



**UNIVERSIDADE  
FEDERAL DO CEARÁ**  
CAMPUS DE QUIXADÁ

**CURSO DE SISTEMAS DE INFORMAÇÃO**

**RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE  
Medical Clinic**

**Equipe:**

**Letícia França Farias - 415097**

**Érica Miranda de Sousa - 418376**

**Professora:**

**Carla Ilane Moreira Bezerra**

**QUIXADÁ**

**Julho, 2021**

## **SUMÁRIO**

<b>DESCRIÇÃO DO PROJETO</b>	<b>2</b>
<b>AVALIAÇÃO DO PROJETO</b>	<b>2</b>
Medição 1 – Antes de refatorar o projeto	2
Detecção dos Code Smells	3
Medição 2 – Após Refatorar Code Smell Feature Envy	4
Medição 3 – Após Refatorar Code Smell Long Method	5
Medição 4 – Após a refatoração de todos os code smells do projeto	5
<b>COMPARAÇÃO DOS RESULTADOS</b>	<b>5</b>
<b>REFERÊNCIAS</b>	<b>6</b>

## 1 DESCRIÇÃO DO PROJETO

Medical Clinic(MC) é um projeto Java de código aberto disponibilizado no GitHub. Trata-se de um sistema simples de gerenciamento de Clínicas Médicas, onde é possível cadastrar médicos e pacientes, além de agendar consultas e exames. Medical Clinic foi construído utilizando o banco de dados MySQL e JDBC para persistência de dados, telas criadas utilizando o Java Swing. O mesmo ainda consta como módulo de testes unitários utilizando JUnit 5.

- Link do repositório original:
  - <https://github.com/Eddusr/ProjetoEngSoft3>
- Link do repositório da Disciplina:
  - <https://github.com/EriccaSousa/QualidadeDeSoftware-TrabalhoFinal>

Tabela 1 – Características do Projeto

Projeto	LOC	Nº de classes	Nº de releases
Medical Clinic (MC)	4 923	979	1 387

## 2 AVALIAÇÃO DO PROJETO

### 2.1 Medição 1 – Antes de refatorar o projeto

A Tabela 2 apresenta os números totais da análise dos atributos de qualidade do projeto Medical Clinic, realizado na ferramenta Understand. Os dados da análise podem ser encontrados com mais detalhes no seguinte link: <https://drive.google.com/file/d/10OrXomzGnHKHtealSFnEJdqknPJc1dN/view?usp=sharing>

O Anexo 1 apresenta as métricas e definição de cada atributo interno de qualidade analisado

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

Antes das Refatorações			
Atributo Interno de Qualidade	Métrica	Valor da Métrica	Total do Atributo

<b>Coesão</b>	<i>LCOM</i>	1 745	1 745
<b>Complexidade</b>	<i>ACC</i>	46	768
	<i>SCC</i>	587	
	<i>EVG</i>	99	
	<i>Nesting</i>	36	
<b>Herança</b>	<i>DIT</i>	49	88
	<i>NOC</i>	1	
	<i>Base Classes</i>	38	
<b>Acoplamento</b>	<i>CBO</i>	95	95
<b>Tamanho</b>	<i>LOC</i>	4923	7441
	<i>CLOC</i>	979	
	<i>NIM</i>	1387	
	<i>CDL</i>	152	

## 2.2 Detecção dos Code Smells

Para detecção de *code smells* no projeto Medical Clinic, foram utilizadas as ferramentas *JSpirit* e *JDeodorant*. Como resultado desta análise, foram reportados um total de 45 *code smells*, divididos em três tipos: Feature Envy, Long Method e Dispersed Coupling. A Tabela 3 apresenta uma descrição detalhada da distribuição dos *code smells* identificados.

Tabela 3 – Code smells do projeto.

Nome do Code Smell	Quantidade
Feature Envy	22
Long Method	10
Dispersed Coupling	13
Total	45

### 2.3 Medição 2 – Após Refatorar Code Smell Feature Envy

Antes da refatoração, o projeto possuía 22 ocorrências de Feature Envy, as quais foram completamente refatoradas nessa etapa. Para realizar as refatorações foram utilizadas as técnicas Extract Method e Move Method, onde:

**Extract Method**, a equipe moveu trechos de código para um novo método, substituindo o código antigo por uma chamada para o novo método.

**Move Method**, onde a equipe buscou identificar métodos que usavam excessivamente recursos de outras classes, para assim mover essas operações para a classe mais relacionada, tornando-as mais coerentes internamente.

A tabela 4 apresenta o impacto dessa refatoração nos Atributos Internos de Qualidade. Consegue-se perceber que **Coesão e Herança aumentaram**, **Complexidade e Tamanho diminuíram** e o **Acoplamento se manteve**.

Tabela 4 – Atributos de Qualidade após refatoração de Feature Envy.

Após refatorar Feature Envy			
Atributo Interno de Qualidade	Métrica	Valor da Métrica	Total do Atributo
<b>Coesão</b>	<i>LCOM</i>	1 787	1 787
<b>Complexidade</b>	<i>ACC</i>	46	683
	<i>SCC</i>	494	
	<i>EVG</i>	99	
	<i>Nesting</i>	44	
<b>Herança</b>	<i>DIT</i>	49	137
	<i>NOC</i>	50	
	<i>Base Classes</i>	38	
<b>Acoplamento</b>	<i>CBO</i>	95	95
<b>Tamanho</b>	<i>LOC</i>	4996	6583
	<i>CLOC</i>	977	
	<i>NIM</i>	458	
	<i>CDL</i>	152	

	Aumentou
	Igual
	Diminuiu

## 2.4 Medição 3 – Após Refatorar Long Method

Antes da refatoração, o projeto possuía 10 ocorrências de Long Method, contidos na camada DAO do projeto, os quais foram completamente refatorados nessa etapa. Para realizar as refatorações foi utilizado a técnica Extract Method, onde a equipe moveu trechos de código para um novo método, substituindo o código antigo por uma chamada para o novo método.

A tabela 5 apresenta o impacto dessa refatoração nos Atributos Internos de Qualidade. Consegue-se perceber que **Coesão, Complexidade e Tamanho aumentaram**, **Herança e Acoplamento se mantiveram**.

Tabela 5 – Atributos de Qualidade após refatoração de Long Method.

Após refatorar Long Method			
Atributo Interno de Qualidade	Métrica	Valor da Métrica	Total do Atributo
<b>Coesão</b>	<i>LCOM</i>	1 876	1 876
<b>Complexidade</b>	<i>ACC</i>	45	692
	<i>SCC</i>	504	
	<i>EVG</i>	99	
	<i>Nesting</i>	44	
<b>Herança</b>	<i>DIT</i>	49	137
	<i>NOC</i>	50	
	<i>Base Classes</i>	38	
<b>Acoplamento</b>	<i>CBO</i>	95	95
<b>Tamanho</b>	<i>LOC</i>	5034	6631
	<i>CLOC</i>	977	
	<i>NIM</i>	468	
	<i>CDL</i>	152	

	Aumentou
	Igual
	Diminuiu

## 2.5 Medição 4 – Após a refatoração de todos os code smells do projeto

Após todos os code smells refatorados, deverá ser realizada a medição final do projeto conforme as métricas da Tabela 2. Deve também ser feita a análise final se as métricas pioram ou melhoram de acordo com a retirada dos code smells.

Tabela 6 – Comparação entre medição inicial (antes das refatorações) e medição final (após última refatoração)

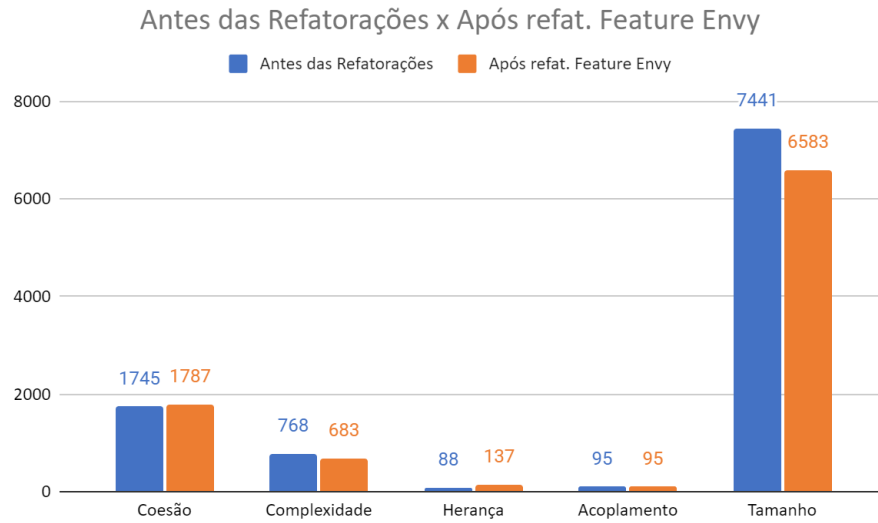
Resultado Final			
Atributo Interno de Qualidade	Métrica	Valor da Métrica	Total do Atributo
<b>Coesão</b>	<i>LCOM</i>	1 876	1 876
<b>Complexidade</b>	<i>ACC</i>	45	692
	<i>SCC</i>	504	
	<i>EVG</i>	99	
	<i>Nesting</i>	44	
<b>Herança</b>	<i>DIT</i>	49	137
	<i>NOC</i>	50	
	<i>Base Classes</i>	38	
<b>Acoplamento</b>	<i>CBO</i>	95	95
<b>Tamanho</b>	<i>LOC</i>	5034	6631
	<i>CLOC</i>	977	
	<i>NIM</i>	468	
	<i>CDL</i>	152	

	Aumentou
	Igual
	Diminuiu

## 3 COMPARAÇÃO DOS RESULTADOS

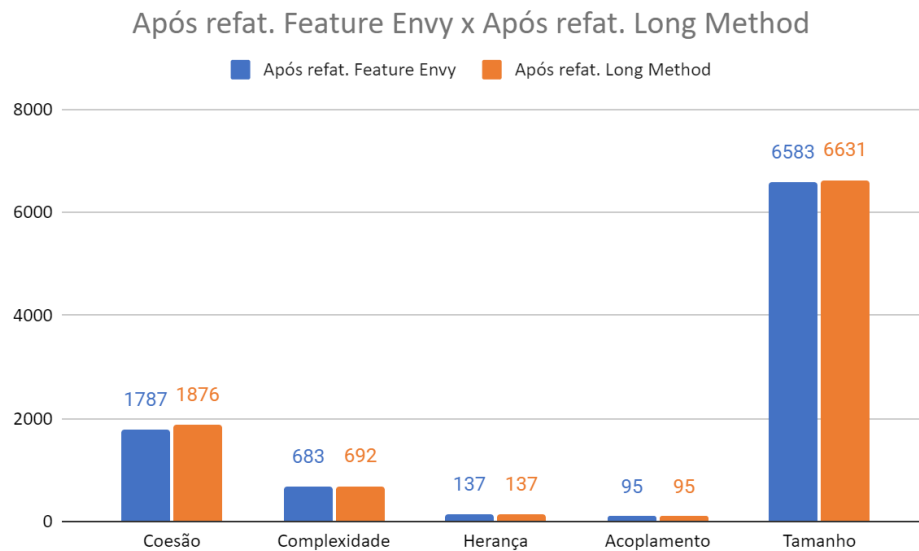
Analisando os dados obtidos, foi possível observar que após refatorar Feature Envy as métricas de Coesão e Herança apresentaram aumento de 2,4% e 55,68% respectivamente, Complexidade apresentou queda de 11,06%, Tamanho apresentou queda de 11,53%. Já o Acoplamento não apresentou mudanças, conforme mostrado na Figura 1.

Figura 1



Comparando os resultados da refatoração de Feature Envoy com a refatoração de Long Method foi possível identificar um aumento nas métricas de Coesão, Complexidade e Tamanho - 4,9%, 1,31% e 0,72%, respectivamente -, Acoplamento e Herança não apresentaram alterações, conforme mostrado na Figura 2.

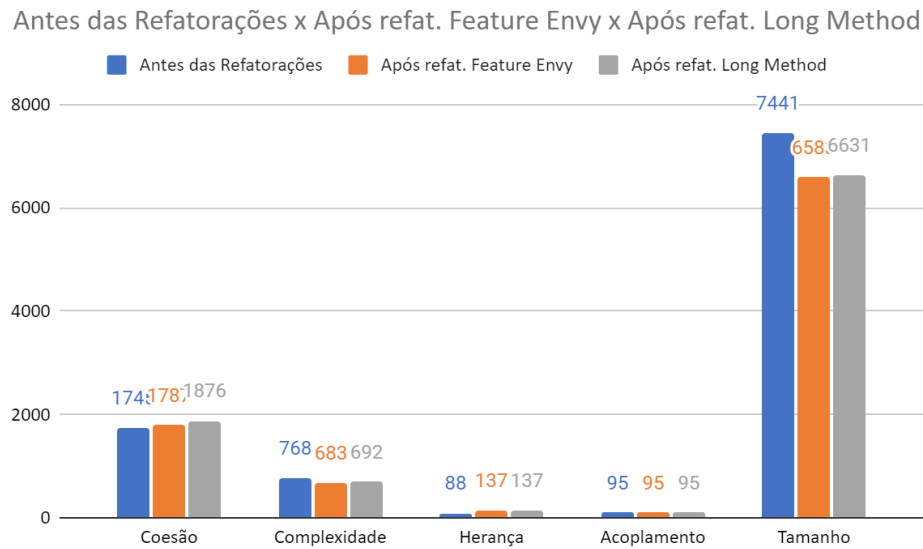
Figura 2



Realizando uma comparação geral, entre as 3 medições, podemos notar que Coesão apresentou aumento de 7,5%, Herança apresentou aumento de 55,6%, Complexidade apresentou queda de 9,9%, Tamanho apresentou queda de 10,8% e Acoplamento se manteve constante durante todo o processo, conforme ilustrado na Figura 3.



Figura 3



O resultado da medição realizada após a remoção pode ser visto na Tabela 7. Onde é possível perceber que houve um aumento em todos os atributos de qualidade, mas isso não configura necessariamente uma piora da qualidade do projeto, pois em contrapartida também houveram melhorias em alguns atributos. Ocorreu aumento na LCOM, Nesting, NOC e LOC. Já as métricas ACC, SCC, CLOC e NIM apresentaram melhoria, Base Classes, CBO e CDL não sofreram alterações durante o processo.

Tabela 6 – Comparação Geral entre as métricas coletadas

	Feature Envy	Long Method	Mudança Final
<b>Coesão</b>	<i>LCOM</i>	<i>LCOM</i>	131
<b>Complexidade</b>	<i>ACC</i>	<i>ACC</i>	1
	<i>SCC</i>	<i>SCC</i>	83
	<i>EVG</i>	<i>EVG</i>	99
	<i>Nesting</i>	<i>Nesting</i>	8
<b>Herança</b>	<i>DIT</i>	<i>DIT</i>	49
	<i>NOC</i>	<i>NOC</i>	49
	<i>Base Classes</i>	<i>Base Classes</i>	38
<b>Acoplamento</b>	<i>CBO</i>	<i>CBO</i>	98
<b>Tamanho</b>	<i>LOC</i>	<i>LOC</i>	111
	<i>CLOC</i>	<i>CLOC</i>	2
	<i>NIM</i>	<i>NIM</i>	919
	<i>CDL</i>	<i>CDL</i>	152

As mudanças vistas confirmam os resultados do artigo Refactoring Effect on Internal Quality Attributes: What Haven't They Told You Yet?, onde vimos que Move Method melhora a complexidade geral do código, assim como há uma melhora em Acoplamento quando Extract Method não apresenta melhorias.

## 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.

## ANEXO 1

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.