

Um Estudo das Características de Qualidade de Sistemas Java: Relatório Final

Laboratório de Experimentação de Software

2021/1

PUC Minas

Guilherme Diniz

Arthur Branco

Introdução

Neste laboratório foi proposto uma análise da qualidade de código a partir da métricas CK de repositórios da linguagem Java e de sua correlação com certas características do seu processo de desenvolvimento.

Como objetivos específicos, deveriam ser formuladas as respostas das 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 dos 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?

A partir dessas indagações, foi criado uma série de hipóteses utilizando conhecimentos de desenvolvedores de sistemas para que fosse possível comparar com os resultados adquiridos posteriormente. Essas hipóteses foram:

RQ 1: De maneira geral, não acredita-se que haja qualquer relação entre a popularidade de um repositório e suas características de qualidade de código. Salvo exceções de repositórios que tem como objetivo demonstrar exemplos de boas práticas e/ou projetos de grandes empresas, não é a qualidade de seu código que é o principal chamativo dos repositórios.

RQ 2: Acredita-se que, conforme a maturidade dos repositórios aumenta, suas características de qualidade tendem a piorar. Afinal, com o passar do tempo, mais código deve ser integrado ao projeto por um número mais extenso de desenvolvedores, dificultando a manutenção de boas práticas.

RQ 3: Não acredita-se que haja qualquer relação entre a atividade dos repositórios e suas características de qualidade de código. Não é porque a comunidade está interagindo com certo sistema que boas práticas estão sendo adotadas em seu desenvolvimento e vice e versa.

RQ 4: Acredita-se que, conforme o tamanho dos repositórios aumenta, suas características de qualidade tendem a piorar. A explicação para isso é a mesma da relação de sua maturidade e das características de qualidade. Mais código apenas dificulta a manutenção de boas práticas.

Metodologia

Utilizando a API GraphQL pública do GitHub, foram coletados dados através do endpoint disponibilizado e um script em python [criado pelos pesquisadores](#). A coleta de métricas de códigos java foi realizada a partir da interação sobre os repositórios encontrados utilizando a ferramenta CK disponibilizada no repositório aberto [mauricioaniche/ck](#) e o script python mencionado anteriormente. O código foi executado dentro de uma pod com recursos ilimitados dentro de um cluster kubernetes. A partir dos dados obtidos, utilizando as bibliotecas pandas e matplotlib, foram gerados os gráficos de tipo scatterplot apresentados nos resultados da pesquisa.

Resultados

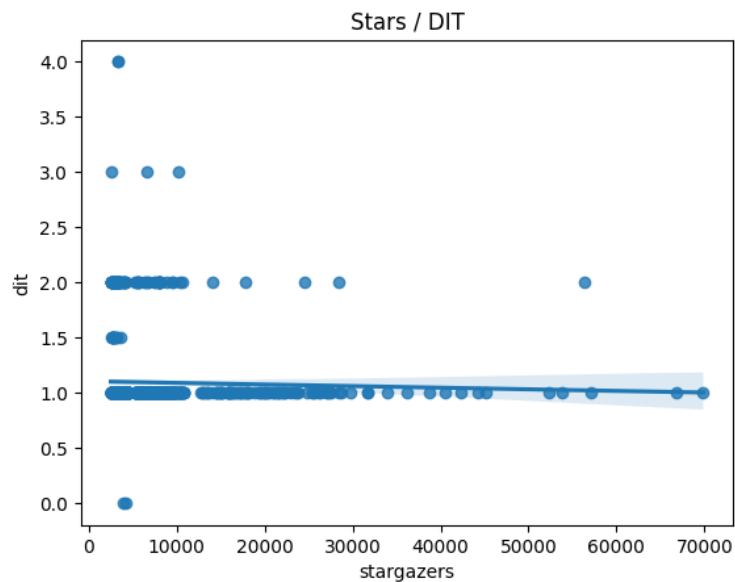
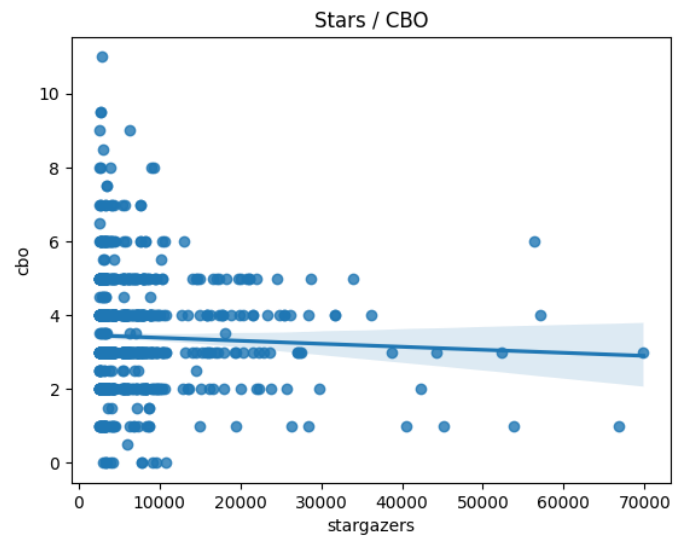
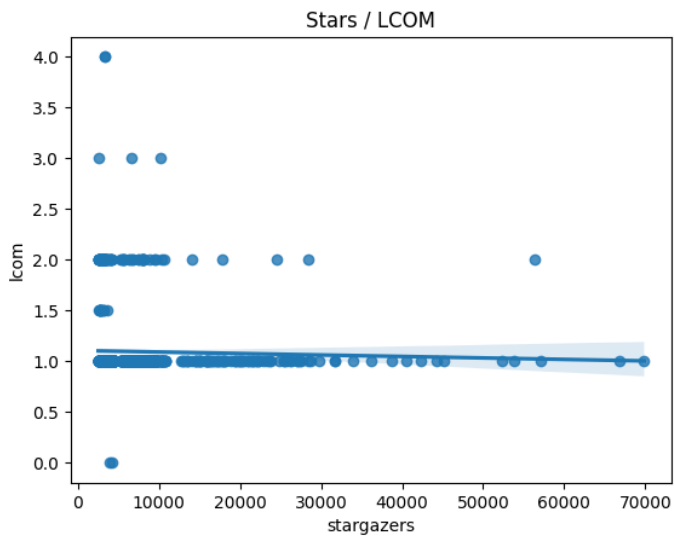
Como resultados preliminares, é ressaltado como os valores das métricas calculadas de LCOM e DIT surpreendem de maneira geral. A seguir foi feita uma pequena tabela com os valores de medida central de cada métrica:

Métrica	Mediana	Média	Desvio Padrão
CBO	3	3.42	1.44
DIT	1.0	1.10	0.33
LCOM	1.0	1.10	0.33

Como é possível verificar na tabela, os valores encontrados de DIT e LCOM foram extremamente próximos, sendo que grande parte dos repositórios apresentaram números próximos a 1. Uma análise literal desses dados seria de que a maioria dos repositórios apresentam métodos extremamente coesos (com apenas uma “intercessão” sem utilização de mesmos atributos por classe), representados pela métrica LCOM, e classes com “profundidade de herança” quase que inexistente, representado pela métrica DIT. Isso pode se dar pelo fato de que existem repositórios com código extremamente simples e que apresentam apenas códigos de exemplo. Outra possibilidade é de que simplesmente a biblioteca de análise de código utilizada não foi capaz de produzir resultados satisfatórios, o que comprometeria toda a interpretação das possíveis respostas das RQs propostas.

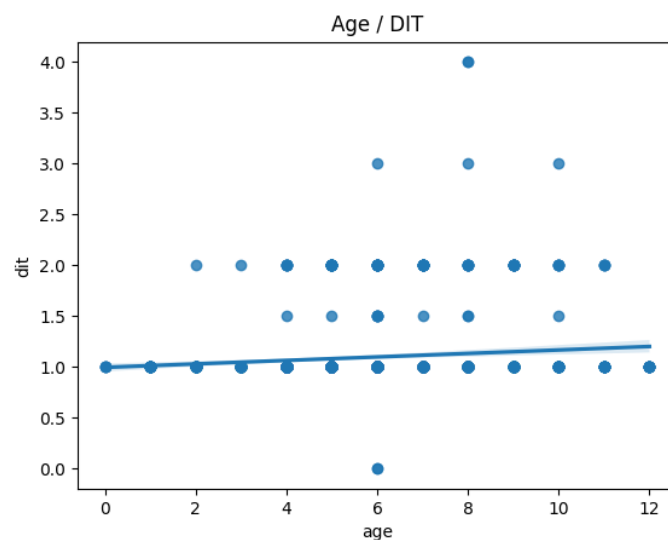
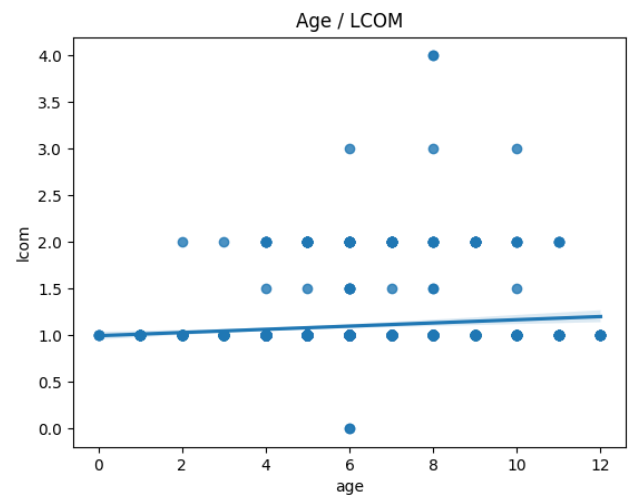
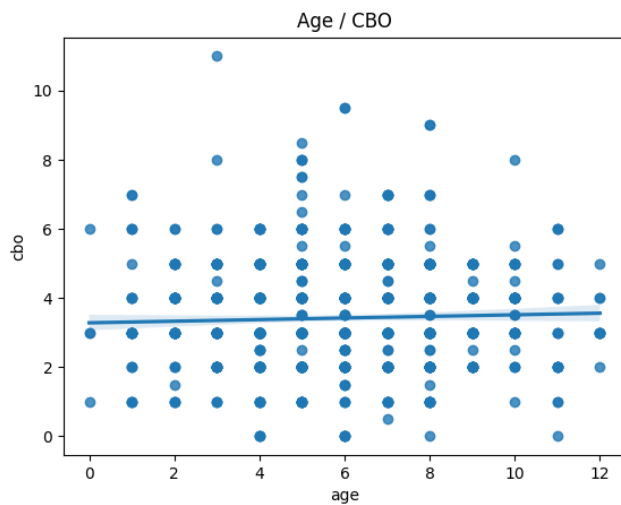
Já os resultados da análise da métrica CBO se mostraram mais diversificados, com valores médios próximos de 3 e desvio padrão próximo de 1.5. Ou seja, na média, há uma utilização de 3 classes externas por cada classe dos repositórios.

RQ 1:



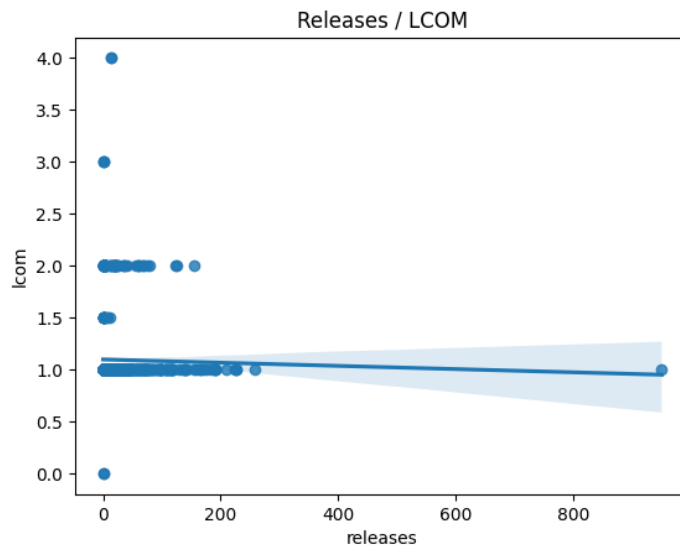
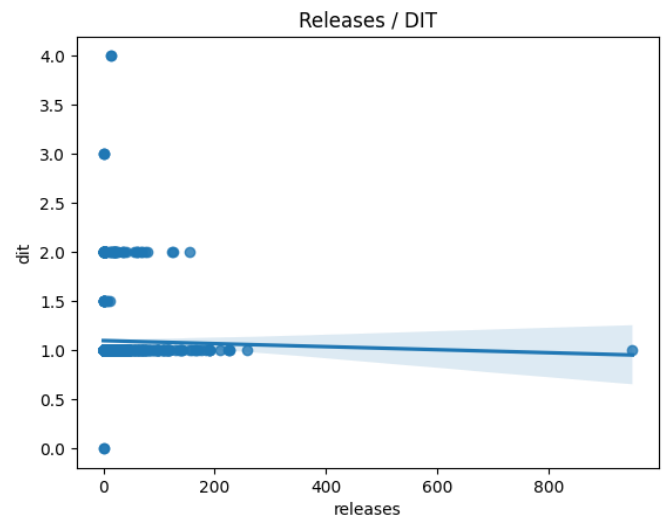
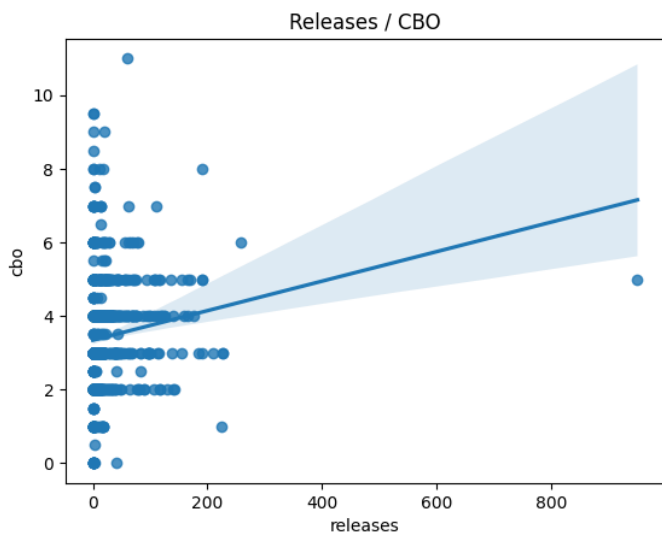
Como é possível observar nos gráficos, há uma concentração maior de repositórios com um número de estrelas no GitHub abaixo de 30000. Além disso, há uma leve variação negativa de todas as métricas conforme o número de estrelas aumenta. Como esse comportamento se apresenta extremamente suave, não é possível tirar grandes conclusões dos dados para que seja traçada uma relação fixa de popularidade e características de qualidade, assim como foi previsto

RQ 2:



A partir das análises dos gráficos, percebe-se que com o crescimento da idade dos repositórios acontece um pequeno aumento de heranças geral entre os métodos e falta de coesão, o que não pode ser afirmado para a métrica de acoplamento (CBO), que apresentou resultados muito amplos apresentando possivelmente ser uma métrica sem relação com a idade do repositório, conclusão geral que está de acordo com a previsão de queda de qualidade do código proposta pela pesquisa.

RQ 3:

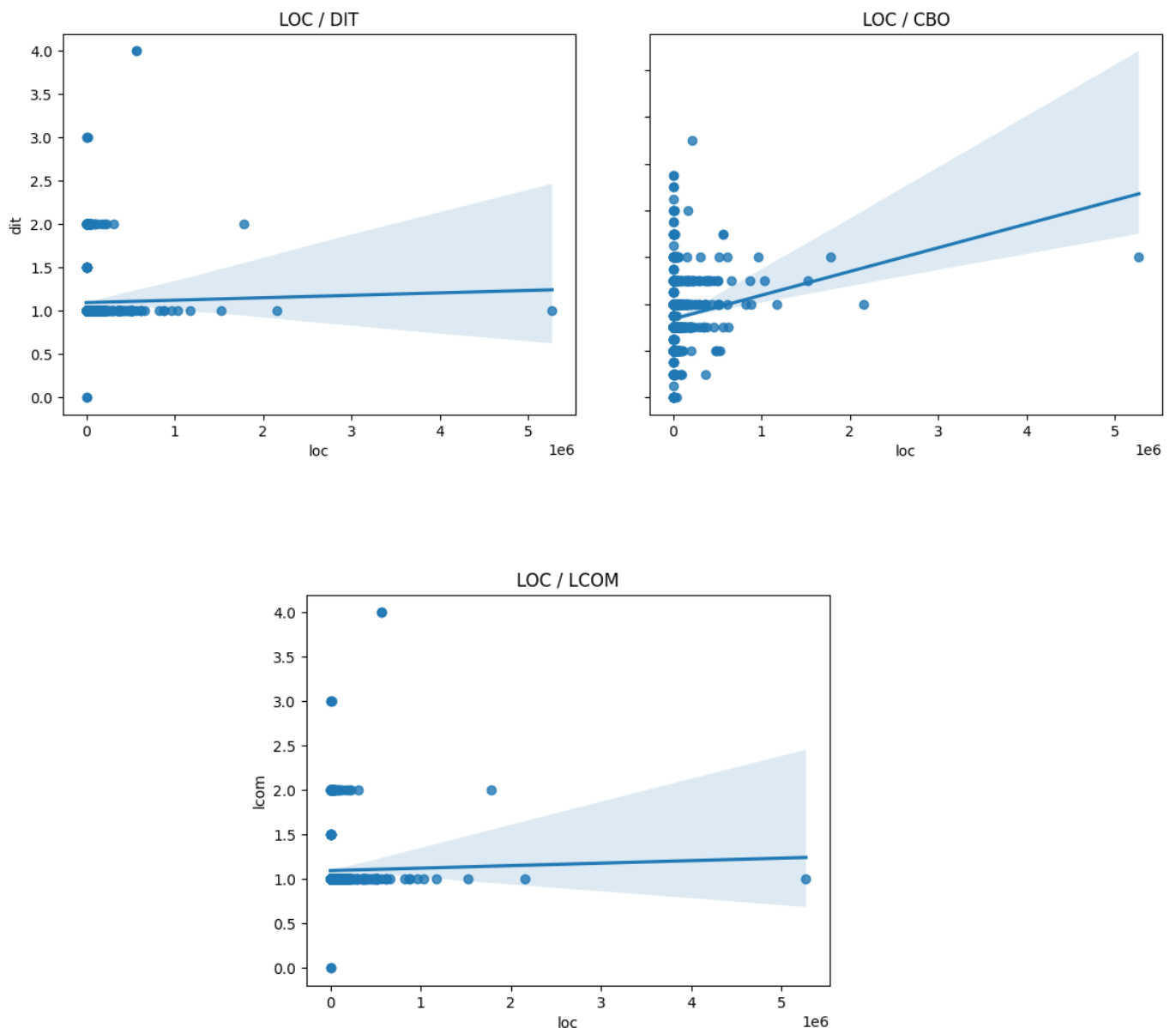


Como é possível verificar nos gráficos, existe uma enorme concentração de repositórios com menos de 200 releases. É interessante verificar, inclusive, que apenas um repositório acima de 400 releases e seu número é ainda maior que 900.

Apesar de que, novamente, as métricas de LCOM e DIT não permitam uma análise muito detalhada do comportamento da qualidade de código conforme o número de releases aumenta, é possível verificar que, conforme o número de releases aumenta, o valor médio de acoplamento entre classes aumenta com inclinação considerável. Sendo assim, uma

conclusão possível é de que a atividade dos repositórios e sua qualidade de código apresentam uma relação inversamente proporcional.

RQ 4:



Analisando os gráficos, percebe-se que o tamanho dos repositórios não teve correlação com o aumento da falta de coesão ou profundidade grande o suficiente para considerar que os mesmos possuem qualquer relação e que possivelmente estes dados são inerentes da técnica ou projeto pretendido para os repositórios, porém pode-se observar que aconteceu uma tendência de acoplamento proporcional com o tamanho dos repositórios, o que

alinha-se com a previsão da pesquisa de queda da qualidade com o aumento do numero de linhas de código.