

# TesterDS: usando Teste de Software para estimular o aprendizado de Estruturas de Dados

Helder Linhares Bertoldo dos Reis (helder.bertoldo@ice.ufjf.br)  
Vânia de Oliveira Neves (vania@ice.ufjf.br)  
Departamento de Ciência da Computação - Universidade Federal de Juiz de Fora

## Resumo

TesterDS é uma ferramenta em desenvolvimento que tem como objetivo simplificar o aprendizado de disciplinas voltadas para a área da Computação que, em geral, os alunos têm mais dificuldade como Estruturas de Dados. Apesar dessas disciplinas ganharem maior enfoque na ferramenta, ela também pode ser utilizada em outras de desenvolvimento de Algoritmos.

A ferramenta baseia-se em três abordagens computacionais: TDD, Teste Estrutural e Ludificação. **TDD - Test Driven Development** [1], em português Desenvolvimento Guiado por Testes, é uma técnica ágil de desenvolvimento de software relacionada com o conceito de verificação e validação, baseado num curto ciclo de repetições: primeiro é escrito um caso de teste automatizado para determinada funcionalidade do sistema e, em seguida, o código é desenvolvido e deve ser validado pelo teste criado; por fim, é feita a refatoração desse código. **Ludificação** – termo anteriormente designado à Gamificação, é uma abordagem que visa a utilização de elementos de jogos no ensino para engajar as pessoas em determinado objetivo. Essa abordagem tem se tornado tendência na educação pela sua maior capacidade de interatividade, em relação aos tradicionais métodos de ensino, ampliando a criatividade em busca de soluções. **Teste Estrutural** - é uma técnica de teste também conhecida como caixa branca e que realiza a análise da estrutura do código-fonte para derivar requisitos de teste. Para isso, utiliza grafos de fluxo de controle (GFC) que são capazes de avaliar aspectos como teste de caminhos lógicos, teste de fluxo de dados, teste de condição, teste de ciclos e códigos nunca executados [2].

Com a combinação dessas técnicas no TesterDS, pretende-se estimular e facilitar a implementação desses algoritmos uma vez que o jogo possibilitará dividir o algoritmo em problemas menores e mais simples de implementar além de possibilitar um maior engajamento dos envolvidos. A aplicação é baseada em níveis e subníveis, onde os níveis representam algoritmos a serem implementados e os subníveis são as etapas necessárias para que esse algoritmo seja desenvolvido com sucesso. Os primeiros subníveis são simples e de fácil implementação e vão se tornando mais complexos à medida que o estudante avança no jogo. A Figura 1 apresenta o esquema do TesterDS. Ao entrar em um nível o usuário recebe um código base para iniciar o desenvolvimento do desafio proposto no subnível atual. Terminado o desenvolvimento, ele submete o código para que seja feita a instrumentação deste e é verificado se passou ou não nos casos de testes previamente propostos pelo professor. Caso o retorno seja positivo, compara-se

então a complexidade ciclomática [3] do código do aluno com o código base do professor e, se o código do aluno for mais complexo que o do professor, o aluno deverá fazer a refatoração do código antes de avançar para o subnível seguinte. Nessa etapa também é gerado o GFC e o aluno pode observar quais requisitos do teste estrutural foram cobertos pelos casos de teste fornecidos. O ciclo segue dessa forma até concluir todos os subníveis e, conseqüentemente, tendo desenvolvido seu algoritmo de forma correta. A cada novo subnível, são realizados testes de regressão e, se falharem, o usuário retorna para o último subnível em que os testes foram positivos.

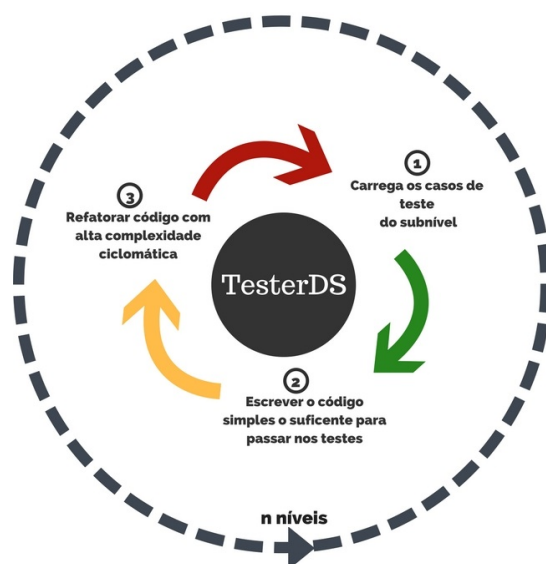


Figura 1: Esquema do TesterDS

O TesterDS é uma aplicação web que está sendo desenvolvida baseando-se em web services Restful como API e utilizando o Spring<sup>1</sup>, um framework do Java. A interface de usuário está sendo criada utilizando Angular<sup>2</sup> na versão 4, um framework *Front-end* em Typescript<sup>3</sup>.

## Referências bibliográficas

- [1] Beck. 2002. Test Driven Development: By Example. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [2] E. F. Barbosa, M. L. Chaim, A. M. R. Vincenzi, M. E. Dela-maró, M. Jino, and J. C. Maldonado. 2007. Teste Estrutural. Elsevier, 47–76.
- [3] Thomas J. McCabe. 1976. A Complexity Measure. In Proceedings of the 2Nd International Conference on Software Engineering (ICSE '76). IEEE Computer Society Press, Los Alamitos, CA, USA, 407–.

<sup>1</sup><https://spring.io>

<sup>2</sup><https://angular.io>

<sup>3</sup><http://www.typescriptlang.org>