



**UNIVERSIDADE
FEDERAL DO CEARÁ**
CAMPUS DE QUIXADÁ

CURSO DE ENGENHARIA DE SOFTWARE

**RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE
JAVA-PROJECT-2017 (Flight-System)**

Equipe:

Diego Feitoza de Oliveira - mat. 412981

Mônica Y. C. da Silva - mat. 469914

Professora:

Carla Ilane Moreira Bezerra

QUIXADÁ

Agosto, 2021

SUMÁRIO

DESCRIÇÃO DO PROJETO	1
AVALIAÇÃO DO PROJETO	1
Medição 1 – Antes de refatorar o projeto	3
Detecção dos Code Smells	4
Medição 2 – Após Refatorar o Code Smell Feature Envy	4
Medição 3 – Após refatorar o Code Smell Intensive Coupling	6
Medição 4 – Após refatorar o Code Smell Brain Method	7
Medição 5 – Após refatorar o Code Smell Dispersed Coupling	8
Medição Final – Após refatorar o Code Smell God Class	9
COMPARAÇÃO DOS RESULTADOS	11
REFERÊNCIAS	12

1 DESCRIÇÃO DO PROJETO

O projeto selecionado é um projeto Java orientado a objeto, que está disponível publicamente no Github e é descrito da seguinte forma pelo autor:

“Este é um projeto de sistema de voos que foi desenvolvido na SUSTC (Southern University of Science and Technology).

As características deste sistema de voo são: contém todas as funções necessárias; utiliza arquivos xml para salvar dados; salva dados em arquivo e altera o status do voo automaticamente; é capaz de pesquisar a estação de transferência; permite a busca por data; possui alta extensibilidade; usa exceção personalizada para lidar com problemas de status e permissão.”

Basicamente é um sistema por linha de comando que permite cadastrar usuários, voos, passageiros, emitir bilhetes e efetuar pagamentos, porém com informações e recursos bem detalhados.

Link do projeto (origem): <https://github.com/dong-zeyu/java-project2017>.



Link do Fork: <https://github.com/DiFeitoza/java-project2017>.

A tabela a seguir traz algumas informações sobre o projeto.

Tabela 1 – Características do Projeto

Projeto	LOC	# de classes	# de releases
java-project-2017 (Flight-System)	2076	19	0

2 AVALIAÇÃO DO PROJETO

Foram geradas duas planilhas principais para acompanhamento das etapas de refatoração e medições. A primeira,  histórico de refatorações - code smells , contém dados para rastreio dos code smells existentes, removidos e gerados a cada sessão de refatoração de um tipo específico de code smells. A segunda,  metricas de qualidade - java-project2017 , traz todas as métricas geradas, incluindo a medição inicial mais métricas de cada refatoração e também tabelas resumo sobre os resultados a cada nova medição. Ao todo foram feitas seis

medições, a inicial e outras cinco, sendo cada uma feita após a refatoração de um tipo x de code smell e sua finalização.

Para melhor compreensão das métricas e dos atributos de qualidade utilizados para avaliar o impacto das refatorações no projeto, leia a tabela abaixo:

Tabela 2 - Métricas usadas para medir os atributos internos de qualidade.

Atributo	Métrica	Descrição
Coesão	LCOM	Mede a coesão de uma classe. Quanto maior o valor dessa métrica, menos coesiva é a classe.
Complexidade	ACC	Média da complexidade ciclomática de todos os métodos. Quanto maior o valor dessa métrica, mais complexos são as classes e métodos.
Herança	SCC	Somatório da complexidade ciclomática de todos os métodos. Quanto maior o valor dessa métrica, mais complexos são as classes e métodos.
	EVG	Mede o grau na qual um módulo contém construtores não estruturados. Quanto maior o valor dessa métrica, mais complexos são as classes e métodos.
	Nesting	Nível Máximo de Aninhamento de construções de controle. Quanto maior o valor dessa métrica, maior é a complexidade de classes e métodos
	DIT	O número de níveis que uma subclasse herda de métodos e atributos de uma superclasse na árvore de herança Quanto maior o valor dessa métrica, maior é o grau de herança de um sistema.
Acoplamento	NOC	Número de subclasses de uma classe. Quanto maior o valor dessa métrica maior é o grau de herança de um sistema.

	Base Classes	Número imediato de classes base. Quanto maior o valor dessa métrica, maior o grau de herança de um sistema.
	CBO	Número de classes que uma classe está acoplada. Quanto maior o valor dessa métrica, maior é o acoplamento de classes e métodos.
Tamanho	LOC	Número de linhas de código, excluindo espaços e comentários. Quanto maior o valor dessa métrica, maior é o tamanho do sistema.
	CLOC	Número de linhas com comentários. Quanto maior o valor dessa métrica, maior é o tamanho do sistema.
	NIM	Número de métodos de instância. Quanto maior o valor dessa métrica, maior é o tamanho do sistema.
	CDL	Número de classes. Quanto maior o valor dessa métrica, maior é o tamanho do sistema.

2.1 Medição 1 – Antes de refatorar o projeto

Para compreendermos o impacto das refatorações que faríamos no projeto, precisamos analisar as métricas iniciais que são: coesão, acoplamento, complexidade, herança e tamanho. Para realizar a detecção dos dados, foi utilizada a ferramenta “Understand”. A tabela abaixo apresenta a descrição das métricas:

Tabela 3 – Medição dos atributos antes de refatorar o projeto.

MEDIÇÃO 1 - ESTADO INICIAL DO SOFTWARE			
Atributo Interno de Qualidade	Métrica	Medição	Total do Atributo
Coesão	LCOM	934	934
Complexidade	ACC	520	2.249
	SCC	1.443	
	EVG	82	

	Nesting	204	
Herança	DIT	25	54
	NOC	2	
	Base Classes	27	
Acoplamento	CBO	56	56
Tamanho	LOC	2.076	2.426
	CLOC	163	
	NIM	168	
	CDL	19	

2.2 Detecção dos Code Smells

Após a detecção dos code smells com a ferramenta Understand, utilizamos a ferramenta JSpirit para conseguirmos elaborar a tabela abaixo. Ela apresenta os tipos de code smells, sua quantidade e a quantidade total de code smells do projeto. No total foram identificados 6 tipos de code smells e o total de 41 code smells.

Tabela 4 – Code Smells do projeto.

Nome do Code Smell	Quantidade	Total
Intensive Coupling	3	41
God Class	1	
Feature Envy	23	
Dispersed Coupling	9	
Data Class	1	
Brain Method	4	

2.3 Medição 2 – Após Refatorar o Code Smell Feature Envy

O primeiro code smell a ser refatorado foi do tipo Feature Envy. Foram feitas refatorações até que ele não fosse mais detectado pela ferramenta JSpirit. Os dados da Tabela 3 foram gerados após a segunda medição ser realizada, que é referente a essa refatoração.

A técnica de refatoração utilizada para retirar esse code smell foi essencialmente aplicar Extract Method em todas as ocorrências. Pela métrica NIM

percebemos o total de métodos criados a partir da refatoração, que ao todo foram 29 novos.

Em geral, todas as métricas tiveram aumento, com exceção da ACC que teve uma redução de 448 unidades. Apesar do aumento expressivo nas métricas de complexidade, a herança e o acoplamento foram as únicas métricas não afetadas pela refatoração.

Como resultado desta etapa, foram removidas 23 Features Envy, 1 Intensive Coupling, 2 Dispersed Coupling e 1 Brain Method, porém foi dado origem a 1 Intensive Coupling e 1 God Class. Partindo de 41 code smells para 16 ao todo.

Tabela 5 – Métricas do projeto após refatoração das Features Envy.

MEDIÇÃO 2 - REFATORAÇÃO 1 - FEATURE ENVY					
Atributo Interno de Qualidade	Métrica	Medição	Variação na Métrica	Total do Atributo	Variação no Atributo
Coesão	LCOM	966	32	966	32
Complexidade	ACC	72	-448	2.426	177
	SCC	2.032	589		
	EVG	87	5		
	Nesting	235	31		
Herança	DIT	25	0	54	0
	NOC	2	0		
	Base Classes	27	0		
Acoplamento	CBO	56	0	56	0
Tamanho	LOC	2.165	89	2.545	119
	CLOC	164	1		
	NIM	197	29		
	CDL	19	0		

Legenda:

	Métricas que melhoraram
	Métricas sem alterações
	Métricas que pioraram

2.4 Medição 3 – Após refatorar o Code Smell Intensive Coupling

A segunda etapa de refatoração consistiu em refatorar o code smell Intensive Coupling. Ele também foi refatorado até não ser mais detectado pela JSpirit e corresponde aos valores da Medição 3. As métricas da Tabela 5 são correspondentes a essa etapa.

As técnicas utilizadas para sanar as ocorrências desse code smell foram Extract Class e Extract Method. A partir deles foram gerados 23 novos métodos e 2 novas classes.




Não foram observadas melhorias nos atributos de qualidade após a refatoração. Com a piora em todos os atributos de qualidade, NOC e CLOC foram as únicas métricas inalteradas.

Como resultado desta etapa foram removidos 3 Intensive Coupling, 2 Dispersed Coupling e 1 Brain Method, porém foi dada origem a mais 1 dispersed coupling. Com isso, partimos de 16 para 11 code smells ao todo.

Tabela 6 – Métricas do projeto após refatoração dos Intensive Couplings.

MEDIÇÃO 3 - REFATORAÇÃO 2 - INTENSIVE COUPLING					
Atributo Interno de Qualidade	Métrica	Medição	Variação na Métrica	Total do Atributo	Variação no Atributo
Coesão	LCOM	1.143	177	1.143	177
Complexidade	ACC	76	4	2.549	123
	SCC	2.128	96		
	EVG	95	8		
	Nesting	250	15		
Herança	DIT	27	2	58	4
	NOC	2	0		
	Base Classes	29	2		
Acoplamento	CBO	63	7	63	7
Tamanho	LOC	2274	109	2679	134
	CLOC	164	0		
	NIM	220	23		
	CDL	21	2		

Legenda:

	Métricas que melhoraram
	Métricas sem alterações
	Métricas que pioraram

2.5 Medição 4 – Após refatorar o Code Smell Brain Method

Esta etapa foi iniciada com a tentativa de eliminar os code smells Dispersed Coupling, pois já havíamos resolvido alguns até aqui. Acontece que após remover duas ocorrências na classe Main, ela passou a se tornar uma Brain Class, um tipo de code smell que não havia sido identificado até então.

O surgimento deste novo code smell fez com que parássemos de resolver os dispersed coupling e voltássemos nossos esforços para ele e consequentemente para os dois Brain Methods que estavam na classe.

Para resolver o problema foi necessário extrair uma nova classe a partir da Main, optamos por separar alguns métodos que faziam CRUD e enviar para a nova classe, nomeada como MainCRUD.

Na nova classe utilizamos algumas vezes a técnica Extract Method para resolver os casos de Brain Method, assim solucionando o problema. Ao todo, aumentamos em um o número de classes e em quinze o número de métodos.

Em decorrência da refatoração foi observado uma queda significativa nas métricas de complexidade: ACC, SCC, EVG e Nesting, com destaque para SCC, somatório da complexidade ciclomática de todos os métodos, que reduziu 476 pontos. Ao todo a complexidade foi reduzida em 548 pontos.

As outras métricas, em geral, tiveram aumento, porém, não tão expressivas como nas refatorações anteriores. Como resultado da refatoração, foram removidos 3 Dispersed Coupling, 2 Brain Methods e a Brain Class que havia sido criada. Ao fim sobraram 6 code smells.

Tabela 7 – Métricas do projeto após refatoração dos Brain Methods.

MEDIÇÃO 4 - REFATORAÇÃO 3 - BRAIN METHOD

Atributo Interno de Qualidade	Métrica	Medição	Variação na métrica	Total do Atributo	Variação no Atributo
Coesão	LCOM	1.206	63	1.206	63
Complexidade	ACC	49	-27	2.001	-548
	SCC	1.652	-476		
	EVG	62	-33		
	Nesting	238	-12		
Herança	DIT	28	1	60	2
	NOC	2	0		
	Base Classes	30	1		
Acoplamento	CBO	67	4	67	4
Tamanho	LOC	2331	57	2752	73
	CLOC	164	0		
	NIM	235	15		
	CDL	22	1		

Legenda:

	Métricas que melhoraram
	Métricas sem alterações
	Métricas que pioraram

2.6 Medição 5 – Após refatorar o Code Smell Dispersed Coupling

Na quinta medição, optamos por eliminar os casos de dispersed coupling, pois haviam restado apenas 3 code smells. Optamos por utilizar Extract Method nos 3 casos, especializando partes dos métodos que poderiam ser divididas.

Aqui notamos um aumento expressivo na métrica SCC (Complexidade), acrescida em 560 pontos, além do aumento de todas as métricas, com exceção das métricas de herança que se mantiveram iguais. O que indica um possível caso onde seria melhor optar por não refatorar o code smell.

Como resultado desta etapa, saímos de 6 code smells para 3 restantes.

Tabela 8 – Métricas do projeto após refatoração dos Dispersed Couplings.

MEDIÇÃO 5 - REFATORAÇÃO 4 - DISPERSED COUPLING					
Atributo Interno de Qualidade	Métrica	Medição	Variação na Métrica	Total do Atributo	Variação no Atributo
Coesão	LCOM	1.214	8	1.214	8
Complexidade	ACC	77	28	2.650	649
	SCC	2.212	560		
	EVG	97	35		
	Nesting	264	26		
Herança	DIT	28	0	60	0
	NOC	2	0		
	Base Classes	30	0		
Acoplamento	CBO	68	1	68	1
Tamanho	LOC	2352	21	2779	27
	CLOC	164	0		
	NIM	241	6		
	CDL	22	0		

Legenda:

	Métricas que melhoraram
	Métricas sem alterações
	Métricas que pioraram

2.7 Medição Final – Após refatorar o Code Smell God Class

Por fim, optamos por refatorar as duas God Classes restantes. Utilizamos o método extract class para separar o atributos de dados básicos, como nome, id, status dos atributos mais complexos ou mais requisitados na classe principal.

Deste modo a partir da classe Flight foi criada a classe FlightData e da classe FlightDaemon foi gerada a classe FlightDaemonData. Como ambas as classes possuíam estrutura muito parecida, a solução de uma facilitou a solução das duas.

Após o Extract Class foram geradas duas novas Feature Envies em cada classe, que foram resolvidas através do método move method das classes principais

para as novas classes, necessitando apenas de ajustes pois os métodos precisavam de dados de ambas as classes em conjunto.

Conforme a tabela expõe, a extração das classes faz computar, além delas próprias, todos os seus métodos, aumentando em 36 o número de métodos. Em geral todas as métricas aumentaram, ainda que pouco, com exceção das métricas Nesting (Complexidade) , NOC (Herança) e CLOC (Tamanho) que se mantiveram iguais.

Tabela 9 – Métricas do projeto após refatoração das God Classes.

MEDIÇÃO 6 - REFATORAÇÃO 5 - GOD CLASS					
Atributo Interno de Qualidade	Métrica	Medição	Variação na Métrica	Total do Atributo	Variação no Atributo
Coesão	LCOM	1380	166	1380	166
Complexidade	ACC	81	4	2.802	152
	SCC	2356	144		
	EVG	101	4		
	Nesting	264	0		
Herança	DIT	30	2	64	4
	NOC	2	0		
	Base Classes	32	2		
Acoplamento	CBO	73	5	73	5
Tamanho	LOC	2469	117	2934	155
	CLOC	164	0		
	NIM	277	36		
	CDL	24	2		

Legenda:

	Métricas que melhoraram
	Métricas sem alterações
	Métricas que pioraram

3 COMPARAÇÃO DOS RESULTADOS

Após concluídas as refatorações e as respectivas métricas, foram geradas tabelas comparativas (tabela 10 e 11, abaixo) para analisar os resultados obtidos.

Foi observado a partir dos resultados que, em geral, todos os atributos de qualidade interna sofreram aumento considerável: Coesão aumentou em 446 pontos, Complexidade em 553 pontos, Herança em 10 pontos, Acoplamento em 17 pontos e Tamanho em 508 pontos.

A única melhora observável foi em relação à métrica de complexidade ACC (média da complexidade ciclomática de todos os métodos), quanto menor o valor dessa métrica, menos complexas são as classes e métodos. Esta melhoria se deu essencialmente pela refatoração de 23 Feature Envies iniciais do projeto.

Enquanto a única métrica que não apresentou mudança foi a métrica de herança NOC (número de subclasses de uma classe), visto que de fato não houve alterações de código para criação de subclasses, apesar de ter sido cogitado nos casos de God Classes.

Tabela 10 – Resumo comparativo das métricas iniciais e finais do projeto em decorrência das refatorações

MÉTRICAS INICIAIS E FINAIS							
Atributo Interno de Qualidade	Métrica	Medição Inicial	Medição Final	Variação na Métrica	Total do Atributo (Inicial)	Total do Atributo (Final)	Variação no Atributo
Coesão	LCOM	934	1.380	446	934	1.380	446
Complexidade	ACC	520	81	-439	2.249	2.802	553
	SCC	1.443	2.356	913			
	EVG	82	101	19			
	Nesting	204	264	60			
Herança	DIT	25	30	5	54	64	10
	NOC	2	2	0			
	Base Classes	27	32	5			
Acoplamento	CBO	56	73	17	56	73	17
Tamanho	LOC	2.076	2.469	393	2.426	2.934	508

	CLOC	163	164	1			
	NIM	168	277	109			
	CDL	19	24	5			

Legenda:

	Métricas que melhoraram
	Métricas sem alterações
	Métricas que pioraram

Tabela 11 – Visão geral das métricas por Code Smell

VISÃO GERAL DAS ALTERAÇÕES POR CODE SMELL					
Atributo Interno de Qualidade	Feature Envoy	Intensive Coupling	Brain Method	Dispersed Coupling	God Class
Coesão	LCOM	LCOM	LCOM	LCOM	LCOM
Complexidade	ACC	ACC	ACC	ACC	ACC
	SCC	SCC	SCC	SCC	SCC
	EVG	EVG	EVG	EVG	EVG
	Nesting	Nesting	Nesting	Nesting	Nesting
Herança	DIT	DIT	DIT	DIT	DIT
	NOC	NOC	NOC	NOC	NOC
	Base Classes	Base Classes	Base Classes	Base Classes	Base Classes
Acoplamento	CBO	CBO	CBO	CBO	CBO
Tamanho	LOC	LOC	LOC	LOC	LOC
	CLOC	CLOC	CLOC	CLOC	CLOC
	NIM	NIM	NIM	NIM	NIM
	CDL	CDL	CDL	CDL	CDL

REFERÊNCIAS

AZEEM, Muhammad. Machine learning techniques for code smell detection: A systematic literature review and meta-analysis. Information and Software Technology, v. 108, p. 115-138, 2019.

SABIR, Fatima. A systematic literature review on the detection of smells and their evolution in object-oriented and service-oriented systems. *Software: Practice and Experience*, v. 49, n. 1, p. 3-39, 2019.