

Relatório de cobertura de testes

Tal como indicado no enunciado precisávamos de atingir uma cobertura igual ou superior a 50% nos testes JUnit numa das três métricas indicadas, Linhas de código, Instruções ou Complexidade ciclomática. Escolhemos como métrica a Complexidade Ciclomática.

É necessário ter em conta que uma grande parte do nosso projeto consiste em classes de interfaces gráficas, feitas para representar calendários, menus entre outras funcionalidades ao utilizador, tais como:

- Calendar;
- CalendarViews;
- MonthlyCalendar;
- MembersList;
- ReunionWindow;
- Login;

Todas as classes acima mencionadas não foram submetidas aos testes JUnit, pois seria impossível obter resultados válidos através dos mesmos.

Tendo em conta o mencionado acima decidimos submeter apenas as classes que consideramos serem “Back-End”, estas são as classes:

- CalendarEvent;
- Event;
- Week;
- Schedule;
- DayCalendar;
- WeekCalendar;
- Element;
- TimeOfDay;
- CalendarEventClickEvent;
- CalendarEmptyClickEvent;
- IntegerFilter;

Com estas classes conseguimos testar e obter resultados válidos que podem ser representados em código escrito e não graficamente / visualmente.

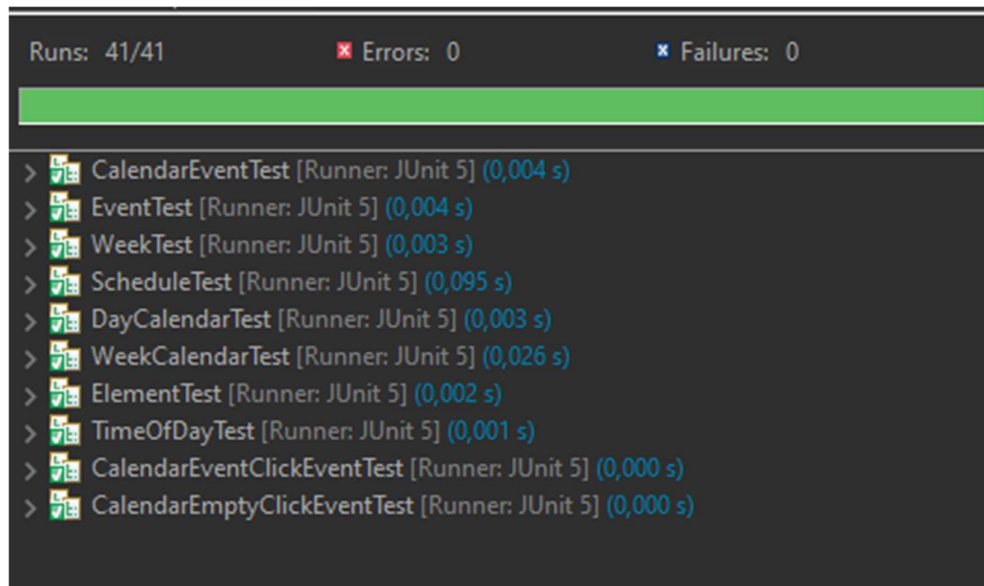


Figura 1. Resultados dos testes

Como é possível verificar na **Figura 1.**, todos os testes que subtemos as nossas classes Back-End retornaram positivos e sem nenhum erro.

Element	Coverage	Covered Complexity	Missed Complexity	Total Complexity
ES_LETI_1Semestre_2022_GRUPO_07	35,7 %	166	299	465
src/test/java	100,0 %	61	0	61
src/main/java	26,0 %	105	299	404
ES_LETI_1Semestre_2022_GRUPO_07.ES_LET	26,0 %	105	299	404
CalendarEmptyClickEvent.java	100,0 %	2	0	2
CalendarEventClickEvent.java	100,0 %	2	0	2
TimeOfDay.java	100,0 %	6	0	6
Element.java	66,7 %	8	4	12
Week.java	66,7 %	8	4	12
DayCalendar.java	60,0 %	6	4	10
WeekCalendar.java	58,3 %	7	5	12
Event.java	55,3 %	21	17	38
CalendarEvent.java	45,0 %	9	11	20
Schedule.java	39,7 %	31	47	78
Calendar.java	7,8 %	5	59	64
CalendarViews.java	0,0 %	0	34	34
CalendarWindow.java	0,0 %	0	17	17
IntegerFilter.java	0,0 %	0	6	6
Login.java	0,0 %	0	18	18
Main.java	0,0 %	0	2	2
MembersList.java	0,0 %	0	2	2
MonthlyCalendar.java	0,0 %	0	51	51
ReunionWindow.java	0,0 %	0	18	18

Figura 2. Cobertura das classes do projeto seguindo a métrica Cognitive Complexity

Referindo o citado no início do relatório decidimos contar para a percentagem da cobertura apenas as classes Back-End. Observando a **Figura 2.** verificamos que:

Total Complexity: 198 | Covered Complexity: 115 | Missed Complexity: 83 (Back-End)

Isto significa que a cobertura é de **58%**, cumprindo com o requisito de estar acima dos 50%.