INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA - PRAÇA DA LIBERDADE

CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

Matheus Santos Rosa Carneiro

Raíssa Carolina Vilela da Silva

Vitor Augusto Alves de Jesus

TRABALHO PRÁTICO DE LABORATÓRIO DE EXPERIMENTAÇÃO DE SOFTWARE

BELO HORIZONTE

2021

1. INTRODUÇÃO

O GitHub é uma plataforma de armazenamento e versionamento de códigos que permite a criação, gestão e compartilhamento de repositórios gratuitamente a partir de qualquer máquina que tenha conexão à internet. Essa ferramenta oferece uma API gratuita que permite a obtenção de dados sobre qualquer repositório público utilizando *queries* com parâmetros bem definidos.

Este trabalho tem o intuito de analisar os dados presentes nos 1.000 primeiros repositórios mais populares do GitHub. Esta análise será feita através de um experimento utilizando **GraphQL** e **Python**. O estudo dos dados será realizado por meio de uma ***research question*** e informações coletadas dos repositórios existentes.

Temos como objetivo, ampliar a percepção sobre o comportamento de sistemas ***open-source*** em larga escala e o uso da experimentação como instrumento para tomada de decisões mais eficazes. O repositório deste trabalho está disponível em: **https://github.com/mcarneirobug/lab-exp-software**.

Este trabalho prático foi proposto pelo professor orientador da disciplina de Laboratório de Experimentação de Software, com o intuito de fazer com que respondêssemos as seguintes perguntas:

1.1. Sistemas populares são maduros/antigos?

1.1.1. Hipótese: A métrica proposta irá indicar se a popularidade de um repositório está diretamente relacionada com a idade do mesmo, ou seja, quanto mais velho o repositório, mais popular ele é. Logo, poderemos pressupor que um sistema maduro, deve ter a idade mínima de cinco anos, desta forma, poderemos afirmar que eles são a maioria em comparação aos sistemas mais novos, devido ao fato de ter coletado os mil primeiros repositórios mais populares.

1.2. Sistemas populares recebem muita contribuição externa?

1.2.1. Hipótese: Pressupõe-se que quanto mais tempo de vida tem um repositório, mais contribuição externa ele recebe, pois, ele está disponível para contribuição há mais tempo do que novos

repositórios, ou seja, quanto mais velho é um repositório mais haverá contribuição dos usuários do GitHub.

1.3. Sistemas populares lançam releases com frequência?

1.3.1. Hipótese: Considerando o contexto da pergunta, podemos supor que os repositórios mais novos não lancem tantos *releases* com muita frequência, uma vez que repositórios mais novos, em teoria, são menos populares e consequentemente tendem a ter menos contribuição externa.

1.4. Sistemas populares são atualizados com frequência?

1.4.1. Hipótese: Levando em consideração que os dados coletados foram dos mil repositórios mais bem avaliados do GitHub, podemos considerar que todos os repositórios que estão sendo analisados neste estudo são repositórios populares, desta forma, iremos pressupor que todos os repositórios são atualizados com frequência.

1.5. Sistemas populares são escritos nas linguagens mais populares?

1.5.1. Hipótese: Pegando como base o esquema de *ranking* do GitHub (**https://octoverse.github.com**), será possível supor que a maior parte dos repositórios estão utilizando as linguagens *JavaScript*, *Python* e *Java.*

1.6. Sistemas populares possuem um alto percentual de *issues* fechadas?

1.6.1. Hipótese: Se considerarmos que os sistemas populares são os sistemas mais utilizados do GitHub, podemos pressupor que os repositórios mais velhos, pelo fato de ter mais contribuições têm tendência a ter mais *issues* abertas e consequentemente fecha-las na mesma proporção.

1.7. Sistemas escritos em linguagens mais populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência?

1.7.1. Pressupondo que as linguagens mais conhecidas têm apoio maior da comunidade de desenvolvedores que utilizam a

plataforma GitHub, podemos dizer que os repositórios analisados tendem a ter maior contribuição, *releases* e estejam mais atualizados do que os repositórios que não utilizam as linguagens mais populares dentro do *ranking* de linguagens do GitHub.

2. METODOLOGIA

Este estudo de cunho descritivo consiste na realização de uma abordagem quantitativa. Essa abordagem, surge da mineração dos mil repositórios de software mais populares e bem avaliados da plataforma GitHub. A análise dos dados coletados então, é realizada a partir das hipóteses anteriormente descritas, elaboradas a partir do estudo das *questions* propostas pelo professor orientador durante a disciplina de Laboratório de Experimentação de Software.

Afim de realizar experimentos através do estudo dos dados coletados, foi elaborado um *script* na linguagem de programação *Python* para que fosse possível realizar a extração e tratativa dos dados. As informações, foram obtidas através da API que o GitHub disponibiliza, conhecida como *GraphQL*. Todos os artefatos gerados através deste estudo, estão armazenados em: **https://github.com/mcarneirobug/lab-exp-software**.

Algumas métricas foram estipuladas para que fosse possível auxiliar na análise dos resultados de cada *question*. A coleta dos dados foi feita através da API citada e em seguida os dados foram exportados para um arquivo com a extensão *.csv* para que fosse mais fácil realizar a análise desses dados. As métricas utilizadas neste estudo, estão enumeradas de acordo com a ordem das hipóteses citadas anteriormente:

1. Idade do repositório em anos (calculado a partir da data de sua criação).
2. Total de *pull requests* aceitas.
3. Total de *releases*.
4. Tempo até a última atualização em dias (calculado a partir da data de última atualização).
5. Linguagem primária de cada repositório.
6. Razão entre o número de *issues* fechadas pelo total de *issues*.

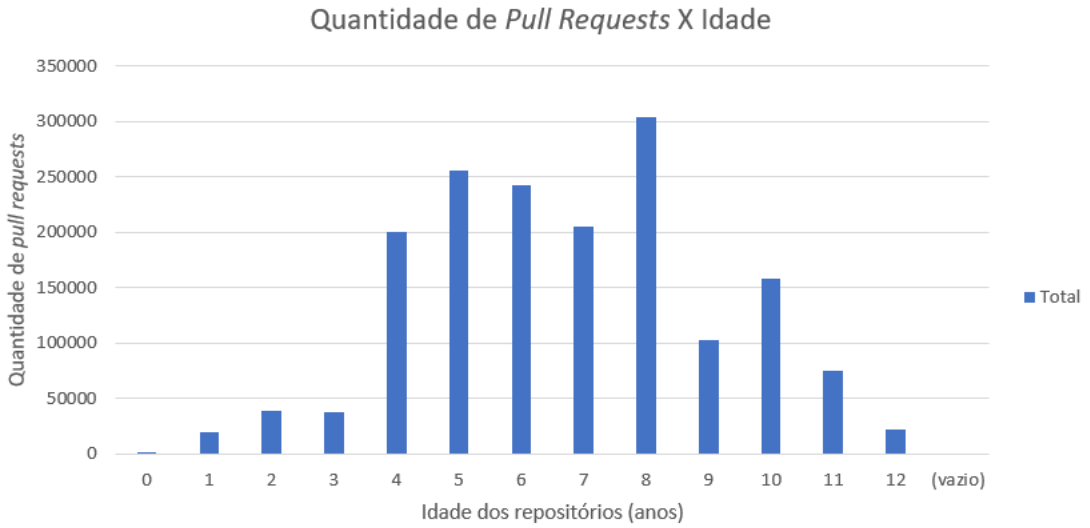
7. *Issues* fechadas por total de issues, dias percorridos da última atualização, quantidade de releases, *pull requests* aceitos e idade em anos de repositórios feitos em linguagens populares e não populares.

3. RESULTADOS OBTIDOS

Os resultados obtidos estão respectivamente relacionados às métricas citadas no tópico anterior.

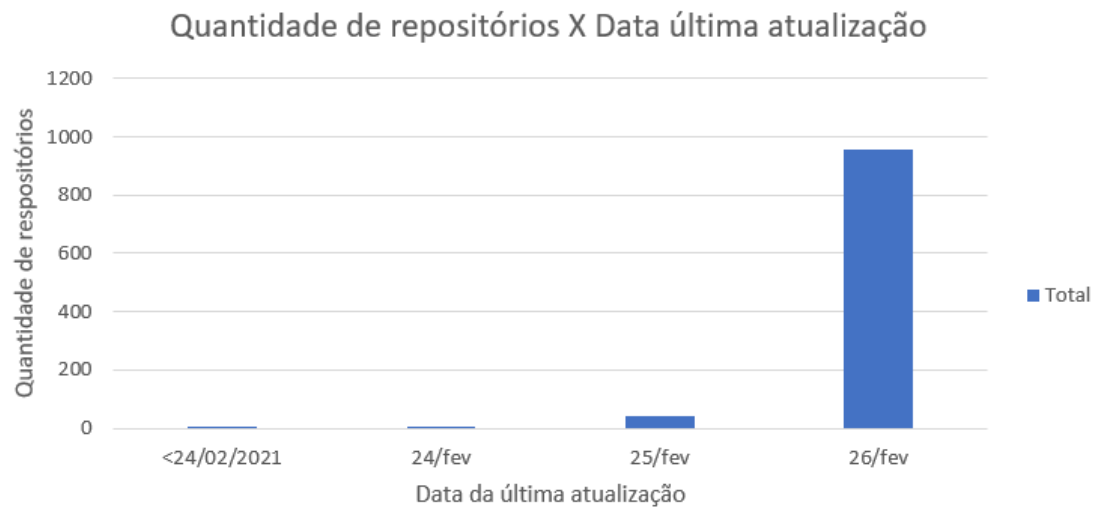
3.1. **Conclusão:** Através dos dados obtidos podemos afirmar que os repositórios mais populares estão presentes na faixa etária de quatro a cinco anos. Dessa forma, conclui-se que a hipótese é falsa, devido ao fato de ter sistemas populares com menos de cinco anos.

3.2. **Conclusão:** A hipótese está inválida, pois, não necessariamente os repositórios mais maduros recebem mais contribuição externa. Analisando os dados, vemos que os repositórios que mais recebem contribuição, são os repositórios com idade intermediária.



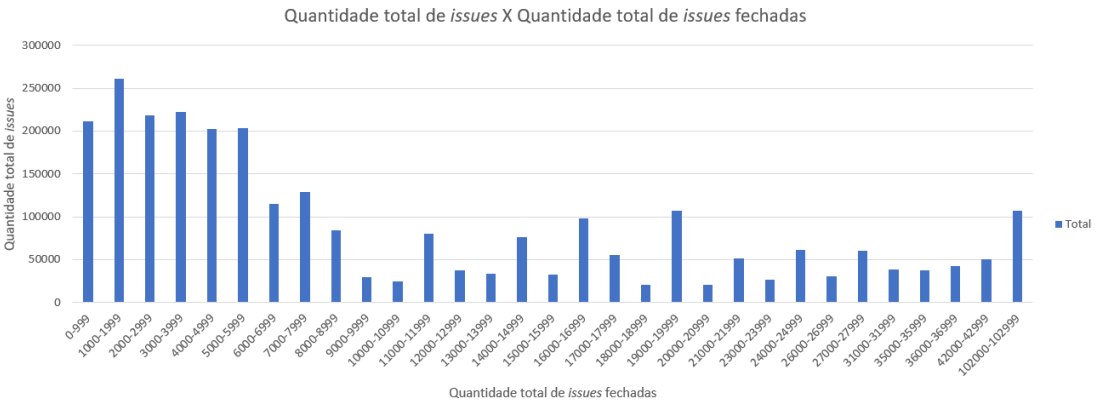
3.3. **Conclusão:** Pode-se observar que a hipótese é verdadeira, devido ao fato de os repositórios novos terem tido menos tempo para lançar grandes números de *releases*.

3.4. **Conclusão:** A hipótese é verdadeira, pois, a maioria dos sistemas foram atualizados próximo a data de coleta dos dados (26/02/2021).

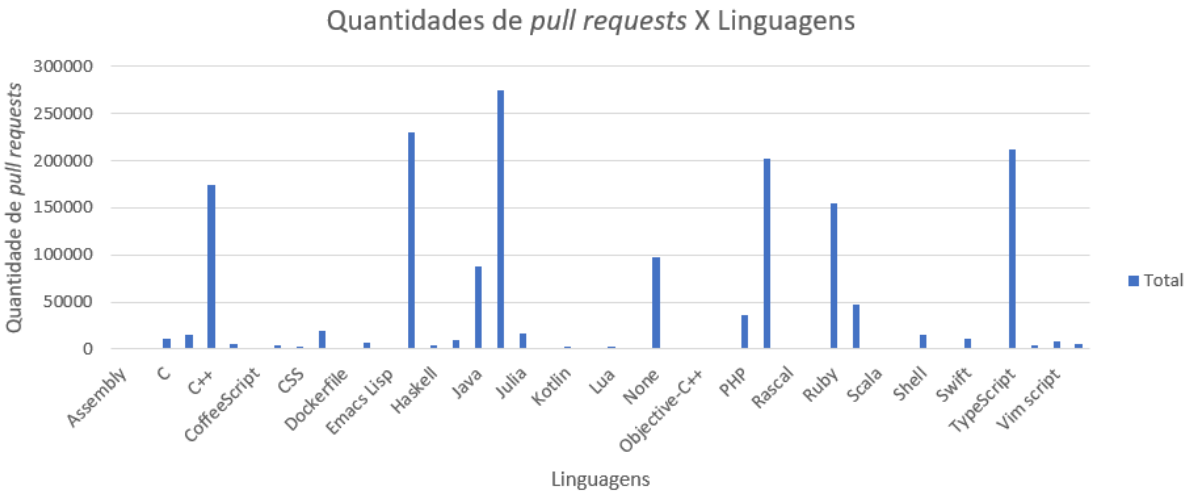


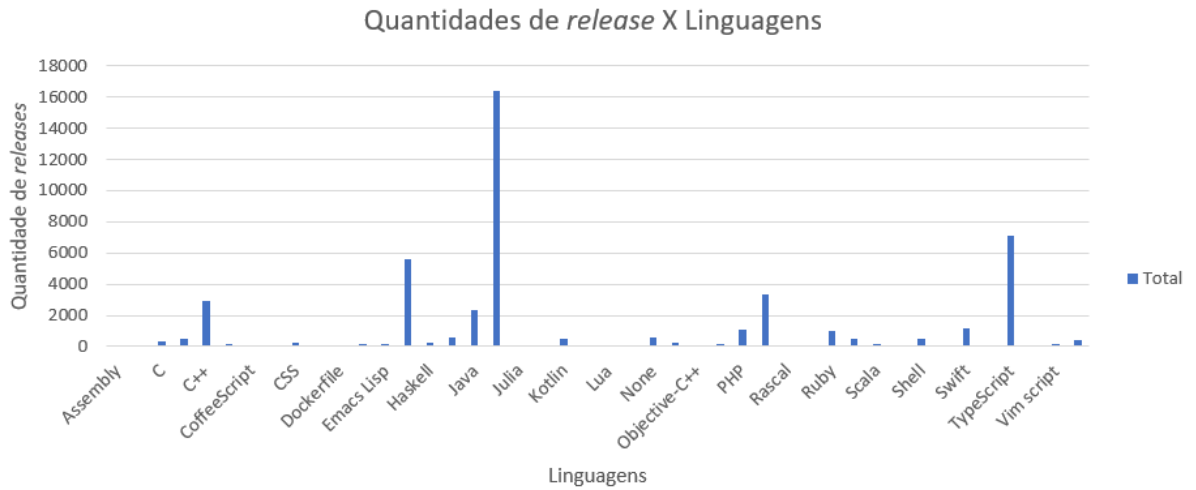
3.5. **Conclusão:** A hipótese é verdadeira, pois, as linguagens mais populares correspondem às mais utilizadas nos repositórios encontrados.

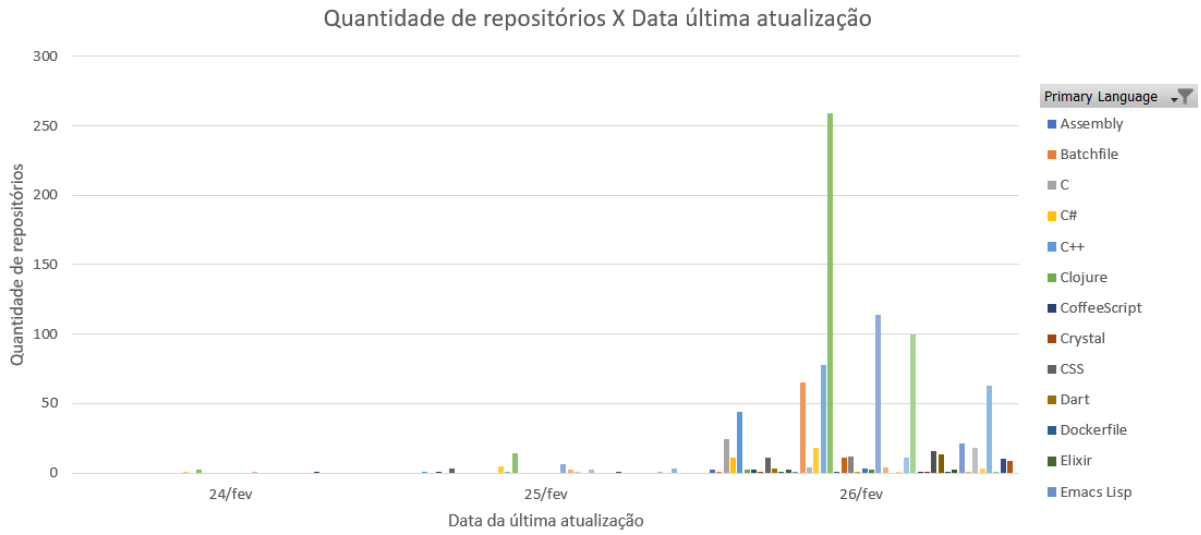
3.6. **Conclusão:** A hipótese é falsa, pois, é perceptível que a maior parte dos repositórios tem grande quantidade de *issues* abertas que ainda não foram fechadas.



3.7. **Conclusão:** A hipótese é falsa, pois, a maior parte dos repositórios que tem grande quantidade de *pull requests* e *quantidade de releases* estão utilizando a linguagem *Julia*, *Haskell* e *TypeScript*. Além disso, os repositórios que possuem as linguagens mais populares não necessariamente foram os mais atualizados até o dia da coleta de dados.


Quantidades de *pull requests* X Linguagens
Quantidade de *pull requests*
300000
250000
200000
150000
100000
50000
0
Assembly
C
C++
CoffeeScript
CSS
Dockerfile
Emacs Lisp
Haskell
Java
Julia
Kotlin
Lua
None
Objective-C++
PHP
Rascal
Ruby
Scala
Shell
Swift
TypeScript
Vim script
Linguagens
Total


Quantidades de *release* X Linguagens
Quantidade de *releases*
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Assembly
C
C++
CoffeeScript
CSS
Dockerfile
Emacs Lisp
Haskell
Java
Julia
Kotlin
Lua
None
Objective-C++
PHP
Rascal
Ruby
Scala
Shell
Swift
TypeScript
Vim script
Linguagens
Total


Quantidade de repositórios X Data última atualização
Quantidade de repositórios
300
250
200
150
100
50
0
24/fev
25/fev
26/fev
Data da última atualização
Primary Language
Assembly
Batchfile
C
C#
C++
Clojure
CoffeeScript
Crystal
CSS
Dart
Dockerfile
Elixir
Emacs Lisp

4. CONCLUSÃO

A partir do trabalho realizado, pôde-se compreender alguns comportamentos a respeito dos repositórios populares do GitHub. A partir das métricas estipuladas, conseguimos observar que a maioria dos repositórios populares *open-source,* possuem de 4 a 8 anos. Além disso, essa faixa possui maior contribuição da comunidade, medida através da quantidade de *pull requests,* e também maior quantidade de releases.

Também foi possível analisar que a maior parte dos repositórios pesquisados possuíram atualizações no dia, ou próximo ao dia, da coleta dos dados. Outro fato importante que foi verificado é que a linguagem principal dos repositórios, condizem com as linguagens mais utilizadas do mercado. Por fim, o experimento mostrou que ainda existe muito trabalho sendo realizados nesses repositórios, tendo em vista o número de *issues* abertas que os repositórios ainda possuem.