



**CURSO DE ENGENHARIA DE SOFTWARE**

**RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE**  
**State Progress Bar**

**Equipe:**

**Douglas da Silva Holanda**  
**Wesley Itallo Vieira da Silva**

**Professora:**

**Carla Ilane Moreira Bezerra**

**QUIXADÁ**

**Junho, 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
	REFERÊNCIAS	4

## 1 DESCRIÇÃO DO PROJETO

StateProgressBar é uma biblioteca Android para exibir e transitar entre os vários status de uma barra de progresso. Desenvolvida por Kofi Gyan, a biblioteca oferece diversas funcionalidades para barras de progresso, podendo ser utilizada para personalizar da melhor maneira. O projeto possui 2265 linhas de código Java e 1173 linhas de código xml.

Link do projeto: <https://github.com/kofigyan/StateProgressBar>

Tabela 1 – Características do Projeto

Projeto	LOC	# de classes	# de releases
StateProgressBar	2165	57	1

## 2 AVALIAÇÃO DO PROJETO

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

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

Sistema	Coesão	Complexidade	Herança			Acoplamento	Tamanho			
	LCOM	CC	DIT	NOC	FANIN	CBO	LOC	CLOC	NIM	CDL
Antes da refatoração	-187	401	33	0	55	199	1721	444	287	57

Tabela 3 - Métricas da ferramenta Designite que não foram relacionadas com as métricas fornecidas pelo Understand.

Sistema	NOF	NOPF	NOPM	WMC	FANOUT	CC	PC
Antes da refatoração	157	12	175	401	55	401	274

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 as 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

Tabela 3 – Code smells do projeto.

Nome do Code Smell	Quantidade
Long Identifier	26
Magic Number	58
Long Statement	40
Unutilized Abstraction	36
Complex Method	2

## 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.