



RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE SisCom

Equipe:

Francisco Thierry Barros Oliveira

Davi Teixeira

Professora:

Carla Ilane Moreira Bezerra

QUIXADÁ

Agosto, 2021

SUMÁRIO

1	DESCRIÇÃO DO PROJETO	2
2	AVALIAÇÃO DO PROJETO	4
2.1	Medição 1 – Antes de refatorar o projeto	4
2.2	Detecção dos Code Smells	5
2.3	Medição 2 – Após Refatorar Code Smell Feature Envy	6

1 DESCRIÇÃO DO PROJETO

O SisCom é um sistema comercial de compra e venda, desenvolvido em Java OO para desktop, em que basicamente temos subdividido os clientes, fornecedores, vendedores e produtos, onde cada um possui um crud básico de cadastro, listagem e remoção em tempo de execução, assim o administrador do sistema pode cadastrar um produto, após isso o fornecedor pode adquirir esse produto, repassando-o ao vendedor, e por fim o vendedor realizar a venda do determinado produto a um usuário cadastrado no sistema. É um projeto aberto o qual foi desenvolvido por um estudante de Belo Horizonte, mas ainda se encontra incompleto, e seu código fonte está disponível no link abaixo.

Link do projeto: <https://github.com/TierryBr/Qualidade-SmellsAndRefactoring>

Assim o projeto tem algumas características e informações como:

Cadastro:

Nome; Telefone; E-mail; CNPJ; Limite de Crédito.

Consulta:

Pesquisa pelo CPF; Funcionalidade de deletar um cliente; Funcionalidade de listar todos.

Na aba de Fornecedor, temos quatro funcionalidades, o cadastro de fornecedor, a consulta de fornecedor, uma aba de compras e uma aba de estatísticas, assim cada uma delas tem as seguintes informações a serem cadastradas:

Cadastro:

Nome; Telefone; E-mail; CNPJ; Contato

Consulta:

Pesquisa pelo CNPJ; Funcionalidade de deletar um cliente; Funcionalidade de listar todos.

Compra:

Aqui temos a funcionalidade de pesquisar um produto pelo cnpj do fornecedor, exibindo assim uma tabela com todos os produtos cadastrados naquele cnpj; Temos a opção de pesquisar pelo nome do produto; Uma área que mostra a lista de compras daquele fornecedor.

Na aba de Vendedor, temos quatro funcionalidades, o cadastro de vendedor, a consulta de vendedor, uma aba de vendas e uma aba de estatísticas de vendas, assim cada uma delas tem as seguintes informações a serem cadastradas:

Cadastro:

Nome; Telefone; E-mail; CPF; Meta Mensal

Consulta:

Pesquisa pelo CPF; Funcionalidade de deletar um vendedor; Funcionalidade de listar todos.

Venda:

Aqui temos a funcionalidade de pesquisar um vendedor pelo cpf, exibindo assim suas informações; Temos a opção de pesquisar pelo nome do produto; Uma área que mostra a lista de vendas daquele vendedor.

Por fim, na aba de Produtos, temos duas funcionalidades, o cadastro de produtos e a consulta de produtos, tendo as seguintes informações a serem cadastradas:

Cadastro:

Nome; Preço unitário; Estoque; Estoque Mínimo

Consulta:

Pesquisa pelo nome do produto; Funcionalidade de listar todos, tanto por ordem alfabética ou pelo preço mínimo.

Também temos algo que é padrão para todos, que é a:

Estatística de Compras:

Uma funcionalidade que mostra a lista de compras de todos os clientes, por um período estabelecido; Uma funcionalidade que mostra todas as estatísticas por um período fornecido.

A seguir, está uma tabela que mostra algumas propriedades do projeto:

Tabela 1 – Características do Projeto

Projeto	LOC	# de classes	# de releases
SisCom	5676	93	1

2 AVALIAÇÃO DO PROJETO

2.1 Medição 1 – Antes de refatorar o projeto

A tabela a seguir (Tabela 2) mostra a medição dos atributos de qualidade na ferramenta Understand antes de refatorar o projeto, o qual destaca a coesão, complexidade, herança, acoplamento e tamanho em cada classe diferente do código, e por fim mostra o somatório dessas métricas, como pode ser observada mais em detalhes na tabela 3.

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

Pacotes/classe	COESÃO	COMPLEXIDADE				Herança			ACOPLAMENTO	TAMANHO			
	LCOM	ACC	SCC	EVG	Nesting	DIT	NOC	Base Cla	CBO	LOC	CLOC	NIM	CDL
"controller"	89	2	87	11	4	1	0	2	12	5676	1123	358	93
Comercial													
"error"													
SisComException	75	1	8	1	0	2	0	1	0				
"model"													
Cliente	70	1	11	1	1	2	0	2	2				
Compra	72	1	11	1	0	1	0	2	2				
Estatística	71	1	12	1	0	1	0	2	0				
Fornecedor	70	1	11	1	1	2	0	2	2				
ItemCompra	62	1	8	1	0	1	0	2	1				
ItemVenda	66	1	9	1	0	1	0	2	1				
Pessoa	75	1	14	1	0	1	3	3	2				
Produto	77	1	17	1	0	1	0	3	0				
Venda	78	1	15	1	0	1	0	2	4				
Vendedor	73	1	13	1	1	2	0	2	3				
"utilitarios"													
Console	0	2	13	3	3	1	0	1	0				
CustomTransferHandler	100	3	16	5	3	2	0	1	3				
InterfaceUtil	0	1	7	1	0	1	0	1	0				
LtpLib	99	3	145	11	4	1	0	1	1				
"view"													
Colors	0	0	0	0	0	1	0	1	0				
InterfaceBase	75	1	3	1	1	2	0	1	6				
InterfaceComercial	76	3	94	1	2	1	0	1	12				
InternalFrameVendaCliente	58	3	10	1	4	2	0	1	11				
PanelClienteCenter	81	3	15	1	2	2	0	1	7				
PanelFornecedorCenter	89	3	33	1	5	2	0	1	11				
PanelProdutosCenter	56	2	4	1	1	2	0	1	5				
PanelVendedorCenter	88	3	26	1	2	2	0	1	9				
Total	1600	40	582	49	34	35	3	37	94				
Total All Metrics	1600		705				75						

Tabela 3 – Resultado da medição antes das refatorações.

Atributos Internos de Qualidade	Métricas	Valor da Métrica	Total do Atributo
Coesão	LCOM	1600	1600
Complexidade	ACC	40	705
	SCC	582	

	EVG	49	
	Nesting	34	
Herança	DIT	35	75
	NOC	3	
	Base Classes	37	
Acoplamento	CBO	94	94
Tamanho	LOC	5676	7250
	CLOC	1123	
	NIM	358	
	CDL	93	

2.2 Detecção dos Code Smells

Ao todo foram encontrados 55 Code Smells o qual foram detectados pela ferramenta JSPirit. Como vamos refatorar 40 smells, descartamos o Brain Method, Brain Class e Data Class, restando 5 tipos para se trabalhar e dentro da margem dos 40 smells. A tabela 4 abaixo mostra o total de ocorrências para cada tipo de smells detectados no código.

Tabela 4 – Code smells do projeto.

Nome do Code Smell	Quantidade
Feature Envy	24
Dispersed Coupling	13
Shotgun Surgery	7
Intensive Coupling	5
God Class	3
Brain Method	1

Brain Class	1
Data Class	1

2.3 Medição 2 – Após Refatorar Code Smell Feature Envy

Como visto, o nosso código possui em um maior número o Feature Envy, então dos 24 encontrados, foram refatorados 20 deles, assim usamos a técnica do Move Method e Extract Method para aplicar a refatoração, já que são as soluções mais adequadas para esse tipo de smell, que consiste em basicamente mover o método para outra classe que faça sentido, ou fazer um split no método até que ele seja excluído totalmente durante a detecção do JSPirit. Logo após concluir essas 20 refatorações, medimos o projeto novamente na ferramenta Understand e chegamos aos resultados mostrados na tabela 5.

Como observado na tabela 5, é possível perceber que houve um aumento em quase todos os atributos de qualidade, embora isso não signifique uma piora na qualidade geral do projeto. Como notado, tivemos aumento em todas as métricas que envolvem Coesão e Complexidade, como também o Acoplamento, isso se deve ao fato de que usamos o Move Method e Extract Method, que foram os métodos de refatorações mais recomendados para esse caso, em basicamente movemos alguns métodos e outros separamos em dois para uma melhor compreensão, enquanto isso, a Herança permaneceu estável. Por outro lado, podemos notar uma certa melhora no Tamanho, algo que chega a ser um pouco surpreendente, já que teve melhora em basicamente todas as métricas, acreditamos que tenha sido devido a uma melhor distribuição do código ao refatora-lo.

Tabela 5 – Resultado da medição antes das refatorações.

Atributos Internos de Qualidade	Métricas	Valor da Métrica	Total do Atributo
Coesão	LCOM	1640	1640
Complexidade	ACC	41	714
	SCC	583	
	EVG	53	

	Nesting	37	
Herança	DIT	35	75
	NOC	3	
	Base Classes	37	
Acoplamento	CBO	100	100
Tamanho	LOC	5663	7229
	CLOC	1123	
	NIM	352	
	CDL	91	

Legenda: maior, igual, menor