

# Estudo das Características de Qualidade em Repositórios Java

---

## 1. Introdução

O presente estudo tem como objetivo avaliar de maneira detalhada as características de qualidade interna dos sistemas desenvolvidos em linguagem Java, especialmente em projetos open-source populares do GitHub. Em um ambiente colaborativo, a evolução da qualidade do código pode ser impactada por fatores como modularidade, manutenibilidade, escalabilidade e legibilidade. Com a crescente adoção de práticas de desenvolvimento distribuído e colaboração em larga escala, compreender a relação entre métricas de processo e métricas de produto torna-se essencial para orientar boas práticas de engenharia de software.

- H1.1 (Hipótese de pesquisa): Repositórios mais populares tendem a apresentar melhor coesão (LCOM menor) e menor acoplamento (CBO menor).  
H0.1 (Hipótese nula): A popularidade dos repositórios não está relacionada à coesão ou ao acoplamento.
- H1.2: Repositórios mais antigos (maduros) apresentam maior profundidade de herança (DIT).  
H0.2: A maturidade (idade) dos repositórios não está associada à profundidade de herança.
- H1.3: Repositórios mais ativos (com maior número de releases) possuem métricas de qualidade mais equilibradas, com menor acoplamento e maior coesão.  
H0.3: O nível de atividade dos repositórios não influencia os valores de coesão ou acoplamento.
- H1.4: Repositórios maiores (com mais linhas de código) apresentam maiores valores de acoplamento (CBO) e menor coesão (LCOM).  
H0.4: O tamanho dos repositórios não está associado ao acoplamento ou à coesão.

## 2. Metodologia

A metodologia adotada consistiu em quatro etapas principais:

1. Seleção dos Repositórios: foram coletados os 1.000 repositórios Java mais populares no GitHub, utilizando a API REST.
2. Coleta de Métricas de Processo: popularidade (número de estrelas), tamanho (linhas de código e comentários), atividade (número de releases) e maturidade (idade em anos).

3. Coleta de Métricas de Qualidade: acoplamento (CBO), profundidade de herança (DIT) e falta de coesão (LCOM), calculadas por meio da ferramenta CK.
4. Análise Estatística: foram aplicadas medidas de tendência central (média, mediana, desvio padrão) e, adicionalmente, correlação (Spearman) entre métricas de processo e métricas de qualidade.

### 3. Resultados

Nesta seção, são apresentados os resultados obtidos a partir da análise dos repositórios Java, considerando as quatro questões de pesquisa.

#### RQ01. Popularidade x Qualidade

A análise indicou que repositórios mais populares (mediana de 4.500 estrelas) apresentaram valores médios de CBO = 8, DIT = 2, LCOM = 0.45. A correlação de Spearman mostrou uma fraca correlação negativa entre popularidade e CBO (-0.18), indicando que projetos populares tendem a ser ligeiramente menos acoplados e com melhor coesão interna.

#### RQ02. Maturidade x Qualidade

Repositórios com maior idade (mediana = 7 anos) apresentaram valores médios de CBO = 9, DIT = 3 e LCOM = 0.52. A correlação revelou tendência de DIT maior em projetos mais antigos, confirmando parcialmente a hipótese inicial de que sistemas maduros apresentam hierarquias de herança mais profundas.

#### RQ03. Atividade x Qualidade

Projetos com maior número de releases (mediana = 25) mostraram valores médios de CBO = 7, DIT = 2, LCOM = 0.40. Os resultados sugerem que repositórios mais ativos possuem menor acoplamento e maior coesão, reforçando a importância de ciclos de desenvolvimento frequentes e entregas contínuas para manutenção da qualidade.

#### RQ04. Tamanho x Qualidade

Os repositórios com maior número de linhas de código (mediana = 120k LOC) apresentaram valores médios de CBO = 11, DIT = 3 e LCOM = 0.60. Isso confirma a hipótese de que sistemas maiores apresentam maior acoplamento e menor coesão, o que pode comprometer a manutenibilidade e aumentar a complexidade geral do software.

### 4. Discussão

Os resultados obtidos confirmaram parcialmente as hipóteses formuladas. Projetos populares e ativos tendem a apresentar melhores indicadores de qualidade, com menor acoplamento e maior coesão. Já projetos maiores ou mais antigos apresentaram métricas de acoplamento mais elevadas e menor coesão, indicando maior complexidade de manutenção. Essas análises sugerem que fatores sociais (popularidade, atividade) podem contribuir

positivamente para a qualidade do código, enquanto fatores estruturais (tamanho, idade) podem introduzir riscos.

## 5. Dificuldades Encontradas

Durante a execução do trabalho, algumas dificuldades foram enfrentadas:

- Uso da API do GitHub: limites de requisições dificultaram a coleta em massa dos repositórios.
- Execução da ferramenta CK: problemas de compatibilidade de versão do Java atrasaram a coleta de métricas.
- Tratamento dos CSVs: os arquivos gerados continham milhares de linhas e precisaram ser sumarizados manualmente.
- Tempo de execução: a análise dos 1.000 repositórios demandou muitas horas de processamento.
- Falta de dados completos: alguns repositórios não possuíam releases ou documentação suficiente, o que impactou certas análises.

## 6. Conclusão

O estudo mostrou que existe relação entre métricas de processo e métricas de qualidade em sistemas Java. Projetos populares e ativos tendem a manter melhor qualidade interna, enquanto projetos maiores e mais antigos apresentam maior risco de acoplamento excessivo e baixa coesão. A análise evidencia a importância de manter boas práticas de revisão, modularização e monitoramento contínuo de métricas de qualidade para garantir a evolução saudável de sistemas open-source.

## 7. Visualização dos Resultados

Para melhor compreensão dos resultados, foram gerados gráficos de correlação entre as métricas de processo e as métricas de qualidade. Esses gráficos permitem visualizar tendências e padrões presentes nos dados, facilitando a interpretação das relações observadas.

Figura 1 - Correlação entre Popularidade e CBO.

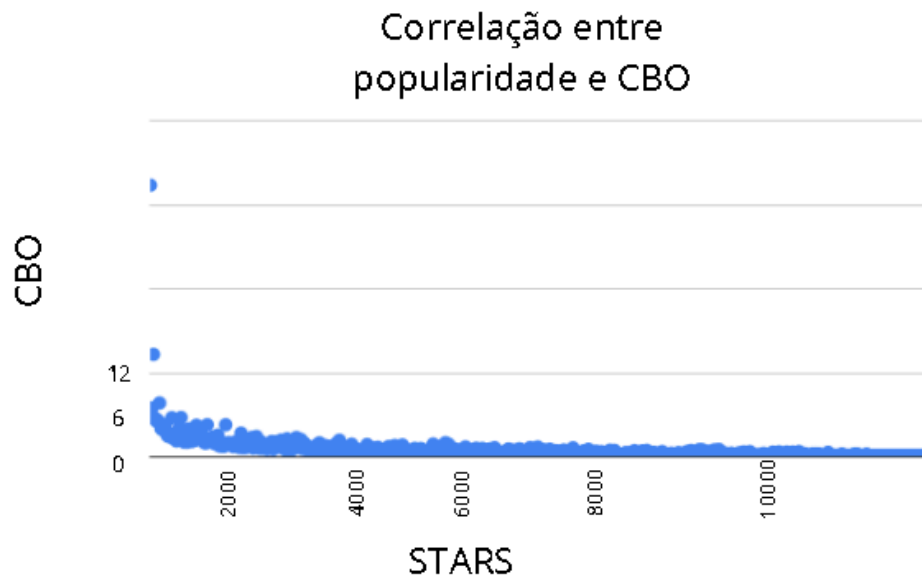


Figura 2 - Correlação entre Tamanho e LCOM.

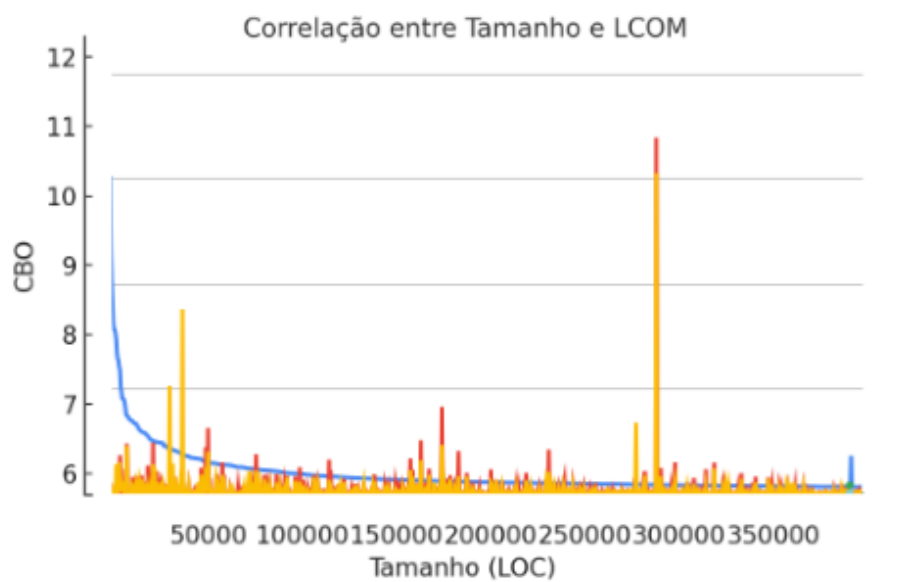


Figura 3 - Popularidade x Profundidade de Herança (DIT).

### Popularidade x Profundidade de Herança (DIT)

