Um Estudo das Características de Qualidade de Sistemas Java

Ana Flávia de Souza Ribeiro

Elian Eliezer Fialho de Castro

Miguel Martins Fonseca da Cruz

Rafael Augusto Vieira de Almeida

Instituto de Ciências Exatas e Informática
Pontifícia Universidade de Minas Gerais (PUC Minas)
Belo Horizonte – MG – Brasil

afsribeiro@sga.pucminas.br

eefcastro@sga.pucminas.br

miguel.cruz@sga.pucminas.br

rafael.almeida.1098763@sga.pucminas.br

O sistema tem como objetivo analisar aspectos da qualidade de repositórios desenvolvidos na linguagem Java, relacionando-os com características do seu processo de desenvolvimento, sob perspectiva de métricas de produto calculadas através da ferramenta CK.

1. Introdução

Os repositórios open source são base para a colaboração e inovação na comunidade de desenvolvimento de software atual. Ao entender as características desses repositórios podemos realizar análises importantes sobre o comportamento nesses ambientes. Um projeto de mineração de dados no GitHub se apresenta como uma oportunidade promissora para explorar esses aspectos, fornecendo não apenas uma visão mais profunda dos repositórios open source, mas também impulsionando o crescimento e a evolução contínua desta comunidade global.

Por esse ponto de vista, neste projeto foi realizada a mineração de repositório do Github com a fim de responder perguntas pré-definidas. O trabalho foi feito a partir da utilização de Python e GraphQL. Ao se destacar a sprint 2, leva-se em conta que 1000 repositórios foram analisados a fim de se responder às seguintes perguntas, com suas respectivas hipóteses estipuladas:

RQ 01. Qual a relação entre a popularidade dos repositórios e as suas características de qualidade?

H: Espera-se que quanto maior a popularidade do repositório, maior a suas características de qualidade, possivelmente porque um dos motivos de suas popularidades serem suas qualidades e por receberem maior atenção e manutenção.

RQ 02. Qual a relação entre a maturidade dos repositórios e as suas características de qualidade ?

H: Espera-se que quanto maior a maturidade de um repositório, maior suas características de qualidade, possivelmente porque devem incluir um código mais estável e uma comunidade mais opinativa em relação a erros.

RQ 03. Qual a relação entre a atividade dos repositórios e as suas características de qualidade?

H: Espera-se que os repositórios mais ativos tenham indicadores mais elevados de qualidade, possivelmente porque o envolvimento da comunidade e a manutenção constante se associam a uma maior qualidade no desenvolvimento de software.

RQ 04. Qual a relação entre o tamanho dos repositórios e as suas características de qualidade?

H: Não teria relação entre tamanho do repositório e característica de qualidade.

2. Metodologia

A metodologia aplicada neste estudo tem como o objetivo extrair os dados dos 1000 repositórios mais avaliados do Github. A fim de responder os questionamentos feitos e aplicar suas respectivas métricas, foi realizada a análise de dados diante de todos os resultados obtidos. Houve uma abordagem quantitativa para a pesquisa, fundamentada em literatura acadêmica especializada que reconhece a relevância e eficácia dessa técnica em projetos focados em coleta e análise de dados para o desenvolvimento de pesquisas acadêmicas, conforme Antonio Carlos Gil(2002).

Com base nas características que uma pesquisa quantitativa pode apresentar segundo Aline Oliva(s.d), analisa-se objetividade e neutralidade na utilização dos métodos e técnicas para a coleta de dados visando recolhe-los de forma imparcial para a pesquisa. Outra análise feita é a estatística para identificar padrões e relações entre todos os dados coletados.

2.1 Tecnologias Utilizadas

As tecnologias selecionadas para fazer a mineração dos dados foram GraphSql para consulta dos dados e o Python para a elaboração do código para extrair os dados da API do GitHub. Segundo Elisandra Botega(2023) o Python é conhecido por sua sintaxe simples e legibilidade, tornando-o acessível a uma ampla variedade de profissionais, desde analista de negócios até cientistas de dados.

Para auxiliar no processo de mineração de dados usando Python, foram utilizadas diversas bibliotecas para desempenhar um papel específico no código. A biblioteca csv foi implementada para ler arquivos csv e extrair as informações necessárias para a análise. A biblioteca shutil foi utilizada para remover diretórios e seus conteúdos recursivamente após a análise de um repositório, ajudando a limpar o espaço de armazenamento. A biblioteca subprocess é utilizada para a execução de processos externos a partir do código Python. No script, ela foi usada para chamar um processo Java, passando argumentos para executar a análise de métricas de código usando a ferramenta CK. A biblioteca tqdm é utilizada para fornecer uma barra de progresso visual durante o processamento dos repositórios, dando uma indicação visual do progresso do processo de análise. Por fim, a biblioteca gitpython foi utilizada para interagir com repositórios Git, mais especificamente clonar os repositórios do Github localmente.

2.2 Sprint 1

A primeira fase do projeto focou na mineração de dados iniciais seguindo as RQs, a partir da lista dos 1.000 repositórios Java + Script de Automação de clone e Coleta de Métricas mais Arquivo .csv com o resultado as medições de 1 repositório.

2.3 Sprint 2

Na Sprint 2, o trabalho evoluiu para a análise e extração de métricas dos repositórios Java coletados anteriormente. Para isso, os repositórios listados no arquivo CSV foram clonados localmente. Em seguida, métricas relevantes, como CBO, DIT e LCOM, foram extraídas utilizando ferramentas apropriadas. Essas métricas foram então armazenadas em arquivos CSV individuais para cada repositório. Posteriormente, foram correlacionadas com as informações originais dos repositórios. O objetivo era calcular somas agregadas das métricas para cada repositório e atualizar o arquivo CSV com essas informações agregadas. O resultado final foi a criação de um novo arquivo CSV, que continha tanto os dados originais dos repositórios quanto as métricas agregadas, proporcionando uma análise mais abrangente e detalhada do conjunto de dados.

3. Resultados

Realizou-se consultas à API do GitHub para coletar informações sobre repositórios que atendessem a certos critérios de seleção. Cada consulta retorna uma lista de repositórios, iteramos sobre essas listas para extrair os dados relevantes, como número de estrelas, número de commits, idade e tamanho de repositórios. Esses dados foram armazenados em arquivos csv para posterior análise.

Para as métricas de qualidade usadas como base, entende-se:

CBO: Acoplamento entre objetos

DIT: Profundidade da Árvore de Herança

LCOM: Falta de Coesão de Métodos

Tais métricas foram utilizadas como base para todas as requisições. Foram adotadas uma vez que são amplamente reconhecidas na engenharia de software como indicadores importantes da qualidade do código. Como uma forma de padronização para o conceito de qualidade necessário, trata-se que quanto menor seus valores, maior o valor de qualidade.

3.1 Popularidade x Qualidade

- **H1:**

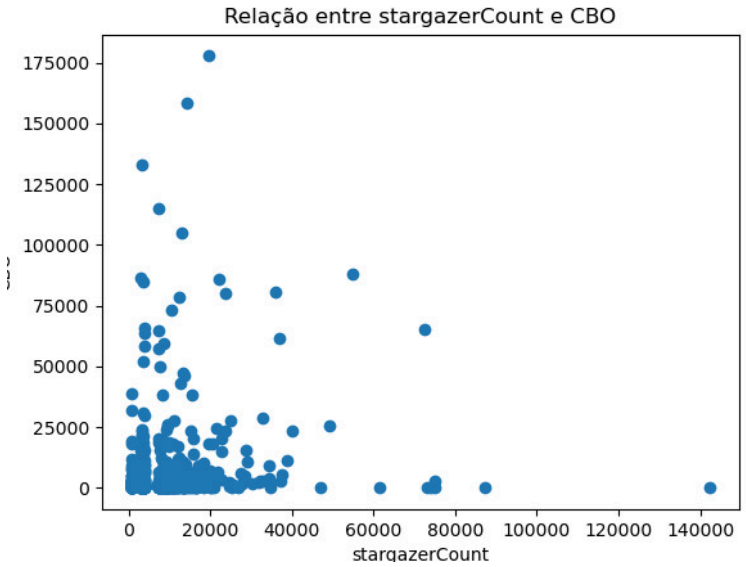

Figura 1 - Gráfico Estrelas X CBO

Tal gráfico representa que quanto maior a popularidade de um repositório, menor tende a ser o acoplamento entre os objetos de seu sistema.

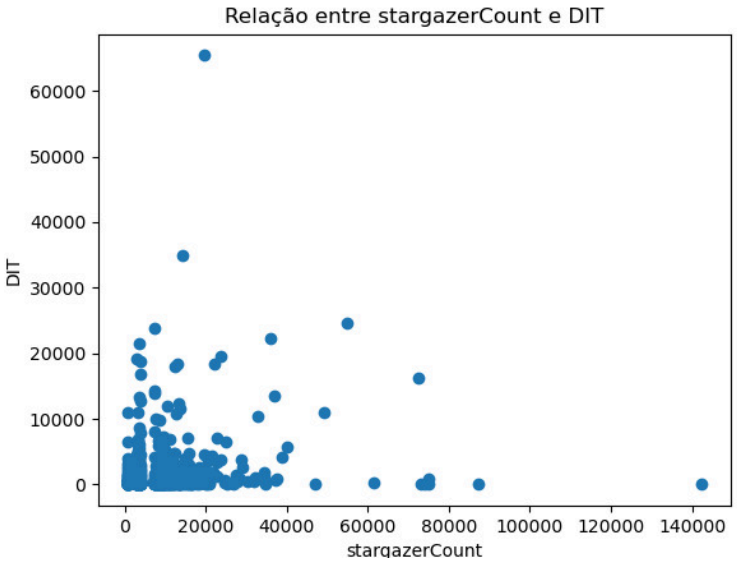

Figura 2 - Gráfico Estrelas X DIT

Tal gráfico representa que quanto maior a popularidade de um repositório, menor tende a ser a profundidade da árvore de herança de seu sistema.

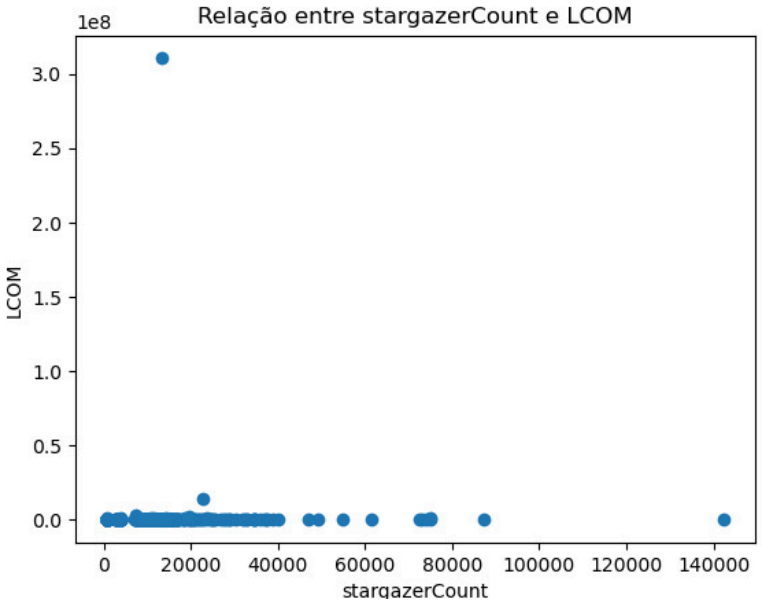

Figura 3 - Gráfico Estrelas X LCOM

Tal gráfico representa que entre os gráficos mais populares, a diferença de popularidade não impacta na análise da falta de coesão entre os métodos.

A partir da definição de qualidade definida pelo projeto usando as 3 métricas, analisa-se que entre os 1000 repositórios mais populares, a popularidade é concorrente com maiores padrões de qualidade.

3.2 Maturidade x Qualidade

- **H2:**

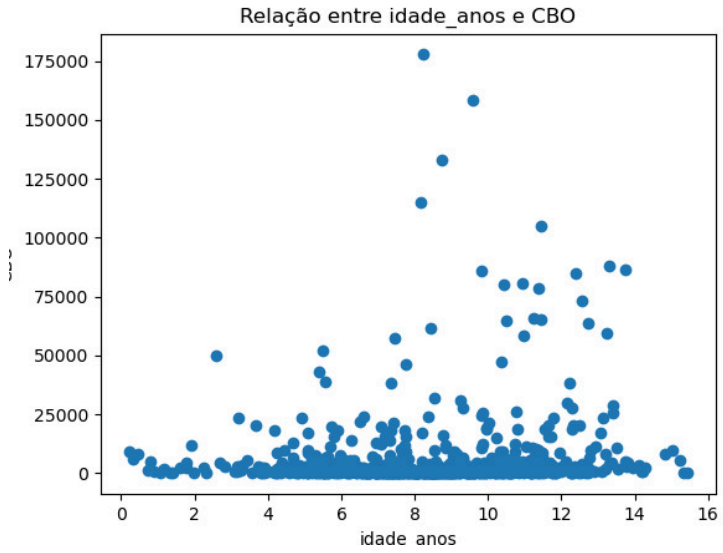

Figura 4 - Gráfico Anos X CBO

Tal gráfico representa que a maturidade de um repositório não impacta na análise do acoplamento entre os objetos de seu sistema.

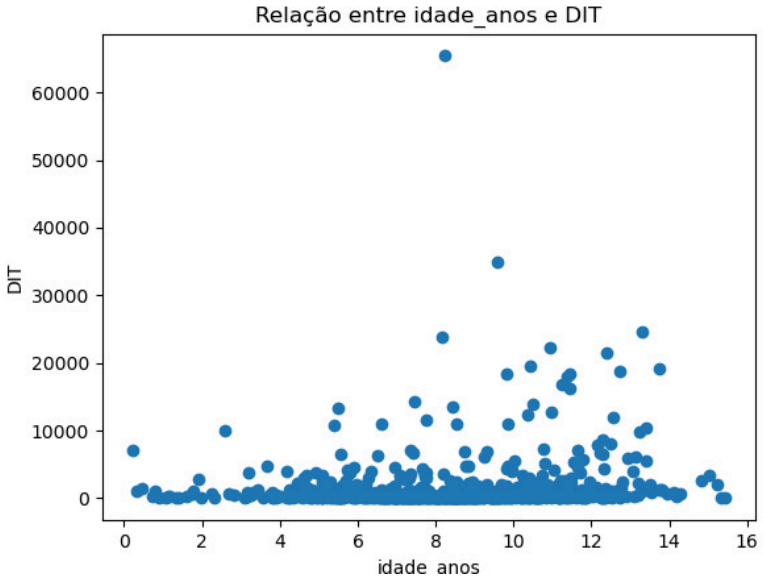

Figura 5 - Gráfico Anos X DIT

Tal gráfico representa que a maturidade de um repositório não impacta na análise da profundidade da árvore de herança de seu sistema.

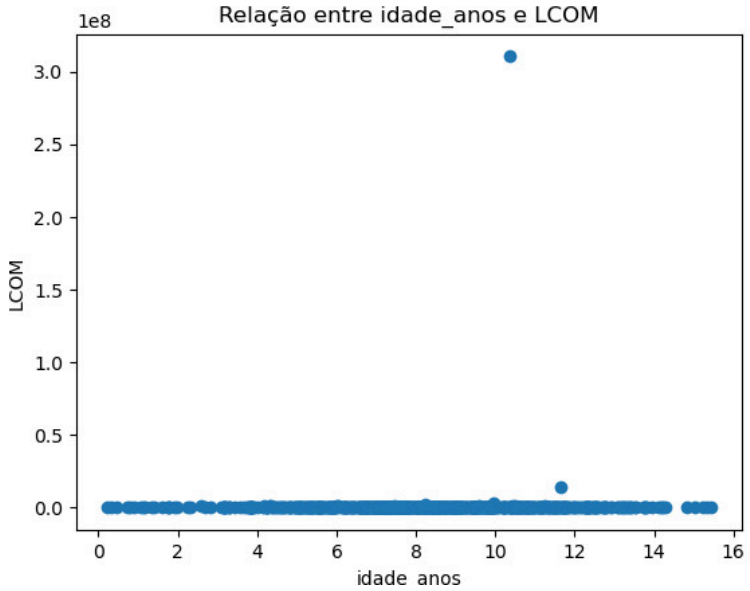

Figura 6 - Gráfico Anos X LCOM

Tal gráfico representa que a maturidade de um repositório não impacta na análise da falta de coesão entre os métodos de seu sistema.

A partir da definição de qualidade definida pelo projeto usando as 3 métricas, analisa-se que a maturidade de um repositório não possui grande impacto nos seus padrões de qualidade.

3.3 Atividade x Qualidade

- **H3:**

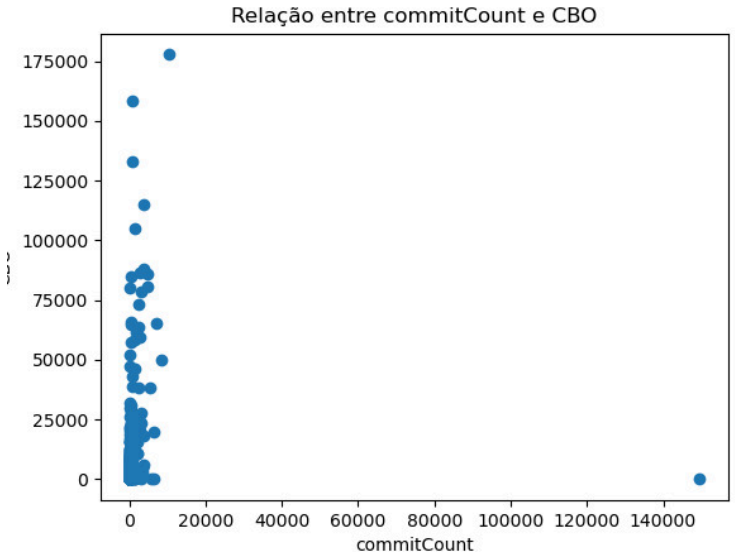

Figura 7 - Gráfico Commits X CBO

Tal gráfico representa que a atividade de um repositório não impacta na análise do acoplamento entre os objetos de seu sistema.

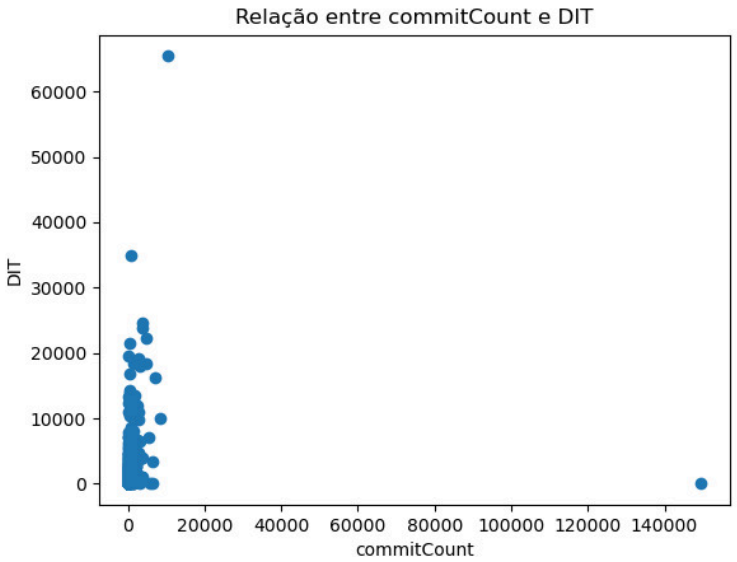

Figura 8 - Gráfico Commits X DIT

Tal gráfico representa que a atividade de um repositório não impacta na análise da profundidade da árvore de herança de seu sistema.

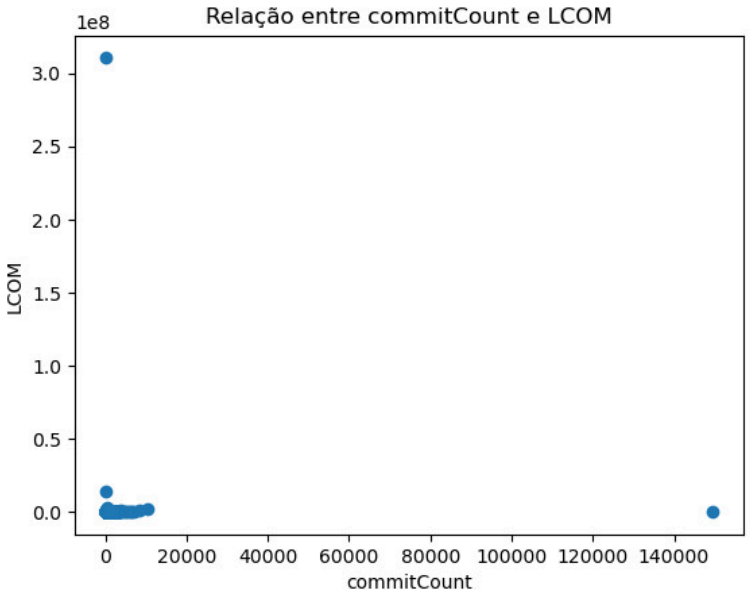

Figura 9 - Gráfico Commits X LCOM

Tal gráfico representa que a atividade de um repositório não impacta na análise da falta de coesão entre os métodos de seu sistema.

A partir da definição de qualidade definida pelo projeto usando as 3 métricas, analisa-se que a atividade de um repositório não possui grande impacto nos seus padrões de qualidade.

3.4 Tamanho x Qualidade

- **H4:**

4. Discussão

Diante dos resultados obtidos, podemos discutir se hipóteses formuladas são verídicas diante dos dados analisados.

4.1 Quanto maior a popularidade do repositório, maior a suas características de qualidade

Pelos resultados analisa-se que repositórios com maior popularidade tendem a apresentar uma qualidade mais elevada, o que pode influenciar tanto as decisões dos desenvolvedores ao escolherem projetos para contribuir quanto dos usuários ao selecionarem soluções para seus problemas.

4.2 Quanto maior a maturidade de um repositório, maior suas características de qualidade

Pelos resultados analisa-se que não há um impacto significativo na relação entre a maturidade de um repositório e suas características de qualidade, sugerindo que a idade do repositório por si só não é um indicador confiável de sua qualidade. Faz-se necessária então o uso de outra métrica para a análise da qualidade de código.

4.3 Os repositórios mais ativos tenham indicadores mais elevados de qualidade

Pelos resultados analisa-se que não há um impacto significativo na relação entre a atividade dos repositórios e seus indicadores de qualidade sugerindo que a quantidade de atividade por si só não é um indicador confiável de qualidade. Embora a participação ativa da comunidade seja importante e possa influenciar positivamente a qualidade de um repositório, outros fatores devem ser considerados e essa métrica se torna inviável para a análise da qualidade de um repositório.

4.4 Não teria relação entre tamanho do repositório e característica de qualidade

Referências

Gil C. A.(2002) “Como Elaborar Projetos de Pesquisa”, In: Editora Atlas, Brasil.

Olivia A.(s.d) “4 características da pesquisa quantitativa”,https://www.questionpro.com/blog/pt-br/caracteristicas-da-pesquisa-quantitativa/..