Análise de Características de Repositórios Populares

**Bruno Gomes Ferreira, João Pedro Mairinque de Azevedo,
Matheus Vieira dos Santos, Marcio Lucas Machado Pereira**

[1] Instituto de Ciências Exatas e Informática (ICEI)
Pontifícia Universidade Católica de Minas Gerais
Engenharia de Software
Belo Horizonte – MG – Brazil

1. Introdução

Os sistemas open-source tem como premissa trazer colaboração e transparência entre os usuários, tendo isso em vista, este projeto tem como objetivo analisar repositórios de código aberto focando em sistemas que possuem uma maior atenção e admiração pela comunidade, ou seja, os repositórios com a maior quantidade de estrelas no GitHub.

Ao longo deste estudo, busca-se percorrer algumas características destes repositórios com o intuito de examinar o processo de desenvolvimento, avaliar a regularidade das contribuições externas, analisar a frequência de lançamentos de versões e investigar outras características associadas a esses elementos. Para compreender melhor essas características foram realizadas as seguintes perguntas de pesquisas (Research Questions):

- RQ 01. Sistemas populares são maduros/antigos?
- RQ 02. Sistemas populares recebem muita contribuição externa?
- RQ 03. Sistemas populares lançam releases com frequência?
- RQ 04. Sistemas populares são atualizados com frequência?
- RQ 05. Sistemas populares são escritos nas linguagens mais populares?
- RQ 06. Sistemas populares possuem um alto percentual de issues fechadas?

Diante as perguntas formuladas, existem algumas hipóteses informais que podem ser levantadas: Sistemas populares têm uma tendência a serem maduros já que passaram por diversas revisões até se tornarem um sistema confiável para a comunidade utilizar no dia-à-dia, por se tratar de um sistema open-source a partir do momento que esse repositório é relevante para a comunidade ele irá receber contribuições externas.

A taxa de releases deve ser frequente, pois isso traria um maior engajamento da comunidade. Esses repositórios também irão ter uma taxa de atualização maior pois a tecnologia está em constante evolução, seja com novas features, correções de bugs ou atualização de dependências.

Repositórios tendem a ser escritos em linguagens populares, pois o engajamento da comunidade muitas vezes está correlacionado a uma tecnologia que se tornou popular e este repositório pode agregar alguma coisa para essa tecnologia. E finalmente, sistemas populares tendem a ter um percentual de issues fechadas, já que ele deverá se manter constantemente atualizado.

2. Metodologia

Com o objetivo de caracterizar os repositórios populares do GitHub verificar a veracidade das hipóteses informais, utilizou-se a linguagem de programação Python, onde foi feito um script que comunica-se com a API do GitHub e faz uma chamada no GraphQL através de uma query para obter as respostas das perguntas de pesquisa. Inicialmente foi utilizado o script para pesquisar os 100 primeiros repositórios com maior quantidade de estrelas no GitHub, em seguida aumentou-se o valor para 1000 repositórios.

3. Resultados

Após obter os 1000 repositórios, foi possível gerar gráficos para melhor visualização dos resultados e por meio disso avaliar as perguntas de pesquisa

3.1. Questão 1: Sistemas populares são maduros/antigos?

Para essa questão traçou-se um gráfico de barras (Figura 1) com a quantidade de repositórios por ano de criação e é possível observar que a distribuição de repositórios ao longo dos anos foi ascendendo até o ano de 2016 e depois disso teve uma queda nos anos seguintes.

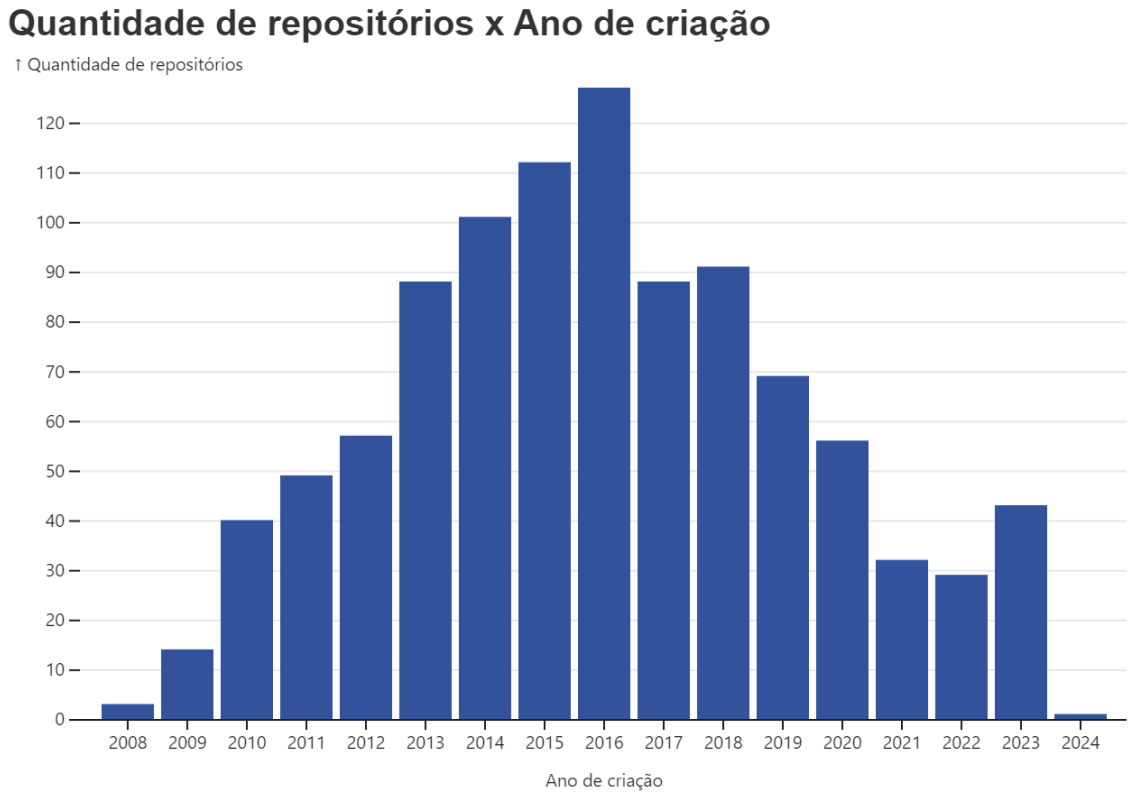

Figure 1. Quantidade de repositórios x Ano de criação

Ainda que tenha tido uma queda, pode-se observar que mesmo nos últimos 5 anos ainda houve uma quantidade expressiva de repositórios populares, o que sugere que sistemas populares não necessariamente precisam ser antigos para ter uma grande comunidade.

3.2. Questão 2: Sistemas populares recebem muita contribuição externa?

Para essa questão foi levantado um gráfico (Figura 2) com a quantidade de repositórios por número de forks, e essa tem pico entre 0 e 10000 forks e decai gradualmente até atingir menos de 50 repositórios a partir de 15000 forks.

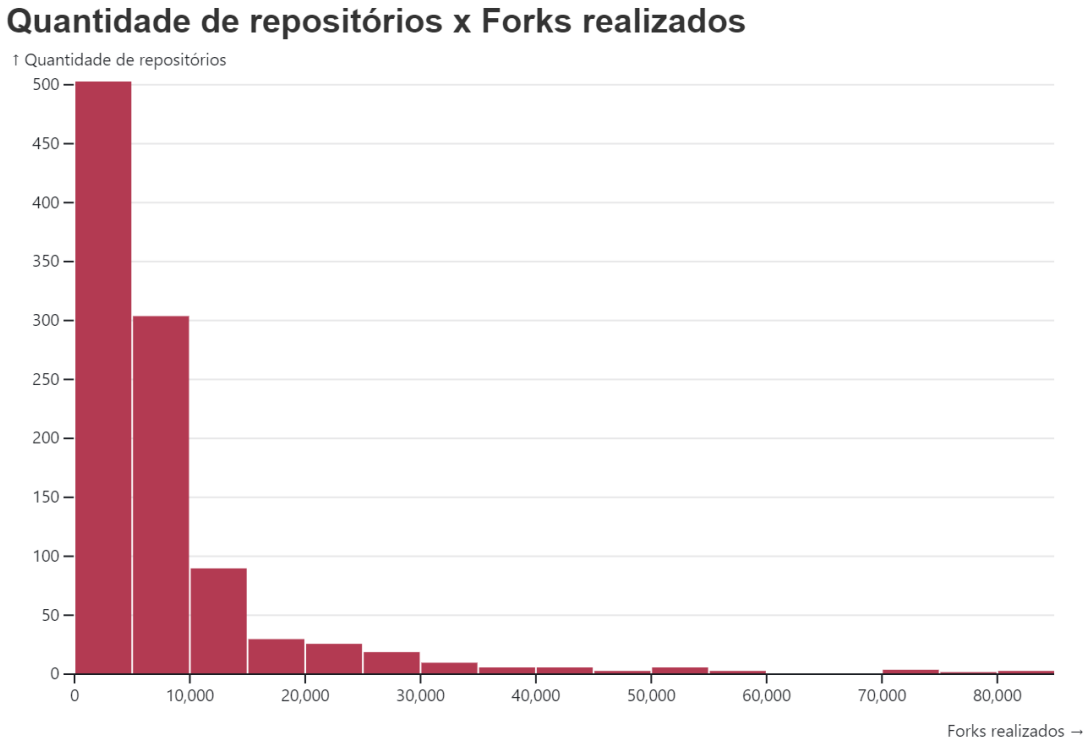

Figure 2. Quantidade de repositórios x Forks realizados

Nota-se que a maioria dos repositórios possuem uma quantidade inferior à 10000 forks, o que não deixa de ser um número alto com relação aos padrões do github. Forks são um indício de contribuições externas realizadas pela comunidade ou outras organizações, portanto estas podem ser inferidas como um número significante para a métrica.

3.3. Questão 3: Sistemas populares lançam releases com frequência?

Para a questão 4, o gráfico compara a quantidade de repositórios em relação ao ano da última release, e aponta que a grande maioria teve seu último lançamento disponibilizado neste ano de 2024.

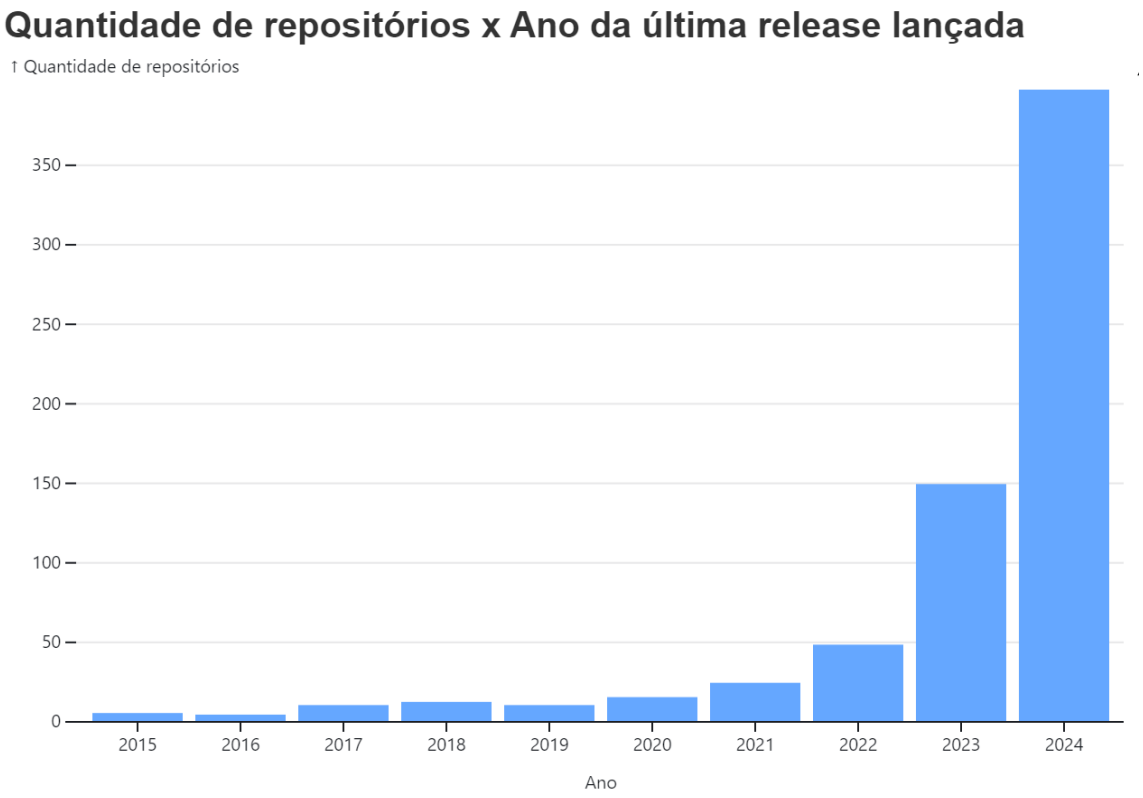

Figure 3. Quantidade de repositórios x Ano da última release lançada

Através dos dados traçados nota-se que os repositórios populares majoritariamente preocupam-se em realizar novos lançamentos com o passar dos anos, já que menos de 50 repositórios realizaram sua última release antes de 2022. É possível então supor que esses projetos continuarão a manter seu fluxo de releases com o passar do tempo.

3.4. Questão 4: Sistemas populares são atualizados com frequência?

Quanto à questão 4, todos os repositórios pesquisados tiveram atualizações no mesmo dia da query realizada (29/02), o gráfico pode ser visto na figura 4 mostrando a faixa de horário em que essas atualizações ocorreram durante o dia.

Figure 4. Tempo da última atualização x Quantidade de repositórios

Isso confirma a teoria de que repositórios populares tendem a ter uma taxa de atualização contínua, seja criando novas branches, fazendo pull requests, fechando issues, ou seja, a movimentação é feita de forma diária pela comunidade.

3.5. Questão 5: Sistemas populares são escritos nas linguagens mais populares?

Para a quinta questão, foi selecionado as 10 primeiras linguagens que mais apareceram como principal linguagem de programação utilizada nos repositórios com maior número de estrelas.

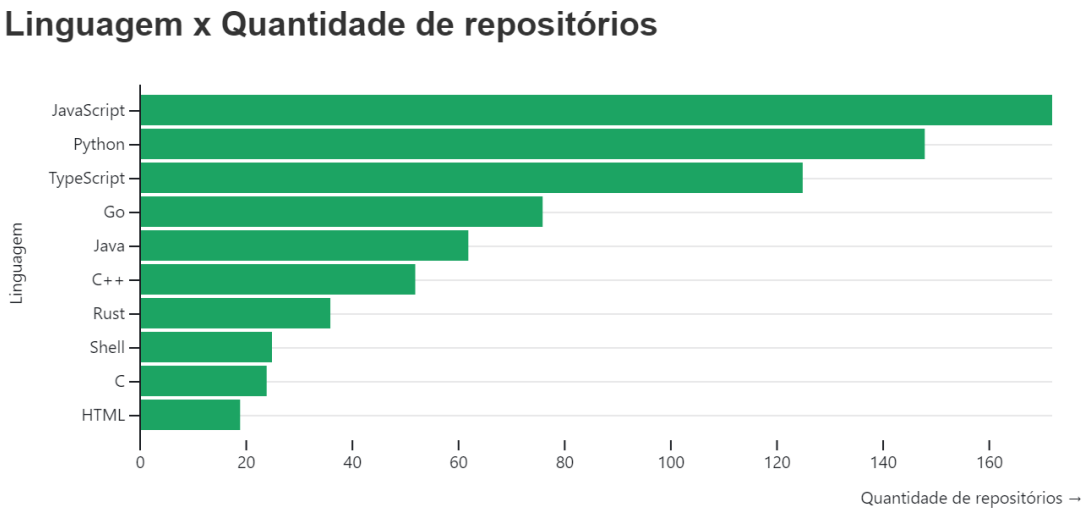

Figure 5. Linguagem x Quantidade de repositórios

Pode-se observar que as linguagens utilizadas nos repositórios com maior número de estrelas, tem uma maior predominância em linguagens mais populares e que já estão bem consolidadas no mercado, como foi o caso do JavaScript, Python, TypeScript, Go, etc.

3.6. Questão 6: Sistemas populares possuem um alto percentual de issues fechadas

Com o objetivo de responder a sexta questão, foi verificado a porcentagem de issues fechadas nos repositórios mais populares, com os dados obtidos foi feito um gráfico de quantidade de repositórios x porcentagem de issues fechadas (figura 6).

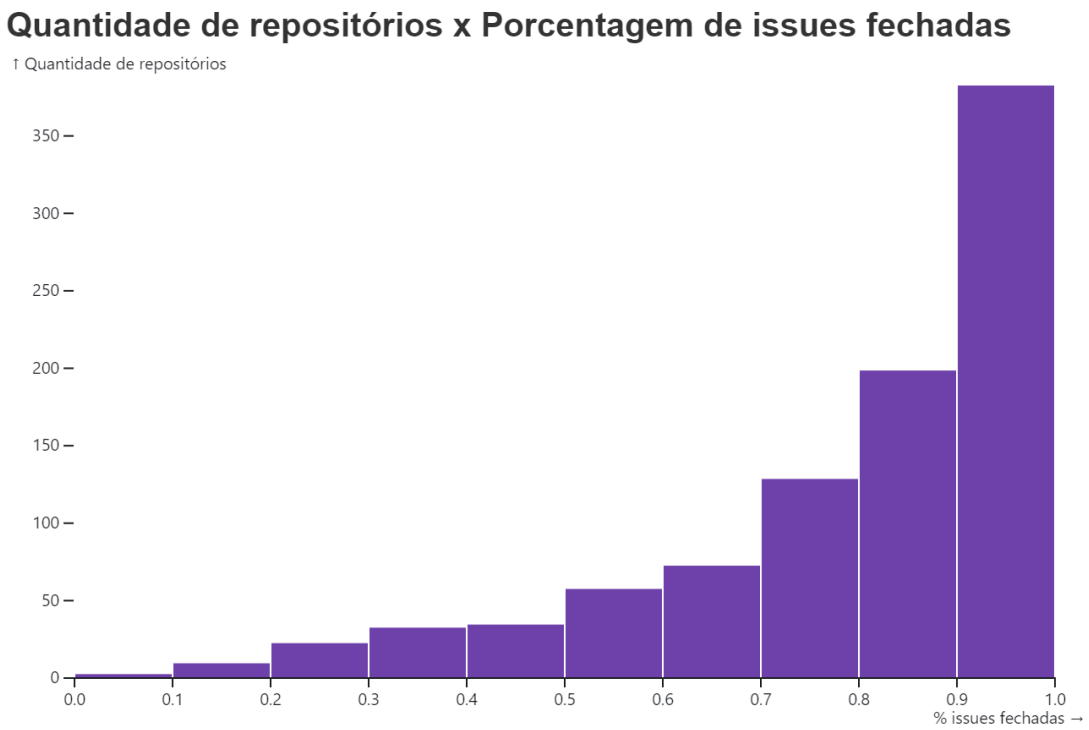

Figure 6. Quantidade de repositórios x Porcentagem de issues fechadas

Pode-se observar pelo gráfico que os repositórios populares tendem a ter uma quantidade alta de issues fechadas, onde a maioria desses repositórios tem uma quantidade superior a 70 por cento de issues fechadas.

4. Discussão

Pode-se verificar que a maioria das hipóteses geradas no início deste relatório de fato ocorreram ao obter os dados dos 1000 repositórios, e outras não. Neste tópico será discutido os resultados obtidos em comparação com o que foi inicialmente proposto pelos autores da pesquisa.

Iniciando pela questão de pesquisa 1, sistemas populares provaram não ter correlação com o tempo que eles existem, indo em contradição com a hipótese proposta inicialmente de que sistemas maduros já são bem estabelecidos e por isso são populares. De acordo com o gráfico, foi possível provar que todos os anos são criados uma quantidade considerável de repositórios que se tornam populares.

Em relação à questão 2, é possível inferir que a quantidade de forks está correlacionado ao quanto aquele repositório se tornou relevante para a comunidade, já que ao utilizar aquele fork, os usuários irão encontrar bugs, sugerir melhorias e abrir issues no repositório principal. Por fim, as hipóteses iniciais estão de acordo com o que foi encontrado.

Ao analisar a questão de pesquisa 3, foi possível observar que a hipótese traçada inicialmente está de acordo com os resultados encontrados, onde observou-se que os repositórios populares tem uma taxa de releases contínuas, com o objetivo de solucionar problemas anteriores alertados pela comunidade, atualizar o projeto com novas features e assim continuar se mantendo relevante para o seu público.

Para a questão 4 foi provado que no mesmo dia em que foi realizado a querry todos esses repositórios tiveram ao menos uma atualização, seja correções de bugs, aberturas de issues, melhorias de códigos, releases, etc. Então é possível observar que uma característica em comum que os repositórios populares possuem é de que eles têm uma taxa de atualização muito grande.

Para a questão 5, foi possível observar que realmente repositórios populares utilizam em sua maioria linguagens que são populares ou bem estabelecidas pela comunidade, isso pode indicar que realmente o engajamento da comunidade está ligado a uma tecnologia que se tornou tendência.

Quanto à questão 6, é possível observar que a quantidade de issues fechadas pode estar relacionada com a quantidade de atividades que um repositório popular recebe por dia. Como a comunidade tende a ser mais ativa, significa que bugs novos são reportados com frequência, assim como discussões sobre determinado bug e sua consequente resolução. Quanto à hipótese inicial, foi possível observar que o gráfico gerado foi condizente com a hipótese de que o percentual de issues fechadas é diretamente proporcional as atualizações daquele repositório.

5. Conclusão

O objetivo principal do trabalho era mapear questões que poderiam dar ideia de quais características repositórios populares no GitHub possuem e formular hipóteses iniciais, após isso através de uma query utilizando o GraphQL poder verificar essas características e comparar através de gráficos se as hipóteses feitas inicialmente são verídicas ou não. Pode-se observar que algumas das hipóteses se tornaram errôneas, como foi o caso da hipótese da questão de pesquisa 1, mas a maioria das hipóteses tiveram um acerto quando se compara com a realidade. Em relação a trabalhos futuros, agora que foi caracterizado os repositórios populares é possível ir um passo além e verificar o engajamento da comunidade em repositórios populares.