




Com base nos dados que foram obtidos pelo Jacoco, pôde-se realizar uma análise mais detalhada acerca da cobertura de testes. Uma vez que, isso nos permite verificar quantas instruções, branches(arestas), linhas de código, métodos e classes estão sendo realmente utilizadas, quando os testes estão sendo executados.

A seguir, pode-se verificar um resumo da análise global do projeto:










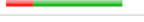







Batalhas

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
 batalha		57%		66%	35 78	59 133	24 48	2 6
Total	253 of 596	57%	20 of 60	66%	35 78	59 133	24 48	2 6

Note que, diante dos dados que estão sendo expostos, que obtivemos uma cobertura de 57% das instruções que foram avaliadas e 66% de cobertura das arestas. E em conjunto, também podemos observar que linhas, métodos e classes foram perdidos durante as análises realizada.

Agora, de maneira mais específicas podemos verificar as métricas relativas a cada classe testada no projeto:

batalha

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Cxty	Missed Lines	Missed Methods	Missed Classes
 Batalha		45%		41%	16 26	25 45	9 14	0 1
 PersonagemBuilder		0%	n/a	n/a	9 9	13 13	9 9	1 1
 Main		0%		0%	4 4	11 11	2 2	1 1
 Personagem		76%		75%	6 23	10 50	4 19	0 1
 Guerreiro		100%		100%	0 8	0 7	0 2	0 1
 Assassino		100%		100%	0 8	0 7	0 2	0 1
Total	253 of 596	57%	20 of 60	66%	35 78	59 133	24 48	2 6

Desses dados, podemos extrair informações acerca da cobertura das instruções. E diante desses dados podemos verificar que a classe Batalha teve 45% de suas instruções cobertas, o que permite sugerir que 55% das instruções não foram executadas durante os testes. Já em relação às classes Main e PersonagemBuider temos uma cobertura de 0%, o que indica que essas classes não foram utilizadas durante a execução dos testes. Diferentemente da classe Personagem que teve 76% de cobertura de suas instruções testadas. O que pode ser compreendido como uma medida razoável, que necessita de mais testes para aumentar a sua confiabilidade, para com isso seguir o mesmo padrão das classes Guerreiro e Assassino que tiveram 100% de cobertura de suas instruções.

Em relação a cobertura de arestas, a classe Batalha obteve 45% de cobertura. O que permite indicar que existem condições presentes no código que não foram testadas. Enquanto que na classe Personagem obtivemos 75% de cobertura das arestas, o que permite sugerir a necessidade de uma reanálise dos testes para obter dados mais robustos e semelhantes às classes Guerreiro e Assassino. Uma vez que obtiveram 100% de cobertura das arestas. O que significa que todas as condicionais presentes nessas classes foram testadas.

Dentre esses dados avaliados também precisamos levar em consideração as linhas de código não cobertas, uma vez que representam as linhas de código que não foram executadas, principalmente nas classes Batalha(35 linhas) e Personagem(23 linhas). E em adição, também temos os métodos não foram executados nas classes PersonagemBuilder e Main.

Portanto, com a obtenção dessa visão global do projeto, conseguimos obter uma rápida análise sobre a cobertura de testes em relação ao código desenvolvido. E para melhorar a cobertura, podemos escrever mais testes, principalmente para as classes que tiveram baixa cobertura, com a finalidade de obter dados mais confiáveis.