PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA

UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE

Bacharelado em Engenharia de Software

Bárbara Bruna Nogueira de Menezes

João Pedro Teles de Andrades Pereira

Relatório de Laboratório de Experimentação de Software

Belo Horizonte

2021

1. Introdução

Este trabalho foi proposto na disciplina de Laboratório de Experimentação de Software com a finalidade de verificar as características dos repositórios dos sistemas populares no GitHub.

Hipóteses e justificativas:

1. Sistemas populares são maduros/antigos?

Métrica: idade do repositório (calculado a partir da data de sua criação).

Hipótese: O GitHub possui 13 anos de existência, levantamos como hipótese que para um repositório ser maduro ele deve ter ao menos 60% da idade do GitHub. Logo, para que um repositório seja considerado maduro, é necessário que ele tenha 8 anos.

2. Sistemas populares recebem muita contribuição externa?

Métrica: Total de Pull Requests aceitas.

Hipótese: A popularidade de um repositório é diretamente proporcional à quantidade de Pull Requests. Tendo isto como base, podemos dizer que a quantidade de Pull Requests de sistemas populares deve ser superior a 500.

3. Sistemas populares lançam releases com frequência?

Métrica: Total de releases

Hipótese: Levando em consideração que um software popular possui, normalmente, uma idade maior e que a idade é diretamente proporcional à quantidade de commits e, consequentemente, melhorias e novas funcionalidades. Temos como hipótese que uma quantidade de releases em um sistema popular é de no mínimo 3 releases/ano.

4. Sistemas populares são atualizados com frequência?

Métrica: Tempo até a última atualização (calculado a partir da data de última atualização)

Hipótese: Um sistema se torna popular por receber e atender aos feedbacks dos usuários, tendo uma necessidade de realizar manutenções de forma frequente e regular. Portanto, um repositório popular deve ser atualizado ao menos uma vez por semana.

5. Sistemas populares são escritos nas linguagens mais populares?

Métrica: Linguagem primária de cada um desses repositórios.

Hipótese: As linguagens de programação mais populares geralmente são as que possuem maior comunidade. Javascript é uma linguagem com grande ascensão no mercado, sendo base para frameworks como React e Angular, seguida de outras linguagens também muito utilizadas como Java e Python. Uma vez que os repositórios mais acessados seguem o mercado, essas linguagens estarão figurando entre as linguagens mais utilizadas nestes repositórios mais populares.

6. Sistemas populares possuem um alto percentual de issues fechadas?

Métrica: Razão entre número de issues fechadas pelo total de issues.

Hipótese: Tendo em vista que um sistema popular busca melhorias e atualizações contínuas, torna-se essencial para estas continuarem evoluindo o software, resolvendo e fechando as issues abertas o mais rápido, cativando sua comunidade. Temos como hipótese que 70% das issues de um repositório popular devem estar fechadas.

2. Método de pesquisa

Para elaborar este trabalho de análise, foi desenvolvido um script em Python que, através de uma query no formato da linguagem de consulta GraphQL, consumia a API do GitHub e, através de paginação, retornava os dados relevantes para a resolução das questões propostas. Ao todo, são recolhidos 1000 repositórios em uma execução do script. Estes repositórios são, posteriormente, exportados para CSV no próprio script. Para visualização do código fonte e do arquivo .csv base utilizado para a análise feita neste trabalho segue o link do meu repositório no GitHub: https://github.com/seteljota/LabEx01.

3. Resultados obtidos

1. Mediana = 6 anos.

2. Mediana de Pull Requests Aceitas = 342,5.

3. A mediana do total de releases de sistemas populares é de 12. Como temos que analisar a quantidade de releases por ano, temos a mediana da idade do repositório igual a 6, como calculado na questão 1. Portanto 12/6 = 2 releases por ano.

4. Os repositórios são atualizados diariamente, pois a mediana do tempo entre atualizações corresponde a 0, ou seja, menos de um dia.

5. Linguagens mais utilizadas de acordo com os repositórios populares analisados (mostrando apenas as 4 mais utilizadas): JavaScript (274), Python (102), Java (78), Go (67).

6. A mediana encontrada para a quantidade total de issues é igual a 1006,5. Para issues fechadas, a mediana é igual a 781,5. A razão entre issues fechadas e issues totais é igual a 0,7764, ou seja, 77,64% das issues totais já foram resolvidas e fechadas.

4. Análise dos resultados em relação à hipótese

1. Através do resultado obtido, concluímos que a mediana dos sistemas populares é de 6 anos, sendo dois anos a menos que os 8 anos esperados para um sistema maduro. Portanto, sistemas populares não são maduros.

2. Através do resultado demonstrou que a mediana de Pull Requests Aceitas foi de aproximadamente 342, corresponde a, aproximadamente, 68% do resultado esperado. Portanto, sistemas populares não recebem muita contribuição externa.

3. A hipótese inicial não foi confirmada, uma vez que a mediana de releases/ano é de 2 releases por ano, sendo maiores que os 3 anos previstos. Portanto, sistemas populares não têm lançado releases com maior frequência.

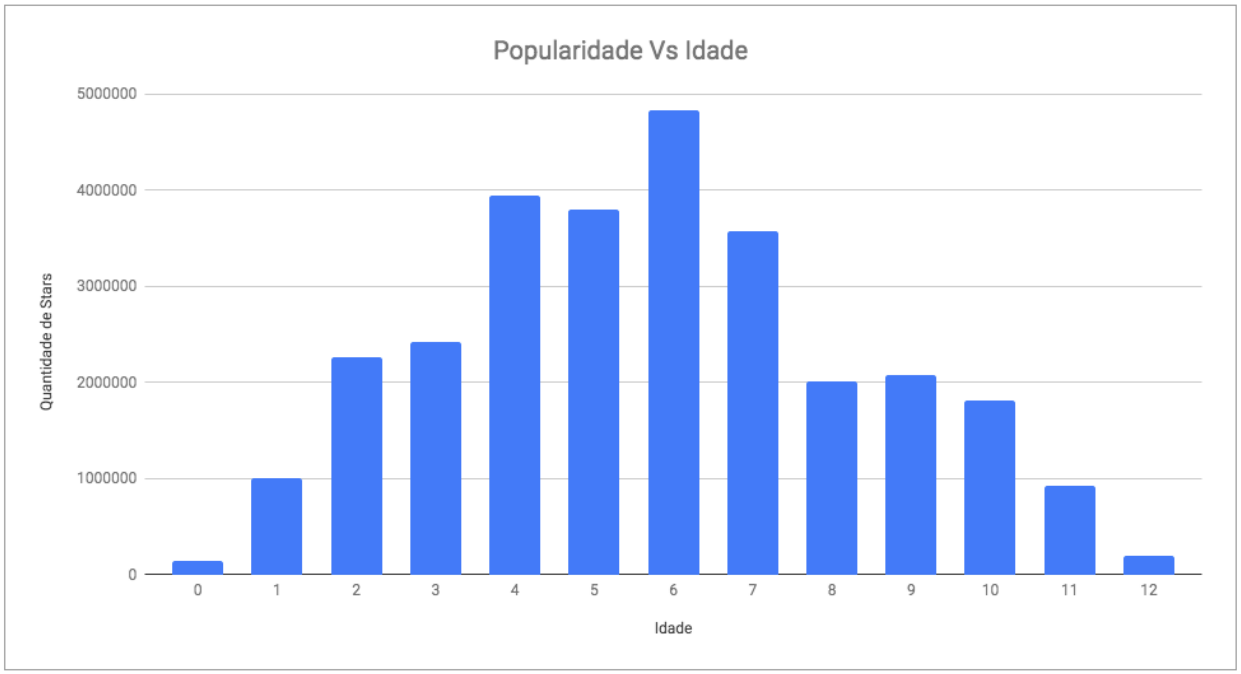
4. A hipótese inicial foi confirmada, uma vez que os sistemas populares são atualizados, de acordo com a mediana, diariamente. Portanto, sistemas populares são atualizados de forma frequente.

5. A hipótese inicial foi confirmada, uma vez que o resultado demonstrou que, JavaScript, Python e Java estavam figurando entre as linguagens primárias mais utilizadas nos repositórios populares. Portanto, sistemas populares são escritos nas linguagens mais populares.

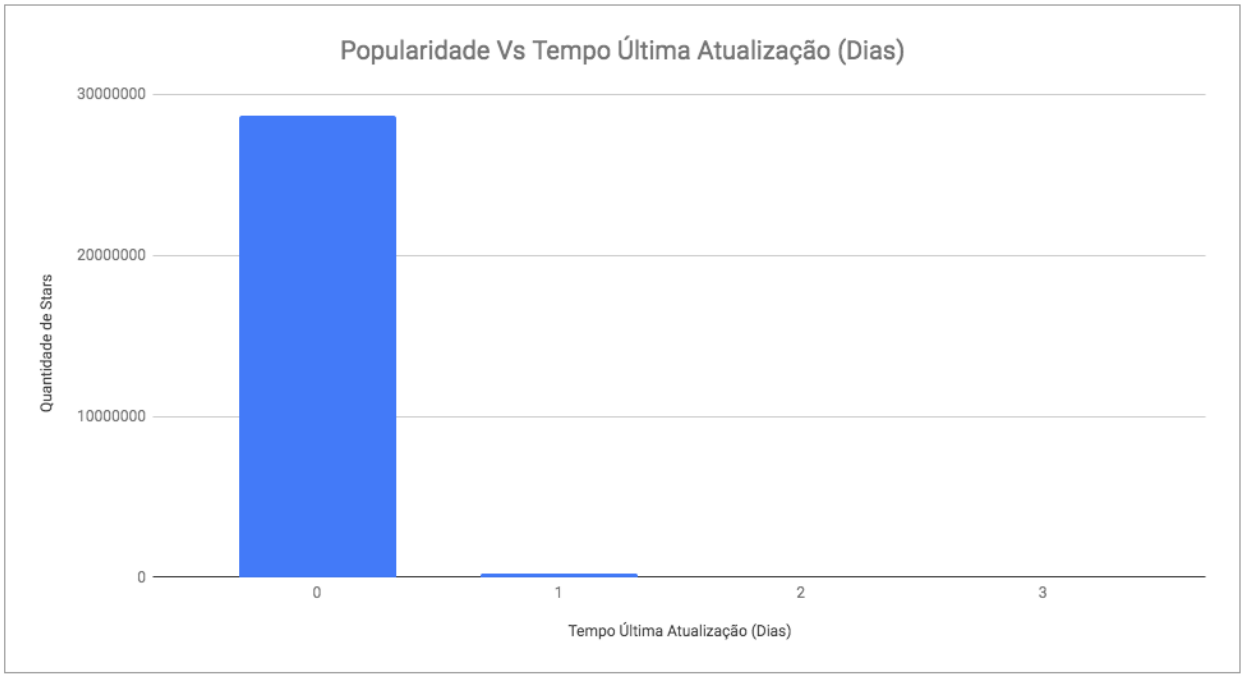
6. A hipótese inicial foi confirmada, uma vez que a porcentagem da razão entre as medianas das issues fechadas e issues totais é de 78%, superando os 70% da hipótese inicial. Portanto, sistemas populares possuem alto percentual de issues fechadas.

5. Gráficos

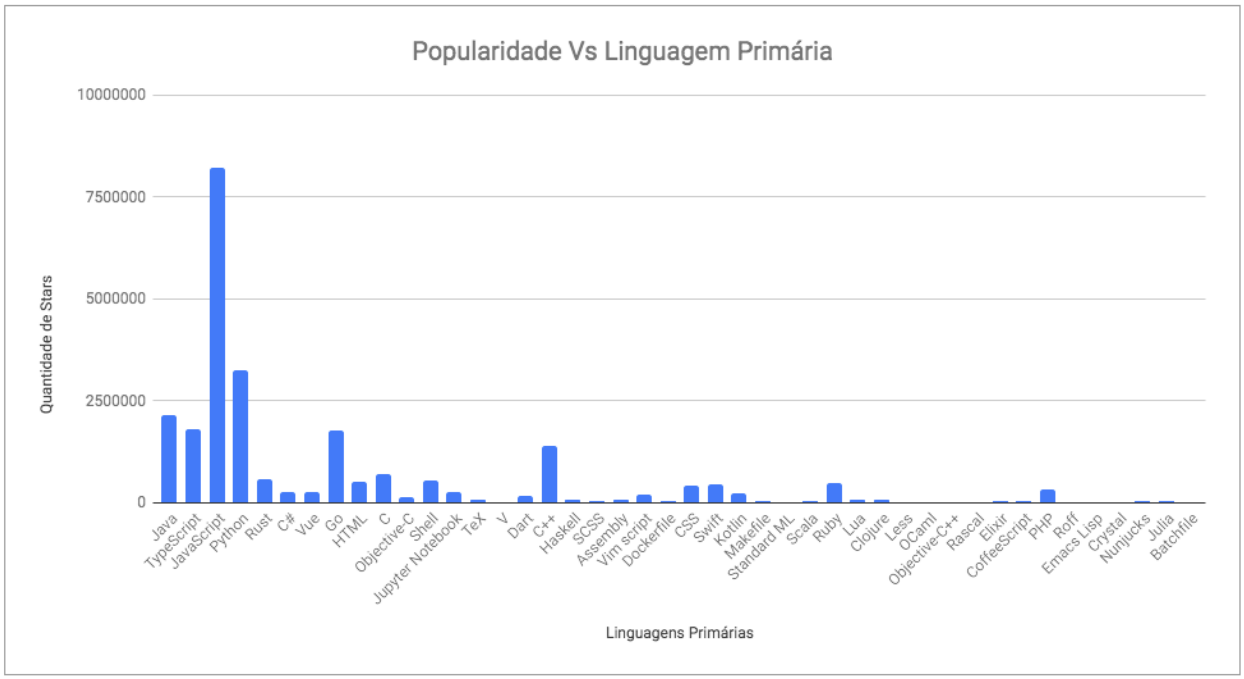
RQ 01:



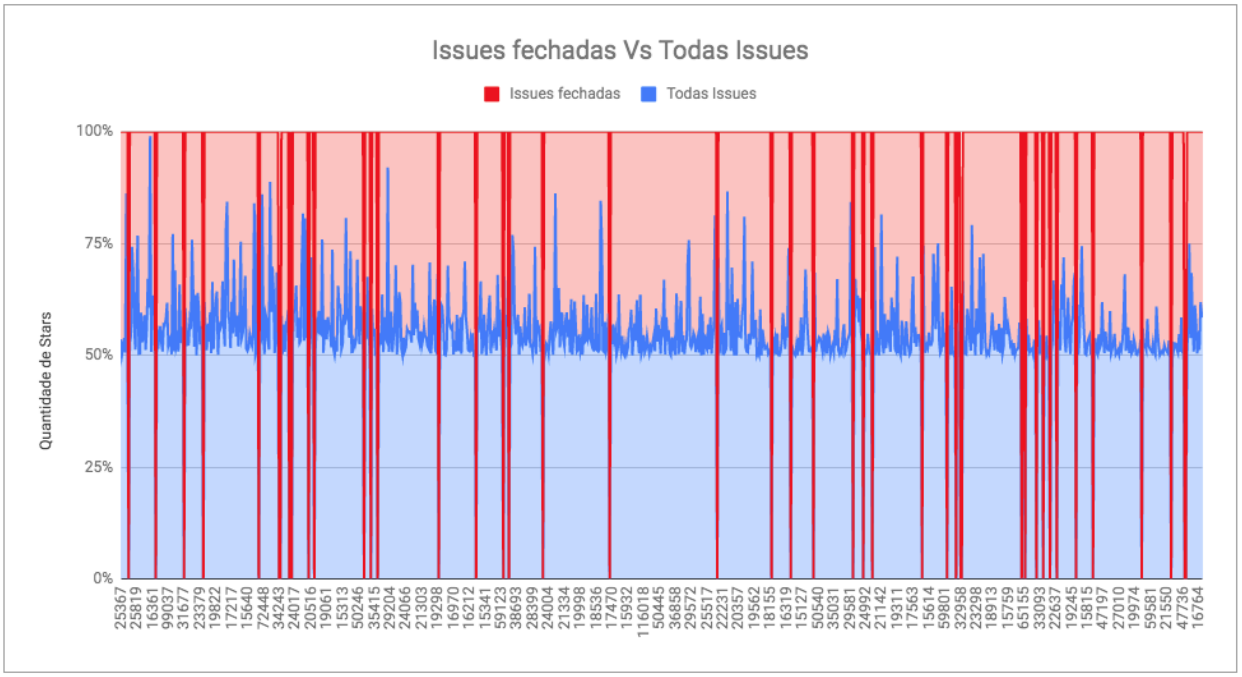
RQ 04:



RQ 05:


Popularidade Vs Linguagem Primária
Quantidade de Stars
10000000
7500000
5000000
2500000
0
Java
TypeScript
JavaScript
Python
Rust
C#
Vue
Go
HTML
C
Objective-C
Shell
Jupyter Notebook
TeX
V
Dart
C++
Haskell
SCSS
Assembly
Vim script
Dockerfile
CSS
Swift
Kotlin
Makefile
Standard ML
Scala
Ruby
Lua
Clojure
Less
OCaml
Objective-C++
Rascal
Elixir
CoffeeScript
PHP
Roff
Emacs Lisp
Crystal
Nunjucks
Julia
Batchfile
Linguagens Primárias

RQ 06:


Issues fechadas Vs Todas Issues
Issues fechadas
Todas Issues
Quantidade de Stars
100%
75%
50%
25%
0%