

# **Um Estudo das Características de Qualidade de Sistemas Java**

**Bruno Gomes Ferreira, João Pedro Mairinque de Azevedo,  
Matheus Vieira dos Santos, Marcio Lucas Machado Pereira**

<sup>1</sup> Instituto de Ciências Exatas e Informática (ICEI)  
Pontifícia Universidade Católica de Minas Gerais  
Engenharia de Software  
Belo Horizonte – MG – Brazil

## **1. Introdução**

No campo do desenvolvimento de softwares open-source, caracterizada pela contribuição descentralizada de desenvolvedores em diferentes partes do código, a preservação de padrões de código pré-estabelecidos e também evolução dos atributos de qualidade interna é um desafio recorrente ao se desenvolver um software. A essência da qualidade interna cobre pontos chave como modularidade, facilidade de manutenção e legibilidade, que podem se perder em meio à colaboração. Para lidar com isso, é possível destacar o papel vital de práticas como as revisões de código e análises estáticas. Essas práticas são frequentemente integradas em processos de Integração Contínua e Entrega Contínua (CI/CD), proporcionando um ciclo de feedback constante e aprimoramento contínuo do software.

A análise de métricas de software o torna-se um elemento crucial para a manutenção da qualidade interna de um software, tendo isso em vista, o presente estudo tem como foco a avaliação da qualidade de repositórios programados em Java, estabelecendo uma relação entre os elementos do processo de desenvolvimento e as métricas de produto obtidas por meio da ferramenta CK. A linguagem de programação Java, por se tratar de uma linguagem Orientada a Objetos é fundamentalmente diferente de softwares desenvolvidos à partir dos demais paradigmas de programação, por essa razão, as métricas devem ser ajustadas para atender a características distintas como: Coesão, abstração e acoplamento [Pressman and Maxim 2016]. De acordo com Pressman (2016), os conjuntos de métricas de classe que recebem maior referência incluem as métricas CK, desenvolvidas por Chidamber e Kemerer em 1994. As métricas CK são utilizadas para analisar classes em sistemas orientados a objetos, visando medir aspectos essenciais como a complexidade, a coesão e o acoplamento.

Para a avaliação das características de qualidade de sistemas programados em Java, traçou-se as seguintes perguntas de pesquisa:

- RQ 01. Qual a relação entre a popularidade dos repositórios e as suas características de qualidade?
- RQ 02. Qual a relação entre a maturidade do repositórios e as suas características de qualidade?
- RQ 03. Qual a relação entre a atividade dos repositórios e as suas características de qualidade?
- RQ 04. Qual a relação entre o tamanho dos repositórios e as suas características de qualidade?

Para os autores, como hipóteses informais, os repositórios populares terão um alto cuidado com modularidade, possuindo baixa taxa de acoplamento e alta coesão. Isso se deve pela quantidade de colaboração e engajamento da comunidade ao passar do tempo. A maturidade tende a ser alta e possui uma alta taxa de atividades dentro desses repositórios. Em termos de tamanho, esses repositórios possuem uma maior tendência a ser grandes por serem mais maduros.

Com o intuito de responder as perguntas de pesquisa, foram utilizadas métricas de qualidade e métricas de processo de forma que seja possível traçar a relação entre elas. Quando se trata de métricas de processo, refere-se as medições que podem ser aplicadas a um processo de software com o objetivo de melhorá-lo de forma contínua. [Pressman and Maxim 2016]. Para este trabalho, foram selecionadas as seguintes métricas de processo:

- Popularidade: número de estrelas
- Maturidade: idade (em anos) de cada repositório coletado
- Atividade: número de releases
- Tamanho: linhas de código (LOC) e linhas de comentários

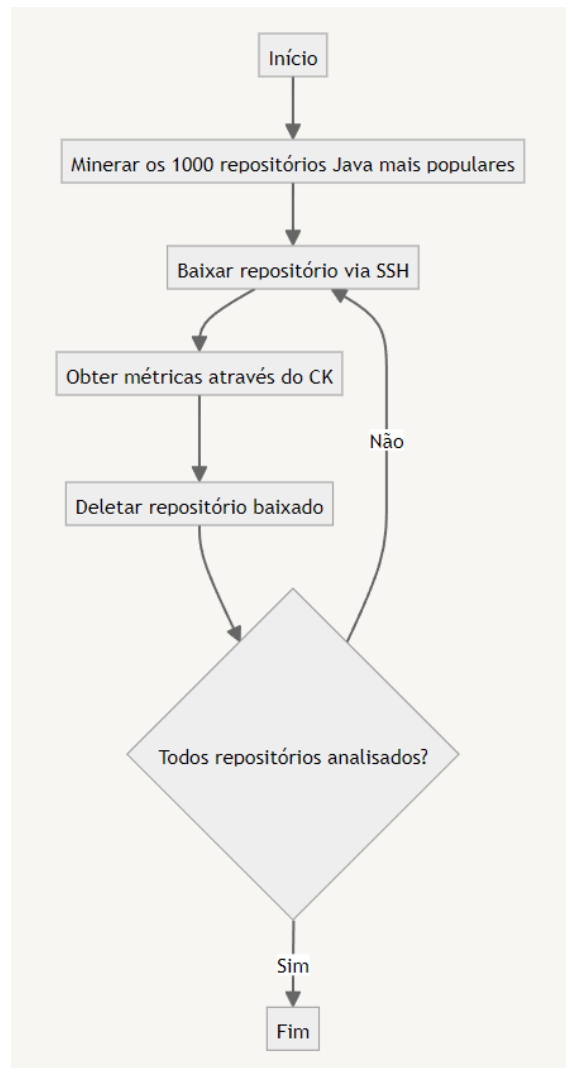
Para as métricas de qualidade, optou-se nesse trabalho por analisar as seguintes métricas:

- CBO: Coupling between objects
- DIT: Depth Inheritance Tree
- LCOM: Lack of Cohesion of Methods

Em relação as métricas de qualidade selecionadas, o CBO mede o nível de acoplamento entre duas classes ou seja, uma classe A é considerada acoplada a classe B se ela utiliza métodos ou atributos pertencentes à classe B. [Oliveira 2010] De acordo com Oliveira (2010), DIT está relacionado ao tamanho da árvore de herança, mede o comprimento da árvore à partir da raiz até o nó folha de maior tamanho, quanto maior o valor de DIT, maior será a complexidade do sistema e também maior dependência. E finalmente, LCOM relaciona com a falta de coesão numa classe, avalia a similaridade entre métodos de uma classe, onde similaridade entre dois métodos é o acesso aos mesmos atributos da classe. Quanto maior a similaridade entre métodos maior será a coesão da classe e menor será o valor de LCOM

## 2. Metodologia

Com o objetivo de obter características de qualidade dos sistemas Java, os autores seguiram a metodologia de pesquisas quantitativas onde através dos valores numéricos obtidos, será possível determinar as respostas das perguntas de pesquisa. Para a obtenção dos dados necessários foi utilizado as seguintes ferramentas: Python e GraphQL para obter os 1000 repositórios Java mais populares, SSH(Security Shell) para poder baixar repositórios do GitHub e o repositório CK Calculator para obter métricas de qualidade dos repositórios. O passo a passo da estratégia utilizada pode ser resumida pelo diagrama de fluxo (Figura 1)



**Figure 1. Diagrama de fluxo para obtenção dos resultados**

Inicialmente, foi minerado 1000 repositórios mais populares que contém a linguagem Java como principal linguagem de desenvolvimento através de GraphQL e a linguagem de programação Python que é responsável por fazer o código que irá fazer a requisição da query do GraphQL, nesta query foi gerado um CSV contendo dados como o link do repositório do GitHub, quantidade de estrelas do repositório e idade do repositório, após isso utilizou-se SSH(Security Shell) para fazer operações de clonagem

dos repositórios e dar prosseguimento a obtenção dos dados.

O SSH, ou Security Shell, é um protocolo de rede que permite a comunicação segura entre dois computadores através de uma conexão criptografada, o GitHub possui comunicação com SSH, desta maneira, é possível realizar comandos, como clonar um repositório para um computador pessoal e também executar outros comandos que também podem ser feitos via terminal. Para este laboratório, foi criado um script em SSH que tem como objetivo ler o CSV gerado com os 1000 repositórios e a partir disto clonar o repositório da lista, gerar uma build do CK Calculator para coletar métricas de qualidade e por fim deletar o repositório, assim repetindo o ciclo até ler os 1000 repositórios.

O repositório CK Calculator, como o nome já remete, é um repositório que consegue fazer análise estática de projetos Java obtendo o resultado das métricas CK para aquele projeto. O CK analisa de forma recursiva os arquivos .java do projeto, bastando apenas executar um comando no terminal onde o programador coloca basicamente qual projeto será analisado e em qual pasta os arquivos gerados pela análise estática serão armazenados. [Aniche 2015]

## References

Aniche, M. (2015). **Ck calculator v0.7.0**. <https://github.com/mauricioaniche/ck>  
Acesso em: 29/03/2024.

Oliveira, J. F. d. (2010). **Métricas para avaliação do grau de quantificação de sistemas orientados por aspectos**. Pós-graduação, Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte - Minas Gerais.

Pressman, R. S. and Maxim, B. R. (2016). ***Engenharia de software: uma abordagem profissional***. AMGH, Porto Alegre, 8 edition.