Relatório Final - Laboratório 02

Gabriel Alejandro Figueiro Galindo, Marcelo Aguilar Araújo D'Almeida,
Philippe Roberto Dutra Chaves Vieira

1 Introdução e Hipóteses

Este estudo vai analisar aspectos da qualidade (modularidade, manutenibilidade, ou legibilidade) de repositórios de sistemas open-source desenvolvidos na linguagem Java, correlacionando-os com características do seu processo de desenvolvimento, sob a perspectiva de métricas de produto calculadas através da ferramenta CK. Antes do início deste trabalho foram propostas questões de pesquisa; e as hipóteses elaboradas para cada uma estão a seguir:

- **RQ 01:** Esperamos que repositórios com maior popularidade sejam aqueles que possuam uma maior qualidade, pois acreditamos que a comunidade não apoiaria aqueles que não possuem um mínimo de qualidade.
- **RQ 02:** A maturidade do repositório (sua idade) deve estar diretamente proporcional à sua qualidade, uma vez que repositórios mais velhos tiveram mais tempo para corrigir seus problemas.
- **RQ 03:** Quando o quesito é a atividade do repositório, acreditamos que aqueles com uma atividade relativamente grande sejam os com maior qualidade; tendo em vista que um maior número de releases significa que o sistema está sempre em atualização e, de maneira optimista, sempre melhorando sua qualidade.
- **RQ 04:** Para a métrica de tamanho, esperamos que os repositórios com uma maior qualidade sejam aqueles com uma menor quantidade de linhas, uma vez que são menos complicados e mais fácil de realizar manutenções de qualidade.

2 Metodologia

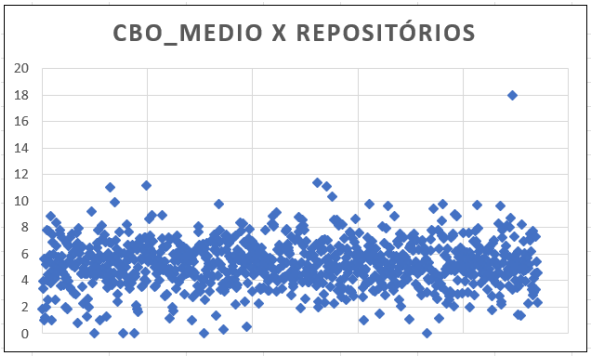
Para responder às questões da pesquisa, foi desenvolvido um código em Python que realiza a coleta de dados por meio da API GraphQL do GitHub. A busca é feita utilizando uma query que recupera os 1000 primeiros repositórios em Java com o maior número de estrelas.
Vale ressaltar que a busca dos dados e métricas foi feita com base na ferramenta CK. Esta ferramenta só funcionou com repositórios que possuíam Maven e versão do Java até a 11.
Os dados extraídos de cada repositório foram os seguintes:

- Nome do repositório
- Número de estrelas
- Linhas de código (kLOC)
- Número de releases
- Idade (em anos)
- CBO: Acoplamento entre objetos
- DIT: Profundidade da Árvore de Herança

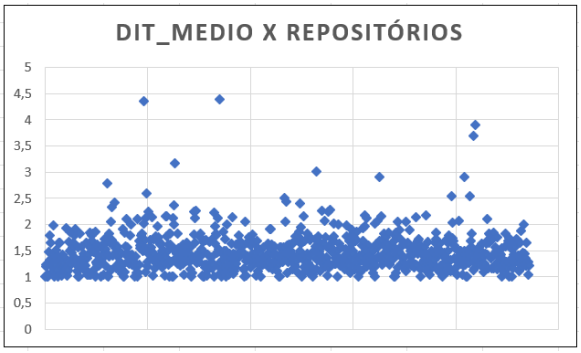
Estes dados são então processados e concatenados em um arquivo no formato CSV.
Com este arquivo, é, então, feita uma exportação e análise dos dados em um arquivo Excel (por meio do Power Query), e em seguida, gráficos foram criados usando estes dados.

3 Resultados Obtidos

Após a análise dos dados e criação dos gráficos, conseguimos visualizar as métricas ressaltadas por cada umas das questões de pesquisa. Abaixo estarão as interpretações dos resultados obtidos em correlação com as hipóteses definidas na seção 1.
Porem, antes das interpretações, mostramos 2 gráficos criados para demonstrar a distribuição das métricas de qualidade de todos os repositórios.

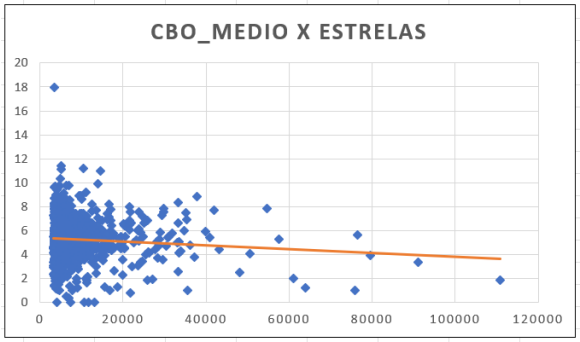


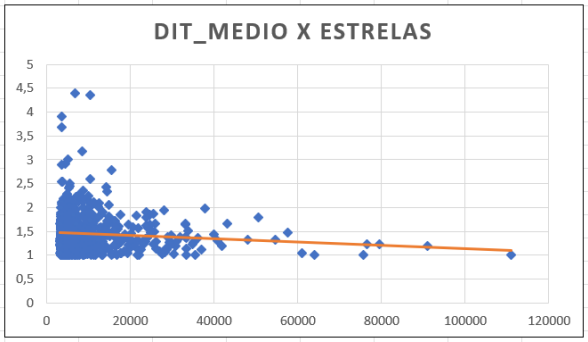
O CBO médio dos repositórios foi de $\simeq 5,27$.



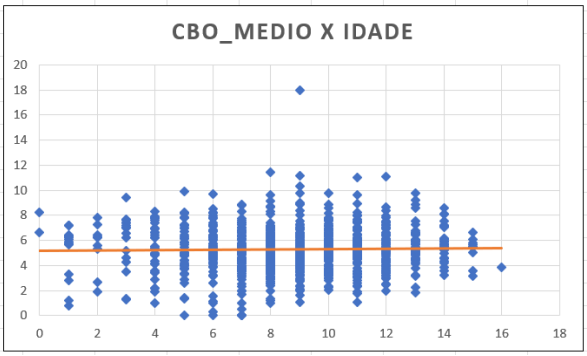
O DIT médio dos repositórios foi de $\simeq 1,46$.

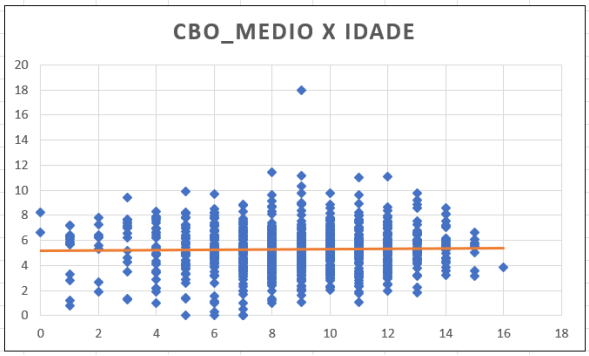
RQ 01:
Em relação à popularidade (n.º de estrelas), os repositórios não têm uma relação direta com as métricas de qualidade. Observamos que, em geral, os repositórios com menos de 10 mil estrelas foram aqueles que apresentaram números maiores nas métricas de qualidade (CBO e DIT).



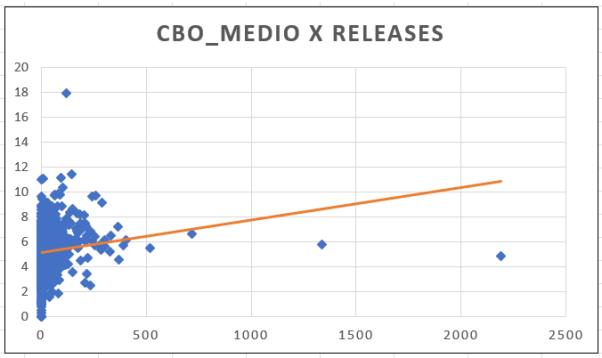


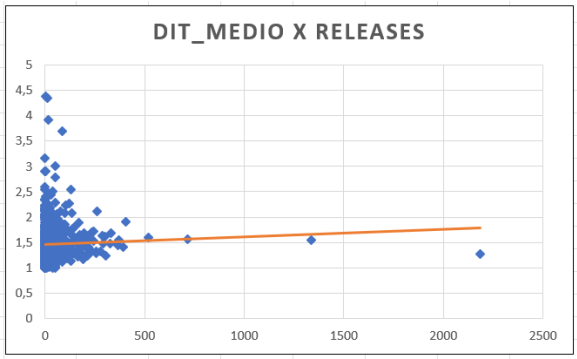
RQ 02:
O quesito idade do repositório não afeta muito a qualidade de tal. É possível identificar repositórios novos que possuem a mesma taxa de qualidade de repositórios velhos.
Portanto, a hipótese que repositórios mais antigos teriam maior qualidade não foi atendida por completo; uma vez que os novos repositórios já surgem seguindo padrões de qualidade preestabelecidos.



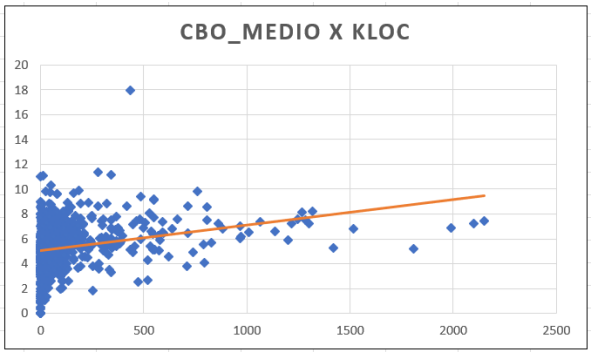


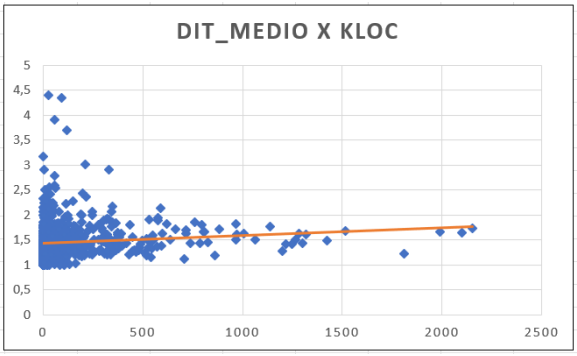
RQ 03:
Quando falamos sobre releases, nossa hipótese se provou errônea. Ficou evidente que o fator qualidade é mantido, mesmo a maioria dos repositórios tendo em torno de até 200 releases.
Acreditamos que isto deve se dar por conta do que comentamos na RQ anterior, isto é, que os repositórios já são criados focados em padrões de qualidade preestabelecidos, o que, no contesto desta questão, significa que não são cometidos muitos erros que prejudicam a qualidade, e, consequentemente, são geradas poucas releases.





RQ 04:
Já para o fator linhas de código (consideradas nesta análise no padrão de 1000 linhas, isto é, kLOC), a nossa hipótese se provou verdadeira. Ficou visível que uma menor quantidade de linha de código ajuda a se manter a qualidade de um sistema; porem também foi possível verificar que sistemas com mais de 2000kLOC (2 milhões de linhas) também conseguem manter o mesmo nível de qualidade que os demais.





4 Conclusão

Como conclusão deste trabalho tiramos que atualmente qualidade já é algo essencial do desenvolvimento de softwares, de forma que já se pensa em formas de garantir a qualidade de um sistema mesmo antes de se quer começar seu desenvolvimento.
Como foi dito na seção anterior, já existem diversos padrões de qualidade extremamente aceitos e exigidos pela comunidade. É importante relembrar que todos os repositórios desta análise possuem mais de 3000 estrelas, e, eles só chegaram a esta marca, por presarem e utilizarem estes padrões de qualidades desde o início do repositório.

References

Dados: https://api.github.com/graphql