Laboratório #02 - Características de Sistemas Populares GitHub

Gabriel Henrique Souza Haddad Campos
(gabriel.campos.1137372@sga.pucminas.br)

Instituto de Ciências Exatas e Informática
Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

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 issue report, 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 Medição e Experimentação de Software com o objetivo de analisar algumas características de sistemas populares de código aberto armazenados em repositórios no GitHub. 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. As demais restrições propostas para a análise dos dados será determinada posteriormente.

Este trabalho prático foi motivado através de perguntas de pesquisa, as quais 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?

Com isso, foram elaboradas hipóteses para cada uma das questões a serem analisadas nesta pesquisa. Estas, por sua vez, seguem enumeradas abaixo com os valores correspondentes às perguntas apresentadas anteriormente:

1) A métrica estabelecida que indicará a popularidade dos repositórios é a quantidade de estrelas, de forma que quanto maior o número, maior a popularidade. Para considerar um sistema maduro, este deve possuir pelo menos cinco anos de idade. 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) Acredito que os repositórios mais populares recebam muitas contribuições externas por serem muito utilizados e por várias pessoas se interessarem neste código. Para metrificar os resultados, é esperado que os sistemas populares tenham mais de 10000 *pull requests* aceitos, determinando um alto número de contribuições externas.

3) No contexto da pergunta que se visa analisar, devem ser ponderar 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. Alguns repositórios também não utilizam a ferramenta de releases do GitHub, optando por não marcar os *commits* como novas versões. Dessa forma, é esperado que os repositórios tenham por volta de 100 *releases*.

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. Com isso, o resultado esperado será de pelo menos uma contribuição enviada nos últimos 7 dias para a maioria dos repositórios.

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 e devido à popularidade da própria linguagem. Sendo assim, se a linguagem é popular, a quantidade de desenvolvedores que se interessa por ela tende a ser alta, o que nos leva a acreditar que sistemas populares são escritos nas linguagens mais populares. A partir disso, espera-se que a maioria dos repositórios tenham sua linguagem principal como uma das

populares de acordo com a pesquisa do StackOverflow de 2019.

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. Portanto, espera-se que a porcentagem de issues fechadas seja acima de 85% para a maioria dos repositórios.

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 devido à maior popularidade da linguagem, atraindo mais desenvolvedores. Com isso, espera-se que os repositórios feitos em linguagens populares cumpram as hipóteses estipuladas anteriormente e possuam métricas superiores às de repositórios feitos em linguagens não populares.

Metodologia

Esta é uma pesquisa de cunho descritivo que realiza uma abordagem quantitativa. Foram minerados os 1000 repositórios de software com mais estrelas 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 para extrair e tratar os dados. Essas informações foram obtidas através da API (GraphQL) do GitHub. Todos os artefatos gerados acompanham este documento num zip e podem ser encontrados em: https://github.com/Haddadson/tp-01-lab-exp.

Foram estipuladas métricas para auxiliar a mensuração dos resultados de cada pergunta. Os resultados necessários para as métricas foram retornados da API citada e tratados para permitir a melhor análise dos dados. As métricas necessárias para responder cada questão estão enumeradas abaixo, sendo cada enumerador referente à questão informada anteriormente.

1) Idade do repositório em anos (calculado a partir da data de sua criação)
2) Total de pull requests aceitos
3) Total de releases
4) Tempo até a última atualização em dias (calculado a partir da data de última atualização - considerando a atualização como o último *push* enviado para o repositório)

5) Linguagem primária de cada repositório
6) Razão entre número de issues fechadas pelo total de issues (percentual)
7) Mediana da razão das *issues* fechadas por total de *issues*, dos dias percorridos da última atualização, da quantidade de releases, dos *pull requests* aceitos e da idade em anos de repositórios feitos em linguagens populares e não populares.

Resultados Obtidos

A partir da execução dos passos anteriormente descritos, foram encontradas as respostas abaixo para cada uma das perguntas enumeradas na Introdução deste documento. É importante ressaltar que a data de execução do script em Python irá interferir nos resultados, considerando as atualizações nos repositórios. A busca dos dados foi realizada no dia 16/09/2020 por volta das 20 horas.

1) Ao analisar as idades dos repositórios obtidos, foi possível identificar a tabela abaixo, sendo X a idade do repositório em anos:

Intervalo de Idade em Anos	Quantidade de Repositórios
x <= 4	386
x = 5	148
5 < x <= 9	402
x >= 10	64
Total Geral	**1000**

Dessa forma, a tabela gerada mostra que a maior parte dos repositórios obtidos possuem 5 ou mais anos de idade, apesar da quantia relevante com menos de 5 anos.

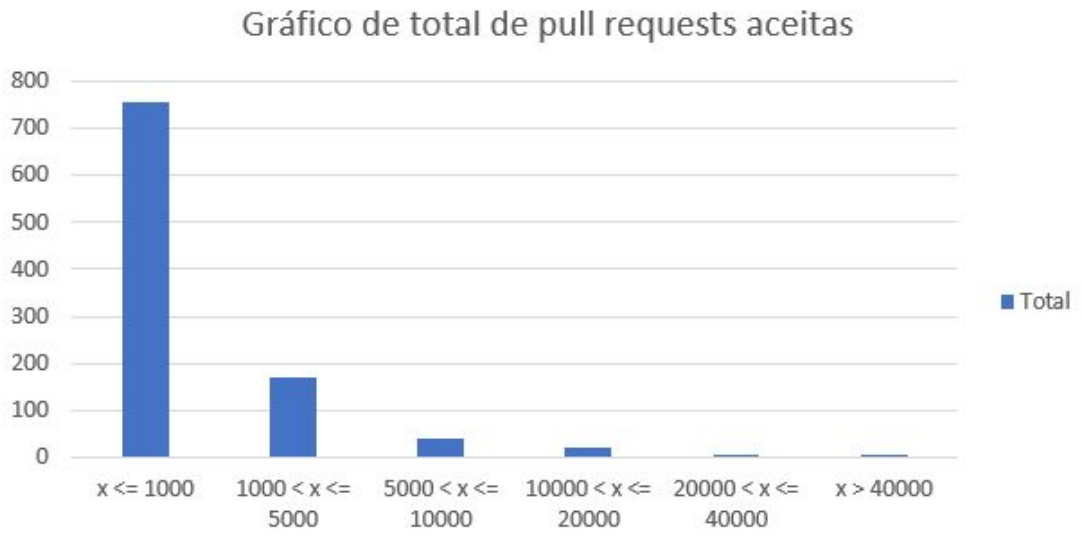
Outra forma de visualização dos dados acima é o gráfico gerado apresentado abaixo.

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

Intervalo de pull requests aceitas	Quantidade de Repositórios
x <= 1000	756
1000 < x <= 5000	172
5000 < x <= 10000	40
10000 < x <= 20000	23
20000 < x <= 40000	6
x > 40000	3
Total Geral	**1000**

Segundo a métrica determinada, o esperado era que a maioria dos repositórios tivesse mais de 10000 *pull requests* aceitas, indicando a quantidade de contribuições externas. Entretanto, ao analisar os dados obtidos, podemos ver que a maior parte dos repositórios possui menos de 1000 *pull requests* aceitas, o que indica que a hipótese mostrou-se inválida.
Ademais, utilizando a métrica da mediana da quantidade de pull requests aceitos, obtém-se um resultado de 311, o qual é um valor abaixo do esperado na hipótese.

O gráfico abaixo possui outra representação dos dados da tabela.

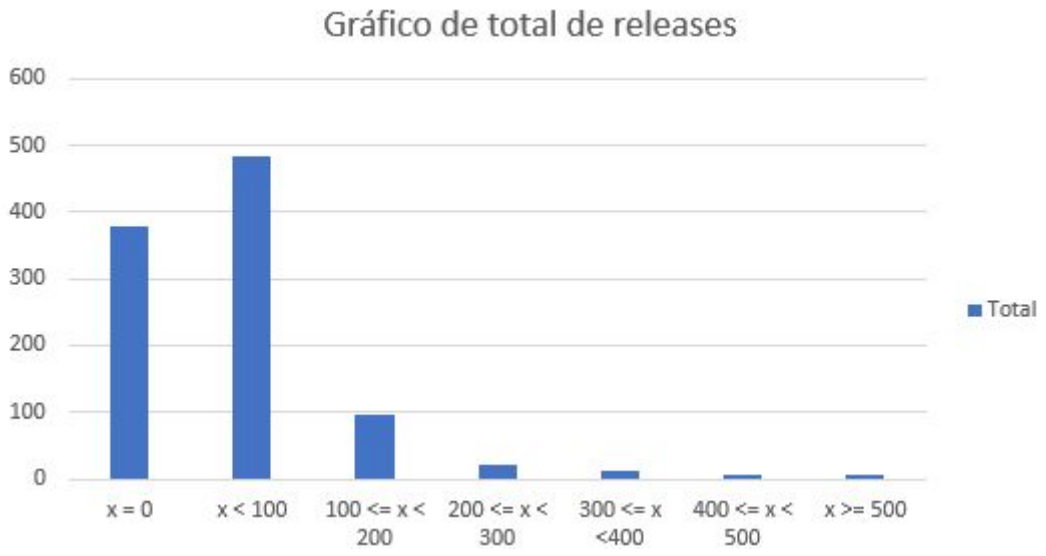


3) Analisando o número de *releases* dos repositórios obtidos, chegamos na seguinte tabela de resultados, sendo X a quantidade de releases:

Intervalo de releases	Quantidade de Repositórios
x = 0	378
x < 100	485
100 <= x < 200	95
200 <= x < 300	20
300 <= x <400	11
400 <= x < 500	5
x >= 500	6
Total Geral	**1000**

Sendo assim, observa-se que cerca de 85% dos repositórios analisados possuem menos de 100 releases. É importante ressaltar que 378 repositórios não utilizam a ferramenta de releases, dessa forma, estes possuem 0 itens. Ademais, utilizando a métrica da mediana da quantidade de releases, obtém-se um resultado de 10.

A representação gráfica da tabela encontra-se abaixo.

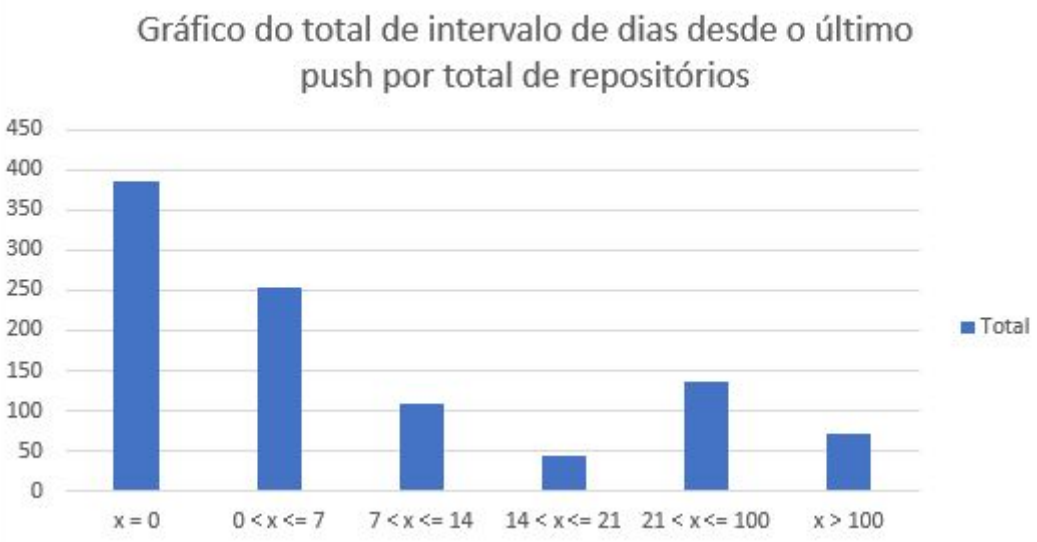


4) Analisando o tempo, em dias, desde a última atualização dos repositórios obtidos em relação a data de mineração dos dados (16/09/2020), chegamos na seguinte tabela de resultados:

Intervalo de dias desde o último push	Quantidade de Repositórios
x = 0	385
0 < x <= 7	253
7 < x <= 14	109
14 < x <= 21	44
21 < x <= 100	137
x > 100	72
Total Geral	**1000**

Dessa forma, observa-se que quase metade dos repositórios analisados possuem um *push* nos 7 dias anteriores à mineração dos dados. Ademais, utilizando a métrica da mediana desses valores, obtém-se 2. Esses dados revelam que boa parte dos repositórios recebem novas atualizações no código com uma frequência semanal, enquanto a outra metade leva mais tempo.

Pode-se analisar a versão gráfica da tabela abaixo.



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/2019/#technology, chegamos nas seguintes tabelas de resultados:

Linguagens mais populares	Quantidade de Repositórios
JavaScript	283
HTML	21
CSS	21
SQL	0
Python	100
Java	80
Bash	1
Shell/PowerShell	22
C#	10
PHP	15
C++	49
TypeScript	62
C	22
Ruby	17
Go	59
Assembly	3
Swift	22
Kotlin	11
R	0
VBA	0
Objective-C	7

Scala	2
Rust	9
Dart	3
Elixir	2
Clojure	2
WebAssembly	0
Nenhuma linguagem identificada	**120**
Total das linguagens populares	**823**

Linguagens não populares analisadas	Quantidade de Repositórios
CoffeeScript	3
Crystal	1
Dockerfile	1
Emacs Lisp	1
Haskell	4
Julia	1
Jupyter Notebook	12
Lua	4
Makefile	2
OCaml	1
Rascal	1
Rich Text Format	1
Standard ML	1
TeX	3
V	1

Vim script	10
Vue	10
Nenhuma linguagem identificada	**120**
Total das linguagens não populares	**57**

Analisando ambas tabelas, as quais se referem à quantidade de repositórios cuja linguagem principal faça parte das mais populares ou não, podemos verificar que a grande maioria dos repositórios analisados foram desenvolvidos com linguagens populares, totalizando uma quantidade de 823. Além disso, a linguagem mais popular da pesquisa, JavaScript, possui a maior quantidade de repositórios.

Também é importante ressaltar que dentre os repositórios analisados, 120 não possuem uma linguagem principal determinada.

6) Analisando a média da razão entre o número de closed issues e o de total issues obtemos um valor de 0,768498397, totalizando 76,85%. Já a mediana da razão entre esses valores resulta no valor de 0,852650396, totalizando 85,26%. É importante destacar que o menor valor obtido da razão resulta em 73,17% e o maior resulta em 100,00%.

7) Analisando a comparação dos repositórios cuja linguagem primária é uma dentre as 25 mais populares segundo a pesquisa do StackOverflow, apresentada anteriormente, com os demais repositórios e segundo todos os parâmetros das perguntas anteriores temos a seguinte tabela:

Medianas		
	Linguagens Populares	**Linguagens Não Populares**
Closed/Total Issues	0,866052936	0,791518738
Dias percorridos desde o último push	1	8
Releases	16,5	3
Pull Requests	378,5	144,5
Idade em anos	5	4

É importante ressaltar que os repositórios sem linguagem principal não foram considerados na criação dessa tabela.

Discussão

Dado esse contexto, comparando as hipóteses com os resultados obtidos, pode-se concluir que:

1) Conforme a hipótese estabelecida anteriormente, era esperado que a maioria dos repositórios possuísse pelo menos cinco anos de idade. De acordo com os resultados, verifica-se que a maior parte dos repositórios possui mais de 5 anos, mas uma quantia relevante possui um valor inferior. 386 repositórios possuem 4 ou menos anos de idade

2) Repositórios populares recebem uma taxa baixa de contribuição com relação à hipótese estabelecida, levando em consideração que a mediana das pull request foram um total de 311, enquanto o maior valor encontrado para essa métrica foi de 69506. Também é importante ressaltar que a grande maioria dos repositórios possui menos de 10000 *pull requests* aceitas, tornando a hipótese equivocada.

3) Repositórios populares possuem poucos releases em relação à hipótese estabelecida, considerando que a mediana desse valor encontrado foi de 10 contra o maior de mais de 643. Cerca de 85% dos repositórios obtidos possuem menos de 100 releases, sendo que 385 dos 1000 analisados não possuem releases. Isso demonstra novamente que a hipótese foi equivocada e não se comprovou.

4) Os dados analisados revelam que boa parte dos repositórios recebem novas atualizações no código com uma frequência semanal visto que receberam um *push* nos últimos 7 dias. Entretanto, a outra parte dos repositórios não recebeu atualizações há mais tempo. Ademais, a mediana encontrada possui valor 2, enquanto o maior valor indica 1192 dias sem um *push* no repositório analisado a partir da data de mineração.

5) Através da análise dos dados, vemos que 823 repositórios foram desenvolvidos com uma das 25 linguagens populares apresentadas na pesquisa do StackOverflow, indicando mais de 80% dos repositórios analisados. Além disso, a linguagem mais popular da pesquisa, JavaScript, possui a maior quantidade de repositórios. Isso comprovou a hipótese formulada.

6) Repositórios populares têm um alto índice de issues fechadas, sendo este de quase 85% em média e 75% de mediana. Dessa forma, a hipótese não se comprovou, visto que esperavam-se valores mais elevados de fechamento de issues.

7) Considerando os valores de medianas obtidas para as linguagens populares, podemos afirmar que a hipótese foi comprovada. As linguagens populares possuem métricas melhores que linguagens não populares dentro das condições estipuladas. Esses repositórios recebem atualizações mais frequentemente, possuem mais releases e mais *pull requests* aceitas, além de uma taxa de fechamento de *issues* superior. Entretanto, repositórios feitos em linguagens populares são um pouco mais antigos, contrariando a hipótese nesse aspecto.