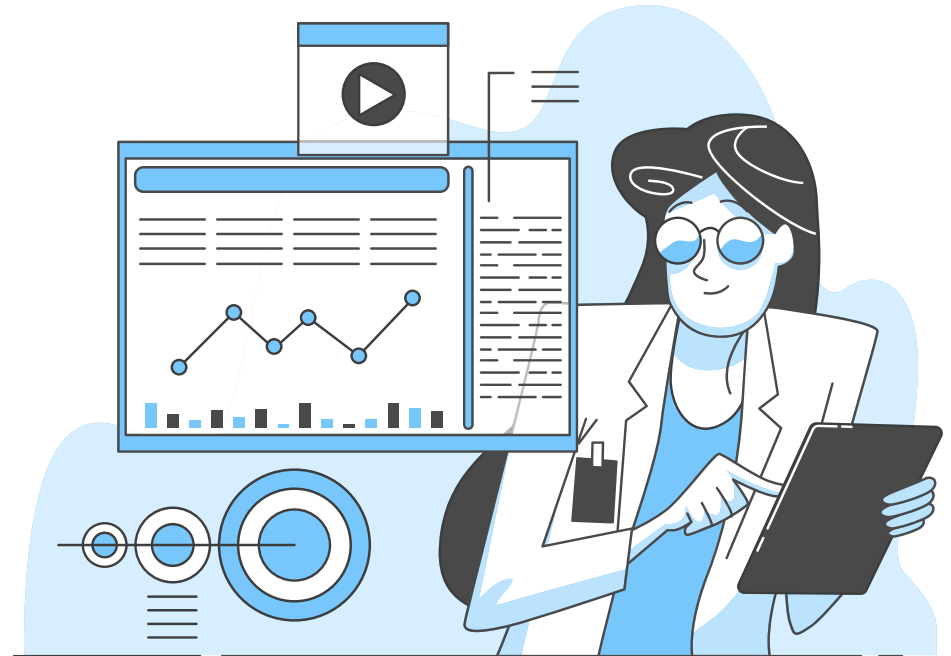
Definições e Conceitos

03

Pesquisas Bases e
Relacionadas

04

Experimento e Resultados





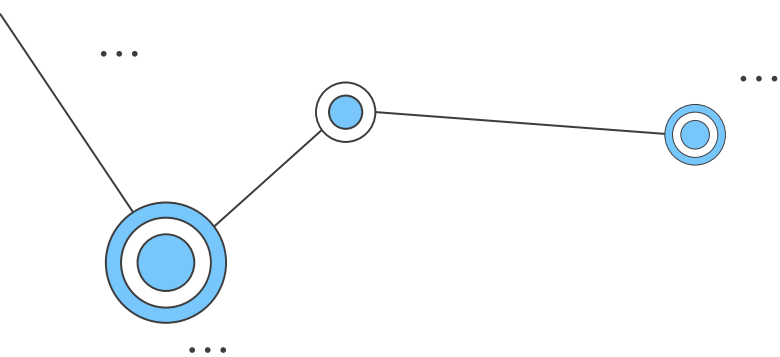
Motivação e Objetivos



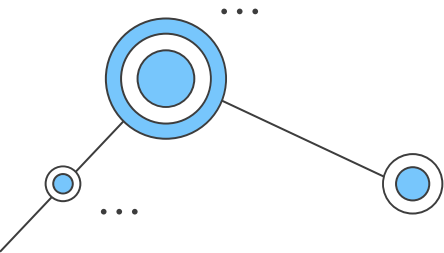
- Poucas evidências **apoando** ou **refutando** o efeito do TDD na qualidade dos testes de unidade em termos de **deteccão de defeitos de código**.
- **Investigar** o impacto do **TDD** na **eficácia** dos casos de **teste de unidade** em comparação com um **desenvolvimento de teste incremental**

Métricas para Testes de Unidade

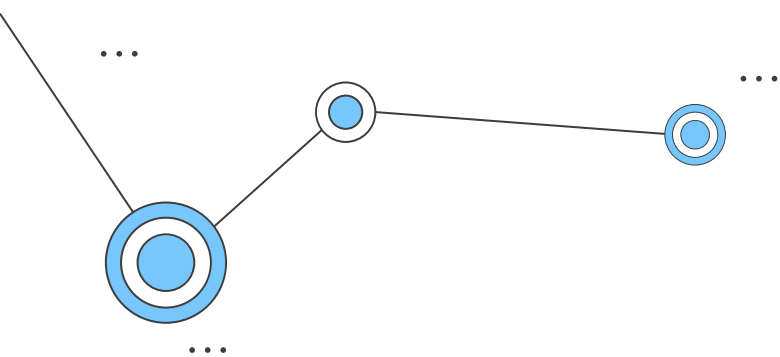




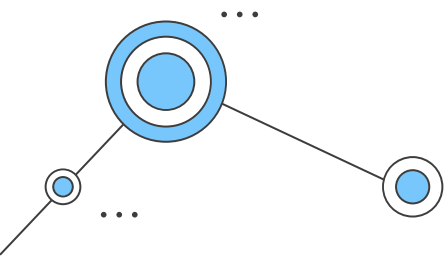
Code Coverage



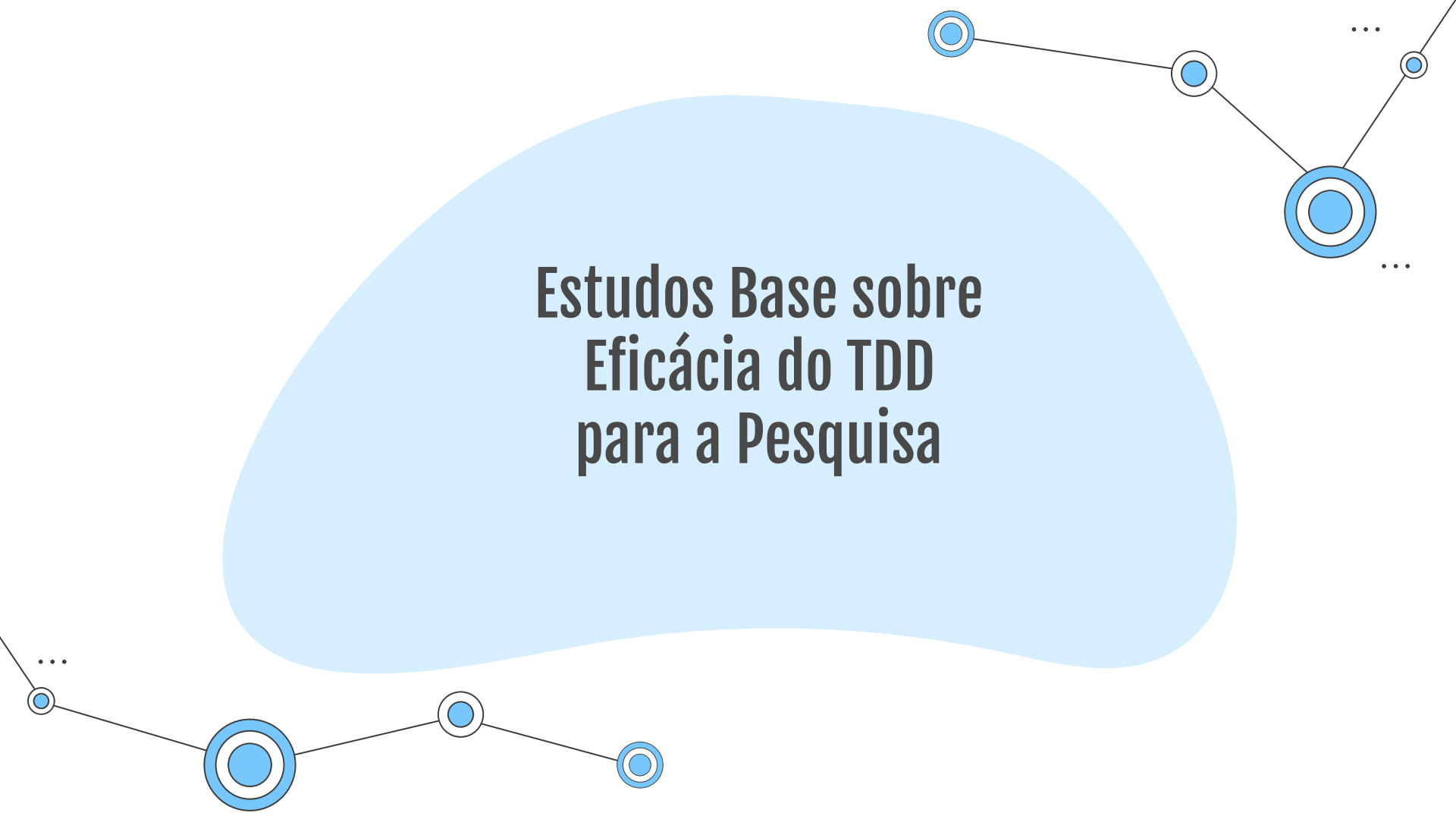
- **Cobertura de código** indica qual parte do código de produção (por exemplo, linha, instruções e ramificações) é exercida / executada durante o teste.
- Diferente formas de medir a **cobertura de código**
 - **Branch coverage**
 - Method coverage
 - Cyclomatic complexity
 - etc
- **Cobertura de código não indica necessariamente a eficácia** de um conjunto de testes; em vez disso, mede a penetração/profundidade (thoroughness) de um conjunto de testes



Mutation Score



- *Mutation testing* é uma forma de medir a **capacidade** de um caso de teste **de detectar defeitos no código**
- Para verificar o quão bom é um conjunto de testes, várias versões do código do programa, nas quais diferentes tipos de defeitos são injetados, podem ser criadas.
- O *mutation score* é calculado como a porcentagem de falhas (mutantes) detectadas pelo conjunto de testes.
- *Mutation score* pode ser usada para medir a eficácia de um conjunto de teste em termos de sua **capacidade de detectar falhas**



Estudos Base sobre Eficácia do TDD para a Pesquisa

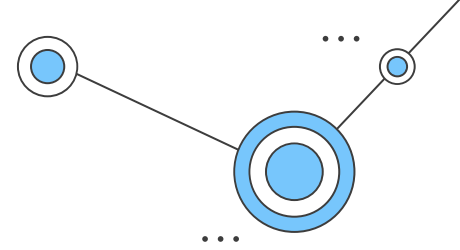
Pesquisas Relacionadas sobre Eficácia do TDD

- O artigo ***Eficácia dos Testes de Unidade no TDD*** de Tosun et al é uma extensão dos seguintes estudos:
 - A. Causevic, D. Sundmark, and S. Punnekkat. 2012. **Quality of testing in test driven development.** In Eighth International Conference on the Quality of Information and Communications Technology.
 - A. Causevic, D. Sundmark, and S. Punnekkat. 2012. **Test case quality in test driven development: A study design and a pilot experiment.** In 16th International Conference on Evaluation & Assessment in Software Engineering (EASE).
 - Madeyski. 2010. **The impact of Test-First programming on branch coverage and mutation score indicator of unit tests: An experiment.** Information and Software Technology 52, 2 (2010).

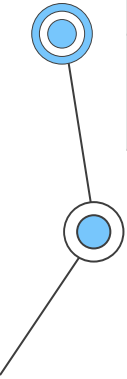
Estudos Base

Titulo	Madeyski	Causevic et al.	Causevic et al.
Objetivo	Comparação entre <i>Test-First</i> (TF) e <i>Test-Last</i> (TL) no que diz respeito à abrangência e eficácia na detecção de falhas dos testes de unidade.	Comparar qualidade dos casos de teste produzidos pelas abordagens Test-first e Test-last	Comparar a eficácia e eficiência do esforço empregado nos testes utilizando Test-first e Test-last
Variável Independente	Abordagem de desenvolvimento: TF ou TL		
Métricas	<i>Branch Coverage</i> e <i>Mutation Score</i>	<i>Statement Coverage</i> e <i>Mutation Score</i>	<i>Failing assertion</i> , <i>Statement Coverage</i> e <i>Mutation Score</i>
Design	Um fator e dois tratamentos		

Estudos Base



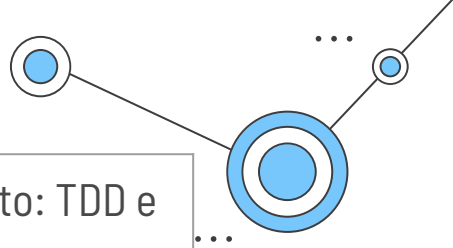
Titulo	Madeyski	Causevic et al.	Causevic et al.
Participantes	22 alunos de mestrado em engenharia de software de terceiro e quarto ano	14 alunos (mestre em engenharia de software)	
Objeto de Estudo	Sistema web para submissão e revisão de artigos	Bowling Score Keeper	
Resultado	Sem diferença significativa	Sem diferença	Sem diferença



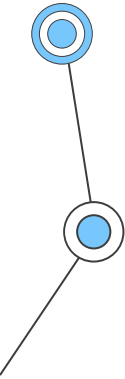


Esperimento

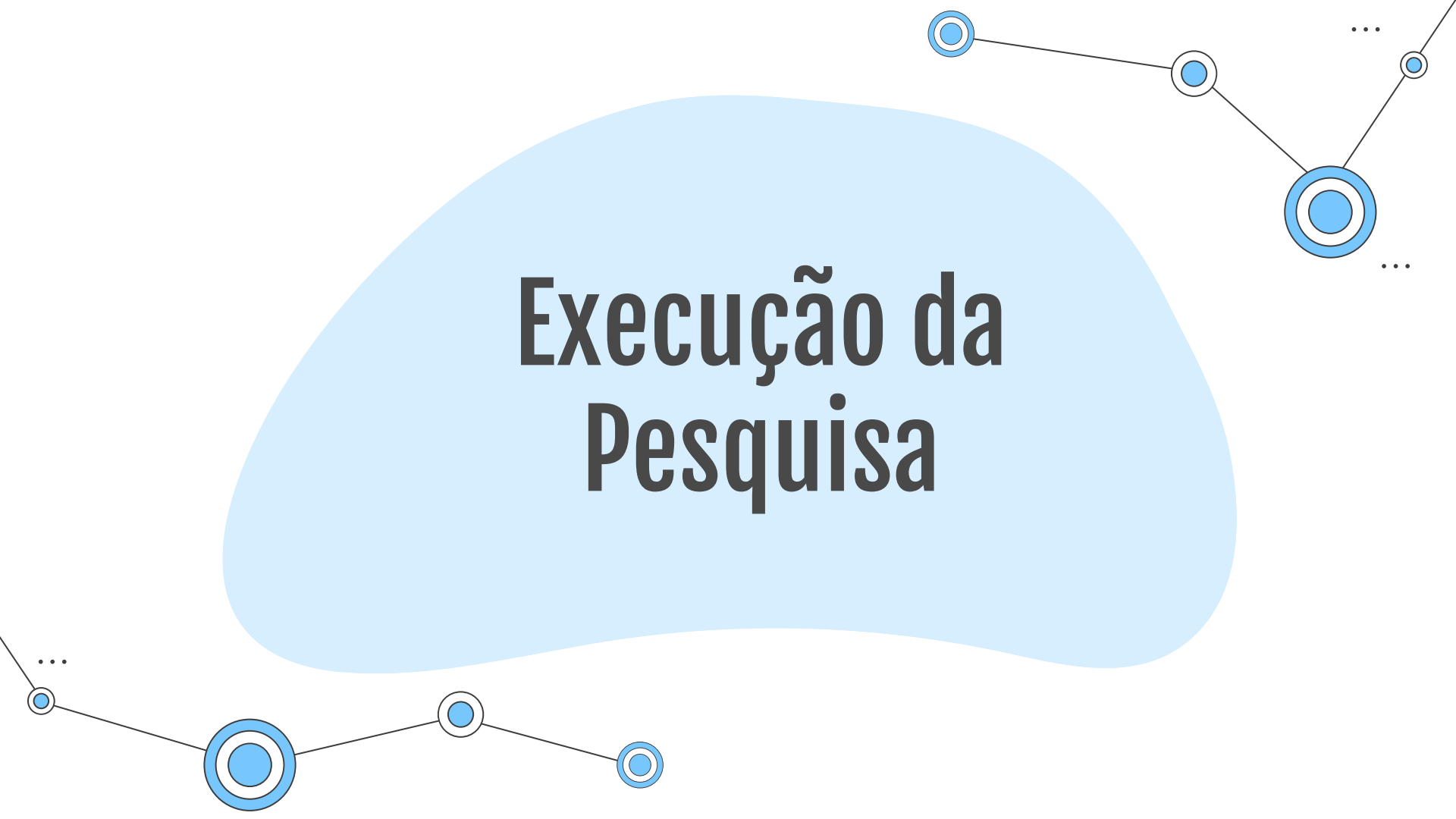




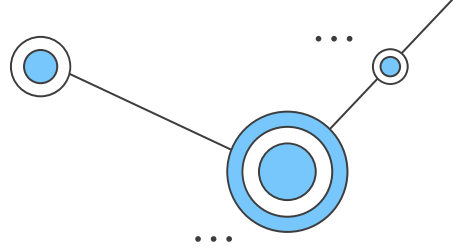
Variável Independente	Abordagem de desenvolvimento: TDD e ITLD
Métricas	<i>Mutation Score, branch coverage e method coverage</i>
Metodologia	Experimento Controlado
Design	Um fator e dois tratamentos
Participantes	24 Professionals sem conhecimento de TDD prévio
Objeto de Estudo	Bowling Score Keeper e Mars Rover API



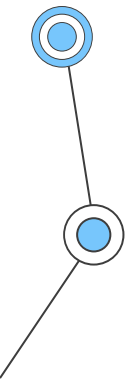
Execução da Pesquisa



Execução da Pesquisa



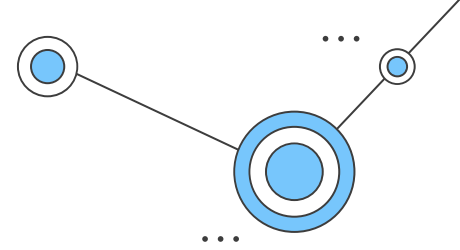
- Todos os participantes foram expostos a ambas as abordagens
 - Participantes receberam treinamento sobre TDD
- Participantes implementaram dois programas utilizando abordagens diferentes
 - MarsRover API utilizando ITLD
 - Bowling Score Keeper utilizando TDD
- Tecnologías utilizadas
 - Java
 - Judy
 - EclEmma



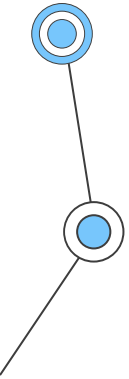


Resultados

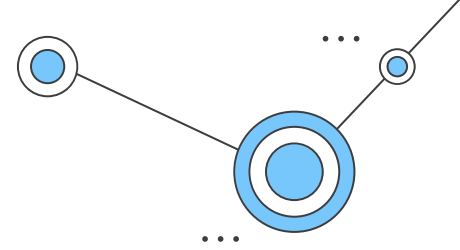
Estatística descritiva – Mutation Scores



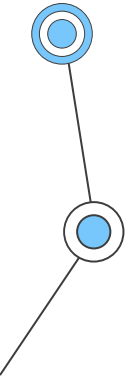
	Número Participantes	Min.	Mediana	Média	Max.	Desvio Padrão
ITLD	13	2	62	54.7	83.0	27.9
TDD	18	6	84	70.6	93.0	28.4



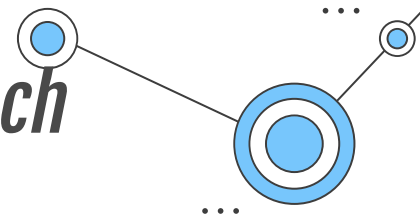
Estatística descritiva – Mutation Scores (Comparação por pares)



	Número Participantes	Min.	Mediana	Média	Max.	Desvio Padrão
ITLD	10	17	67	57.5	83.0	26.17
TDD	10	56	87.5	83.5	93.0	10.69



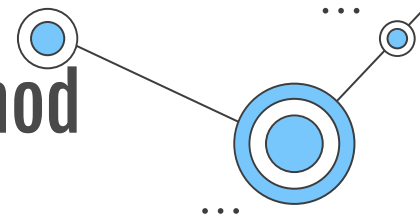
Estatística descritiva – Cobertura de Código *Branch Coverage*



	Min.	Mediana	Média	Max.	Desvio Padrão
ITLD	0	32.5	37.5	96.4	33.8
TDD	0	89.2	59,8	100	45.6



Estatística descritiva – Cobertura de Código Method Coverage



	Min.	Mediana	Média	Max.	Desvio Padrão
ITLD	11.1	87.9	79.1	100	23.8
TDD	0	70.7	61.4	100	30.8

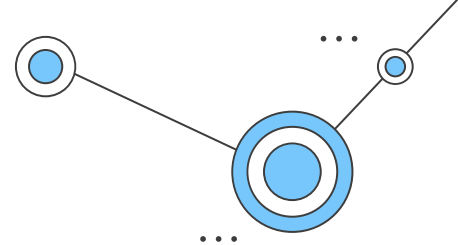


Hipóteses



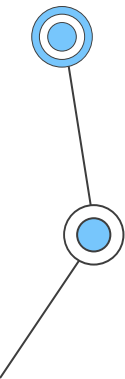
The image features a central light blue cloud-like shape containing the word 'Hipóteses'. Surrounding this central element are several decorative elements: a network of blue circular nodes connected by thin black lines, and three sets of three dots (ellipses) indicating continuation. The nodes vary in size, with some being larger and more prominent than others. The overall design is clean and modern, with a light gray background.

Teste da Hipótese – Mutation Score

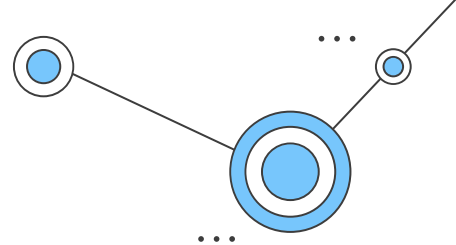


Hipóteses

- $H_0 \mu(MS)_{TDD} = \mu(MS)_{ITLD}$: Não há diferença no mutation score (MS) dos testes de unidade implementados pelos sujeitos durante TDD e ITLD
- $H_1 \mu(MS)_{TDD} \neq \mu(MS)_{ITLD}$: Há uma diferença significativa no mutation score dos testes de unidade implementados pelos sujeitos durante TDD e ITLD

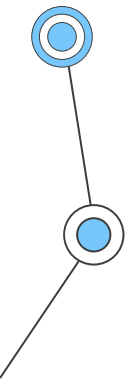


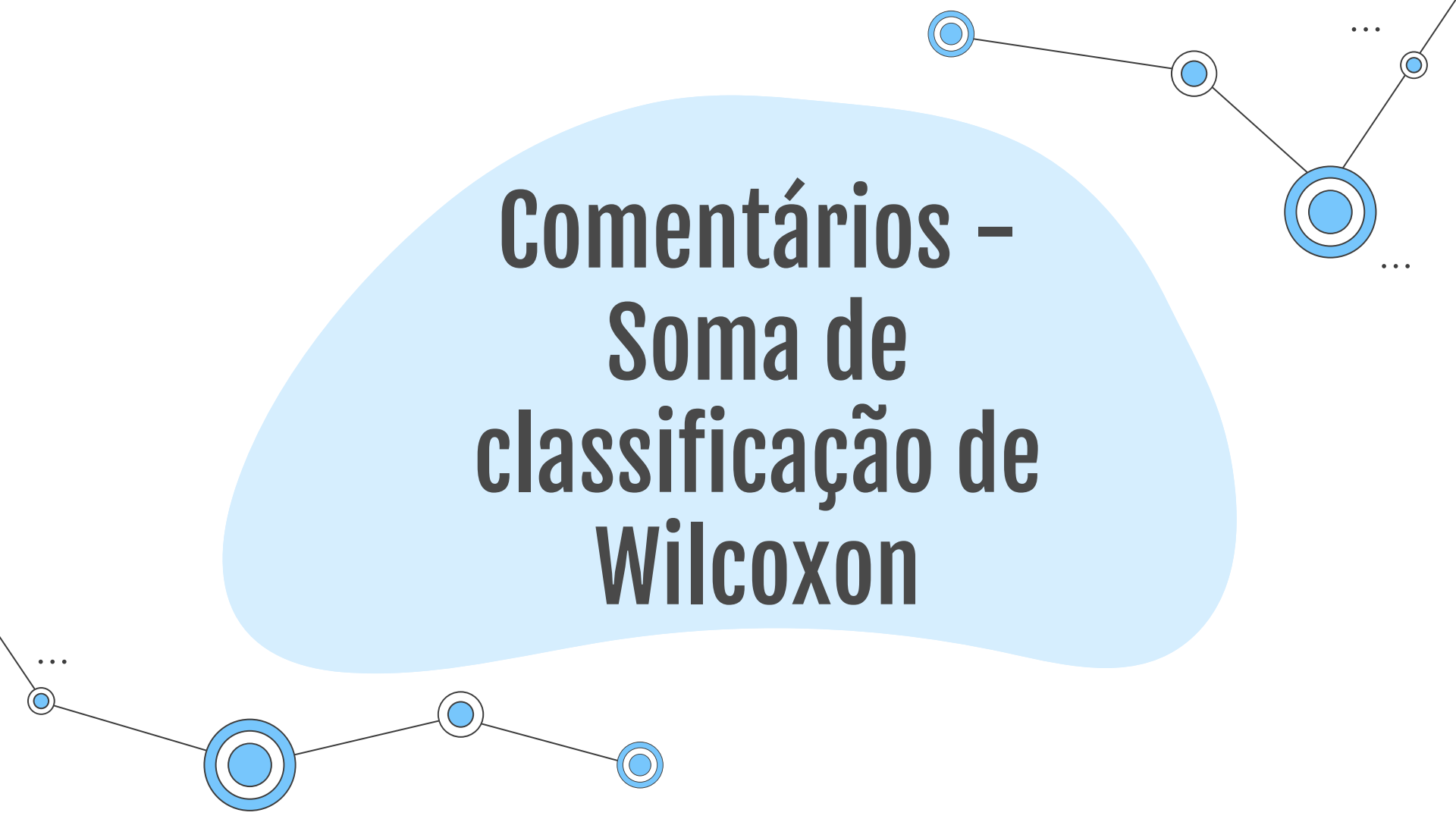
Teste da Hipótese – Cobertura de Código



Hipóteses

- $H_0 \mu(\text{BRCov})_{\text{TDD}} = \mu(\text{BRCov})_{\text{ITLD}}$: Não há diferença entre TDD e ITLD para o branch coverage (BRCov).
- $H_0 \mu(\text{MTCov})_{\text{TDD}} = \mu(\text{MTCov})_{\text{ITLD}}$: Não há diferença entre TDD e ITLD para a cobertura de métodos (MTCov).





Comentários – Soma de classificação de Wilcoxon

Teste da Hipótese

Hipóteses

- $H_0 \mu(MS)_{TDD} = \mu(MS)_{ITLD}$: Não há diferença no mutation score (MS) dos testes de unidade implementados pelos sujeitos durante TDD e ITLD
- $H_1 \mu(MS)_{TDD} \neq \mu(MS)_{ITLD}$: Há uma diferença significativa no mutation score dos testes de unidade implementados pelos sujeitos durante TDD e ITLD
- Nível de significancia = 0.05

- Os resultados do teste de soma de classificação de Wilcoxon entre tarefas ITLD e TDD na pontuação de mutação **mostram que há uma diferença significativa entre a eficácia dos testes de unidade medidos em dois tratamentos (p - valor = 0,04)**.

Teste da Hipótese – Cobertura de Código

Hipóteses

- $H_0\mu(\text{BRCov})_{\text{TDD}} = \mu(\text{BRCov})_{\text{ITLD}}$: Não há diferença entre TDD e ITLD para o branch coverage (BRCov).
- $H_0\mu(\text{MTCov})_{\text{TDD}} = \mu(\text{MTCov})_{\text{ITLD}}$: Não há diferença entre TDD e ITLD para a cobertura de métodos (MTCov).
- Nível de significancia = 0.05

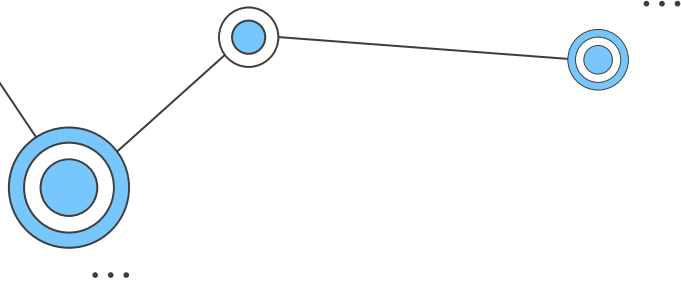
Métricas	p-value	Hipótese
Median _{BRCov}	0.033	Reject
Median _{MTCov}	0.045	Reject



Conclusão

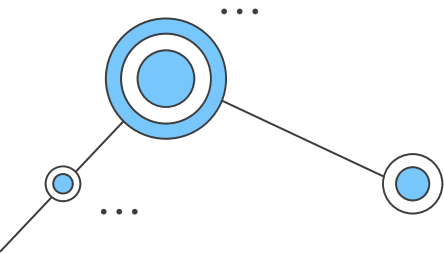


- Os resultados encontrados através da pesquisa **contradizem** os resultados de **experimentos anteriores** sobre a efetividade do TDD em termos de mutation score.
 - TDD melhora o mutation score de testes de unidade em comparação com ITLD
- TDD ajuda na criação de casos de teste mais refinados que cobrem mais *branches*
- ITLD ajuda na criação de casos de teste que cobrem mais métodos do código fonte



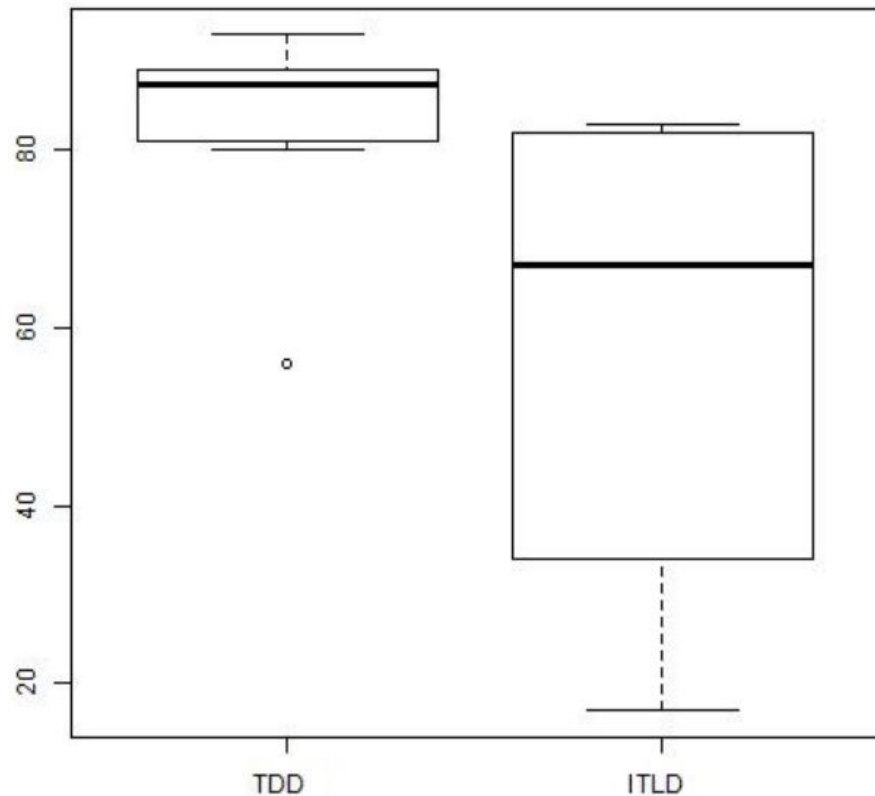
Resultado Obtido vs. Resultado de Pesquisas Anteriores

- Ambas as pesquisas de **Madeysky** e **Causevic et al.** não encontraram nenhuma distinção na qualidade dos testes cases ao seguir a abordagem TDD ou TLD para **branch coverage**.



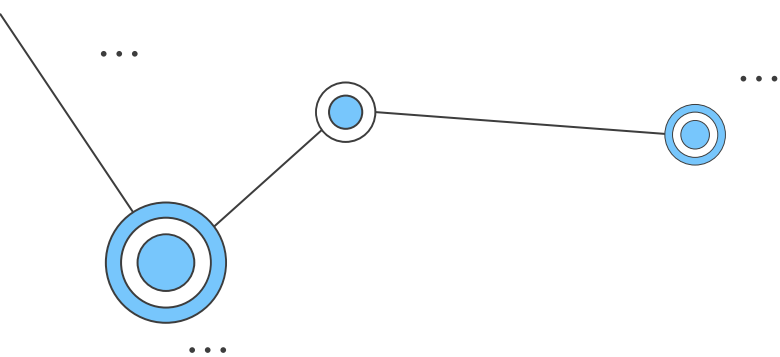


Box plots Mutation Score

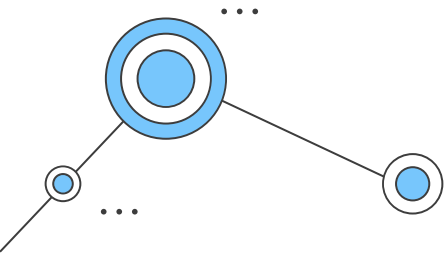




Definições e Conceitos



Teste de Unidade



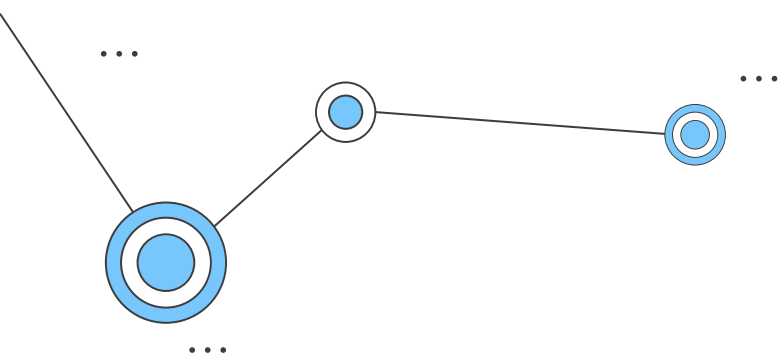
- Hunt e Thomas [22] descrevem os testes de unidade como um pedaço de código escrito por desenvolvedores com a intenção de exercitar / **testar uma funcionalidade específica** em uma pequena área do código.
- Principal **objetivo** é detectar áreas problemáticas do código e reduzir bugs.



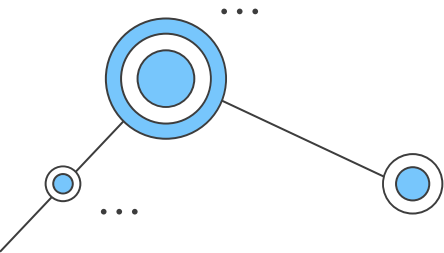
Test-Driven Development (TDD)



- O TDD foi introduzido como uma prática de desenvolvimento de software no início dos anos 1960 durante o projeto Mercury da NASA.
- TDD é um processo de desenvolvimento que requer a escrita e a automatização de testes de unidade antes do início do desenvolvimento de uma unidade do sistema.
- Considerado ser uma metodologia adequada ao desenvolvimento ágil.



Test Smells



- ~ Code Smell
- São más práticas de programação no código de teste de unidade (como a forma como os casos de teste são organizados, implementados e interagem uns com os outros)
- Indicam problemas de design em potencial no código-fonte do teste
- São considerados como **indicadores de qualidade do código de teste** do que uma maneira de entender a eficácia do conjunto de testes