Principais características de projetos open-source no Github

Hugo Poletto Alacoque Gomes [1],Matheus Nolasco [1], Maria Aryene Costa [1], Lucas Santos Rosa [1]

[1] Instituto de Ciências Exatas e Informática
Pontifícia Universidade de Minas Gerais (PUC Minas)
Belo Horizonte – MG – Brasil

{Hugo Poletto Alacoque Gomes, Matheus Nolasco Miranda Soares, Maria Aryene Costa dos Santos,Lucas Santos Rosa}@sga.pucminas.br

Resumo. *O GitHub é a maior plataforma de projetos open source do mundo, sendo usado por diversos desenvolvedores ao redor do mundo para compartilhar o código-fonte do software e criar comunidades. Neste projeto estaremos analisando as principais características dos projetos open source por meio de dados quantitativos coletados diretamente da API oficial do GitHub.*

1. Introdução

Neste projeto pretende-se estudar e analisar um grupo de 1.000 repositórios populares e de *open source* no Github com o intuito de analisar como eles são desenvolvidos, com que frequência recebem contribuições externas, entre outras características. Esse projeto tem como objetivo responder às seguintes questões:

1. Os sistemas populares são maduros/antigos?
2. Os sistemas populares recebem muita contribuição externa?
3. Os sistemas populares lançam releases com frequência?
4. Os sistemas populares são atualizados com frequência?
5. Os sistemas populares são escritos nas linguagens mais populares?
6. Os sistemas populares possuem um alto percentual de issues fechadas?

A partir de cada uma dessas perguntas, uma hipótese inicial foi formulada com o resultado esperado. Essas "hipóteses informais" são:

- Sistemas populares não são necessariamente maduros tendo em vista que a popularização de um sistema não está correlacionada a sua idade, mas sim, a outros fatores.
- Sistemas populares recebem muita contribuição externa pois, por serem populares, possuem muita visibilidade e utilização.
- Sistemas populares lançam release com frequência, pois, como possuem muitas contribuições, estão constantemente sendo atualizados.
- Os sistemas populares são atualizados com frequência, já que a popularidade depende do engajamento dos desenvolvedores com certa frequência.
- Sistemas populares não são necessariamente escritos nas linguagens mais populares, visto que, o que mais impacta na sua popularidade não é a tecnologia de sua utilização.

- Sistema populares não necessariamente possuem alto percentual de issues fechadas pois, por atraírem muitos contribuidores, também atraem desenvolvedores que começam alguma issue mas não finalizam ou demoram muito tempo para finalizá-las.

Para responder às perguntas da pesquisa e confirmar ou não as hipóteses, os dados coletados do github serão analisados em um arquivo CSV.

2. Metodologia

O trabalho apresentado foi conduzido utilizando a abordagem empírica quantitativa. De acordo com (Gomez & Reidl, 2010), a abordagem empírica engloba observação, medição e experimentação. Em complemento, a pesquisa quantitativa, para Mattar (2001), busca a validação das hipóteses mediante a utilização de dados estruturados, estatísticos, com análise de muitos casos representativos.

A coleta foi realizada no Visual Studio Code consultando a API do GitHub via GraphQL. Segundo (Samer, 2012), a linguagem de consulta GraphQL simplifica as interações com servidores web, possibilitando consultas de API mais inteligentes que podem melhorar enormemente a eficiência das solicitações de dados.

A implementação do experimento foi dividida em três sprints. Na primeira sprint, foram coletados via GraphQL os 100 repositórios com maior número de estrelas no GitHub. O trabalho desenvolvido utilizou como roteiro as seguintes perguntas e os parâmetros do GitHub descritos abaixo:

RQ 01. Sistemas populares são maduros/antigos?

Métrica: idade do repositório

RQ 02. Sistemas populares recebem muita contribuição externa?

Métrica: total de pull requests aceitas

RQ 03. Sistemas populares lançam releases com frequência?

Métrica: total de releases

RQ 04. Sistemas populares são atualizados com frequência?

Métrica: data até a última atualização

RQ 05. Sistemas populares são escritos nas linguagens mais populares?

Métrica: linguagem primária de cada um desses repositórios

RQ 06. Os sistemas populares possuem um alto percentual de issues fechadas?

Métrica: razão entre número de issues fechadas pelo total de issues

Na segunda sprint, a consulta foi realizada com 1.000 repositórios, sendo desenvolvidas nessa fase, as hipóteses informais supracitadas na introdução. Ainda na segunda fase, foi realizada a exportação dos dados dos 1.000 repositórios em formato de arquivo CSV para análise posterior.

Na terceira e última sprint, foi desenvolvida a documentação do experimento, contendo metodologia, resultados, discussão e trabalhos futuros.

3. Resultados

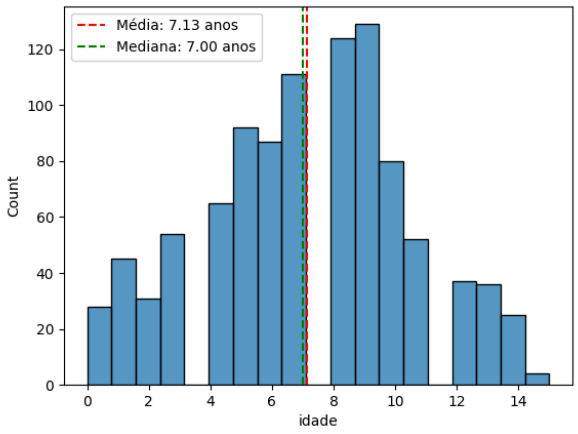
Triola (2023) define Medidas de Tendência Central (MTC) como "valores numéricos que caracterizam o centro de um conjunto de dados". Elas fornecem uma ideia geral de como os dados estão distribuídos e facilitam a comparação entre diferentes conjuntos de dados" (p. 31).

Neste estudo, as MTC foram utilizadas para analisar a distribuição dos dados.

3.1 Apresentação dos resultados

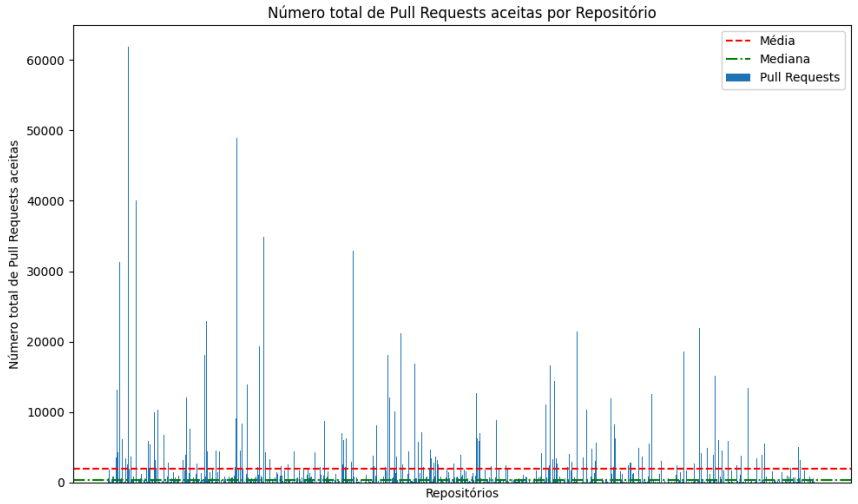
RQ 01. Sistemas populares são maduros/antigos?

Métrica: idade do repositório



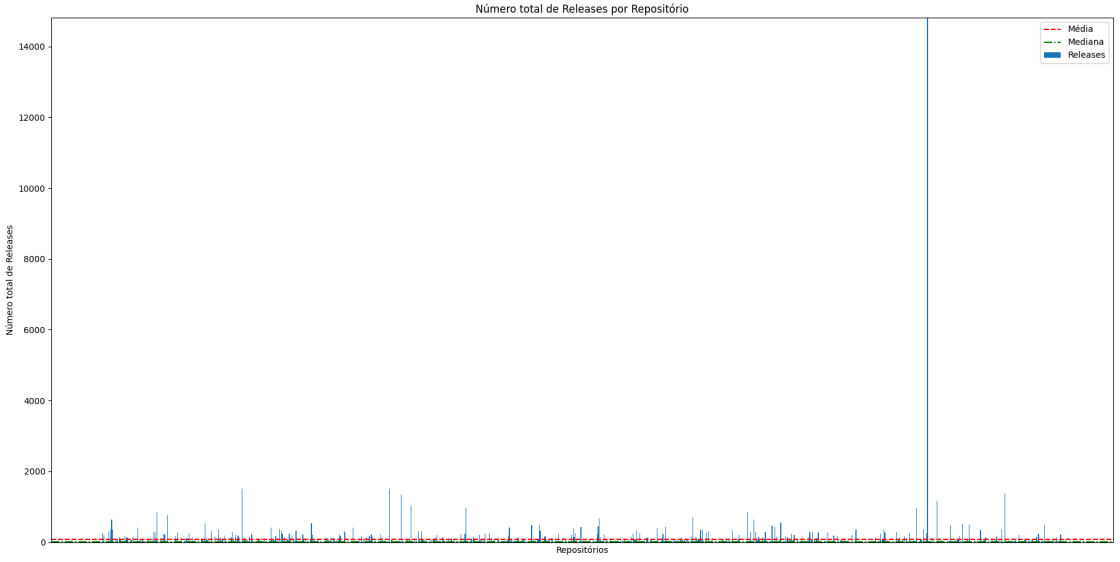
RQ 02. Sistemas populares recebem muita contribuição externa?

Métrica: total de pull requests aceitas



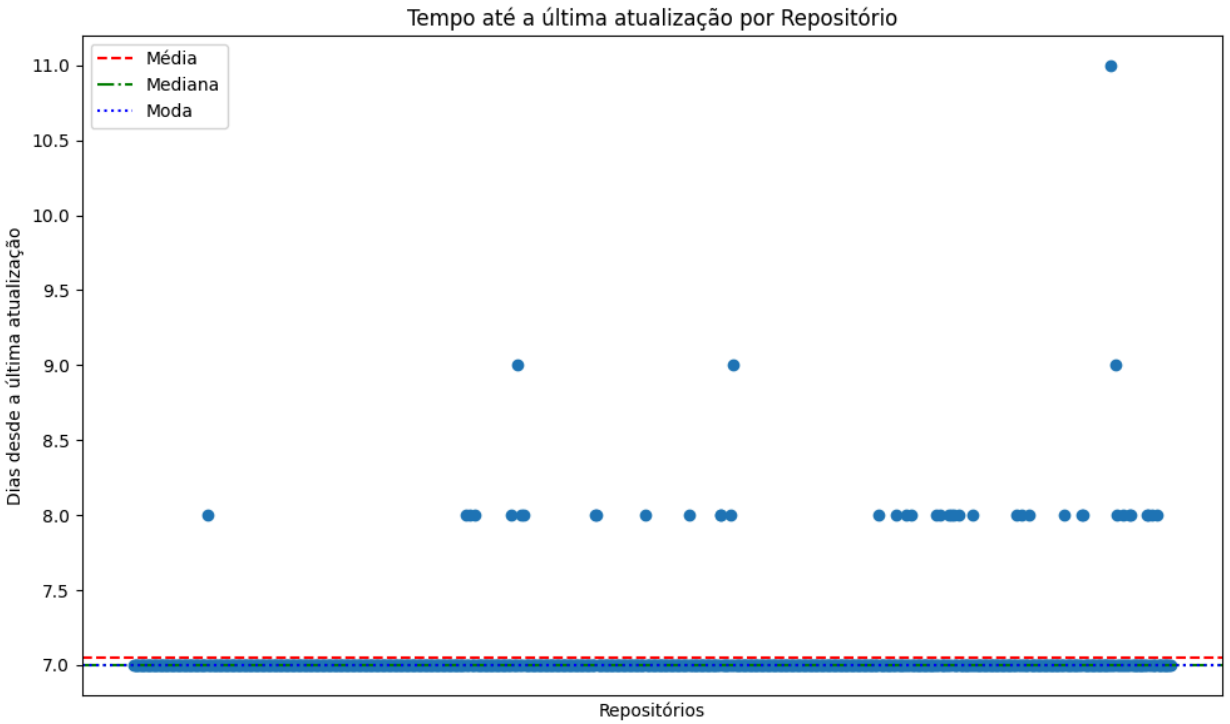
RQ 03. Sistemas populares lançam releases com frequência?

Métrica: total de releases



RQ 04. Sistemas populares são atualizados com frequência?

Métrica: Data até a última atualização



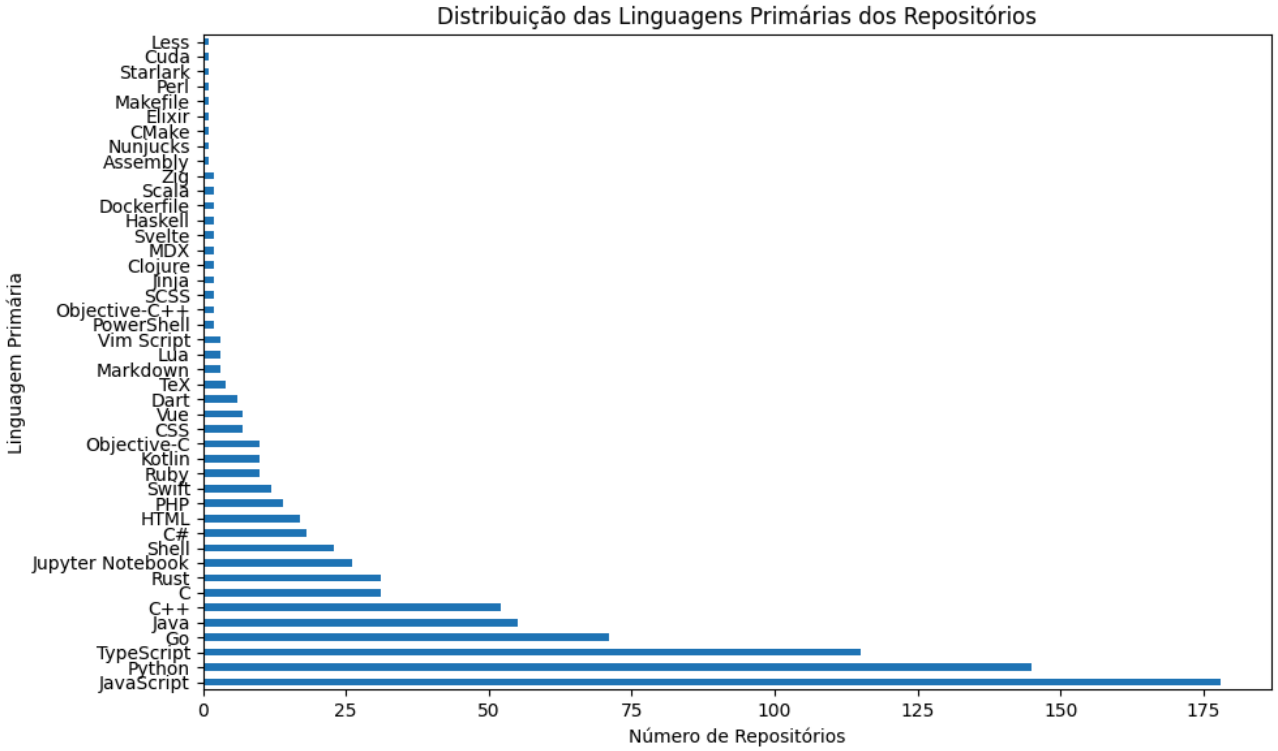
Média: 7.049 dias

Mediana: 7.0 dias

Moda: 7 dias

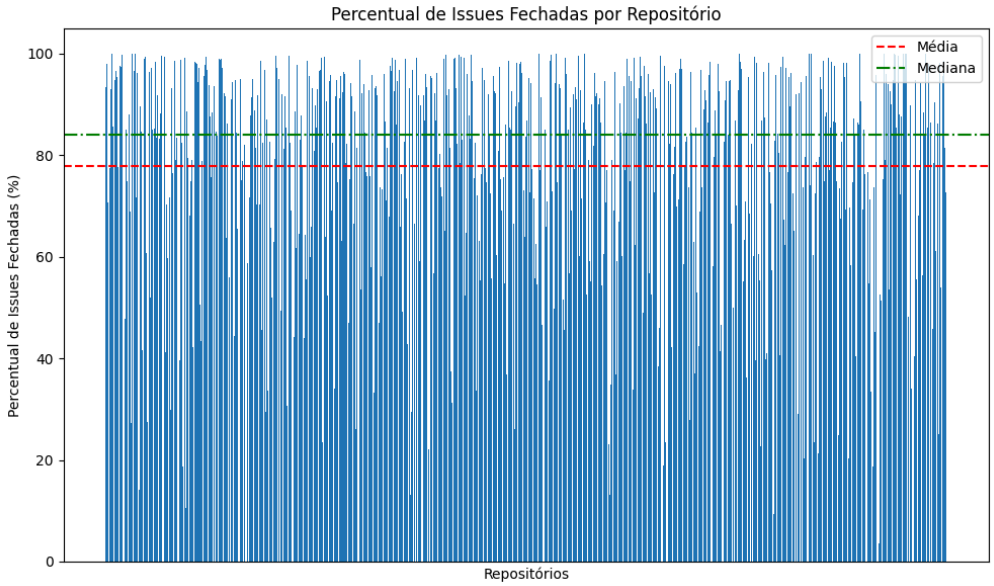
RQ 05. Sistemas populares são escritos nas linguagens mais populares?

Métrica: linguagem primária de cada repositório



RQ 06. Os sistemas populares possuem um alto percentual de issues fechadas?

Métrica: razão entre número de issues fechadas pelo total de issues



Média: 77.91

Mediana: 84.05

3. Discussão

De acordo com os resultados obtidos, torna-se possível observar se as hipóteses iniciais condizem com os resultados, sendo eles:

1. Os sistemas mais populares tendem a se popularizar quando chegam em 7 anos e decair a sua popularidade ao passar do tempo.
2. É possível observar que os sistemas mais populares recebem mais contribuições externas, já que o número de pull requests é maior em repositórios com mais estrelas no GitHub.
3. A quantidade de releases em um repositório não está diretamente relacionada à sua popularidade.
4. Os repositórios mais populares do Github não são necessariamente atualizados com mais frequência.
5. As linguagens de programação mais populares do GitHub como Javascript, Python, Java, C#, C++, entre outros, estão constatadas no topo dos repositórios mais populares, logo pode-se observar que a popularidade do repositório é ligada à sua linguagem de programação primária.
6. O percentual de issues fechadas não é maior para projetos open-source mais populares do GitHub.

4. Conclusões e trabalhos futuros

A partir da aplicação das métricas determinadas e da análise dos resultados obtidos pela API do GitHub, foi possível responder a cada uma das perguntas que faziam parte do objetivo deste trabalho. Sendo assim, as seguinte conclusões foram alcançadas:

1. Sistemas populares possuem uma média de 7 anos de idade, o que leva a crer que em sua maioria, os sistemas necessitam de um tempo de maturação para se tornarem populares.
2. Sistemas populares recebem em média 1974 Pull Requests, comparando esse número com a média de idade dos repositórios, pode-se estimar que um repositório popular recebe em média 23 Pull Requests por mês. Dito isso, pode-se concluir que a popularidade de um sistema está diretamente relacionada à quantidade de Pulls Request que ele recebe.
3. Sistemas populares lançaram em média 86 Releases, comparando esse número com a média de idade dos repositórios, é possível estimar que um repositório popular realiza em média 1 release por mês. Tendo isso em vista, é possível concluir que sistemas populares, por receberem muitos pull requests, lançam muitos releases, ou seja, ambos também são diretamente proporcionais.
4. Sistemas populares tiveram sua atualização mais recente lançada, em média, a 7 dias antes da coleta dos dados. Relacionando esse número com o tempo de vida médio dos sistemas, é possível concluir que mesmo os sistemas precisando de

um tempo de maturação para se tornarem populares, essa popularidade não se esvai, ou seja, eles continuam recebendo atualizações constantemente.

5. Ao levar em conta as informações coletadas e o gráfico a seguir que foi disponibilizado no relatório anual de desenvolvimento do GitHub:

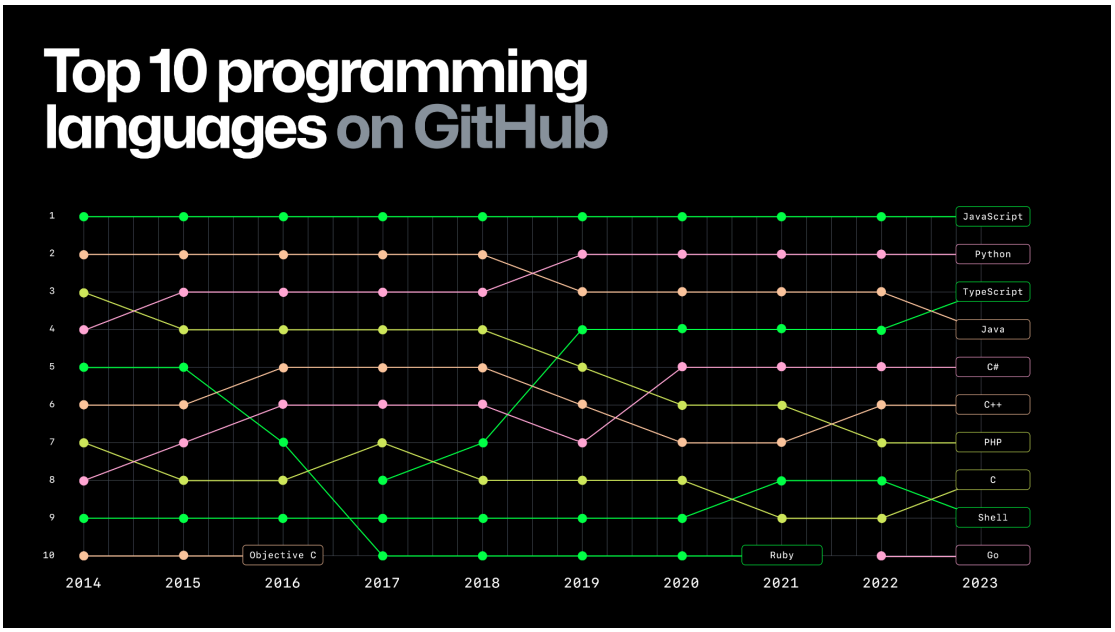


Imagem 1: Linguagens mais populares segundo o "GitHub Octoverse". fonte: https://github.blog/2023-11-08-the-state-of-open-source-and-ai/

é possível verificar que as linguagens mais populares dos últimos 10 anos estão entre as linguagens predominantes dentre os sistemas mais populares. Com isso, pode-se concluir que a utilização de uma linguagem popular em um sistema, o torna também mais popular.

6. Sistemas populares tiveram em média 77% das suas issues fechadas. Comparando esse resultado com os demais, é possível concluir que, repositórios populares recebem muitas contribuições e, por conta disso, possuem grande parte de suas issues fechadas.

Em trabalhos futuros, pretende-se destrinchar melhor o que necessariamente torna um sistema popular, e se os sistemas que alcançam esse estado, tendem a se manter assim por quanto tempo.

Referências

C. E. Anchundia and E. R. Fonseca C., "Resources for Reproducibility of Experiments in Empirical Software Engineering: Topics Derived From a Secondary Study," in IEEE Access, vol. 8, pp. 8992-9004, 2020, doi: 10.1109/ACCESS.2020.2964587.

MATTAR, F. N. Pesquisa de marketing. 3.ed. São Paulo: Atlas, 2001.

Samer Buna, GraphQL in Action , Manning, 2021.

Triola, Mario F.. Estatística. 14ª ed. Rio de Janeiro: LTC, 2023. (Capítulo 3: Medidas de Tendência Central)