



RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE

Sistema de Restaurante para Clientes

Equipe:

Maxwell Félix dos Santos

Professora:

Carla Ilane Moreira Bezerra

QUIXADÁ

Agosto, 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.....	5
2.3	Medição 2 - Após Refatorar CS. Featury Envy	6
2.4	Medição 3 - Após Refatorar CS. God Class	6
2.5	Medição 4 - Após Refatorar CS. Dispersed Coupling.....	7
2.6	Medição 5 - Após Refatorar CS. Refused Parent Bequest.....	7
2.7	Medição 6 - Após Refatorar CS. Shotgun Surgery.....	8
2.8	Medição 7 - Após Refatorar CS. Long Method.....	8
3	COMPARAÇÃO DOS RESULTADOS.....	10
	REFERÊNCIAS.....	11

1 DESCRIÇÃO DO PROJETO

O projeto se trata de um sistema de restaurante para clientes, que possui como principal objetivo fornecer o atendimento completo ao cliente, como o cardápio, fornecendo informações sobre os tipos de alimentos disponíveis em uma lista, delivery do produto requisitado pelo cliente, a reserva de uma mesa, entre outros aspectos..

Para o desenvolvimento do sistema foram utilizadas as tecnologias Spring MVC, Java Server Pages (JSP), JSPA e Hibernate.

Link do projeto: <https://github.com/jonatasjp/projeto-restaurante>

Tabela 1 – Características do Projeto

Projeto	LOC	# de classes	# de releases
Sistema Restaurante	3686	68	60

2. AVALIAÇÃO DO PROJETO

2.1 Medição 1 – Antes de refatorar o projeto

Sistema	Coesão	Complexidade				Herança			Acoplament o	Tamanho			
proj.unipe.controllers	LCOM2	ACC	SCC	EVG	MaxNet	DIT	NOC	IFANIN	CBO	LOC	CL ₃ OC	NIM	C
AutorizadorInterceptor	0	2	1	1	3	4	3	2	3	26	5	6	3
JPAUtil	0	0	1	1	1	5	2	6	0	9	2	3	4
CardapioController	60	3	2	1	1	7	8	2	13	12	4	7	8
CategoriaController	22	2	5	6	1	6	4	2	9	63	3	4	6
ClienteController	25	0	1	2	3	8	6	5	10	9	8	9	4
DeliveryController	80	4	5	2	1	4	3	6	11	4	5	8	9
FuncionarioController	16	3	2	1	4	9	8	7	9	131	3	4	5
LayoutController	0	2	3	1	4	8	9	4	1	3	11	2	3
LoginController	66	6	7	8	1	8	7	2	3	144	2	3	7
MesaController	28	1	3	2	1	3	4	5	11	9	8	7	2
PerdidoController	78	4	4	3	2	7	0	2	18	7	1	2	3
ReservaController	57	2	3	4	5	8	9	2	12	75	2	4	8
AbstractDAO	25	7	2	1	3	4	5	2	4	323	5	5	5
AbstractDAO.T	0	5	3	9	1	6	2	1	0	2	3	4	2
CardapioDAO	100	3	4	2	1	7	8	2	5	85	2	1	2
CategoriaDAO	83	2	4	1	3	8	4	1	4	14	3	4	1
ClienteDAO	100	5	6	7	8	6	4	7	4	27	6	4	7
DeliveryDAO	100	1	2	4	5	6	5	2	2	13	1	2	0
FuncionarioDAO	75	4	3	7	1	0	4	5	4	63	4	5	7
GerenteDAO	100	1	2	4	1	8	5	2	2	8	6	2	1
ItemCardapioDAO	100	1	6	7	1	9	7	1	5	45	3	4	2
MesaDAO	66	8	1	2	3	9	2	1	3	44	5	2	1
PedidoDAO	90	1	3	4	5	8	7	4	5	2	3	4	6
ReservaDAO	83	3	1	4	5	2	3	4	8	26	9	8	2
TradicionalDAO	100	7	1	4	5	6	4	2	2	74	4	8	6
UsuarioDAO	100	8	2	3	4	8	2	3	4	5	2	4	6
AbstractEntity	0	7	8	4	1	9	4	5	2	107	1	2	3
CardapioDAO	83	1	2	4	7	8	4	2	4	3	2	1	2
CategoriaDAO	75	6	2	3	4	10	11	2	3	9	5	6	7
ClienteDAO	50	5	4	2	3	9	4	5	2	3	4	2	6
DeliveryDAO	50	7	2	3	4	8	9	4	1	78	3	4	4
EndereçoDAO	80	8	6	4	5	8	6	6	1	7	7	8	1
FuncionarioDAO	80	4	0	0	3	8	2	1	3	172	2	1	2
GerenteDAO	0	2	4	3	2	9	4	7	0	9	8	7	1

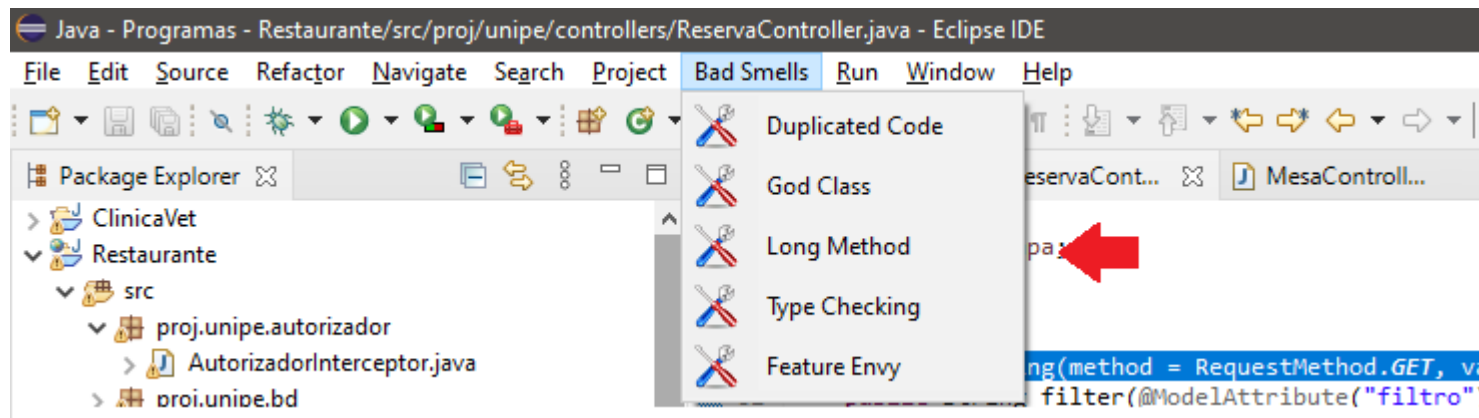
[illegible]

2.2 Detecção dos Code Smells

Tabela 1 – Code smells do projeto.

Nome do Code Smell	Quantidade
Feature Envoy	11
Dispersed Coupling	3
Refused Parent Bequest	2
Shotgun Surgery	1
God Class	9
Long Method	12

OBS: Foram encontrados mais 21 code smells, de 2 tipos diferentes, por meio da ferramenta JDeodorant, totalizando 38 code smells.



1. Ferramenta “JDeodorant” com CS adicionais

Kind of Design Flaw	Java Element	#Ranking	Ranking Value
Shotgun Surgery	AbstractEntity.temIdValido	1	1.0
Dispersed Coupling	DeliveryController.finalizarPedido	2	1.0
Feature Envoy	PedidoDAO.buscarPedido	3	1.0
Feature Envoy	FuncionarioController.filtrar	4	1.0
Feature Envoy	CardapioController.filtrar	5	1.0
Feature Envoy	CardapioController.save	6	1.0
Dispersed Coupling	CardapioController.save	7	1.0
Feature Envoy	CategoriaController.filter	8	1.0
Feature Envoy	CategoriaController.save	9	1.0
Dispersed Coupling	CategoriaController.save	10	1.0
Feature Envoy	AutorizadorInterceptor.preHandle	11	1.0
Feature Envoy	LoginController.logar	12	1.0
Refused Parent Be...	ItemCardapio	13	1.0
Feature Envoy	ReservaController.filter	14	1.0
Refused Parent Be...	Reserva	15	1.0
Feature Envoy	Pedido.getTotal	16	1.0
Feature Envoy	ClienteController.filter	17	1.0

2. CodeSmells encontrados utilizando a ferramenta JSPIRIT.

2.3 Medição 1 – Após Refatorar CS. Featury Envy

A métrica **Coesão**, teve um aumento expressivo de 1481, para 2503 (Após a refatoração), ou seja, houve um resultado negativo, pois as classes ficaram menos coesas.

Para a métrica de **Acoplamento**, ocorreu um aumento expressivo de 144 para 257, o resultado é negativo, uma vez que quanto maior o numero total das métricas, maior o acoplamento entre classes e métodos.

Para a métrica **Complexidade**, houve um aumento relevante de 452 para 484, ou seja, houve um resultado negativo, pois as métricas foram altas em todos os casos, logo, a complexidade entre classes e métodos é maior.

Para a métrica **Herança**, houve também um aumento no valor total, de 427 para 442, logo, um resultado negativo, pois quanto maior o valor total das métrica, maior o grau de herança do sistema.

E por fim, para a métrica de **Tamanho**, houve um aumento no valor total de 4033 para 4084. Resultado negativo, pois houve aumento no tamanho do sistema integrando as métricas avaliadas.

Tipo de Refatoração utilizado: **Move Method**

Quantidade de CS. Featury Envy refatorados: 11

2.4 Medição 2 – Após Refatorar CS. God Class

Para a métrica **Coesão**, houve um aumento de 2503, para 2603 (Após a refatoração), o resultado foi negativo, uma vez que quanto maior o valor, menor a coesão entre as classes.

Para a métrica **Acoplamento**, também houve um pequeno aumento, de 257 para 264. Resultado negativo, pois há um maior acomplamento entre classes e métodos.

Para a métrica de **Complexidade**, houve uma diminuição de 484 para 525. Resultado positivo, pois a complexidade do sistema tornou-se menor.

Para a métrica **Herança**, houve novamente um aumento relevante no valor total, de 442 para 481. Resultado negativo, pois menor é o grau de herança do sistema

Para a métrica de **Tamanho**, houve também uma diminuição no valor total de 4084 para 4231, ou seja, resultado negativo, uma vez que houve uma diminuição do sistema.

Tipo de Refatoração utilizado: **Extract Class**

Quantidade Total de CS. God Class refatorados: 9

2.5 Medição 3 – Após Refatorar CS. Dispersed Coupling

Para a métrica **Coesão**, houve um aumento de 2603, para 2665 (Após a refatoração) no valor total. Um resultado negativo, uma vez que, quanto maior o valor total, menor a coesão entre classes.

Para a métrica de **Acoplamento**, ocorreu um aumento, de 264 para 347. Resultado negativo, maior acoplamento entre classes e métodos do sistema

Para a métrica **Complexidade**, houve um aumento de 525 para 579. Resultado negativo, pois aumentou a complexidade do sistema, para as métricas respectivas: **ACC**, **SCC**, **EVG** e **MaxNet**.

Para a métrica **Herança**, ocorreu um aumento de 481 para 506, resultado negativo, menor grau de herança do sistema entre classes e subclasses, e por fim, para a métrica de **Tamanho**, houve um aumento no valor total de 4231 para 4264. Resultado negativo, pois quanto maior o valor, maior o tamanho do sistema.

Tipo de Refatoração utilizado: **Move Method**

Quantidade Total de CS. Dispersed Coupling refatorados: 3

2.6 Medição 4 – Após Refatorar CS. Refused Parent Bequest

Para a métrica **Coesão**, houve novamente um aumento de 2665, para 2741 (Após a refatoração) no valor total. Resultado negativo, uma vez que, quanto maior o valor total, menor a coesão entre classes.

Para a métrica de **Acoplamento**, ocorreu um aumento, de 347 para 387. Resultado negativo, maior acoplamento entre classes e métodos do sistema

Para a métrica **Complexidade**, houve uma leve diminuição de 579 para 564. Resultado positivo, menor a complexidade do sistema.

Para a métrica **Herança**, ocorreu um aumento de 506 para 523, resultado negativo, o grau de herança do sistema é maior, e por fim, para a métrica de **Tamanho**, houve um aumento no valor total de 4264 para 4277. Resultado negativo, pois quanto maior o valor, maior o tamanho do sistema.

Tipo de Refatoração utilizado: **Extract SuperClass**

Quantidade Total de CS. Refused Parent Bequest refatorados: 2

2.7 Medição 5 – Após Refatorar CS. Shotgun Surgery

Para a métrica **Coesão**, houve um aumento de 2741, para 2778 (Após a refatoração) no valor total. Resultado negativo, uma vez que, quanto maior o valor total, menor a coesão entre classes.

Para a métrica de **Acoplamento**, ocorreu um aumento de 387 para 408. Resultado negativo, maior acoplamento entre classes e métodos do sistema

Para a métrica **Complexidade**, houve um aumento de 564 para 701. A complexidade do sistema aumentou, logo, o resultado é negativo

Para a métrica **Herança**, ocorreu um aumento de 523 para 546, Resultado negativo, pois o grau de herança das classes e sub-classes aumentou. e por fim, para a métrica de **Tamanho**, houve também um aumento no valor total de 4277 para 5100. Resultado negativo, pois quanto maior o valor, maior o tamanho do sistema.

Tipo de Refatoração utilizado: **Inline Class**

Quantidade Total de CS. Shotgun Surgery refatorados: 1

2.8 Medição 7 – Após Refatorar CS. Long Method

Para a métrica **Coesão**, houve uma aumento de 2778 para 2853 (Após a refatoração) no valor total. Resultado negativo, uma vez que, quanto maior o valor total, menor a coesão entre classes.

Para a métrica de **Acoplamento**, ocorreu um aumento de 408 para 453. Resultado negativo, maior acoplamento entre classes e métodos do sistema

Para a métrica **Complexidade**, houve um aumento de 701 para 729. A complexidade do sistema aumentou, logo, o resultado é negativo, pois mais complexo o sistema.

Para a métrica **Herança**, ocorreu um aumento de 546 para 571, Resultado negativo, pois o grau de herança das classes e sub-classes aumentou consideravelmente, e por fim, para a métrica de **Tamanho**, houve um aumento no valor total de 5100 para 5324. Resultado negativo, pois quanto maior o valor, maior o tamanho do sistema.

Tipo de refatoração utilizado: **Move Method**

Quantidade Total de CS. Long Method refatorados: 12

2.8 Medição 8 – Após a refatoração de todos os Code Smells do Projeto.

Sistema Restaurant e	Coesão	Complexidade				Herança			Acoplament o	Tamanho			
		LCOM	AC	SC	EV	MaxNe	DI	NO		IFANI	CBO	LO	CLO
	2	C	C	G	t	T	C	N		C	C	M	L
Total	2853	121	85	75	448	103	86	382	453	1231	875	188	133
										0			8
Total	2853	729				571			453	5324			
Attributes													
Total ALL	9930												
Metrics													

3. Tabela após Refatoração de todos os Code Smells

Como pode ser visto acima, a tabela contém um resumo de todos os valores totais, para todas as cinco métricas após a refatoração de todos os code smells.

Houve um resultado negativo de uma visão geral após a refatoração total, para cada conjunto de métrica, por exemplo, para a métrica de Coesão(LCOM2), houve um aumento bastante expressivo se comparado ao valor total antes da refatoração, de 1481 para 2853, isso significa um resultado negativo, uma vez que quanto maior o valor, menos coesa são as classes.

Para a Complexidade(ACC, SCC, EVG e MaxNet) houve também um resultado negativo, pois ocorreu um aumento bastante relevante se comparado ao valor total antes da refatoração, de 452 para 729, ou seja, a complexidade entre classes e métodos aumentou, a abstração das classes e métodos tornou-se maior, logo de difícil compreensão para quem for ler o código.

Para o Acomplamento(CBO), houve novamente um resultado negativo no valor total da métrica, de 144(Antes da refatoração), para 571, uma vez que quanto maior o número, maior o acoplamento entre classes e métodos.

E por fim, novamente um aumento bastante expressivo no tamanho, resultando em um resultado negativo para a métrica do Tamanho(LOC, CLOC, NIM e CDL), se comparado ao valor total do conjunto de atributos antes da refatoração(4033), para 5324, maior o tamanho do sistema.

3. COMPARAÇÃO DOS RESULTADOS

Relacionando o artigo “Refactoring effect on internal quality attributes: What haven’t they told you yet?”, com o resultado das refatorações do presente sistema, o estudo do efeito das refatorações não coincide com o resultado das métricas, pois das cinco métricas avaliadas, todas as cinco apresentaram resultado negativo após a refatoração, logo é válido argumentar, que a análise de atributos internos não melhoraram com a aplicação de refatoração.

Link para artigo: [Refactoring effect on internal quality attributes: What haven’t they told you yet?](#)

Referências

Zhat, Dmitry, Guru R. – **“Code Smells”**, Disponível em: <<https://refactoring.guru/pt-br/refactoring/smells>>, Acesso em: 10/08/2021

Lacerda G., Petrilio F., Pimenta M. Guéhéneuc Gael Y. **“Code smells and refactoring: A tertiary systematic review of challenges and observations”**, 2020

Fowler, Martin. Refatoração: **“Aperfeiçoando o design de códigos existentes.”** Novatec Editora, p. 25-33, 2020.

Chiele, Cristiano. **"Estudo sobre práticas ágeis de refatoração e testes automatizados no desenvolvimento de software para melhoria da qualidade de sistemas legados."**, 2017