



**UNIVERSIDADE  
FEDERAL DO CEARÁ**  
CAMPUS DE QUIXADÁ

**CURSO DE QUALIDADE DE SOFTWARE**

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

**Equipe:**

**Luis Gomes - 428223**

**Professora:**

**Carla Ilane Moreira Bezerra**

**QUIXADÁ**

**Junho, 2021**

**SUMÁRIO**

<u>1</u>	<u>DESCRIÇÃO DO PROJETO</u>	2
<u>2</u>	<u>AVALIAÇÃO DO PROJETO</u>	2
<u>2.1</u>	<u>Medição 1 – Antes de refatorar o projeto</u>	2
<u>2.2</u>	<u>Deteção dos Code Smells</u>	3
<u>2.3</u>	<u>Medição 2 – Após Refatorar Code Smell X</u>	4
<u>2.4</u>	<u>Medição 3 – Após Refatorar Code Smell Y</u>	4
<u>2.5</u>	<u>Medição Z – Após a refatoração de todos os code smells do projeto</u>	4
<u>3</u>	<u>COMPARAÇÃO DOS RESULTADOS</u>	4
	<u>REFERÊNCIAS</u>	4
	<u>APÊNDICE A</u>	5

## • DESCRIÇÃO DO PROJETO

PubNub foi desenvolvido em HTML, CSS, JavaScript, Java e PHP, sendo então uma plataforma de API para desenvolvedores que potencializa a infraestrutura em tempo real em aplicativos para criar espaços virtuais envolventes onde as comunidades online podem realmente se conectar. O PubNub serve como base para bate-papo on-line, eventos ao vivo, geolocalização, controle remoto de IoT e atualizações em tempo real em milhares de organizações em uma ampla variedade de setores, tudo em uma escala global massiva. Empresas notáveis como Adobe, Atlassian, DocuSign, Peloton, TicketMaster e RingCentral confiam na plataforma e infraestrutura global do PubNub para alimentar suas plataformas e aplicativos.

Link do projeto: <https://github.com/pubnub/java>

Tabela 1 – Características do Projeto

Projeto	LOC	# de classes	# de releases
Pubnub	157533	1292	59

- **Medição 1 – Antes de refatorar o projeto**

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

[illegible]

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.
	<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 (EVC)</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>Basees 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.

- Detecção dos Code Smells**

Nessa Seção deve ser indicado quais e quantos code smells foram detectados no projeto. Faça uma Tabela indicando os code smells detectados pela ferramenta JSPirit e quantos code smells para cada tipo foram detectados.

Tabela 3 – Code smells do projeto.

Nome do Code Smell	Quantidade
Long Statement	1016
Magic Number	938
Complex Conditional	20
Complex Method	25

Long Identifier	78
Long Parameter List	57
Empty Catch Clause	8
Long Method	6
Missing Default	6
Abstract Function Call from Constructor	0

- **Medição 2 – Após Refatorar Code Smell X**

Nessa Seção você deve indicar os valores de todas as métricas da Tabela 2, após refatorar um determinado code smell. Esse code smell deve ser totalmente refatorado até não ser mais detectado pela JSPirit. Você deve também incluir a técnica de refatoração utilizada para retirar o code smell. Isso deve ser feito para cada code smell detectado no projeto. Após a refatoração de cada code smell deve ser realizada uma nova medição na ferramenta Understand. Deve ser realizada também uma análise dos 5 atributos de qualidade e que métricas pioram ou melhoram de acordo com a retirada desses code smells.

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

Sistema	Coesão	Complexidade				Herança			Acoplament o	Tamanho			
	LCOM 2	ACC	SC C	EVG	MaxNe t	DI T	NO C	IFANI N	CBO	LO C	CLO C	NI M	CD L
S1 antes da refatoraça o	10742	102 0	166 84	989	1429	109 5	471	859	5884	157 533	4862	299 5	129 2
S1 após refat. CS X	10896	102 5	165 80	101 0	1444	110 4	471	867	5914	159 214	5176	301 0	130 8

S1 após refat. CS X													

- **Medição 3 – Após Refatorar Code Smell Y**

.....

- **Medição Z – 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 pioraram ou melhoraram de acordo com a retirada dos code smells.

- **COMPARAÇÃO DOS RESULTADOS**

Leia o artigo:

[https://www.sciencedirect.com/science/article/pii/S0950584920301142?casa\\_token=xcwL1BwaRFUAAAAA:wZjXB0Wx-0FiMSpZSzyi0b7iRe7ZJOr8FdwiHzEkvzeQHh0Iz6mxPCF769JgRiZ69TyfI5l8BP0](https://www.sciencedirect.com/science/article/pii/S0950584920301142?casa_token=xcwL1BwaRFUAAAAA:wZjXB0Wx-0FiMSpZSzyi0b7iRe7ZJOr8FdwiHzEkvzeQHh0Iz6mxPCF769JgRiZ69TyfI5l8BP0)

Faça uma comparação dos resultados do seu projeto de acordo com esse artigo.

- **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**

Incluir possíveis documentos que possam ser gerados no desenvolvimento do sistema.