Laboratório #03 - Análise Comparativa de Qualidade de Repositórios Java e Python

Gabriel Henrique Souza Haddad Campos[1]
[1] Bacharelado em Engenharia de Software
Instituto de Ciências Exatas e Informática – PUC Minas

gabriel.campos.1137372@sga.pucminas.br

1. Introdução

O GitHub é uma plataforma de armazenamento e versionamento de códigos que permite a criação, gestão e compartilhamento de repositórios de autoria própria gratuitamente a partir de qualquer máquina com acesso à internet. Essa ferramenta disponibiliza estatísticas sobre as atividades nos repositórios, participação em repositórios de terceiros, abertura de issues, realização de pull requests etc. Além disso, o GitHub oferece uma API gratuita que permite a obtenção de dados sobre qualquer repositório público utilizando queries com parâmetros bem definidos.

Dado esse contexto, será desenvolvido este trabalho da disciplina de Laboratório de Experimentação de Software com o objetivo de analisar algumas características de sistemas populares de código aberto armazenados em repositórios no GitHub, comparando-as entre repositórios com linguagem principal Java e Python. O critério para definição da popularidade de um repositório será sua quantidade de estrelas, sistema da própria ferramenta que indica que um usuário tem interesse naquele código.

Sendo assim, serão coletados os top-100 repositórios Python e os top-100 repositórios Java mais populares do GitHub, calculando cada uma das métricas definidas abaixo:

- Popularidade: número de estrelas, número de watchers, número de forks;
- Tamanho: linhas de código (LOC);
- Atividade: número de releases, frequência de releases (número de releases/dias);
- Maturidade: idade (em anos).

Dessa forma, as perguntas que motivam este trabalho estão listadas a seguir:

1) Quais as características dos top-100 repositórios Java mais populares?
2) Quais as características dos top-1000 repositórios Python mais populares?
3) Repositórios Java e Python populares possuem características de “boa qualidade” semelhantes?
4) A popularidade influencia nas características dos repositórios Java e Python?

Nesse sentido, foram elaboradas hipóteses para cada uma dessas questões a serem testadas nesta pesquisa. Estas seguem enumeradas abaixo, com os seus numeradores correspondentes às perguntas anteriormente demonstradas:

1) Considerando que Java é a quinta linguagem mais popular de acordo com a pesquisa do StackOverflow de 2019, espera-se que os seus repositórios possuam grandes valores para os números de estrelas, watchers, forks, releases, linhas de código e idade. Também espera-se que seus repositórios possuam uma alta porcentagem de comentários no código, com relação ao total de LOC.

2) Considerando que Java é a quarta linguagem mais popular de acordo com a pesquisa do StackOverflow de 2019, espera-se que os seus repositórios possuam grandes valores para os números de estrelas, watchers, forks,

releases, linhas de código e idade. Também espera-se que a quantidade de LOC não seja muito elevada, visto que é uma linguagem de alto nível. Dessa forma, a mediana de LOC esperada é de 10000 LOC.

3) Considerando a porcentagem de linhas de código referentes à comentários do código como indicativo de qualidade, espera-se que os repositórios possuam pelo menos 25% do total de LOC como comentários.

4) Considerando que com um maior número de usuários de um repositório os valores de estrelas, watchers, forks, releases, linhas de código e idade tendem a crescer constantemente, acredita-se que a popularidade influência nas características de repositórios Java e Python.

2. Metodologia

Esta é uma pesquisa de cunho descritivo que realiza uma abordagem quantitativa. Para análise das métricas de popularidade, atividade e maturidade, foram coletadas informações dos repositórios de ambos os conjuntos (top-100 Java e top-100 Python) utilizando a API GraphQL do GitHub através de um script Python. Para calcular os valores de LOC, foi utilizada uma ferramenta de análise estática de código, a biblioteca Python-Loc-Counter, a qual disponibiliza a quantidade de linhas de código-fonte, de comentários e de linhas em branco de um arquivo.

Foi necessário efetuar o *clone* de cada um dos repositórios do *dataset* para realizar a análise e contagem do número de LOC, verificando os dados de cada arquivo com extensão *".java"* para repositórios com linguagem primária Java e de arquivos com extensão *".py"* para repositórios com linguagem primária Python.

Para obter a métrica de atividade dos repositórios, calculou-se a frequência de publicação de releases utilizando o total de releases dividido pela idade do repositório em dias, indicando quantas releases por dia foram publicadas, aproximadamente. Dessa forma, quanto mais próximo de 1, mais frequentes são as publicações do repositório.

Nesse sentido, as questões de pesquisa 1 e 2 serão respondidas a partir da análise quantitativa de cada uma métricas, utilizando de valores medianos. Para a 3 e 4, os valores obtidos nas 1 e 2 devem ser comparados e discutidos individualmente.

A coleta dos dados foi realizada no dia 05/10/2020, durante a segunda sprint do trabalho prático. A análise dos dados foi realizada no dia 14/10/2020. O repositório com o script, os dados obtidos e a análise dos dados podem ser encontrados no GitHub através desta URL: https://github.com/Haddadson/tp-03-lab-exp

3. Resultados

A partir da execução dos passos anteriormente descritos, foram encontradas as respostas abaixo para cada uma das perguntas enumeradas na Introdução deste documento:

1) Analisando todas as características dos 100 repositórios mais populares de Java, temos as seguintes tabelas:

Métricas	Estrelas	Watchers	Forks	Releases	Idade(dias)	Idade(anos)	Releases/dias
Mediana	17070,5	900,5	4743,5	7,5	2027,5	5	0,003867663
Máximo	112090	5229	36601	201	3893	10	0,116463415
Mínimo	12032	179	588	0	130	0	0
Média	23008,52	1206,41	6910,5	24,39	1979,62	4,87	0,012493252
Desvio padrão	15631,92	912,1472	6760,1	39,72173	929,097311	2,52524626	0,019685534

Métricas	SLOC	Linhas de comentário	Linhas em branco	LOC	Percentual de comentários (%)
Mediana	24540	43,5	4545,5	29279	0,117479665
Máximo	2153923	18217	311482	2466128	2,615664845
Mínimo	0	0	0	0	0
Média	165101	929,8	26743,22	191907,8	0,267094306
Desvio padrão	370705	2789,789714	57628,91023	427756,7	0,400222661

Sendo assim, a partir da mediana dos valores, pode-se observar que é retornada uma quantidade de 17070,5 estrelas, 900,5 watchers, 4743,5 forks, 7,5 releases e 29279 linhas de código, para a mediana de 5 anos de idade. Ademais, visualiza-se uma grande diferença entre os valores máximos e mínimos de cada uma das características, bem como de sua média, demonstrando um conjunto de dados diversificado. O desvio padrão é muito elevado, reforçando essa variação nos valores. Podemos ver que alguns repositórios não possuem linhas de código e não utilizam a ferramenta de releases do GitHub.

2) Analisando todas as características dos 100 repositórios mais populares de Python, temos as seguintes tabelas:

Métricas	Estrelas	Watchers	Forks	Releases	Idade(dias)	Idade(anos)	Releases/dias
Mediana	20898,5	855,5	4521,5	2,5	1880,5	5	0,0018692
Máximo	108620	5887	40212	657	4493	12	0,255046584
Mínimo	13796	181	661	0	376	1	0
Média	28556,14	1225,29	6535,8	28,46	2057,88	5,15	0,012652647
Desvio padrão	19443,41	1068,533	6826,2	83,32854	1040,00877	2,88631393	0,034791283

Métricas	SLOC	Linhas de comentário	Linhas em branco	LOC	Percentual de comentários (%)
Mediana	6001	1796	1667	10229	18,96364981
Máximo	610222	171123	148808	834814	99,43790434
Mínimo	0	0	0	0	0
Média	48630,2	14775,07	11513,13	74261,9	19,11366692
Desvio padrão	101025	31960,90568	24406,69845	150759,9	12,98815931

linhas

Dessa forma, a partir da mediana dos valores, pode-se observar que é retornada uma quantidade de 20898,5 estrelas, 855,5 watchers, 4521,5 forks, 2,5 releases e 1880,5 linhas de código, para a mediana de 5 anos de idade. Ademais, visualiza-se uma grande diferença entre os valores máximos e mínimos de cada uma das características, bem como de sua média, demonstrando um conjunto de dados diversificado. O desvio padrão é muito elevado, reforçando essa variação nos valores. Podemos ver que alguns repositórios também não possuem linhas de código e não utilizam a ferramenta de releases do GitHub.

3) Assumindo como referência o percentual de comentários no código para a definição de “boa qualidade” do código do repositório, pode-se verificar que a mediana deste percentual é muito mais elevada para projetos desenvolvidos em Python. O valor para os repositórios Python se aproxima de 19%, enquanto para repositórios Java mostra-se com apenas 0,0038%.

4) Apesar das métricas de estrelas dos repositórios Python mostrarem-se superiores aos de Java, indicando maior popularidade, as demais métricas mostraram-se menores. A mediana de Watchers, Forks e Releases são inferiores em projetos com linguagem Python. A mediana indica que projetos Java são mais velhos que os desenvolvidos em Python, apesar do mais novo obtido ter apenas 130 dias e ser desenvolvido em Java.

4. Conclusão

Nesse contexto, comparando as hipóteses com os resultados, pode-se concluir que:

1) De acordo com a hipótese formulada, era esperado que os valores retornados para stars, watchers, forks, releases, linhas de código e idade dos repositórios de Java fossem altos. Tais valores mostraram-se elevados, todavia, o percentual de comentários no código com relação ao total de LOC teve valores extremamente baixos, com a mediana não chegando a 1%. O repositório com maior percentual atingiu 2,61%.

2) De acordo com a hipótese formulada, era esperado que os valores retornados para stars, watchers, forks, releases, linhas de código e idade dos repositórios Python mais populares fossem altos. Todavia, conforme evidenciado anteriormente, apenas parte dos valores assim se comportaram. Foram retornados valores zerados para releases e uma idade pequena para o repositório da mediana, se comparado ao de maior idade. A quantidade mediana de LOCs se aproximou do esperado, atingindo 10229 linhas. O percentual de comentários mostrou-se inferior ao mencionado na hipótese, atingindo o valor de 18,96% na mediana. Entretanto, o repositório com maior valor possui 99% do total das linhas de código como comentários, representando um possível projeto com diversas descrições e pequenos trechos de código-fonte.

3) Hipótese não comprovada. Os repositórios Java possuem como mediana apenas 0,11% do total de LOCs como comentários. Projetos Python atingem 18,96%, sendo um valor muito superior ao do outro *dataset*, mas ainda insuficiente para atingir o valor esperado.

4) Hipótese parcialmente comprovada, pois a mediana de estrelas de repositórios em Python é superior à projetos Java, o que indica que são projetos mais populares de acordo com a definição estabelecida. Isso implicou em números elevados de métricas de releases, releases por dia, forks e watchers. Entretanto, mesmo em projetos de menor popularidade feitos em Java, os valores das medianas obtidos foram maiores. Apenas a métrica de releases por dia mostrou-se superior.