



**UNIVERSIDADE
FEDERAL DO CEARÁ**
CAMPUS DE QUIXADÁ

CURSO DE ENGENHARIA DE SOFTWARE

**RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE
InfoSaúde**

Equipe:

**Lucas do Nascimento Diniz
João Teixeira do Nascimento**

Professora:

Carla Ilane Moreira Bezerra

QUIXADÁ

Março, 2021

SUMÁRIO

1	DESCRIÇÃO DO PROJETO	2
2	AValiação DO PROJETO	2
2.1	Medição 1 – Antes de refatorar o projeto	2
2.2	Detecção dos Code Smells	3
2.3	Medição 2 – Após Refatorar o Code Smell Brain Method	4
2.4	Medição 3 – Após Refatorar o Code Smell Shotgun Surgery	5
2.5	Medição 4 – Após a refatoração o Code Smell Feature Envy	5
2.6	Medição 5 – Após a refatoração o Code Smell Intensive Coupling	6
2.7	Medição 6 – Após a refatoração o Code Smell Dispersed Coupling	6
3	COMPARAÇÃO DOS RESULTADOS	7
	REFERÊNCIAS	8
	APÊNDICE A	8

1 DESCRIÇÃO DO PROJETO

O sistema que será abordado neste relatório é uma aplicação desenvolvida com finalidade de digitalizar uma carteira de vacinação. O sistema se chama InfoSaúde e foi desenvolvido na linguagem de programação Java, utilizando o Postgresql para persistir os dados. O sistema foi desenvolvido por um time composto de três programadores e utilizando o GitHub para fazer o controle de versões.

Link do projeto: <https://github.com/GustavoSantosCS/InfoSaude>

Tabela 1 – Características do Projeto

Projeto	LOC	# de classes	# de releases
InfoSaúde	6599	112	1

2 AVALIAÇÃO DO PROJETO

2.1 Medição 1 – Antes de refatorar o projeto

A seguir está descrita uma tabela que apresenta informações relacionadas com a medição das métricas de coesão, acoplamento, complexidade, herança e tamanho, antes do projeto ser refatorado. Para isso será utilizada a ferramenta Understand.

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

Sistema	Coesão	Complexidade				Herança			Acoplamento	Tamanho			
	LCOM2	ACC	SCC	EVG	MaxNet	DIT	NOC	IFANIN	CBO	LOC	CLOC	NIM	CDL
S1 antes da refatoração	2323	110	775	122	57	85	8	98	229	6599	531	381	112
S1 após refat. CS Brain Method	2328	105	783	125	57	85	8	98	229	6621	531	388	112
S1 após refat. CS Shotgun Surgery	2388	105	778	124	55	85	8	98	215	6611	531	389	112

S1 após refat. CS Feature-envy	2459	105	778	124	55	85	8	98	215	6785	531	441	112
S1 após refat. CS Intensive Coupling	2824	104	881	125	64	95	11	105	232	7081	536	496	119
S1 após refat. CS Dispersed Coupling	2795	105	891	126	64	101	14	105	230	7116	536	505	119

Tabela 3 – Métricas dos atributos internos de qualidade (MCCABE, 1976; CHIDAMBER; KEMERER, 1994; LORENZ; KIDD, 1994; DESTEFANIS *et al.*, 2014)

Atributos	Métricas	Descrição
Coesão	<i>Lack of Cohesion of Methods (LCOM2)</i> (CHIDAMBER; KEMERER, 1994)	Mede a coesão de uma classe. Quanto maior o valor dessa métrica, menos coesiva é a classe.
Acoplamento	<i>Coupling Between Objects (CBO)</i> (CHIDAMBER; KEMERER, 1994)	Número de classes que uma classe está acoplada Quanto maior o valor dessa métrica, maior é o acoplamento de classes e métodos.
Complexidade	<i>Average Cyclomatic Complexity (ACC)</i> (MCCABE, 1976)	Média da complexidade ciclomática de todos os métodos. Quanto maior o valor dessa métrica, mais complexa são a classes e métodos.
	<i>Sum Cyclomatic Complexity (SCC)</i> (MCCABE, 1976)	Somatório da complexidade ciclomática de todos os métodos. Quanto maior o valor dessa métrica, mais complexos são as classes e métodos.
	<i>Nesting (MaxNest)</i> (LORENZ; KIDD, 1994)	Nível máximo de aninhamento de construções de controle. Quanto maior o valor dessa métrica, maior é a complexidade de classes e métodos.
	<i>Essential Complexity (EVG)</i> (MCCABE, 1976)	Mede o grau na qual um módulo contém construtores não estruturados. Quanto maior o valor dessa métrica mais complexas são as classes e métodos.
Herança	<i>Number Of Children (NOC)</i> (CHIDAMBER; KEMERER, 1994)	Número de subclasses de uma classe. Quanto maior o valor dessa métrica maior é o grau de herança de um sistema.
	<i>Depth of Inheritance Tree (DIT)</i> (CHIDAMBER; KEMERER, 1994)	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.
	<i>Bases Classes (IFANIN)</i> (DESTEFANIS <i>et al.</i> , 2014)	Número imediato de classes base. Quanto maior o valor dessa métrica, maior o grau de herança de um sistema.
Tamanho	<i>Lines of Code (LOC)</i> (LORENZ; KIDD, 1994)	Número de linhas de código, excluindo espaços e comentários. Quanto maior o valor dessa métrica, maior é o tamanho do sistema.
	<i>Lines with Comments (CLOC)</i> (LORENZ; KIDD, 1994)	Número de linhas com comentários. Quanto maior o valor dessa métrica maior o tamanho do sistema.
	<i>Classes (CDL)</i> (LORENZ; KIDD, 1994)	Número de classes. Quanto maior o valor, maior o tamanho do sistema.
	<i>Instance Methods (NIM)</i> (LORENZ; KIDD, 1994)	Número de métodos de instância. Quanto maior o valor dessa métrica maior é o tamanho do sistema.

2.2 Detecção dos Code Smells

A ferramenta JSPiRIT detectou 5 tipos de code smells, todos esses tipos de code smells compõem um conjunto de 58 ocorrências de code smells.

Tabela 3 – Code smells do projeto.

Nome do Code Smell	Quantidade
Dispersed Coupling	21
Feature Envy	16
Intensive Coupling	13
Shotgun Surgery	5
Brain Method	3

2.3 Medição 2 – Após Refatorar Code Smell Brain Method

Nessa seção estão descritos os resultados da medição realizada após refatorar os code smells referentes ao tipo Brain Method. Os code smells foram identificados utilizando o JSpirit, esse tipo de code smell possui 3 ocorrências. Foi possível observar que o valor da LCOM2 aumentou, indicando que o sistema ficou menos coeso. A segunda medição revelou que os métodos ficaram mais complexos após retirar essas instâncias do Brain Method, apenas a métrica ACC indica o contrário. As métricas relacionadas com a herança permaneceram as mesmas da mesma forma que a métrica de acoplamento não foi alterada. Quando observamos o tamanho, podemos concluir que houve um acréscimo, apenas as métricas CLOC e CDL permaneceram as mesmas.

Tabela 4 – Code smells do projeto após refatorar o Brain Method.

Nome do Code Smell	Quantidade
Dispersed Coupling	25
Feature Envy	18
Intensive Coupling	11
Shotgun Surgery	5

2.4 Medição 3 – Após Refatorar Code Smell Shotgun Surgery

Nessa seção estão descritos os resultados da medição realizada após refatorar os code smells referentes ao tipo Shotgun Surgery. Os code smells foram identificados utilizando o JSpirt, esse tipo de code smell possui 5 ocorrências. Foi possível observar que o valor da LCOM2 aumentou, indicando que o sistema ficou menos coeso. A segunda medição revelou que os métodos ficaram menos complexos após retirar essas instâncias do Shotgun Surgery. As métricas relacionadas com a herança permaneceram as mesmas, já a métrica de acoplamento indicou que o sistema ficou menos acoplado. Quando observamos o tamanho, podemos entender que houve um acréscimo na métrica de NIM e LOC, apenas as métricas CLOC e CDL permaneceram as mesmas.

Tabela 5 – Code smells do projeto após refatorar o Shotgun Surgery.

Nome do Code Smell	Quantidade
Dispersed Coupling	17
Feature Envy	22
Intensive Coupling	18

2.5 Medição 4 – Após Refatorar Code Smell Feature envy

Nessa seção estão descritos os resultados da medição realizada após refatorar os code smells referentes ao tipo Feature Envy. Os code smells foram identificados utilizando o JSpirt, esse tipo de code smell possui 22 ocorrências. Foi possível observar que o valor da LCOM2 aumentou, indicando que o sistema ficou menos coeso. As métricas de complexidade, herança e acoplamento seguiram com os mesmos valores. Quando observamos o tamanho, podemos ver que houve um acréscimo nas métricas de NIM e LOC, apenas as métricas CLOC e CDL permaneceram as mesmas.

Tabela 6 – Code smells do projeto após refatorar o Feature Envy.

Nome do Code Smell	Quantidade
Dispersed Coupling	17

Intensive Coupling	18
God Class	1

2.6 Medição 5 – Após Refatorar Code Smell Intensive Coupling

Nessa seção estão descritos os resultados da medição realizada após refatorar os code smells referentes ao tipo Intensive Coupling. Os code smells foram identificados utilizando o JSpirt, esse tipo de code smell possui 18 ocorrências. Foi possível observar que o valor da LCOM2 aumentou, indicando que o sistema ficou menos coeso. As métricas de complexidade aumentaram com exceção da métrica ACC. A herança e o acoplamento também aumentaram. Quando olhamos para o tamanho, podemos observar que houve um acréscimo em todas as métricas.

Tabela 7 – Code smells do projeto após refatorar o Intensive Coupling.

Nome do Code Smell	Quantidade
Dispersed Coupling	27
God Class	1

2.7 Medição 6 – Após Refatorar Code Smell Dispersed Coupling

Nesta seção estão descritos os resultados da medição realizada após refatorar os code smells referentes ao tipo Dispersed Coupling. Os code smells foram identificados utilizando o JSpirt, esse tipo de code smell possui 27 ocorrências. Foi possível observar que o valor da LCOM2 diminuiu, indicando que o sistema ficou mais coeso. As métricas de complexidade aumentaram os seus valores. As métricas de herança aumentaram com exceção da IFANIN. Já o acoplamento diminuiu. Quando observamos o tamanho, podemos identificar que houve um acréscimo na métrica de NIM e na métrica de LOC, apenas as métricas CLOC e CDL permaneceram as mesmas.

Tabela 8 – Code smells do projeto após refatorar o Dispersed Coupling.

Nome do Code Smell	Quantidade
Dispersed Coupling	16
God Class	1

3 COMPARAÇÃO DOS RESULTADO

Abordando a refatoração dos code smells durante a realização desse experimento, ficou evidente que a refatoração pode impactar os atributos de qualidade de duas formas distintas: aumentando ou diminuindo as métricas. Por outro lado, podemos observar três cenários diferentes quanto aos atributos internos de qualidade, após a refatoração de determinado tipo de code smell: (1) (↑) determinada métrica aumenta; (2) (↓) determinada métrica diminui; (3) (-) determinada métrica permanece inalterada. Essa relação pode ser vista na tabela 9.

Analizando as informações apresentadas na tabela 9, com base nos tipos de code smells, podemos observar que apenas a refatoração do shotgun surgery contemplou os três cenários possíveis. Em contraste, apenas a refatoração do code smell intensive coupling contemplou um cenário.

A proporção de ocorrência desses três cenários tendo como base cada atributo de qualidade e code smell refatorado seguiu da seguinte forma: A métrica de coesão aumentou em 80% das sprints de refatoração e diminuiu em 20% dos casos; A métrica de complexidade aumentou em 60% dos casos e permaneceu a mesma em 20% dos casos; A métrica de herança permaneceu a mesma em 60% dos casos e aumentou nos outros 40%; O acoplamento diminuiu em 40% dos casos e permaneceu a mesma em 40%; O tamanho aumentou em 80% dos casos e diminuiu em 20% dos casos.

Tabela 9 – Relação entre alterações das métricas e code smells detectados.

	Coesão	Complexidade	Herança	Acoplamento	Tamanho
Brain Method	↑	↑	-	-	↑
Shotgun Surgery	↑	↓	-	↓	↓
Feature Envy	↑	-	-	-	↑
Intensive Coupling	↑	↑	↑	↑	↑

Dispersed Coupling	↓	↑	↑	↓	↑
--------------------	---	---	---	---	---

Fazendo uma média geral de ocorrência desses três cenários, sem fazer distinção entre os code smells e métricas de qualidade, temos que as métricas permaneceram as mesmas em 24% de situações, as métricas aumentaram em 56% das situações e as métricas diminuíram em 20% das situações.

Baseado nesses dados, as métricas de qualidade no geral aumentaram em comparação com o sistema não refatorado, os atributos de qualidade como coesão, complexidade, herança, acoplamento e tamanho sofreram um aumento de 20%, 11.4%, 15%, 0.4%, 8.5%, respectivamente. Com isso, fica evidente que a refatoração desempenhada nesse sistema não foi algo positivo quando se leva em consideração os atributos de qualidade. Por outro lado, a refatoração desempenhada evidencia um impacto positivo para eliminar ocorrências de code smells de determinado sistema já que foi possível observar uma diminuição de 70,7% das ocorrências dos code smells do sistema abordado.

4 DIFICULDADES ENCONTRADAS NAS REFATORAÇÕES

Durante as refatorações foram notadas algumas dificuldades dependendo do tipo de code smell que está sendo refatorado. Nas refatorações de Shotgun Surgery, as ocorrências referenciavam funções de bibliotecas que impossibilitava algumas das estratégias para o tipo de code smell, outro problema surgiu quando em uma das medições parciais do JSpirit apareceu uma ocorrência falsa que logo foi removida modificando o método de refatoração.

As ocorrências de intensive coupling e dispersed coupling se mostraram as mais difíceis de refatorar, surgindo muitas ocorrências com o andamento do projeto, cada refatoração individual demorou mais que o dobro das refatorações dos outros tipos de code smells, foi usado um [site](#) como referência para refatorar esses dois tipos de code smells, nele tem dicas, já que não há método oficial, de como refatorar essas ocorrências.

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.

APÊNDICE A

<https://github.com/LNascimentoD/InfoSaude> - Repositório que possui todas as refatorações realizadas nesse experimento.

<https://docs.google.com/spreadsheets/d/1sY3UJhzRIx8CH8ge2Bb9BCVvuf1a6a7B/edit#gid=39146266> - Planilha que possui as medições realizadas durante o experimento.

<https://docs.embold.io/anti-patterns/> - Site com dicas de como refatorar os code smells.