Laboratório #01 - Características de Sistemas Populares GitHub

Maria Verônica Santos Soares

(maria.veronica@sga.pucminas.br)

Instituto de Ciências Exatas e Informática

Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

Introdução

O GitHub é uma plataforma popular de versionamento de software, em que seus usuários podem criar, manter e gerenciar seus próprios repositórios de código na nuvem gratuitamente. A plataforma garante acesso aos repositórios - que podem ser públicos ou privados - por parte de seus usuários a partir de qualquer máquina no mundo, desde que esta esteja conectada à internet. Além disso, esse recurso possui também funcionalidades como: estatísticas sobre as atividades nos repositórios, participação em repositórios de terceiros, abertura de *issue report*, realização de *pull requests* etc. O GitHub também disponibiliza uma API, através da qual é possível obter dados sobre repositórios por meio de queries com parâmetros bem definidos.

Nesse contexto, este é um trabalho de medição e experimentação de software, cujo objetivo é analisar algumas características de sistemas populares de código aberto que estão armazenados em repositórios no GitHub. As perguntas que motivam este trabalho estão listadas a seguir:

1) Sistemas populares são maduros/antigos?
2) Sistemas populares recebem muita contribuição externa?
3) Sistemas populares lançam releases com frequência?
4) Sistemas populares são atualizados com frequência?
5) Sistemas populares são escritos nas linguagens mais populares?
6) Sistemas populares possuem um alto percentual de *issues* fechadas?
7) Sistemas escritos em linguagens mais populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência?

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) A popularidade de um sistema no GitHub pode ser obtida por meio da boa avaliação deste, requerendo um alto número de estrelas e, para considerar um sistema maduro, este deve possuir pelo menos dois

anos de idade, o que totalizam um total de 730 dias (365*2). Sendo assim, levando em consideração que sistemas mais antigos podem possuir uma boa avaliação e ainda que para obter sua popularização é interessante que haja uma divulgação do repositório, é condizente pensar que sistemas populares sejam maduros, uma vez que o processo de reconhecimento e ganho de credibilidade é moroso e demanda tempo.

2) Analisando os repositórios das tecnologias mais populares de código aberto (Linux, NodeJS, VueJS, por exemplo) presentes no mercado, pode-se perceber que existe boa avaliação dos usuários e grande número de contribuições destes em prol da melhoria constante do sistema. Portanto, é esperado que sistemas populares recebam muita contribuição externa.
3) No contexto da pergunta que se visa analisar, devem ser ponderadas as duas hipóteses anteriores, abrangendo a popularidade e as *releases*. Se um sistema recebe muita contribuição externa, espera-se que seu número de *releases* aumente. Porém, parte integrante da maturidade, que de certa forma está relacionada à popularidade, de um software é sua estabilidade. Sendo assim, espera-se um número considerável de releases, mas não muito elevado devido à necessidade de filtro do que deve ser realmente adicionado à sua versão final.
4) Levando em consideração a hipótese levantada para a questão 2, espera-se que sistemas populares sejam atualizados com frequência, devido ao engajamento de suas comunidades contribuintes.
5) Um dos fatores de influência na popularidade de um software é a linguagem em que ele foi desenvolvido, pois, inevitavelmente, esse fato atrai a atenção e o engajamentos de desenvolvedores que trabalham com a linguagem em questão. Sendo assim, se a linguagem é popular, a quantidade de desenvolvedores que se interessa por ela tende a ser popular também, o que nos leva a acreditar que sistemas populares são escritos nas linguagens mais populares.
6) Acredito que a correção de problemas encontrados e reportados e a evolução do código são fatores vitais para a popularidade de um sistema. Sendo assim, não necessariamente um sistema popular terá muitas *issues* reportadas, todavia, dentre as que foram reportadas, espera-se que o índice das fechadas seja alto.
7) Levando em consideração as hipóteses anteriores e todo o contexto em que estas estão inseridas, espera-se que sistemas escritos em linguagens mais populares recebam mais contribuição externa, lancem mais releases e sejam atualizados com mais frequência.

Metodologia

Esta é uma pesquisa de cunho descritivo que realiza uma abordagem quantitativa. Foram minerados os 1000 repositórios de software mais bem

avaliados no GitHub e, a partir da obtenção os dados para análise, foram elaboradas as hipóteses anteriormente descritas. A fim de atingir esse resultado, foi elaborado um script Python com a finalidade de extrair e tratar os dados necessários via API (GraphQL) do GitHub. Todos os artefatos gerados acompanham este documento num zip e podem ser encontrados em: https://github.com/mveronicasoares/LabMedicaoExperimentacao/releases.

Resultados obtidos

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

1) Analisando a idade dos repositórios obtidos, chegamos na seguinte tabela de resultados:

Intervalo de Idade em Dias	Quantidade de Repositórios
x <= 500	69
100 < x <= 1500	294
1500 < x <= 2000	227
2000 < x <= 2500	144
2500 < x <= 3000	59
500 < x <= 1000	203
x > 3000	4
Total Geral	**1000**

Sendo assim, observa-se que mais de 700 repositórios possuem um tempo de vida maior do que dois anos, conforme a hipótese enunciada na introdução. Ademais, utilizando a métrica da mediana das idades dos repositórios pesquisados, encontra-se um resultado de 1415 dias, que também é maior do que 750 dias.

2) Analisando o número de *pull requests* aceitas nos repositórios obtidos na pesquisa, chegamos na seguinte tabela de resultados:

Intervalo de *pull requests*	Quantidade de repositórios
x < 10000	783
10000 <= x < 20000	207
20000 <= x < 30000	5
30000 <= x < 40000	3
40000 <= x < 50000	1
x => 50000	1
Total Geral	**1000**

Sendo assim, observa-se que mais de 700 repositórios possuem menos de 10000 *pull requests* aceitos. Ademais, utilizando a métrica da mediana da quantidade de *pull requests* aceitos, obtém-se um resultado de 281.

3) Analisando o número de *releases* dos repositórios obtidos, chegamos na seguinte tabela de resultados:

Intervalo de *releases*	Quantidade de repositórios
x < 100	883
100 <= x < 200	85
200 <= x < 300	17
300 <= x <400	6
400 <= x < 500	5
x >= 500	4
Total Geral	**1000**

Sendo assim, observa-se que quase 90% dos repositórios analisados possuem menos de 100 *releases*. Ademais, utilizando a métrica da mediana da quantidade de *releases*, obtém-se um resultado de 7.

4) Analisando o tempo, em dias, desde a última atualização dos repositórios obtidos em relação a data de hoje (06/03/2020), chegamos na seguinte tabela de resultados:

Dias percorridos	Quantidade de Repositórios
5	943
6	50
7	7
Total Geral	**1000**

Sendo assim, observa-se que quase 95% dos repositórios analisados possuem sua última entrega realizada em 5 dias. Ademais, utilizando a métrica da mediada desses valores, obtém-se 5.

5) Analisando a linguagem dos sistemas resultantes da pesquisa com o objetivo de verificar se estes foram escritos em uma das 25 linguagens mais populares segundo a pesquisa apresentada em https://insights.stackoverflow.com/survey/2018/#technology, chegamos na seguinte tabela de resultados:

Linguagens mais populares	Quantidade de Repositórios
Assembly	2
C	23
C#	8
C++	45
CSS	25
Go	58
HTML	22
Java	71
JavaScript	301
Kotlin	11
Objective-C	12

Perl	1
PHP	19
Python	95
Ruby	17
Scala	3
Swift	23
TypeScript	48
Total das linguagens populares	**784**
Linguagem não popular	100
Nenhuma linguagem identificada	116
Total Geral	**1000**

Sendo assim, observa-se que 784 repositórios foram desenvolvidos nas 25 linguagens de programação mais populares.

6) Analisando a razão entre o número de *closed issues* e o de *total issues* obtemos um valor de 0,849283827, totalizando 84,93%.

7) Analisando a comparação dos repositórios cuja linguagem primária é uma dentre as 25 mais populares segundo a pesquisa demonstrado no item anterior com os demais repositórios segundo todos os parâmetros das perguntas anteriores temos a seguinte tabela:

	Mediana	
	Linguagens populares	**Linguagens não populares**
closed/total	0,864295753	0,786241411
dias percorridos da última atualização	5	5
releases	14	0
pullRequests	332	123,5
idade em dias	1449	1263,5

Discussão

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

1) Grande parte dos repositórios retornados na pesquisa possui um tempo de criação maior do que dois anos, portanto, como esperado na hipótese, repositórios populares são maduros.

2) Repositórios populares recebem uma taxa consideravelmente baixa de contribuição, levando em consideração que a mediana das *pull request* foram um total de 281 enquanto o maior valor encontrado para essa métrica foi de 59271.

3) Repositórios populares possuem poucos *releases*, considerando que a mediana desse valor encontrado foi de 7 contra o maior de mais de 500.

4) Repositórios populares são atualizados com frequência, já que a mediana desses valores foi de 5.

5) Mais de 75% dos sistemas mais bem avaliados estão escritos em linguagens de programação populares.

6) Repositórios populares têm um alto índice de issues fechadas, sendo este de quase 85%.

7) Os sistemas avaliados desenvolvidos em linguagens populares possuem um número maior de *closed issues*, *releases*, *pull requests* e idade se comparados aos que foram produzidos nas demais linguagens. Todavia, a mediana da quantidade de dias percorridos desde a última atualização dos *softwares* foi a mesma, de 5 dias.