

Relatório de cobertura de testes

Cobertura de testes “JUnit Coverage” – Métricas escolhidas (Métodos e Linhas)

Relativamente à cobertura de testes referente à parte do software desenvolvido para o nosso projeto, como mencionado no enunciado, decidimos utilizar o plug-in Eclipse-EclEmma, este plug-in permitiu-nos avaliar o modo como as classes operavam e forçou-nos a explorar todas as hipóteses possíveis para melhor cobertura de testes relativamente a cada classe em particular. Utilizamos ainda o JUnit5. Ao longo do desenvolvimento das classes “**teste.java” tivemos de adequar os testes realizados de forma a conseguir cobrir todas as possibilidades que a classe iria ter de explorar. Optamos por dar mais ênfase aos pacotes ‘extraction’ e ‘detection’ porque são os pacotes que mais influenciam a cobertura total do projeto. Notámos ainda, que ao realizar testes para determinadas classes estávamos a aumentar indiretamente a cobertura de outras classes.

Passando um pouco para a parte mais prática é de mencionar que optamos por utilizar maioritariamente “AssertEquals” e “AssertNull” ao longo do desenvolvimento das classes de teste.

Ao desenvolvermos a GUI, optámos por criar um pacote de auxílio à biblioteca SWT, biblioteca que utilizámos para o desenvolvimento da GUI, daí a cobertura de testes do projeto em si acabar por ser bastante superior à indicada nas figuras.

De seguida apresentamos as métricas por nós escolhidas e ainda a de instruções como mais um exemplo de cobertura superior a 75%.

Element	Coverage	Covered Methods	Missed Methods	Total Methods
Grupo47	65,7 %	333	174	507
> src/test/java	96,9 %	93	3	96
> src/main/java	58,4 %	240	171	411
> org.eclipse.wb.swt	0,0 %	0	20	20
> gui	0,0 %	0	124	124
> G47.Grupo47	0,0 %	0	2	2
> extraction	100,0 %	33	0	33
> excel	100,0 %	6	0	6
> detection	96,7 %	117	4	121
> classes	80,0 %	84	21	105

Figura 1 – Resultado apresentado no “Coverage” após a realização do teste JUnit – Métodos
Média = 94.175%

Como é possível observar no resultado apresentado na Figura 1, conseguimos obter uma cobertura de 65.7% para o projeto, valor distorcido pelo facto de não terem sido realizados testes de cobertura para os pacotes referentes a classes com código que corresponde à GUI da aplicação.

Element	Coverage	Covered Lines	Missed Lines	Total Lines
Grupo47	65,4 %	2 269	1 201	3 470
> src/test/java	98,1 %	1 447	28	1 475
> src/main/java	41,2 %	822	1 173	1 995
> org.eclipse.wb.swt	0,0 %	0	137	137
> gui	0,0 %	0	901	901
> G47.Grupo47	0,0 %	0	11	11
> extraction	100,0 %	137	0	137
> excel	100,0 %	64	0	64
> detection	96,9 %	434	14	448
> classes	63,0 %	187	110	297

Figura 2 – Resultado apresentado no “Coverage” após a realização do teste JUnit – Linhas

Média = 89.975%

Na figura 2, referente às instruções que temos no projeto, podemos observar que a cobertura de testes não atinge os 75% pelas mesmas razões que na figura 1.











Element	Coverage	Covered Lines	Missed Lines	Total Lines
▼ Grupo47	 65,4 %	2 269	1 201	3 470
> src/test/java	 98,1 %	1 447	28	1 475
▼ src/main/java	 41,2 %	822	1 173	1 995
> org.eclipse.wb.swt	 0,0 %	0	137	137
> gui	 0,0 %	0	901	901
> G47.Grupo47	 0,0 %	0	11	11
> extraction	 100,0 %	137	0	137
> excel	 100,0 %	64	0	64
> detection	 96,9 %	434	14	448
> classes	 63,0 %	187	110	297

Figura 3 – Resultado apresentado no “Coverage” após a realização do teste JUnit – Instruções

Média = 87.25%

Como é possível observar nas figuras acima, atingimos uma média superior a 75% de cobertura de testes para as 3 métricas propostas.