



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ

CURSO DE ENGENHARIA DE SOFTWARE

RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE

Sistema Biblioteca (projeto web do curso de Ciência da Computação - UEPB)

Equipe:

Robertson da Silva Nascimento - 391242

Francisco Bruno Lopes Silva - 428966

Professora: Carla Ilane Moreira Bezerra

QUIXADÁ

Julho, 2021

1. DESCRIÇÃO DO PROJETO

O projeto foi criado para ser um Sistema Biblioteca e foi um projeto web do curso ciência da computação da UEPB. Utiliza majoritariamente JAVA e OO, é de código aberto e como tecnologias ele usa JAVA(51.7%), JavaScript (39,4%) e CSS (8,9%).

Link do projeto: <https://github.com/zemalay/biblioteca>

Tabela 1 – Características do Projeto

Projeto	LOC	# de classes	# de releases
Sistema Biblioteca	4525	56	1

2. AVALIAÇÃO DO PROJETO

2.1. Medição 1 - Antes de refatorar o projeto

Tabela 2 - Medição dos atributos antes de refatorar o projeto.

Atributo Interno de Qualidade	Métrica	Valor da métrica	Total do atributo
Coesão	LCOM	2955	2955
Complexidade	ACC	73	1223
	SCC	638	
	EVG	454	
	Nesting	58	
Herança	DIT	64	140
	NOC	12	
	Base Classes	64	
Acoplamento	CBO	207	207
Tamanho	LOC	4525	6309
	CLOC	1309	
	NIM	419	
	CDL	56	

2.2. Detecção de Codes Smells

Tabela 3 – Code smells do projeto de acordo com a ferramenta JSPIRIT.

Nome do Code Smell	Quantidade
Shotgun Surgery	10
Intensive Coupling	11
God Class	1
Feature Envy	24
Dispersed Coupling	17
Total	63

2.3. Medição 2 - Após refatorar 20 code smells do tipo Feature Envy

O code smell que logo de cara decidimos refatorar foi o Feature Envy, pois foi este o tipo que teve maior número de ocorrências no projeto. Mesmo deixando claro que ainda restaram 4 ocorrências desse code smell, sendo que duas delas optamos por não refatorar por motivos de facilidade na leitura do código. Queremos deixar claro que utilizamos método manual de refatoração, com Extract Method e Move Method, sendo predominante o uso de Extract Method, pois o Move foi usado apenas na reestruturação de alguns construtores.

Tabela 4 – Resultado da medição após a refatoração de Feature Envy.

Atributo Interno de Qualidade	Métrica	Valor da métrica	Total do atributo
Coesão	LCOM	3117	3117
Complexidade	ACC	70	1307
	SCC	682	
	EVG	497	
	Nesting	58	
Herança	DIT	64	140
	NOC	12	
	Base Classes	64	
Acoplamento	CBO	207	207

Tamanho	LOC	4649	6481
	CLOC	1309	
	NIM	467	
	CDL	56	

Legenda: Maior; Igual; Menor

2.4. Medição 3 - Após a refatoração de 20 code smells, sendo 10 Shotgun Surgery e 10 Dispersed Coupling.

Os últimos dois code smells que decidimos refatorar foram o Shotgun Surgery e o Dispersed Coupling levando em conta o total de ocorrência de ambos.

Para refatorarmos Shotgun Surgery foi utilizado Move Method, sendo que majoritariamente criamos uma nova classe estática para tratar a utilização do método no qual está presente o code smell. Desse modo, as classes que utilizavam o método anteriormente, agora se relacionam com um controlador da classe que contém o método shotgun, passando o objeto da classe de origem para um método do seu respectivo controlador, e dentro do controlador é chamado esse método do objeto de origem, que agora já não contém mais Shotgun Surgery.

Apesar de Dispersed Coupling não ter métodos bem definidos de refatoração, conseguimos refatorar utilizando Extract Method e Move Field também para a classe estática criada anteriormente, porém aplicado de maneira um pouco diferente, resumindo, a parte de código na qual estava presente o code smell, nós optamos por movê-la para um método da classe estática, e chamar sua execução no método com dispersed, assim, o eliminando.

Tabela 4 – Resultado da medição após a refatoração de Shotgun Surgery e o Dispersed Coupling.

Atributo Interno de Qualidade	Métrica	Valor da métrica	Total do atributo
Coesão	LCOM	3145	3145
Complexidade	ACC	83	1363
	SCC	701	
	EVG	531	
	Nesting	66	
Herança	DIT	76	
	NOC	12	

	Base Classes	76	164
Acoplamento	CBO	283	283
Tamanho	LOC	4831	6689
	CLOC	1309	
	NIM	481	
	CDL	68	

Legenda: Maior; Igual; Menor

3. ANÁLISE APÓS TODAS AS REFATORAÇÕES

Atributo Interno de Qualidade	Feature Envy	Shotgun Surgery e Dispersed Coupling	Final
Coesão	LCOM	LCOM	+190
Complexidade	ACC	ACC	+10
	SCC	SCC	+63
	EVG	EVG	+77
	Nesting	Nesting	+8
Herança	DIT	DIT	+12
	NOC	NOC	0
	Base Classes	Base Classes	+12
Acoplamento	CBO	CBO	+76
Tamanho	LOC	LOC	+306
	CLOC	CLOC	0
	NIM	NIM	+62
	CDL	CDL	+12

Legenda: Maior; Igual; Menor;

Com base nos resultados mostrados na tabela acima, chegamos a conclusão que ao refatorar Feature Envy:

Coesão: Apenas aumentou;

Complexidade: ACC foi a única medida que apresentou melhora. Porém, de modo geral a complexidade aumentou;

Herança: Se manteve a mesma;

Acoplamento: Se manteve o mesmo;

Tamanho: Se manteve o mesmo em algumas métricas, mas aumentou de modo geral.

Ao refatorar Shotgun Surgery e Dispersed Coupling: Primeiro é importante dizer que decidimos realizar a medição logo da refatoração desses dois code smells por questões de representatividade, e impacto na análise, tendo em vista que as técnicas de refatoração usadas foram parecidas;

Coesão: Apenas aumentou;

Complexidade: Apenas aumentou;

Herança: NOC foi a única medida que se manteve a mesma. Porém, de modo geral a herança aumentou.

Acoplamento: Apenas aumentou.

Tamanho: Se manteve o mesmo em CLOC, mas aumentou de modo geral.

Nesse contexto, o que tiramos de aprendizado é que em certos casos, a refatoração de código tende a aumentar a complexidade, fazendo com que o projeto perca coesão e acoplamento, além de que sem dúvida ocorra aumento de tamanho. Com isso em vista, é importante o demasiado estudo do projeto e quais são suas aplicações, com seus criadores etc, para somente assim, partir para a refatoração. Porém, fica claro que dependendo do tipo de code smell e da técnica de refatoração aplicada, é certo se esperar a melhora de algumas métricas, como por exemplo ao remover 10 aparições de feature envy, detectamos uma melhora na métrica ACC na categoria de complexidade.