Laboratório de Experimentação de Software

Relatório Final: Características de Repositórios GitHub Populares

23/09/2020 | Bruno Marini - 634883

Introdução

Com mais de 40 milhões de usuários e a criação de mais de 44 milhões novos repositórios em 2019 [1], o GitHub acaba sendo a plataforma de armazenamento remoto de código versionado mais popular dentre seus concorrentes, principalmente quando falamos de projetos *open source*.

Se atentando a isso, a plataforma também disponibiliza diversas funcionalidades que fomentam o desenvolvimento de software de maneira colaborativa ao mesmo tempo que organizada. Através de *issues*, *pull requests*, *forks* e *wikis*, milhares de usuários podem identificar, contribuir, evoluir e documentar o seu software, contanto que seja público.

Tido esse contexto, este trabalho - proposto na disciplina de Laboratório de Experimentação de Software do curso de Engenharia de Software da PUC Minas no 2º semestre de 2020 - tem o objetivo de, através de uma análise quantitativa, responder uma série de questões que expõem as principais características dos repositórios públicos mais populares (de acordo com o número de estrelas) do GitHub.

Questões de Pesquisa

As questões que fomentam este trabalho são:

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?

Hipóteses

Com o objetivo de conjecturar em cima das questões de pesquisa propostas, para cada uma, foi elaborado uma hipótese - e sua respectiva justificativa - a ser validada a partir da análise dos resultados obtidos. Com isso, podemos responder cada questão com o seu valor quantitativo enquanto, ao mesmo tempo, identificamos tendências sobre os resultados.

Dito isso, as hipóteses são:

1. Quanto mais popular, mais maduros/antigos são os repositórios públicos.

   Justificativa: por possuírem mais tempo de evolução, contribuição e uso, além de mais softwares dependentes e serem mais difundidos entre os desenvolvedores, os repositórios mais maduros/antigos têm mais tempo para se tornarem populares;

2. Quanto mais popular, maior a contribuição externa em repositórios públicos.

   Justificativa: devido a popularidade, são mais conhecidos e disseminados entre os desenvolvedores, fomentando novas contribuições pela alta exposição.

3. Quanto mais popular, maior a quantidade de *release* lançadas em repositórios públicos;

   Justificativa: como uma via de mão dupla, repositórios públicos podem seguir essa prerrogativa tanto porque é a quantidade de *releases* (manutenção e evolução) que mantém seu software relevante (e com isso popular), quanto porque pela popularidade são mais dependentes e mantidos por sua comunidade de contribuidores, este semelhante à justificativa da hipótese 2.

4. Quanto mais popular, mais frequente é a atualização de um repositório público;

   Justificativa: semelhante à justificativa da hipótese 3, um alto nível de popularidade pode demonstrar que muitos softwares e desenvolvedores dependem e utilizam o que é disponibilizado pelo repositório e, pela grande quantidade de utilizadores, torna mais provável tanto a identificação de *bugs* e sugestões de melhorias a serem feitas quanto a manutenção de outros desenvolvedores, levando a mais atualizações.

5. Baseado no *ranking* de linguagens populares, realizado pelo Stack Overflow em 2020 [2], quanto mais popular a linguagem principal, mais popular é um repositório público;

   Justificativa: um repositório com código escrito numa linguagem popular acaba chamando mais atenção, e ganhando mais popularidade com isso, tanto pela demanda

crescente de soluções escritas em tal linguagem quanto pela grande quantidade de desenvolvedores e usuários adeptos.

6. Quanto mais popular, menor o percentual de *issues* fechadas em repositório públicos;

   Justificativa: ao contrário do que é afirmado pela questão de pesquisa relacionada a esta hipótese, acredito que pela alta popularidade, um repositório público acaba tendo muito mais relatos de *issues* e, pelo grande volume, acaba não conseguindo resolver grande parte do que é relatado; ao contrário de repositórios menos populares com menos *issues*.

7. Os repositórios públicos mais populares são escritos nas linguagens mais populares, recebem mais contribuição externa, lançam mais *releases* e são atualizados com mais frequência.

   Justificativa: como resultado das justificativas das hipóteses 2, 3, 4 e 5.

Métricas

Visando testar a validade das hipóteses e, com isso também, responder as questões de pesquisa, a análise de cada uma será baseada nas seguintes métricas:

1. Nº de estrelas e idade do repositório (a partir da data da sua criação);
2. Nº de estrelas e total de *pull requests* aceitos (*merged*);
3. Nº de estrelas e total de *releases*;
4. Nº de estrelas e tempo até o último *push*;
5. Nº de estrelas e linguagem primária;
6. Nº de estrelas e total de issues fechadas (*closed*) sobre o total de issues;
7. Nº de estrelas, linguagem principal, total de *pull requests* aceitos (*merged*), total de *releases* e tempo até o último *push*.

Metodologia

Foi elaborado um *script* em Node.js que, a partir de um *token* válido a API do GitHub, realiza uma busca paginada - da query GraphQL a seguir - enquanto, paralelamente, salva todos os resultados em um arquivo CSV. Tanto o código quanto a sua documentação pode ser encontrada em: https://github.com/TheMarini/LAB-6/tree/v0.2.0.

```
{
  search(query: "stars:>100", type: REPOSITORY, first: 1000) {
   repositoryCount
   pageInfo {
     endCursor
   }
   nodes {
     ... on Repository {
       nameWithOwner
       createdAt
       pushedAt
       stargazers {
         totalCount
       }
       mergedPullRequests: pullRequests(states: MERGED) {
         totalCount
       }
       releases {
         totalCount
       }
       primaryLanguage {
         name
       }
       closedIssues: issues(states: CLOSED) {
         totalCount
       }
       totalIssues: issues {
         totalCount
       }
     }
   }
  }
}
```

Esta busca foi realizada às 18h57 do dia 22/09/2020 para os 1.000 primeiros repositórios públicos, ordenados pela sua quantidade de estrelas, e almejava todos os atributos necessários para cumprir as métricas de cada questão de pesquisa e, consequentemente, de cada hipótese.

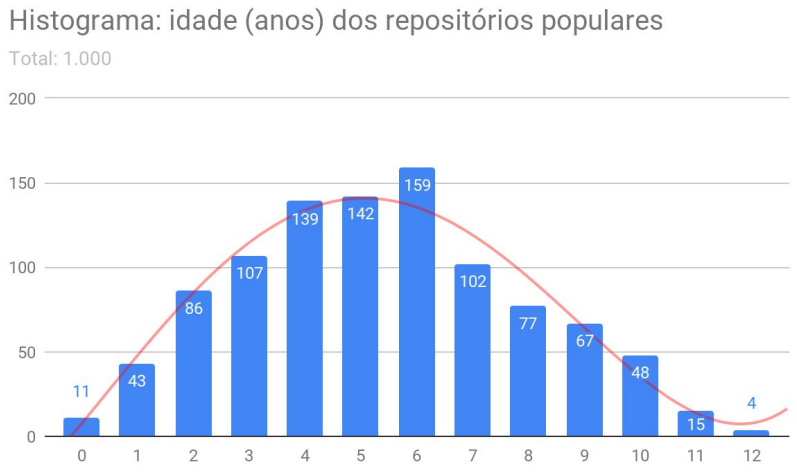
Resultados Obtidos

Como proposto pelo próprio enunciado deste trabalho e para ser justo nos casos que há um repositório com uma quantidade de estrelas muito superior do que os demais na mesma categoria - como por idade, linguagem principal, nº de *releases* e etc. -, os valores a seguir representam a **mediana** [3] dos resultados obtidos para questão de pesquisa e hipótese.

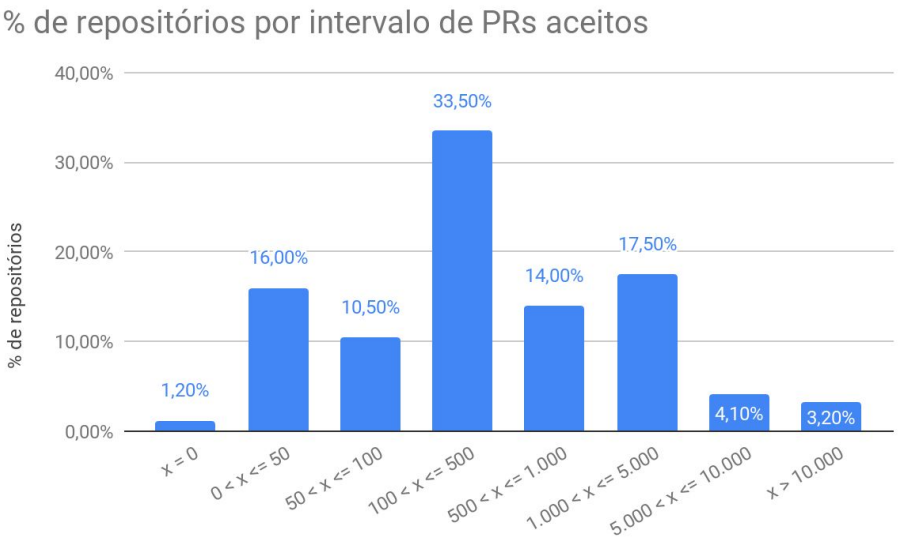
Dito isso, segue os resultados obtidos.

Questões

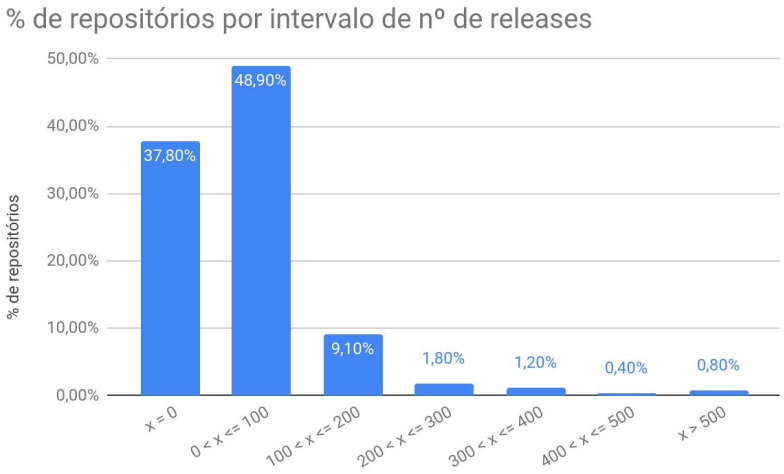
1. ***"Sistemas populares são maduros/antigos?"***
   Sim. Como demonstrado no histograma a seguir, a maior quantidade de sistemas populares se concentra com idade entre 4 e 6 anos.



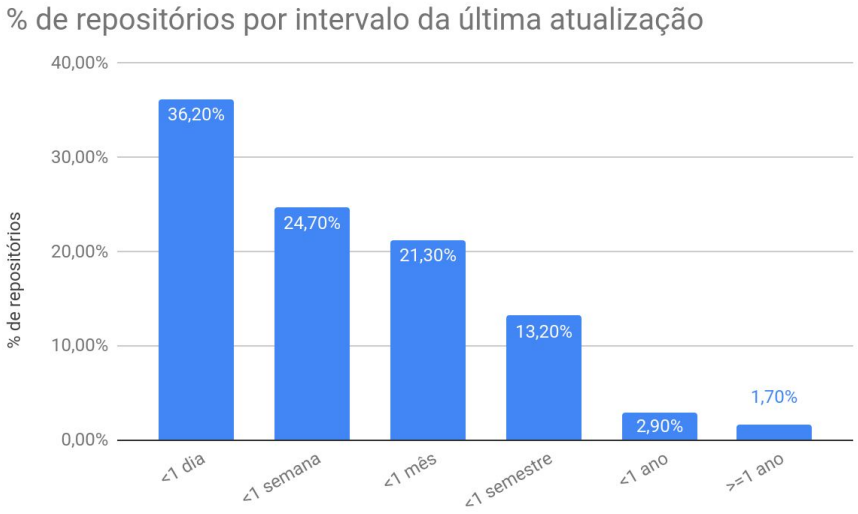
2. ***"Sistemas populares recebem muita contribuição externa?"***
   Sim. Como demonstrado no gráfico a seguir, 47,50% dos repositórios públicos populares possuem entre 100 e 500 PRs aceitos, um número bem significativo de contribuições externas.



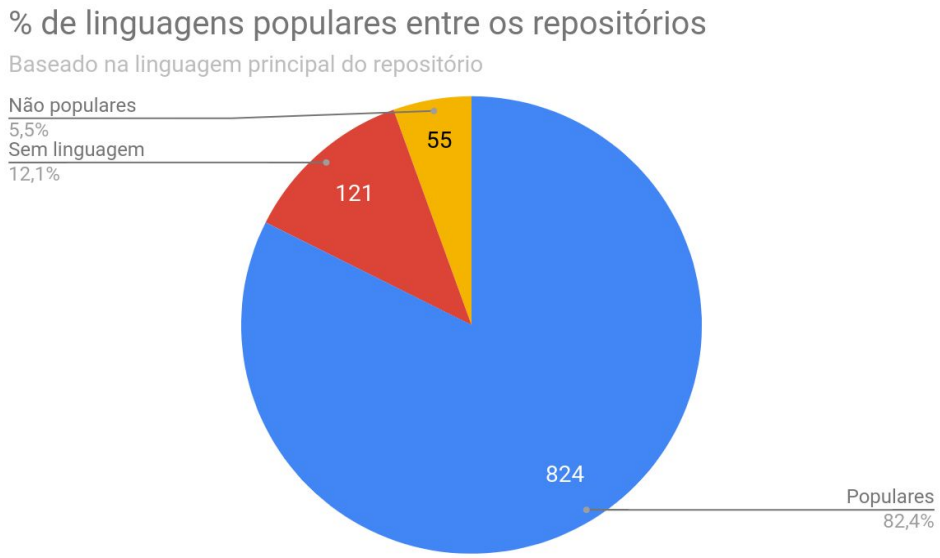
3. ***"Sistemas populares lançam releases com frequência?"***
   Não. Como demonstrado no gráfico a seguir, cerca de 87,70% dos repositórios públicos populares analisados possuem menos que 100 *releases*. Por mais que seja um valor significativo quando lembramos que cada *release* é a disponibilização de uma nova versão em produção, cerca de 13,30% dos repositórios possuem uma quantidade superior de *releases*.

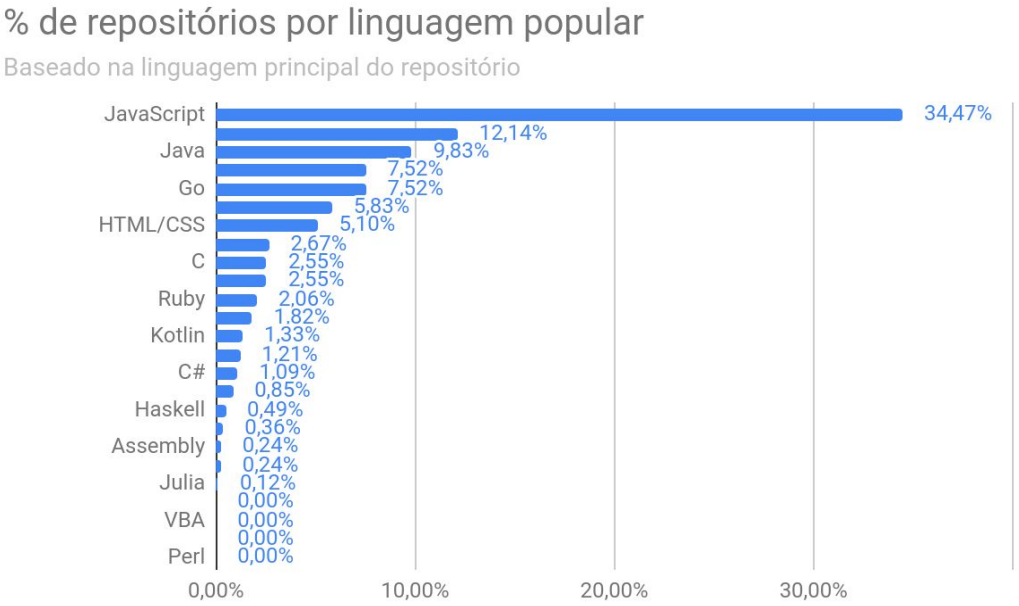


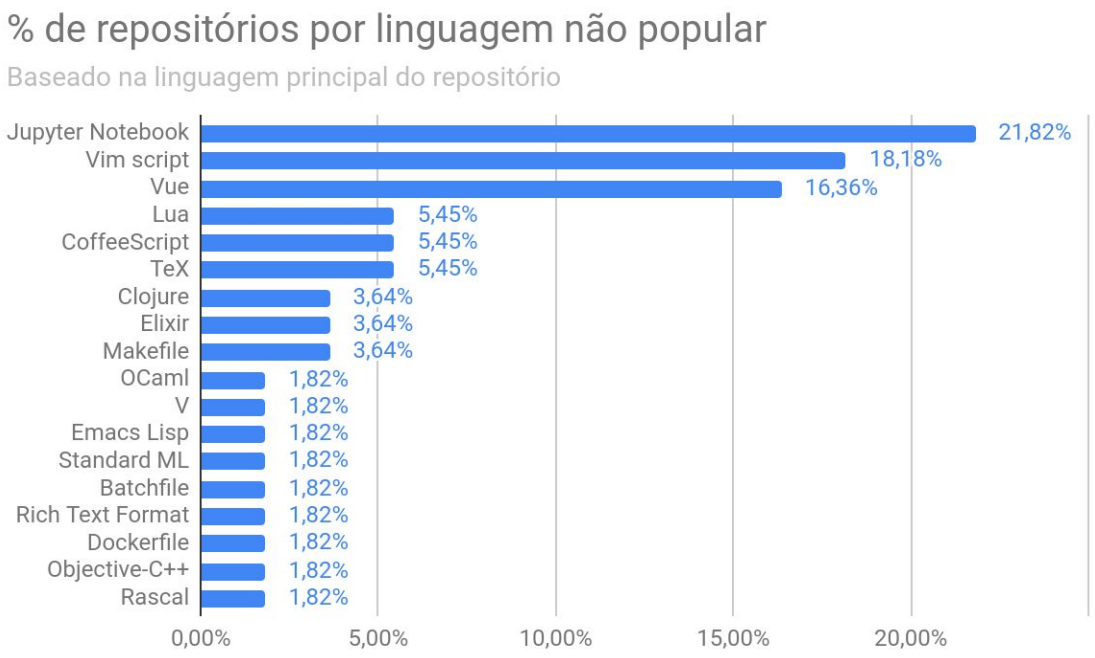
4. ***"Sistemas populares são atualizados com frequência?"***
   Sim. Como demonstrado no gráfico a seguir, cerca de 60,90% dos repositórios públicos populares analisados foram atualizados em um intervalo inferior de uma semana e 82,20% dentro de um mês. Enquanto isso, somente 17,80% foram atualizados em um período superior.



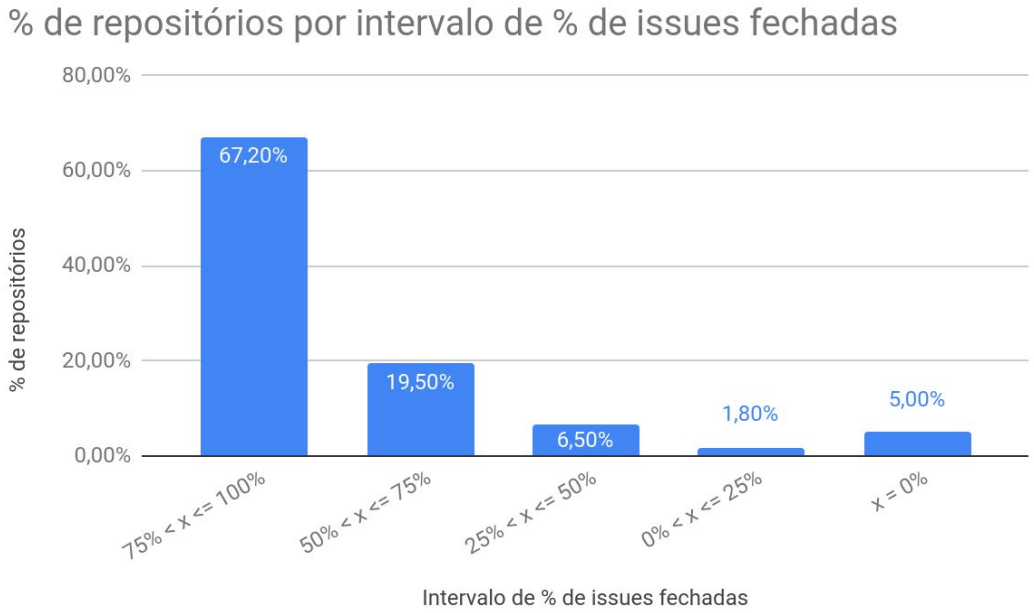
5. ***"Sistemas populares são escritos nas linguagens mais populares?"***
   Sim. Como demonstrado no primeiro gráfico a seguir, 82,40% dos repositórios públicos populares tem como principal linguagem uma que é considerada popular pelo Stack Overflow [2]. Além disso, como demonstrado no segundo e terceiro gráfico, 34,47% dos repositórios com linguagem principal popular são desenvolvidos em Javascript, enquanto 21,82% dos repositórios sem linguagem principal popular são desenvolvidos em Jupyter Notebook.





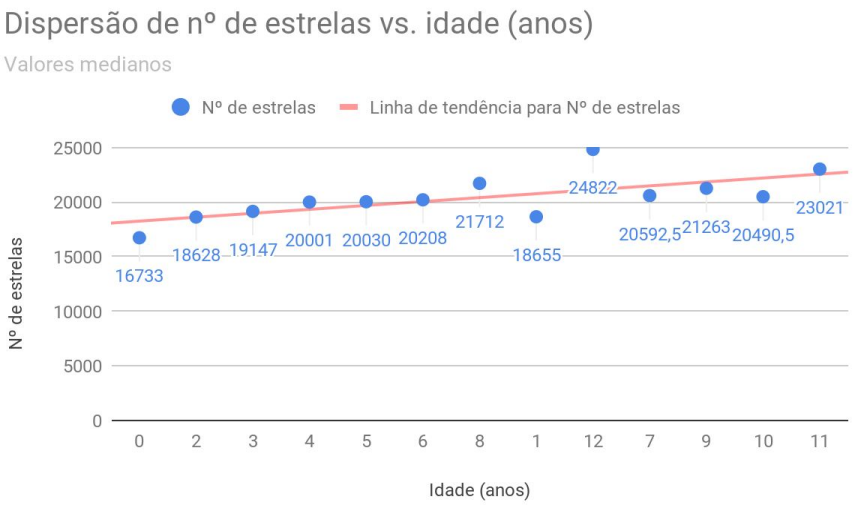


6. ***"Sistemas populares possuem um alto percentual de issues fechadas?"***
   Sim. Como demonstrado no gráfico a seguir, 62,20% dos repositórios públicos populares têm entre 75% e 100% das suas issues fechadas, enquanto menos que ⅓ disso possui um entre 50% e 75%.

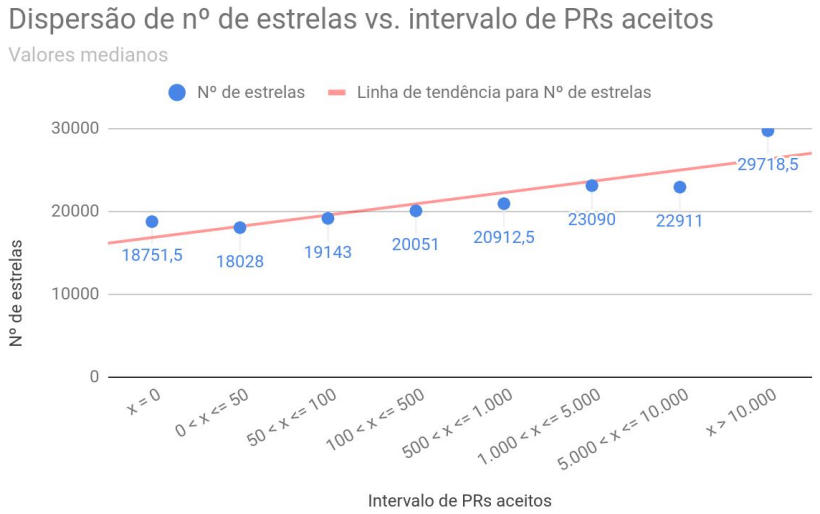


Hipóteses

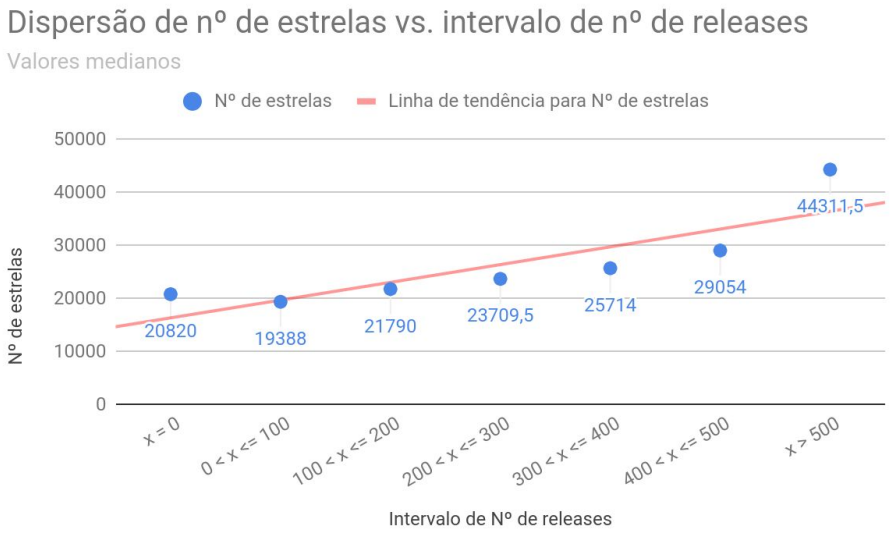
1. ***"Quanto mais popular, <u>mais maduros/antigos</u> são os repositórios públicos"***
   Sim. De acordo com os resultados obtidos e como demonstrado no gráfico a seguir, há uma correlação forte entre a quantidade de estrelas e a idade do repositório. Ou seja, quanto mais velho o repositório maior a tendência de ter um número maior de estrelas, portanto, mais popular.



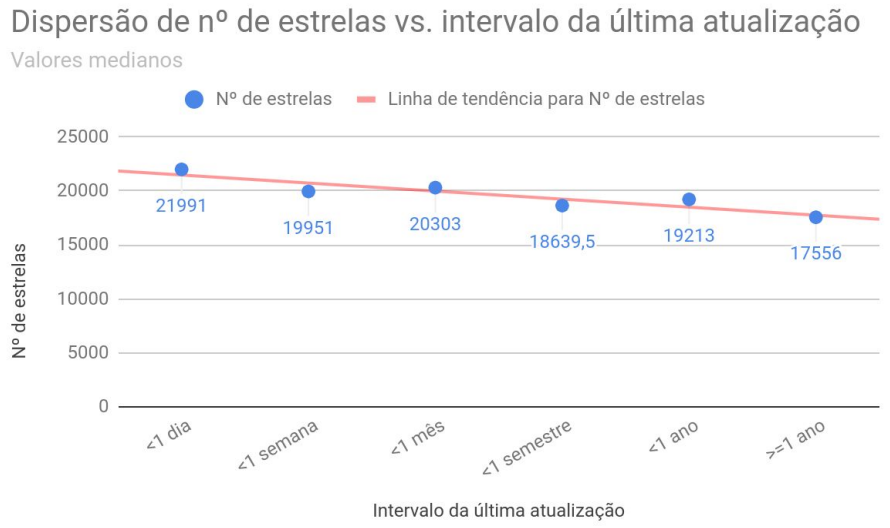
2. ***"Quanto mais popular, <u>maior a contribuição externa</u> em repositórios públicos"***
   Sim. Como demonstrado no gráfico de dispersão a seguir, há uma forte correlação entre a quantidade de PRs aceitas e a popularidade do repositório público (nº de estrelas).



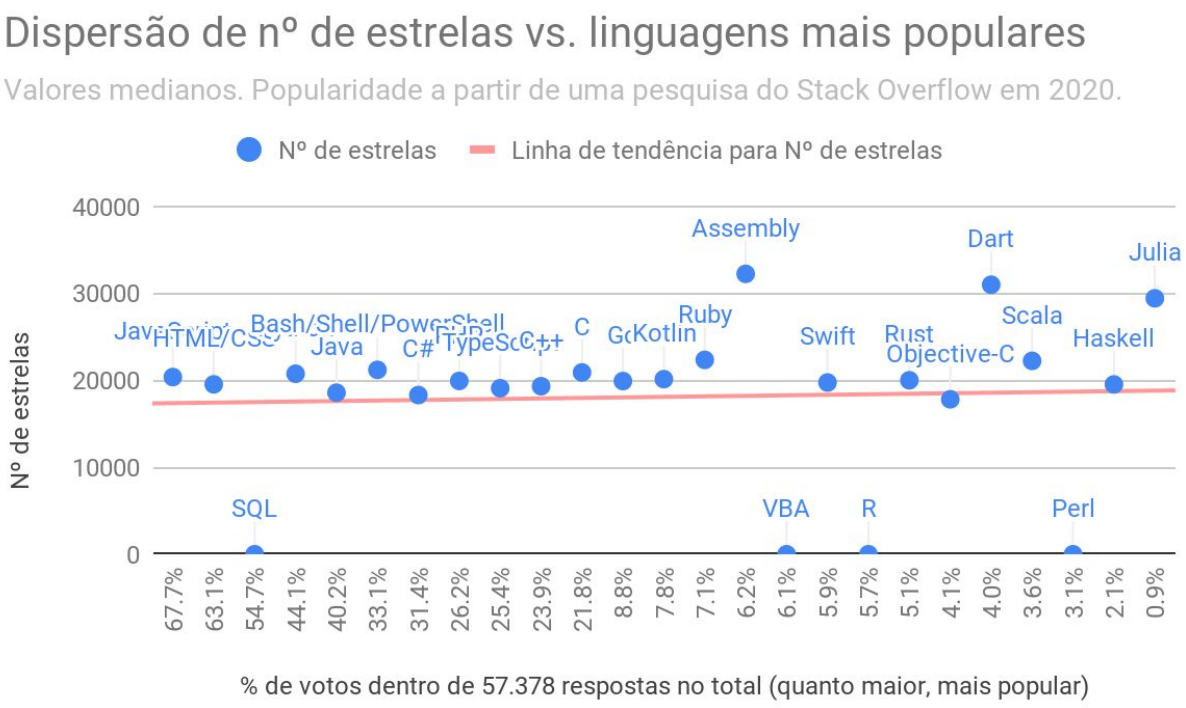
3. ***"Quanto mais popular, <u>maior a quantidade de release lançadas</u> em repositórios públicos"***
   Sim. Como o gráfico de dispersão a seguir demonstra, há uma leve correlação entre a quantidade de releases e a popularidade de um repositório público (nº de estrelas).



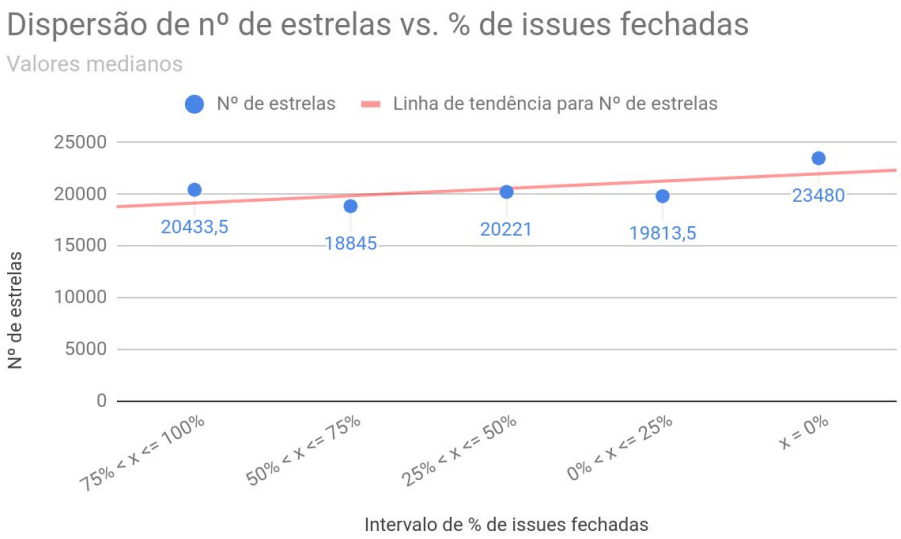
4. ***"Quanto mais popular, <u>mais frequente</u> é a atualização de um repositório público"***
   Inconclusivo. Por mais que o gráfico de dispersão a seguir demonstre uma certa correlação entre a frequência de atualização e a popularidade de um repositório público (nº de estrelas), não há uma diferença considerável no nº de estrelas entre os menos atualizados dos mais atualizados. Muito provavelmente, precisaríamos de um limite superior à 1.000 repositórios para identificarmos uma diferenciação mais abrupta.



5. ***"Baseado no ranking de linguagens populares, realizado pelo Stack Overflow em 2020 [2], quanto mais popular a linguagem principal, mais popular é um repositório público"***
   Não. Como demonstrado no gráfico de dispersão a seguir, não há nenhuma correlação entre o nível de popularidade da linguagem principal de um repositório público com o seu nº de estrelas (popularidade). Não obstante que os repositórios escritos em Javascript, HTML/CSS e SQL - as linguagens mais populares - possuem uma mediana do nº de estrelas muito inferior às linguagens menos populares, como Assembly, Dart e Julia.



6. ***"Quanto mais popular, menor o percentual de issues fechadas em repositório públicos"***
   Não. Como demonstrado no gráfico de dispersão a seguir, não há nenhuma correlação entre o nível de popularidade e o intervalo de % de *issues* fechadas. Não obstante que os repositórios com nenhuma *issue* fechada têm uma mediana do nº de estrelas superior a todos os outros com maior % de *issues* fechadas.



Conclusão

Através de uma análise quantitativa sobre os 1.000 repositório públicos mais populares do GitHub, foi possível validar algumas questões e dúvidas constantes desse ambiente de projetos *open source* enquanto, ao mesmo tempo, conjecturar sobre os outros repositórios que não entraram no escopo deste trabalho. Assim, em caso de dúvidas acerca de algum dos critérios avaliados aqui, é possível inferir conclusões mais assertivas e embasadas pelos dados analisados.

Referências

1. The expanding Octoverse. Disponível em: https://octoverse.github.com (acessado em 23/09/2020);
2. Most Popular Technologies: Programming, Scripting, and Markup Languages. Disponível em: https://insights.stackoverflow.com/survey/2020/#most-popular-technologies (acessado em 23/09/2020);
3. Measures of Central Tendency: Mean, Median, and Mode. Disponível em: https://www.sciencebuddies.org/science-fair-projects/science-fair/summarizing-your-data#meanmedianandmode (acessado em 25/09/2020).