

Testes Unitários Automatizados Utilizando O Framework JUnit

Trabalho II da Disciplina de Qualidade de Software – UFSM/2019-2

Nome do Aluno: Otávio da Cruz Mello

Curso: Sistemas de Informação

Data de Entrega: 27/11/2019

Este relatório tem como intuito apresentar alguns dos resultados obtidos com o uso do framework open-source JUnit para testes de software em Java, bem como relatar a experiência com o uso e passos para realizar os testes desenvolvidos. Para realizar todas as atividades deste projeto foi utilizado o Ambiente de Desenvolvimento Eclipse, que possuía o JUnit integrado em seu sistema.

Foram desenvolvidos no total quarenta e quatro (44) testes diferentes para os métodos relacionados ao jogo, divididos em três grupos (nesse caso, classes) principais. Na primeira classe *CardTest* temos a maior quantidade de testes. Os testes nessa classe estão normalmente associados a questões de gerações de cartas dentro do jogo e controle de pilhas de cartas. Na segunda classe *SolitaireTest*, temos testes mais voltados para elementos de estado e informações do jogo em uma visão geral. Por fim, na terceira classe, *UtilTest*, a menor, se tem testes relacionados a atributos de interface.

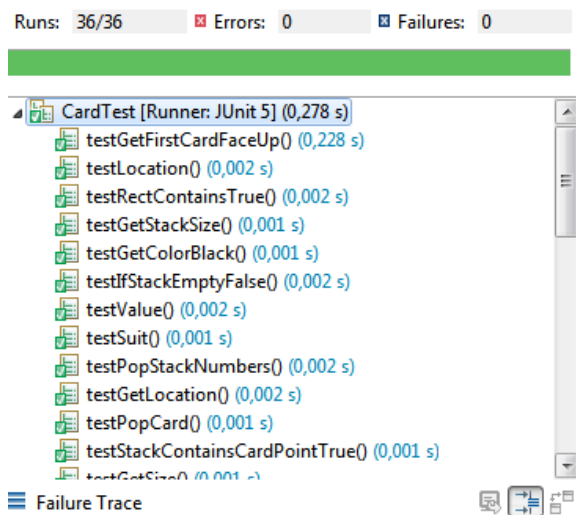


Figura 1 - Resultados dos Testes na classe CardTest

No primeiro caso, tivemos trinta e seis (36) testes diferentes, e em cada um foi atribuído e retornado valores a fim de garantir que os métodos estavam funcionando adequadamente. Obteve-se um índice de 100% de aprovação, com o sistema não apresentando nenhuma falha com a execução dos testes escritos, o que pode ser visualizado na Figura 1.

O segundo caso, na classe de testes voltada para o armazenamento de informações acerca do estado do jogo, foram

escritos um total de sete (7) outros testes adicionais. Assim como na primeira, também se obtiveram resultados muito positivos, com a aprovação máxima em cada um dos testes. O que também não foi diferente na terceira classe de testes, com dois (2) outros testes que também foram aprovados em sua execução. Abaixo, na Figura 2 e na

Figura 3, é possível ter uma visão dos resultados dessas duas outras classes também.

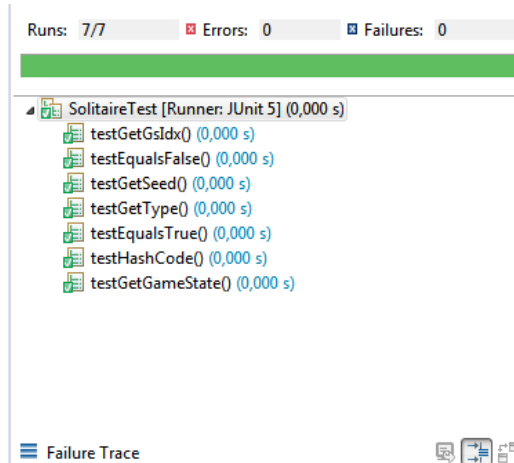


Figura 2 - Resultados na classe SolitaireTest

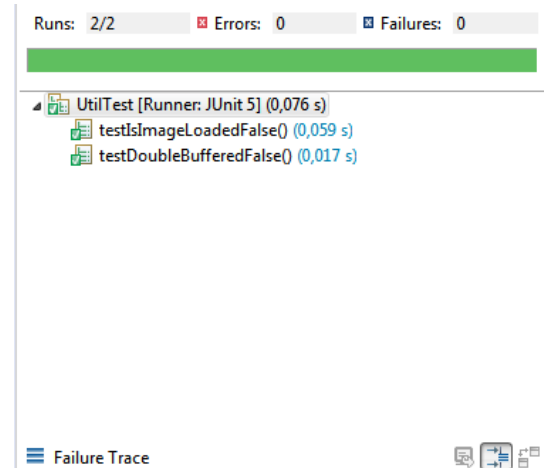


Figura 3 - Resultados dos Testes na classe UtilTest

Existiram alguns poucos casos em que não foi possível se escrever casos de teste, pois algumas classes e métodos necessitavam de atributos que não poderiam ser gerados sem a execução plena do sistema. Se levou em consideração laços de repetição, estruturas de decisão (condicionais), e exceções.