

**CURSO DE ENGENHARIA DE SOFTWARE**

**RELATÓRIO – TRABALHO FINAL QUALIDADE DE SOFTWARE**

**Canarinho**

**Equipe:**

**Eduardo Henrique Brito da Silva**

**537777**

**Francisco Jerônimo da Silva Júnior**

**433399**

**Professora:**

**Carla Ilane Moreira Bezerra**

**QUIXADÁ**

**Junho, 2022**

**SUMÁRIO**

[1 DESCRIÇÃO DO PROJETO 2](#_Toc63168868)

[2 AVALIAÇÃO DO PROJETO 2](#_Toc63168869)

[2.1 Medição 1 – Antes de refatorar o projeto 2](#_Toc63168870)

[2.2 Detecção dos Code Smells 3](#_Toc63168871)

[2.3 Medição 2 – Após Refatorar Code Smell X 4](#_Toc63168872)

[2.4 Medição 3 – Após Refatorar Code Smell Y 4](#_Toc63168873)

[2.5 Medição Z – Após a refatoração de todos os code smells do projeto 4](#_Toc63168874)

[3 COMPARAÇÃO DOS RESULTADOS 4](#_Toc63168875)

[REFERÊNCIAS 4](#_Toc63168876)

[APÊNDICE A 5](#_Toc63168877)

# DESCRIÇÃO DO PROJETO

Nessa Seção deve ser descrito o projeto. Importante incluir uma Tabela com as seguintes informações: Número de Linhas de Código (LOC), Número de Classes e Número de Releases. Incluir o link do github do projeto. Falar sobre a natureza do projeto, se é JAVA OO, se é de código aberto ou fechado, quais tecnologias utilizadas no projeto.

Link do projeto: <https://github.com/concretesolutions/canarinho.git>

Esta biblioteca é um conjunto de utilitários para trabalhar com padrões brasileiros no Android, por exemplo: validação e formatação de CPF, CNPJ e outros cadastros comuns no Brasil. Inspirado em: <https://github.com/caelum/caelum-stella>. O foco aqui é o Android. Portanto, não é compatível com aplicações Java puras.

Tabela 1 – Características do Projeto

|  |  |  |  |
| --- | --- | --- | --- |
| **Projeto** | **LOC** | **# de classes** | **# de releases** |
| Canarinho | 2555 | 62 | 2.0.2 |

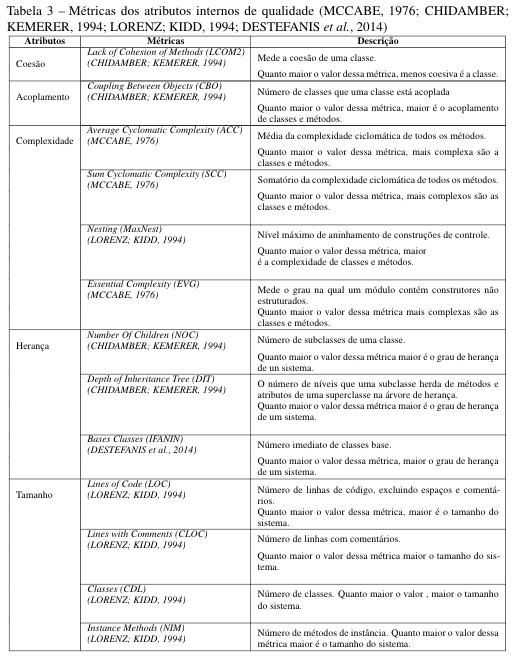
# AVALIAÇÃO DO PROJETO

## Medição 1 – Antes de refatorar o projeto

Nessa Seção deve ser incluída a Tabela com a medição das métricas de coesão, acoplamento, complexidade, herança e tamanho, antes do projeto ser refatorado. Para isso será utilizada a ferramenta Understand. A Tabela 2 apresenta a descrição das métricas, faça uma tabela similar.

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sistema** | **Coesão** | **Complexidade** | | | | **Herança** | | | **Acoplamento** | **Tamanho** | | | |
| LCOM2 | ACC | SCC | EVG | MaxNet | DIT | NOC | IFANIN | CBO | LOC | CLOC | NIM | CDL |
| S1 antes da refatoração | 1071 | 412 | 140 | 1573 | 184 | 288 | 34 | 96 | 90 | 10122 | 2231 | 134 | 739 |
| S1 após refat. CS X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| S1 após refat. CS X |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |



## 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** |
|  |  |
|  |  |

## 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.

## 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>

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.