Pontifícia Universidade Católica de Minas Gerais

Laboratório de Experimentação de Software

Um estudo das características de qualidade de sistema java

Giovanni Bogliolo Sirihal Duarte
Luiz Gustavo Mendes Santos
Pedro Ramos Vidigal

Belo Horizonte
2024

1. Introdução

Este relatório se propõe a investigar a relação entre a qualidade de código e diversos atributos de 1000 repositórios Java de código aberto com maior popularidade. Para isso, serão analisadas métricas de qualidade de código, como acoplamento, coesão e profundidade da árvore de herança, em conjunto com características como número de estrelas, idade do repositório, frequência de releases e tamanho do código.

A partir dessa análise, busca-se responder a questões como: os repositórios mais populares tendem a ter melhor qualidade de código? A maturidade de um projeto influencia sua qualidade? A frequência de atualizações e o tamanho do código estão relacionados à qualidade?

Com o intuito de responder a essas perguntas, este estudo empregará técnicas de mineração de dados e análise estática de código utilizando a ferramenta CK do java, explorando a relação entre diferentes aspectos quantitativos dos repositórios e sua qualidade.

É importante ressaltar que todos os dados foram coletados na data de envio deste relatório e, portanto, podem não refletir o estado atual dos repositórios analisados.

1.1. Hipóteses

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

R: Repositórios com maior número de estrelas tendem a apresentar melhores métricas de qualidade de código, como menor CBO e LCOM , refletindo um design mais modular e manutenível que atrai mais desenvolvedores.

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

R: Repositórios maduros podem ter passado por mais refatorações e aprimoramentos ao longo do tempo, o que pode levar a métricas de qualidade melhores, como menor DIT e LCOM, demonstrando um esforço contínuo para manter o código limpo e organizado.

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

R: Repositórios com maior número de releases podem indicar um desenvolvimento ativo e uma busca por melhorar o código, o que pode implicar em melhores métricas de qualidade, como menor CBO e LCOM, devido à atenção constante à manutenção do código.

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

R: Repositórios com um número elevado de linhas de código podem demonstrar um esforço intencional para manter a qualidade. Métricas como um DIT baixo podem indicar uma estrutura bem organizada e modular.

2. Metodologia

Inicialmente, foram coletados dados de 1000 repositórios Java com maior número de estrelas. Para viabilizar essa etapa, um script em Python, utilizando a linguagem de consulta GraphQL, foi implementado para minerar os dados dos repositórios e armazená-los na pasta de *input*.

Em seguida, a ferramenta CK foi empregada para realizar análises estáticas de código nos repositórios coletados. Essa ferramenta permitiu calcular métricas de qualidade de código ao nível de classe e método, como *Coupling Between Objects* (CBO), *Depth of Inheritance Tree* (DIT) e *Lack of Cohesion of Methods* (LCOM). Os resultados individuais dessas análises foram armazenados em arquivos .csv na pasta de *output*, enquanto os dados originais na pasta de *input* foram deletados para otimizar o espaço de armazenamento.

Posteriormente, um código em Python foi desenvolvido para extrair as métricas de qualidade dos arquivos de resultados individuais e consolidá-las em um único arquivo de resultados gerais. Paralelamente, outro script Python foi utilizado para coletar métricas adicionais dos repositórios, como maturidade, atividade (número de releases) e popularidade (número de estrelas).

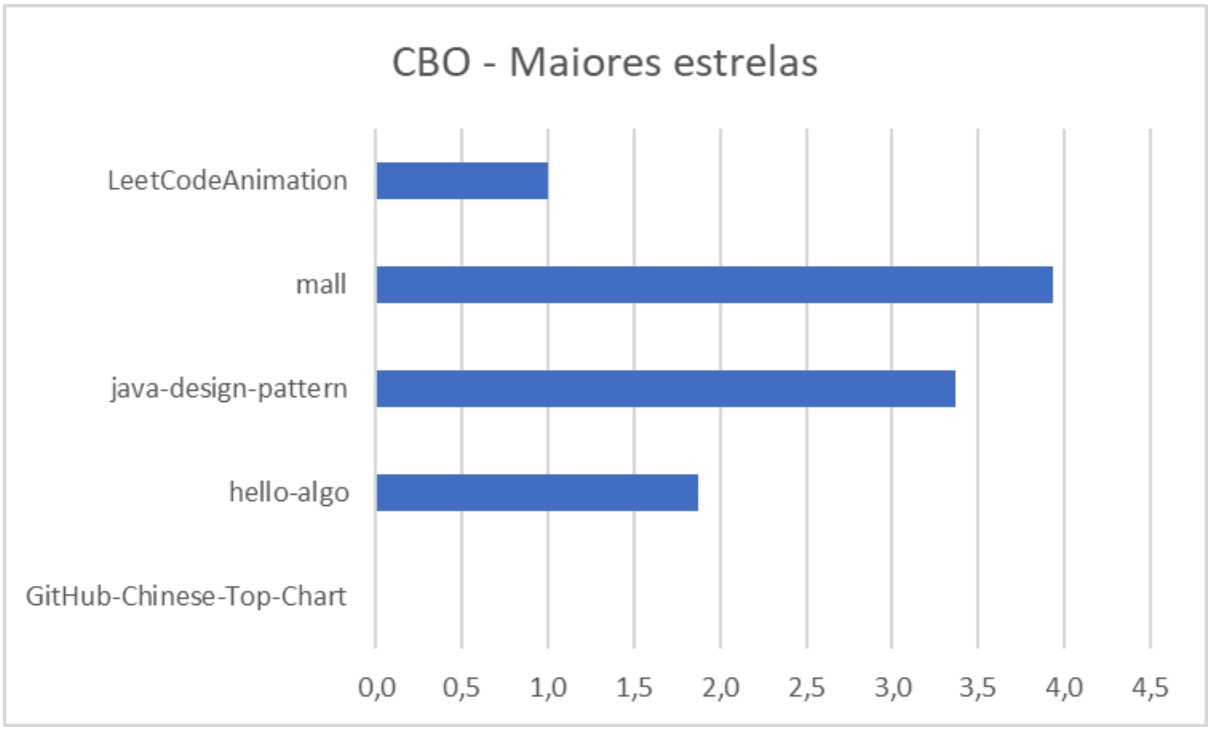
Com base nos dados consolidados, foram construídos gráficos boxplot e histogramas para visualizar e analisar a relação entre as características de qualidade e os demais atributos dos repositórios. Essa etapa permitiu identificar padrões e tendências nos dados, contribuindo para a formulação de hipóteses e conclusões sobre o tema de pesquisa.

3. Resultados obtidos

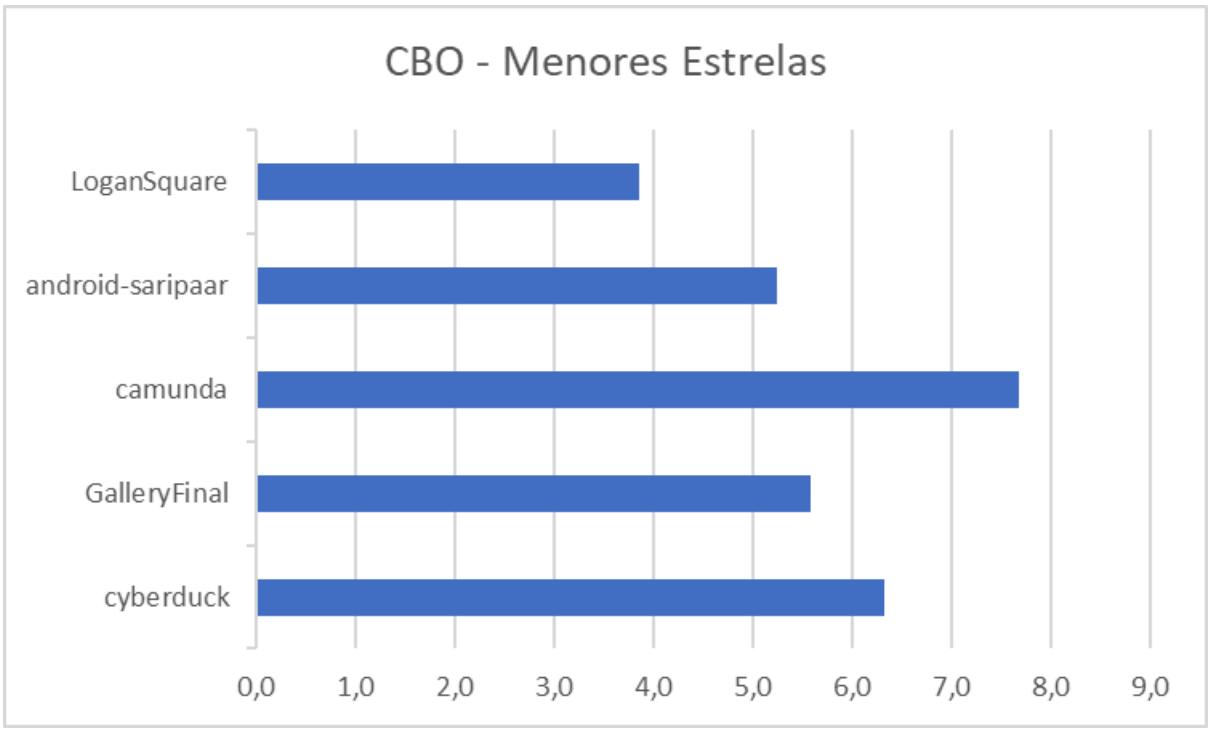
Para analisar os 1000 repositórios, foram calculadas a média, mediana e moda dos 3 atributos de qualidade requisitados:

Métrica	Média	Mediana	Desvio Padrão
LCOM	115.97	22.90	1789.01
DIT	1.46	1.40	0.36
CBO	5.29	5.23	1.81

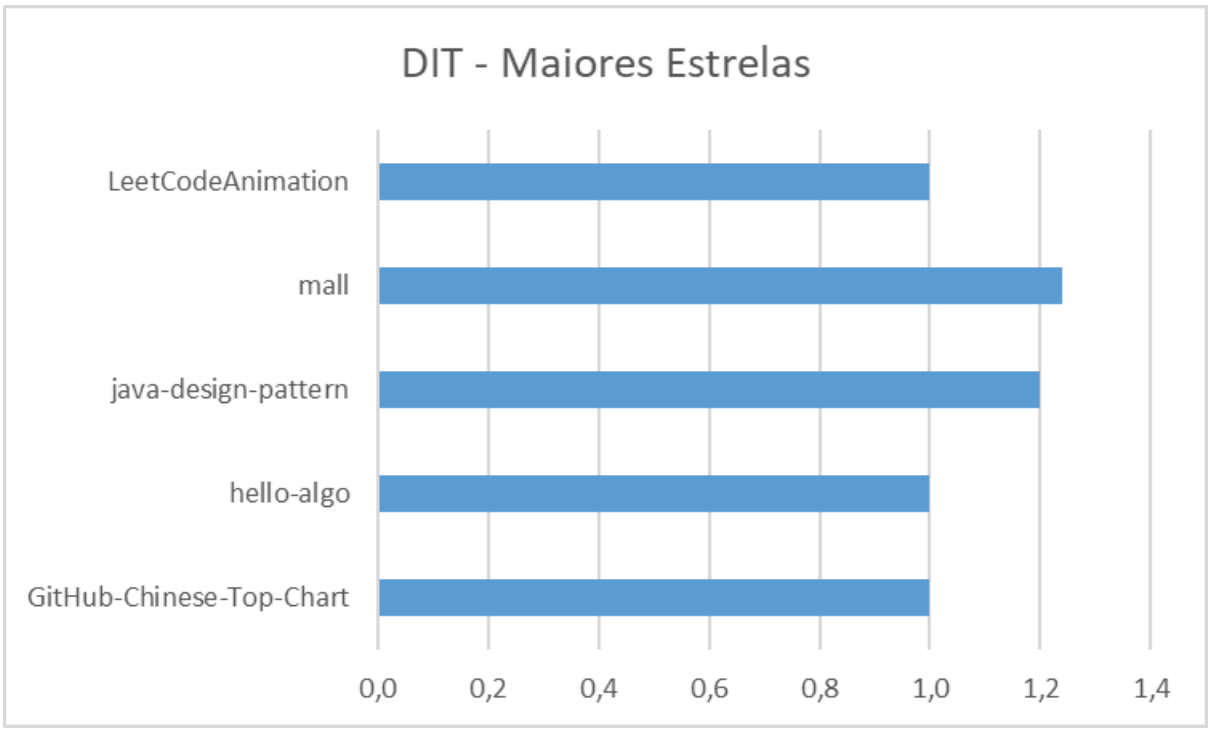
RQ 01.1: 98919 GitHub-Chinese-Top-Chart || 95671 hello-algo || 89400 java-design-pattern || 77356 mall || 75326 LeetCodeAnimation

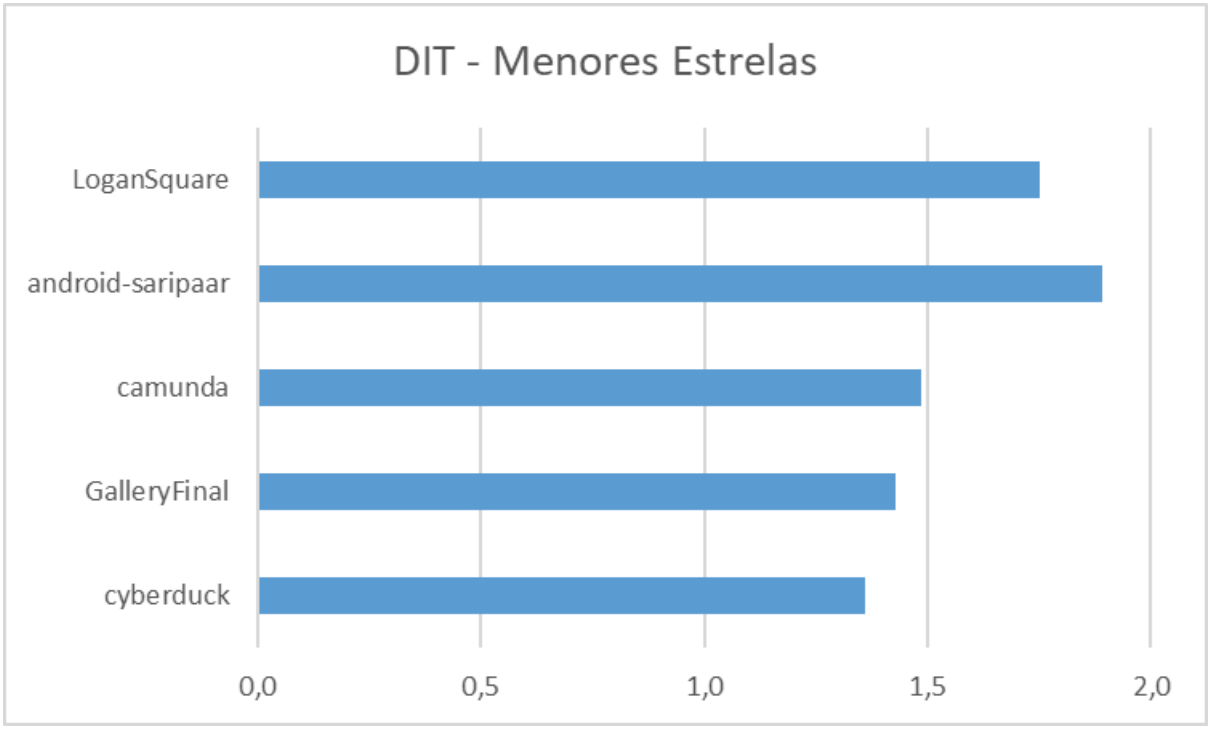


3229 cyberduck || 3228 GalleryFinal || 3224 camunda || 3223 bandroid-saripaar || 3211 LoganSquare

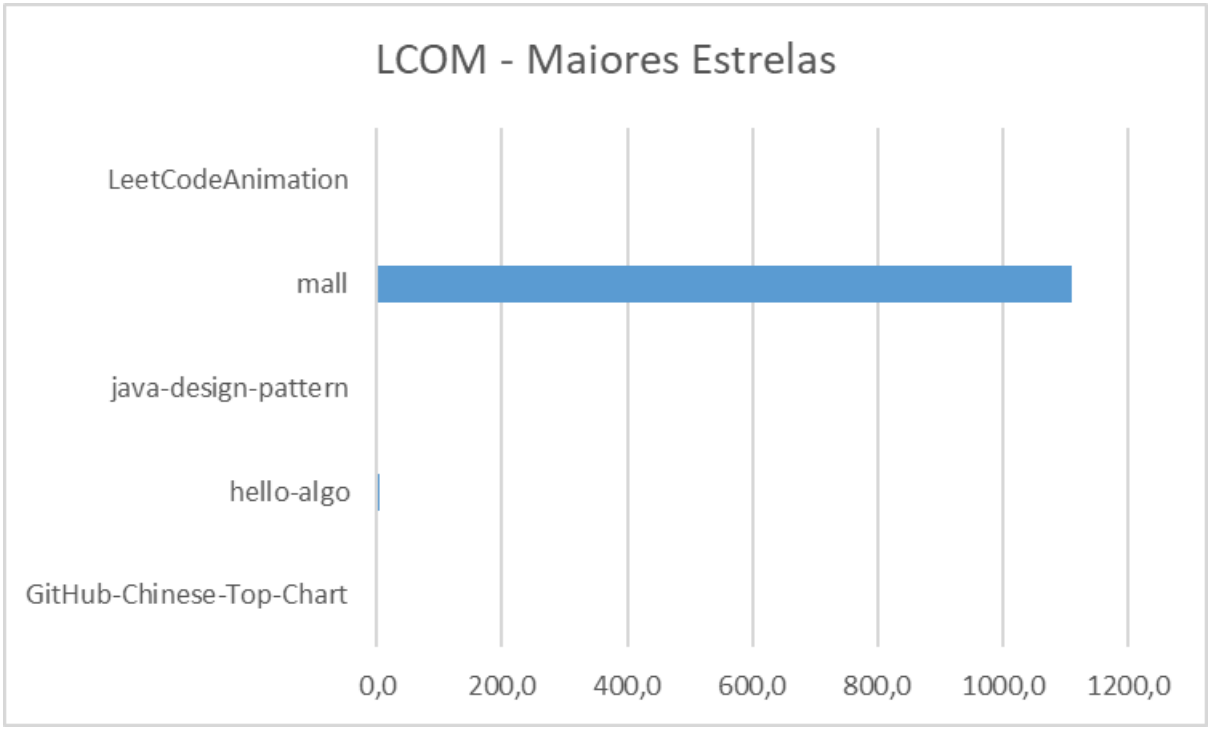


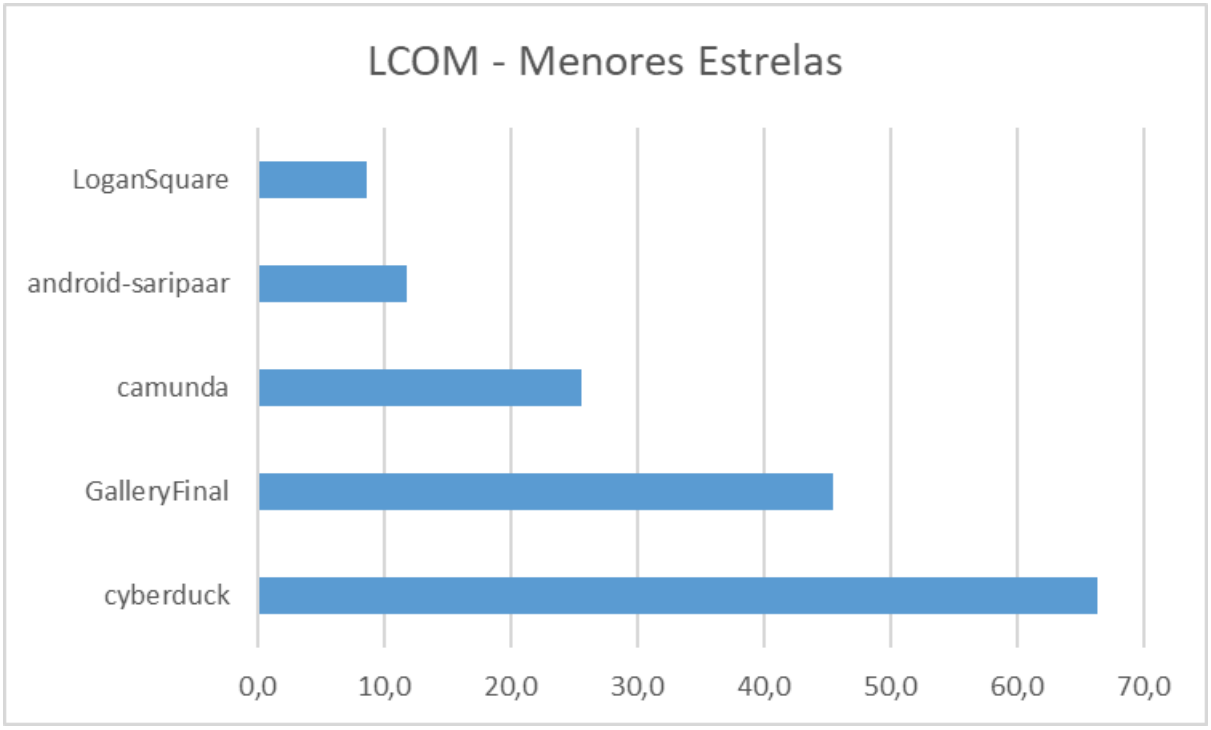
RQ 01.2.




DIT - Menores Estrelas
LoganSquare
android-saripaar
camunda
GalleryFinal
cyberduck
0,0
0,5
1,0
1,5
2,0

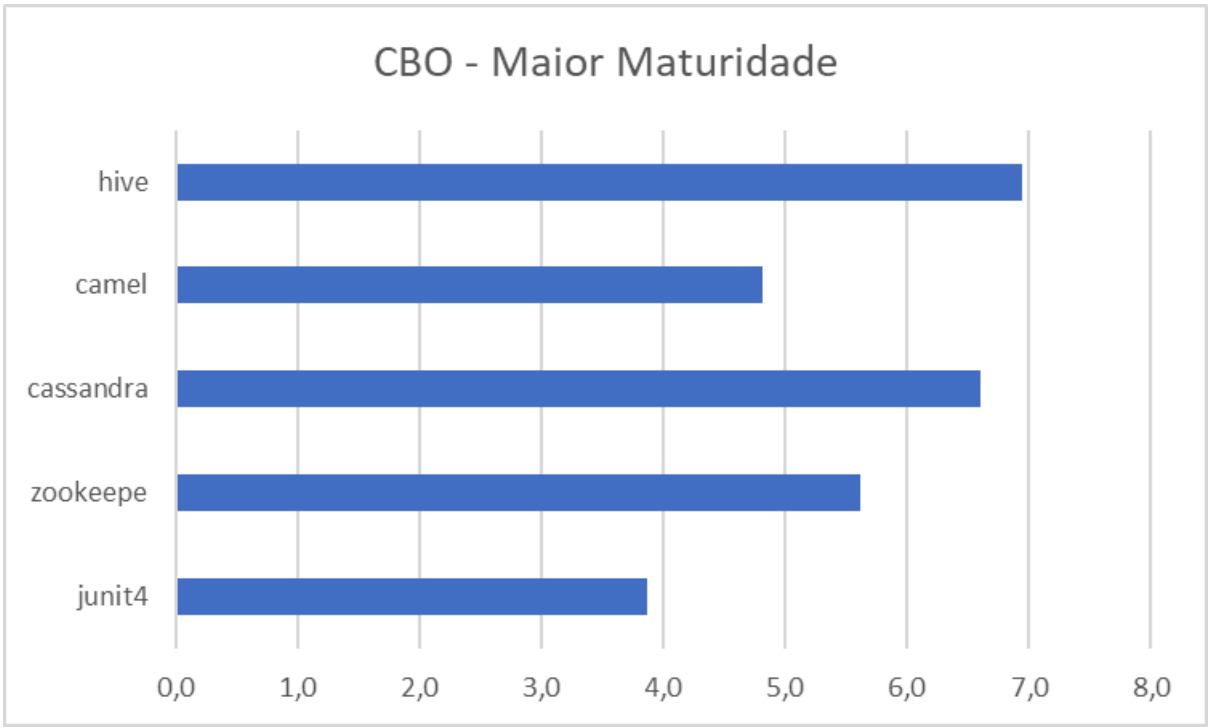
RQ 01.3.


LCOM - Maiores Estrelas
LeetCodeAnimation
mall
java-design-pattern
hello-algo
GitHub-Chinese-Top-Chart
0,0
200,0
400,0
600,0
800,0
1000,0
1200,0

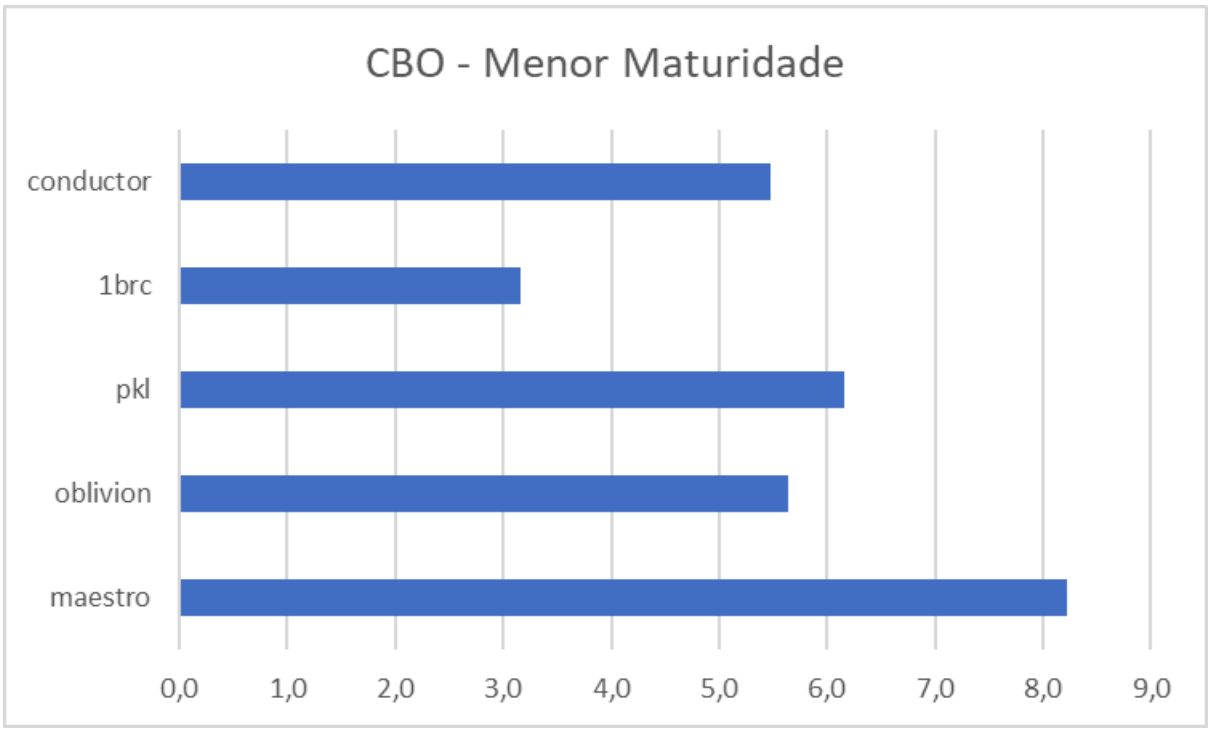


RQ 02.1.

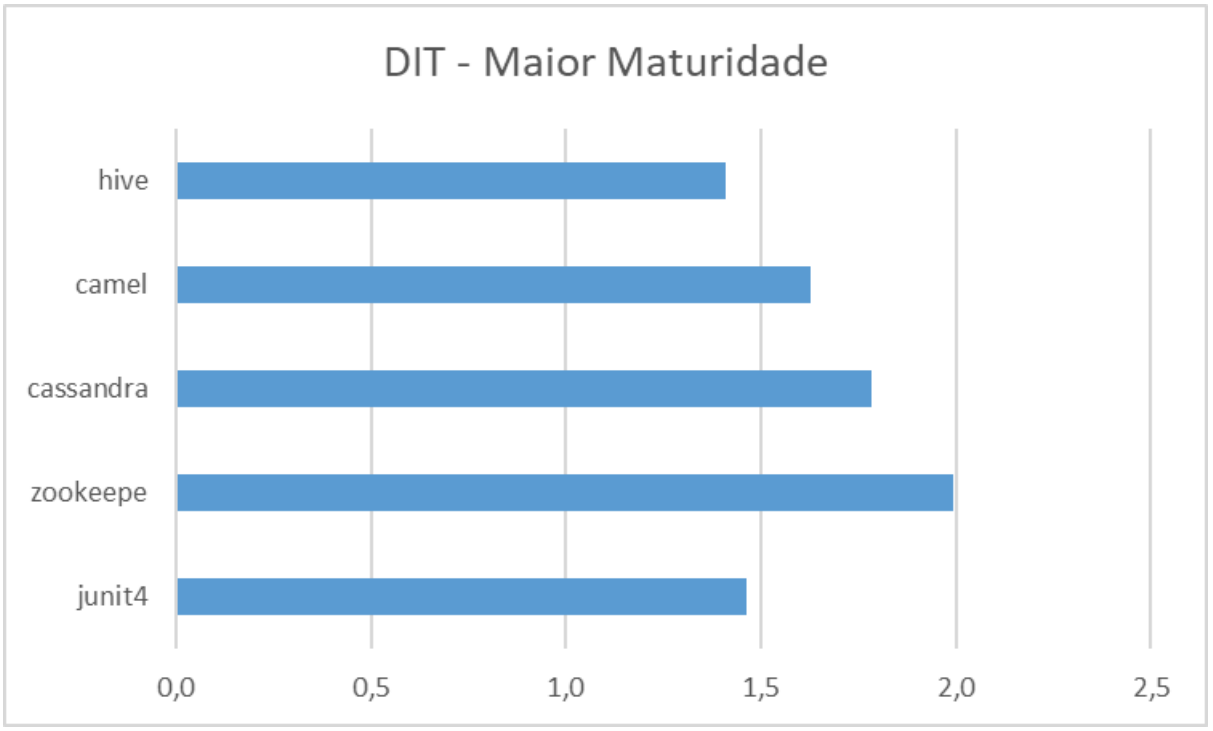
15.70 junit4 || 15.50 zookeepe || 15.35 cassandra || 15.35 camel || 15.35 hive

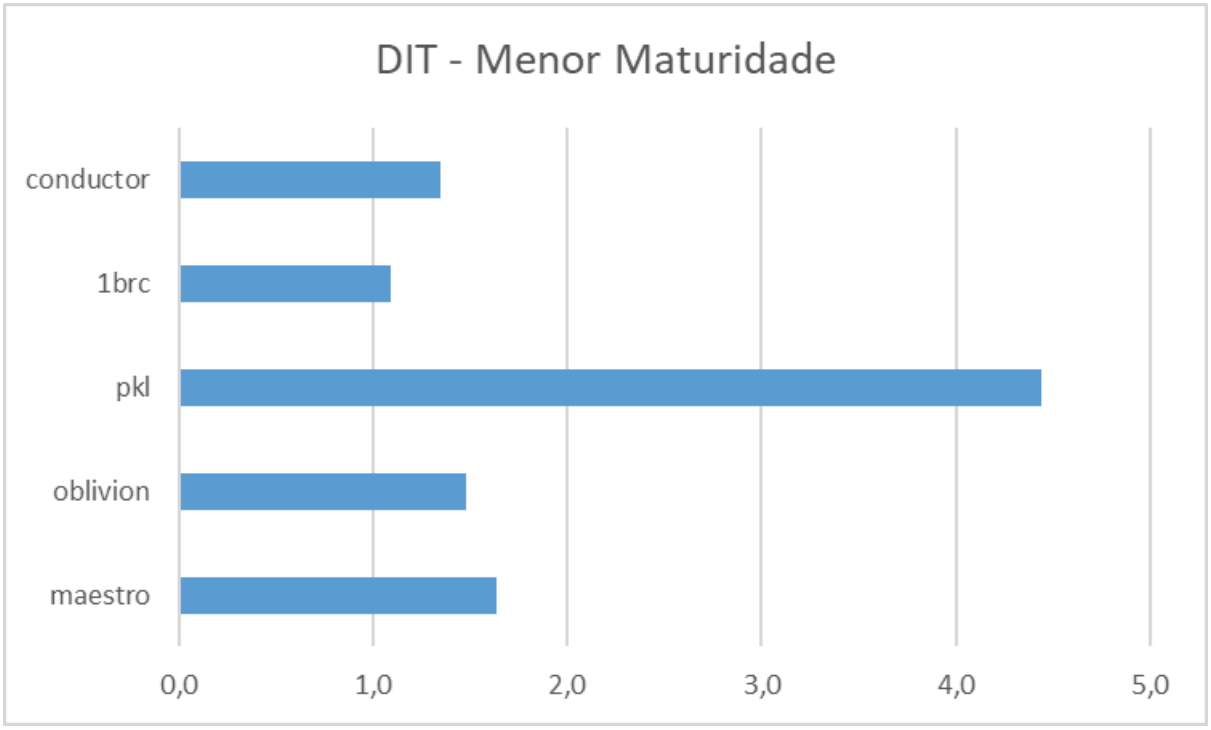


0.79 maestro || 0.73 oblivion || 0.67 pkl || 0.62 1brc || 0.43 conductor

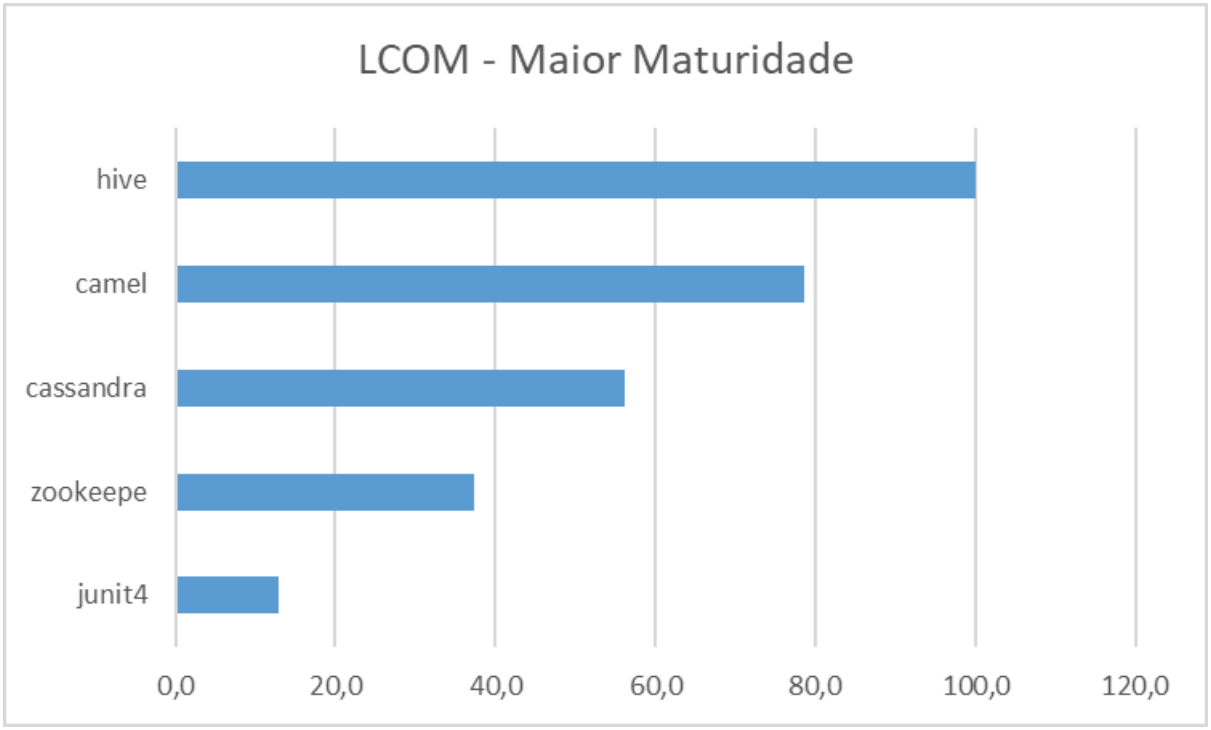


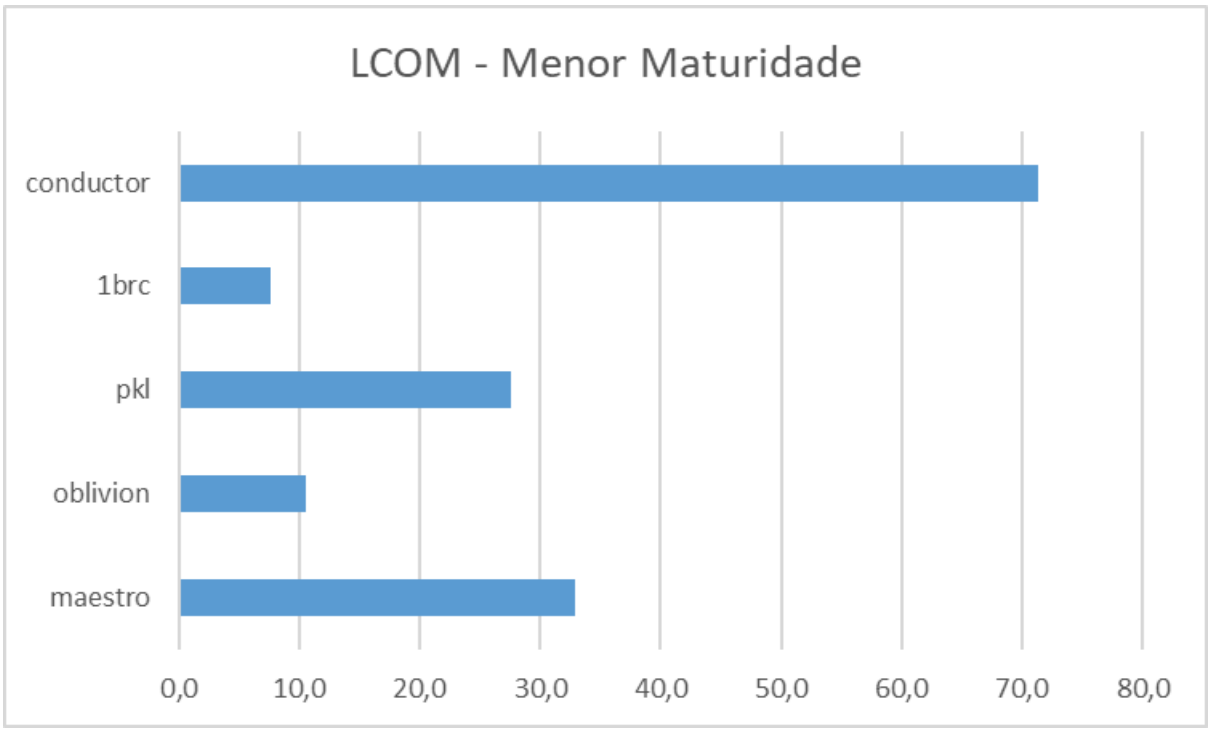
RQ 02.2.




DIT - Menor Maturidade
conductor
1brc
pkl
oblivion
maestro
0,0
1,0
2,0
3,0
4,0
5,0

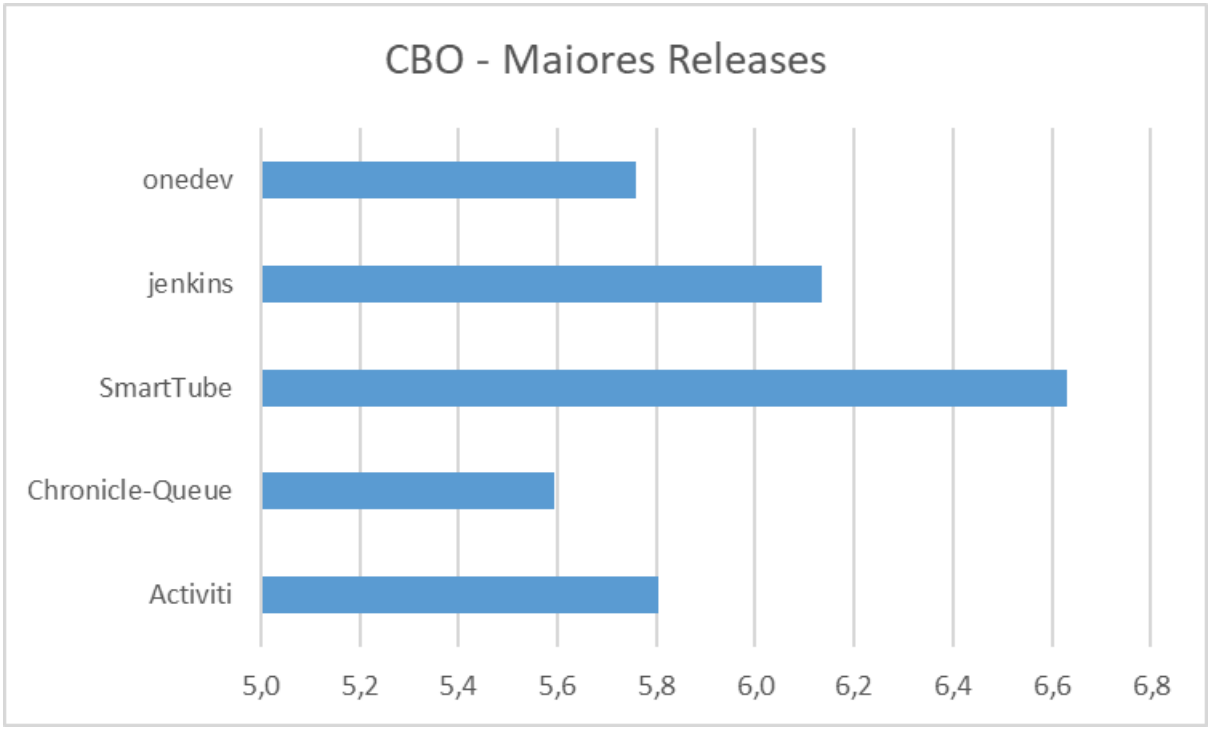
RQ 02.3.


LCOM - Maior Maturidade
hive
camel
cassandra
zookeepe
junit4
0,0
20,0
40,0
60,0
80,0
100,0
120,0

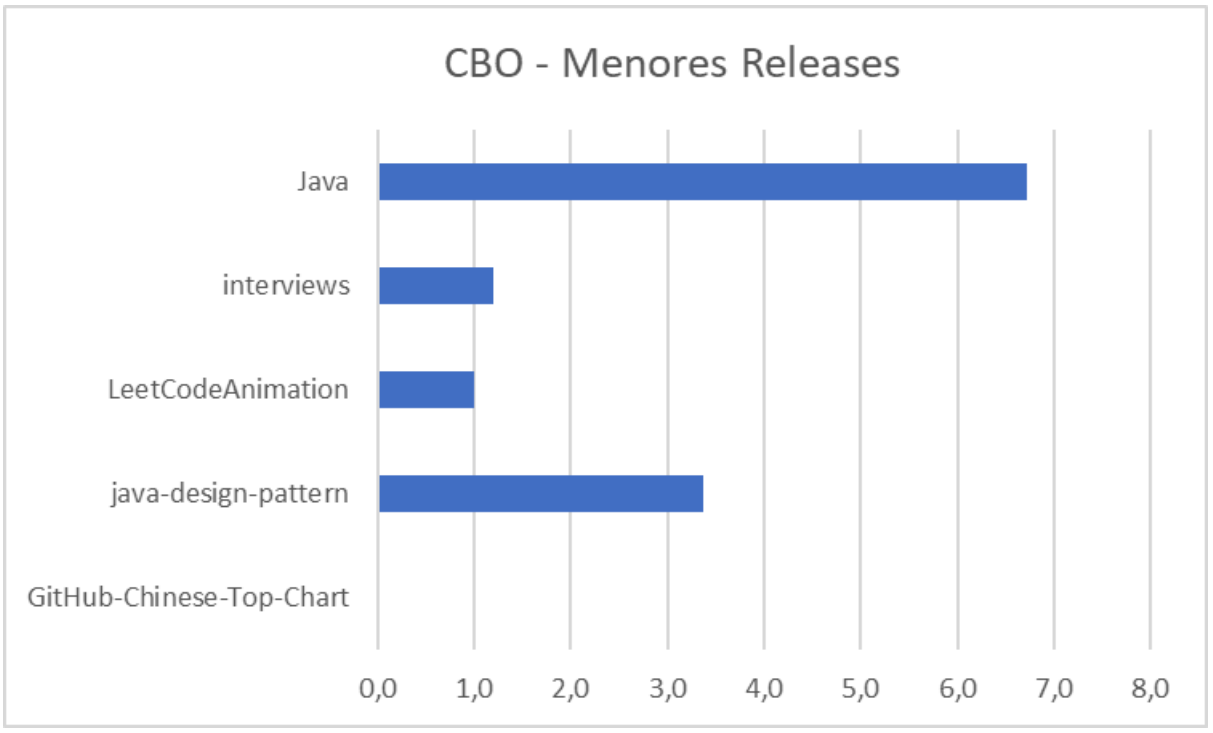


RQ 03.1.

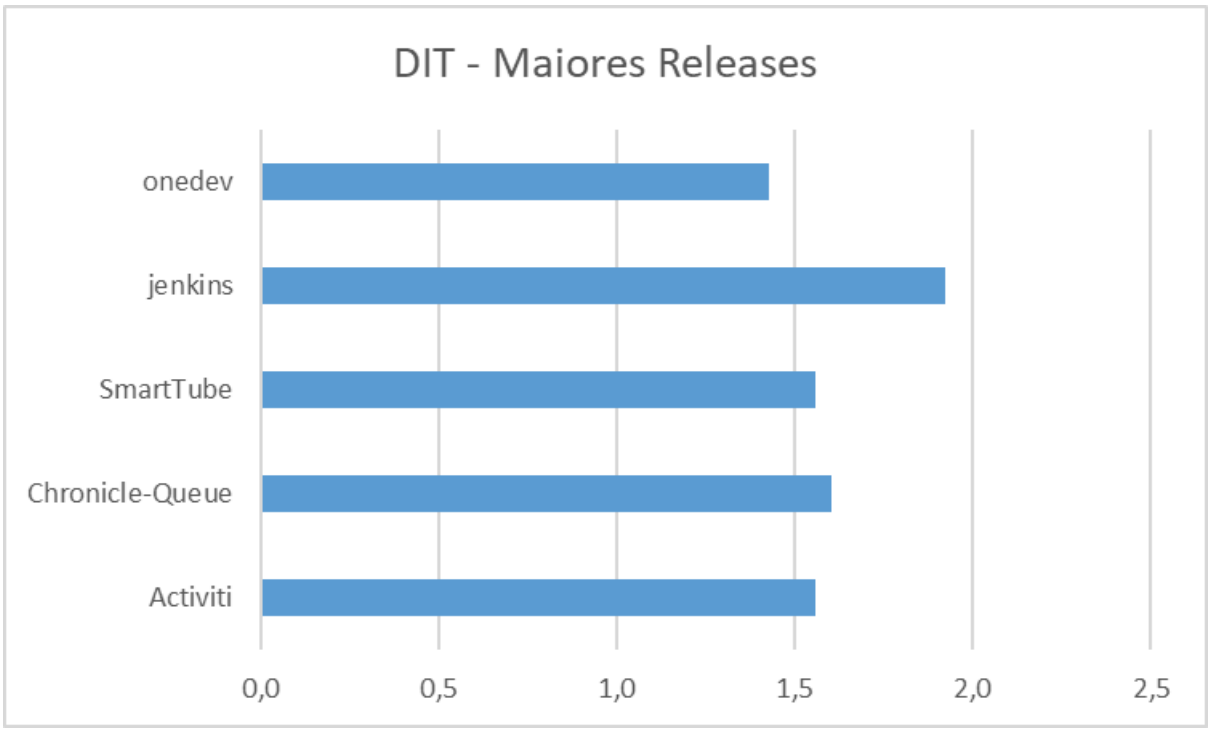
969 Activiti || 511 Chronicle-Queue || 495 SmartTube || 367 jenkins || 355 onedev

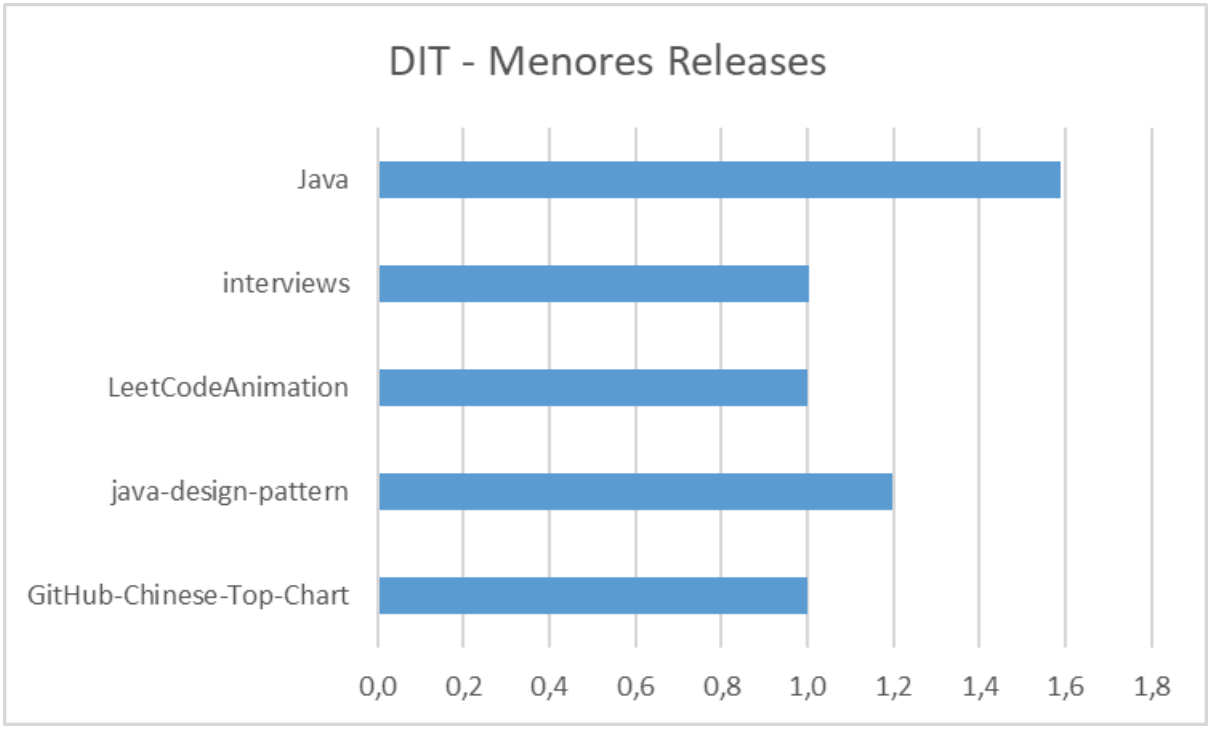


0 GitHub-Chinese-Top-Chart || 0 java-design-pattern || 0 LeetCodeAnimation || 0 interviews || 0 Java

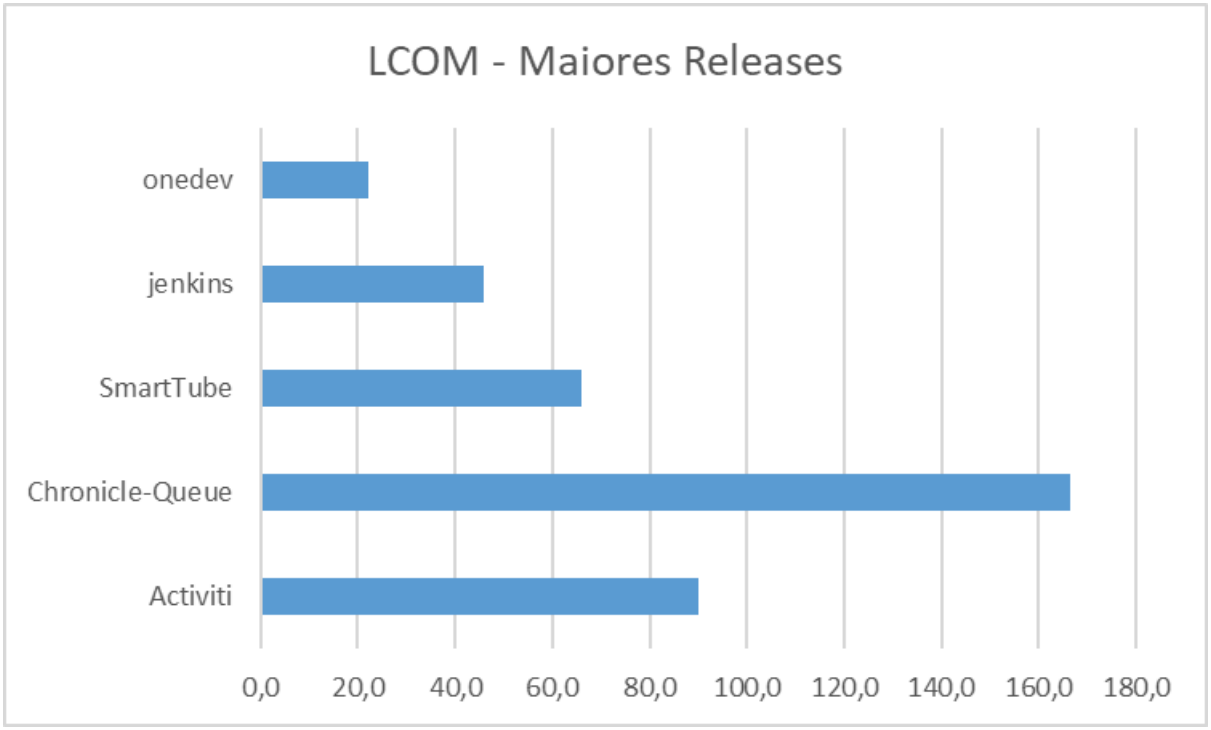


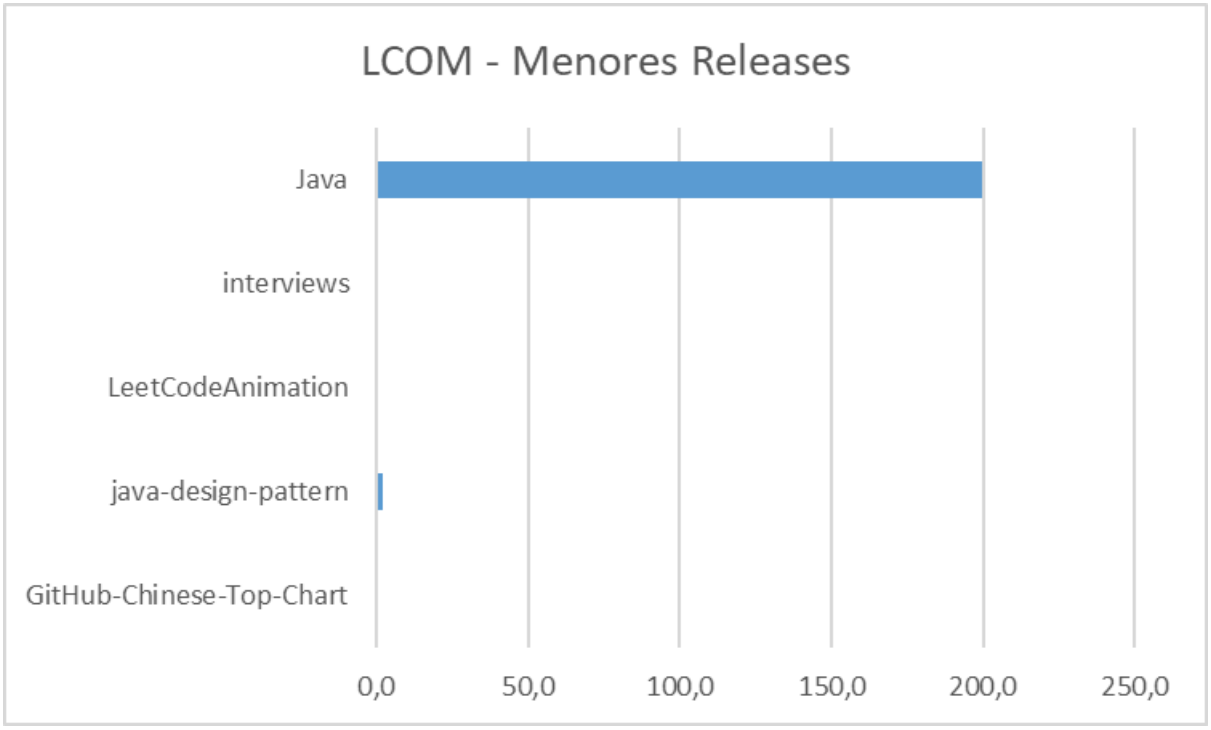
RQ 03.2.




DIT - Menores Releases
Java
interviews
LeetCodeAnimation
java-design-pattern
GitHub-Chinese-Top-Chart
0,0
0,2
0,4
0,6
0,8
1,0
1,2
1,4
1,6
1,8

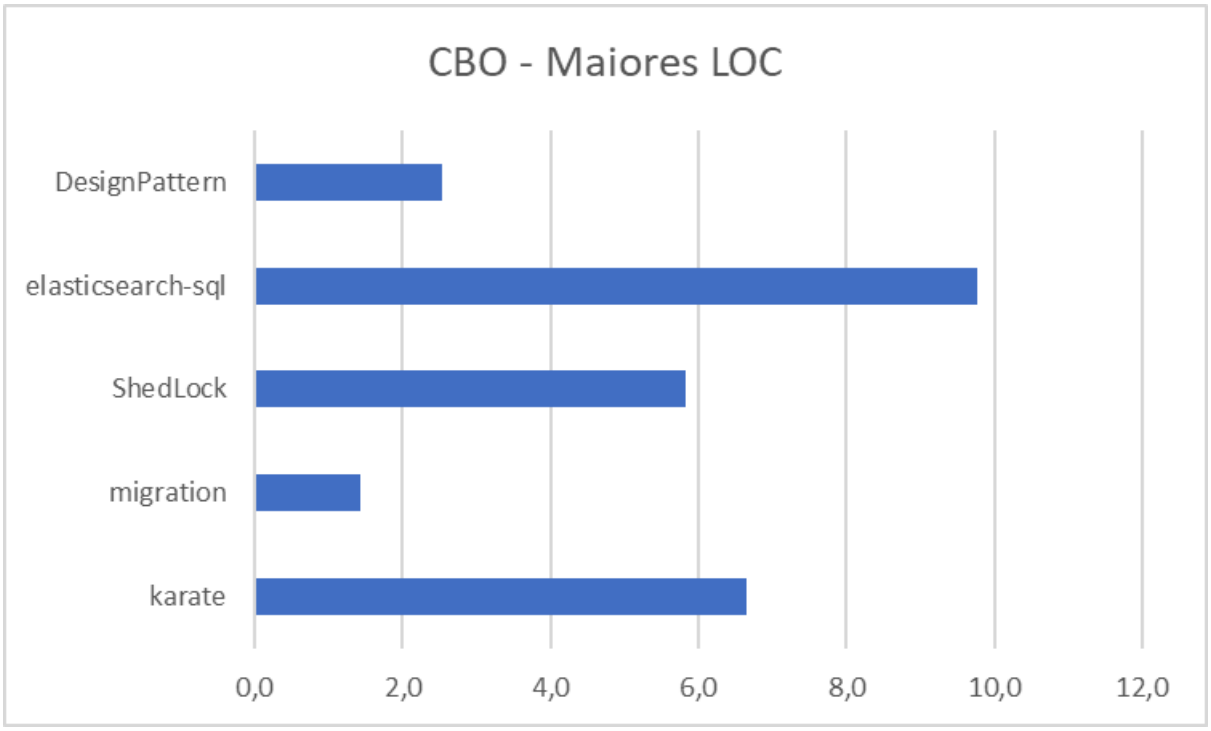
RQ 03.3.


LCOM - Maiores Releases
onedev
jenkins
SmartTube
Chronicle-Queue
Activiti
0,0
20,0
40,0
60,0
80,0
100,0
120,0
140,0
160,0
180,0

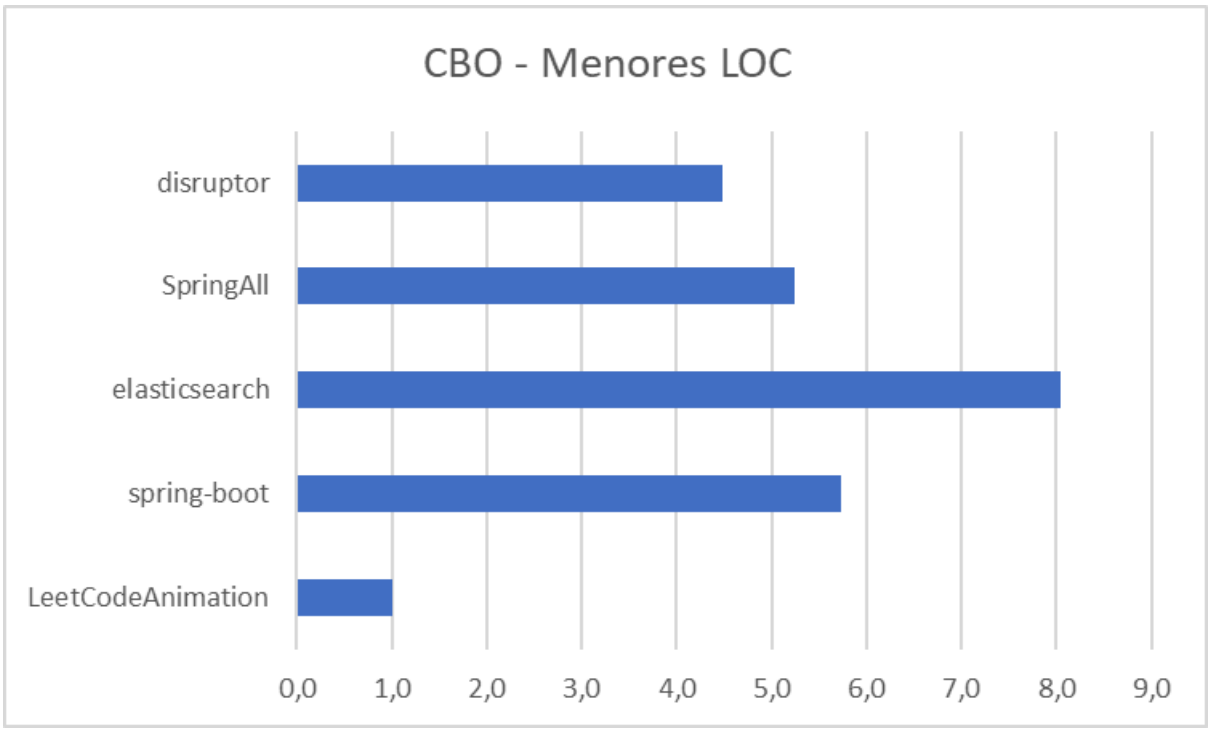


RQ 4.1.

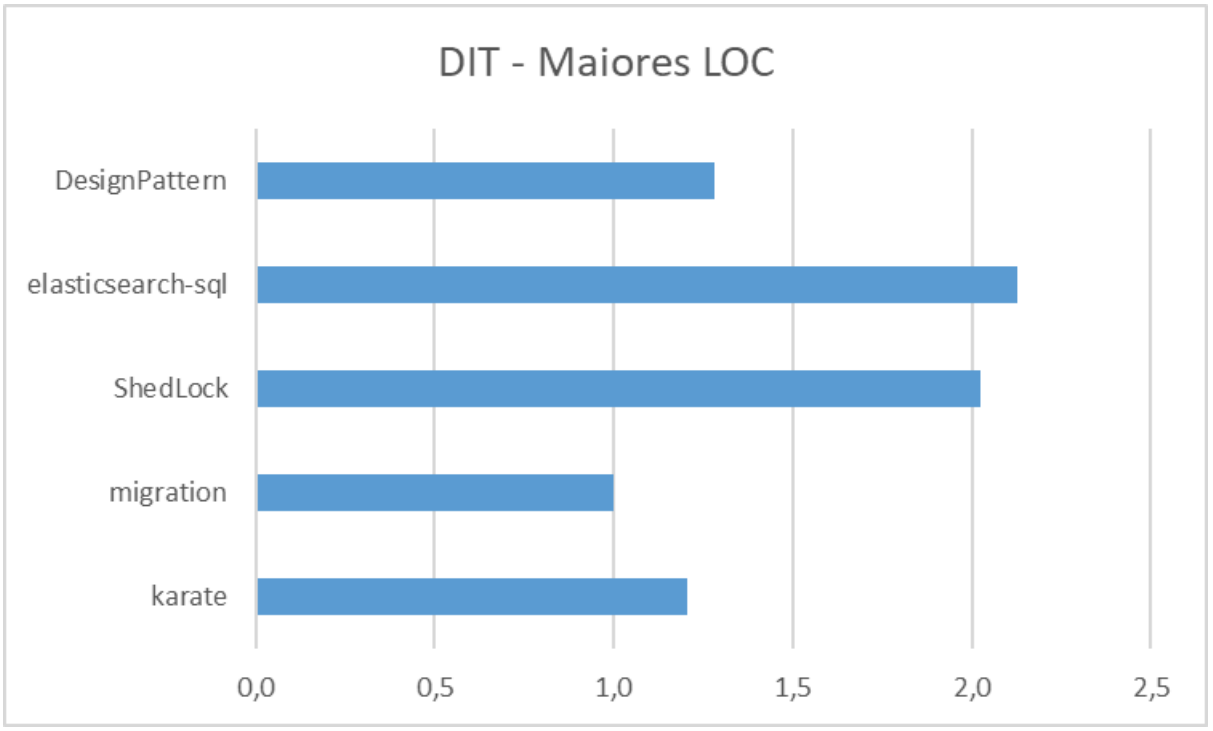
3668 karate || 1486 migration || 1243 ShedLock || 1207 elasticsearch-sql || 1163 DesignPattern

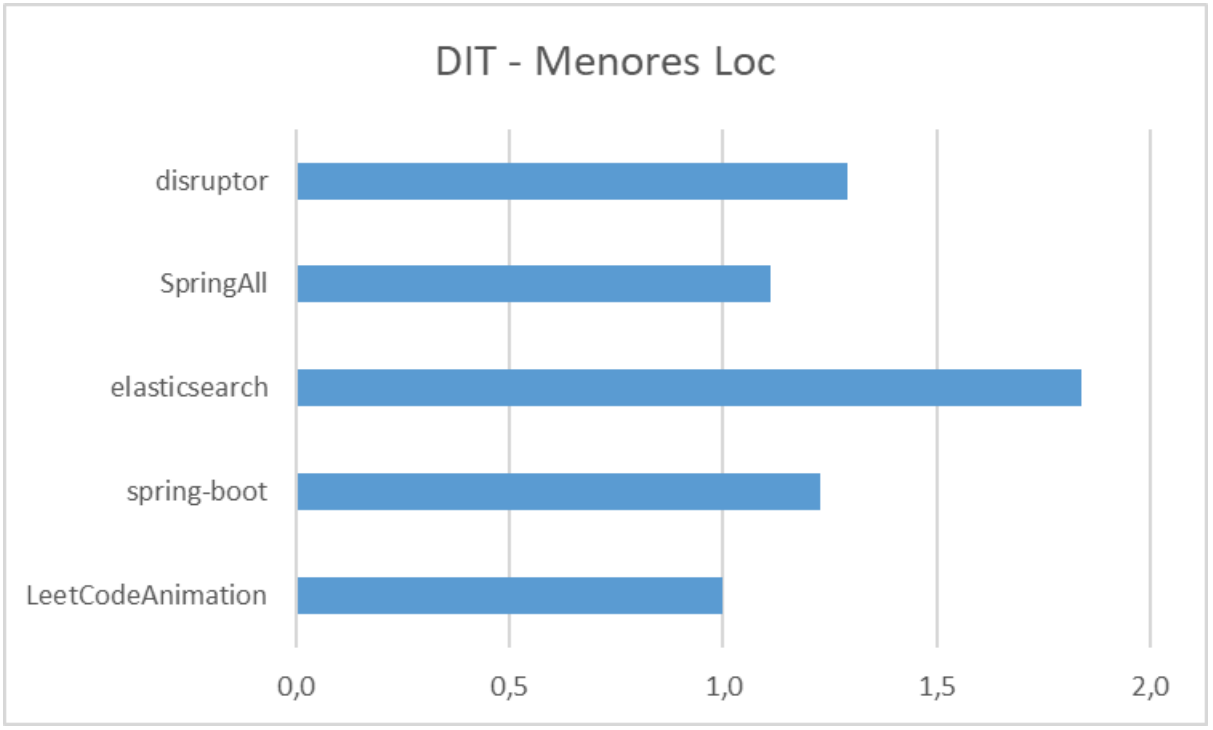


0 LeetCodeAnimation || 0 spring-boot || 0 elasticsearch || 0 SpringAll || 0 disruptor

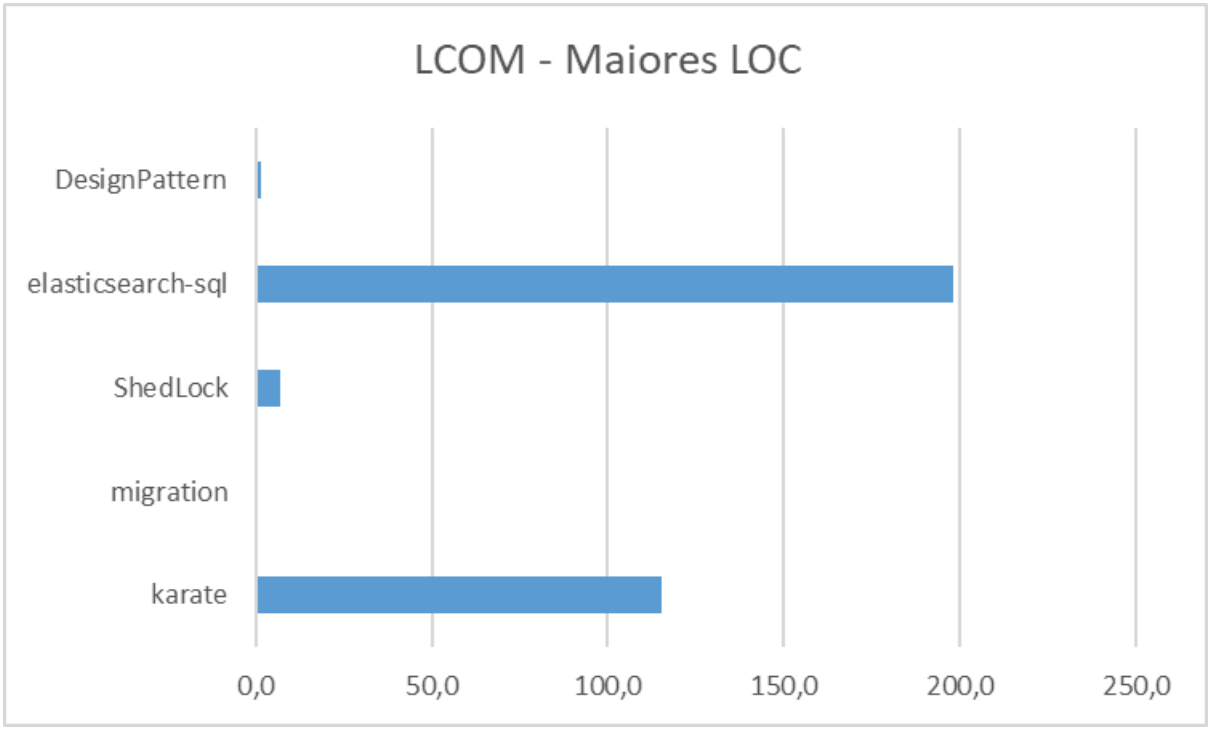


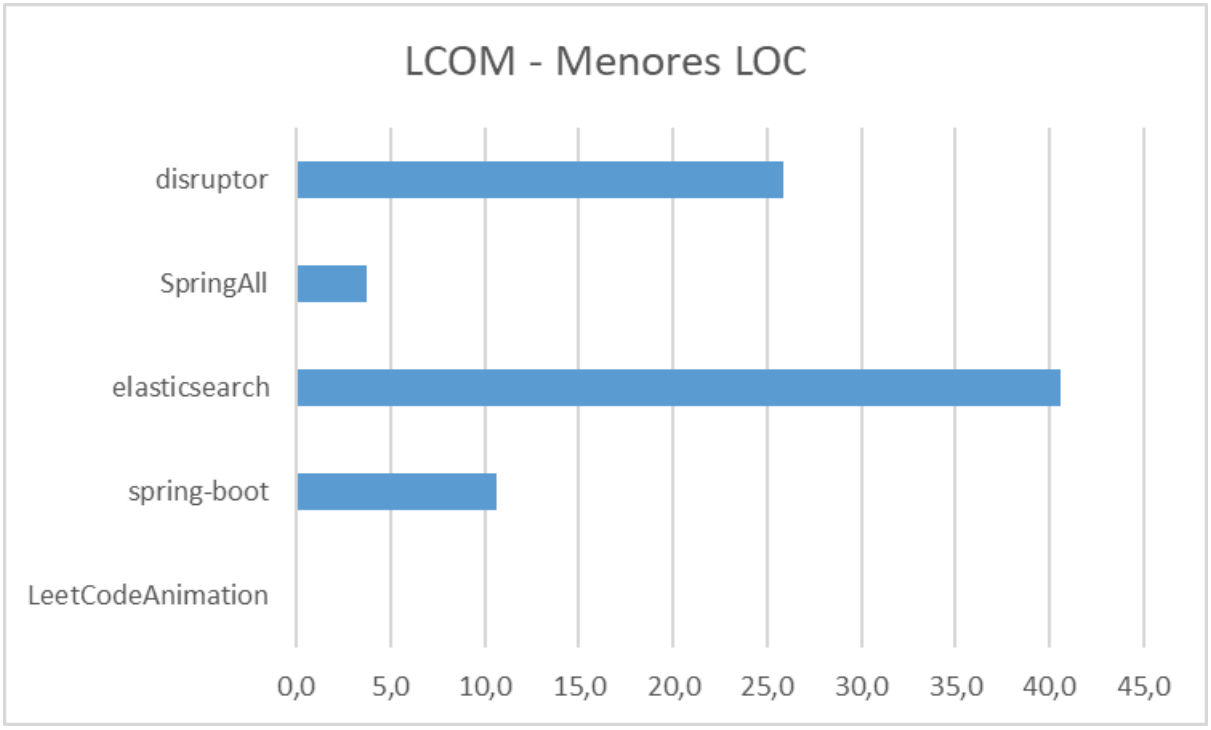
RQ 4.2.




DIT - Menores Loc
disruptor
SpringAll
elasticsearch
spring-boot
LeetCodeAnimation
0,0
0,5
1,0
1,5
2,0

RQ 4.3.


LCOM - Maiores LOC
DesignPattern
elasticsearch-sql
ShedLock
migration
karate
0,0
50,0
100,0
150,0
200,0
250,0



4. Análise dos resultados

Nesta seção, serão apresentadas as análises dos resultados obtidos para cada pergunta de pesquisa.

RQ 01.

Análise: A hipótese inicial de que repositórios mais populares tendem a apresentar melhor qualidade de código foi confirmada. Ao comparar os 5 repositórios mais populares com os 5 menos populares, foi observado que as métricas de qualidade de código (CBO, LCOM, DIT) são consistentemente melhores nos primeiros. Isso sugere que a popularidade pode estar associada a um maior cuidado com a qualidade do código, possivelmente devido a uma maior atenção da comunidade e contribuições de desenvolvedores experientes.

RQ 02.

Análise: A hipótese 2, que sugere que repositórios maduros passam por mais refatorações e aprimoramentos, foi parcialmente confirmada pelos resultados. Ao comparar os 5 repositórios mais antigos com os 5 mais novos, foi notado que a maturidade de um projeto é a métrica coletada que menos interfere nos fatores de qualidade, ao analisar os gráficos podemos perceber que tanto os top 5 com menos maturidade quanto os top 5 com maior maturidade tem atributos de qualidade similares, com o top 5 com maior maturidade possuindo estatísticas um pouco piores.

RQ 03.

Análise: A hipótese 3, que relaciona maior atividade dos repositórios com melhor qualidade de código, se mostrou inválida. Ao comparar os 5 repositórios com maior número de releases recentes com os 5 com menor número de releases, foi constatado que os primeiros tendem a apresentar métricas de qualidade com médias inferiores (maior CBO e LCOM).. Analisamos que um alto número de releases pode levar a um aumento temporário em métricas como DIT e LCOM, possivelmente devido a mudanças frequentes e apressadas.

RQ 4.

Análise: A hipótese 4, que previa que repositórios maiores teriam desafios no gerenciamento da complexidade e, consequentemente, pior qualidade de código, foi confirmada. Ao comparar os 5 repositórios com maior número de linhas de código com os 5 com menor número, foi observado que os primeiros tendem a apresentar métricas de qualidade piores (maior CBO e LCOM). No entanto, é importante ressaltar que os repositórios com menor tamanho foram medidos como tendo 0 linhas pelo código, o que pode alterar os dados analisados.