Daniella Rodrigues
Júlia Martins Reis

Repositórios Populares

Belo Horizonte

2023

Daniella Rodrigues
Júlia Martins Reis

Repositórios Populares

Projeto de Pesquisa apresentado na disciplina Laboratório de Experimentação de Software do curso de Engenharia de Software da Pontifícia Universidade Católica de Minas Gerais.

Belo Horizonte
2023

SUMÁRIO

1 INTRODUÇÃO

O trabalho tem como proposta estudar e apresentar as principais características de sistemas populares open-source. Tais características podem ser citadas como: linguagem mais popular entre os repositórios, número de issues abertas, frequencia de releases e outros aspectos que serão analisados dentre os 1000 projetos com o maior número de estrelas definidos pela plataforma do GitHub.

Essa pesquisa justifica-se pela abrangência de códigos que existem em repositórios open-source que não são estudados e utilizados devidamente -tanto para estudo quanto para trabalhos- para a área de desenvolvimento de software e tecnologias em geral. Assim como, o projeto busca esses resultado com um fim didático para que os alunos compreendam sobre a busca de repositórios utilizando querys e desenvolvam habilidades com a linguagem de programação Python.

Este trabalho está organizado da seguinte forma. A Seção 1.1 exibe as hipóteses geradas antes dos resultados. A Seção 2 apresenta a metodologia usada neste trabalho. A Seção 3 descreve os resultados positivos. A Seção 4 retrata a discussão do que se espera e os valores obtidos.

1.1 Hipóteses Informais

Questão 1 - Os sistemas populares são maduros/antigos?

Hipótese: Não, sistemas populares tendem a ser mais novos. Sistemas mais novos apresentam linguagens mais modernas o que acarreta em uma maior popularidade entre os devs, do que sistemas mais antigos que apresentam linguagens depreciadas.

Questão 2 - Os sistemas populares recebem muita contribuição externa?

Hipótese: Sim, justamente pelo fato de serem populares/conhecidos, muitos usuários desejam contribuir com o código.

Questão 3 - Sistemas populares lançam releases com frequência?

Hipótese: Não, pelo fato de eles serem populares, não necessitam de lançar releases contínuas.

Questão 4 - Sistemas populares são atualizados com frequência?

Hipótese: Sim, pelo fato de serem populares, os autores deveriam atualizar com frequência para não se tornar um código depreciado e com funcionalidades desatualizadas.

Questão 5 - Sistemas populares são escritos nas linguagens mais populares?

Hipótese: Sim, sistemas populares são populares justamente pois quem os busca

procura um sistema com linguagens populares e/ou atuais.

Questão 6 - Os sistemas populares possuem um alto percentual de issues fechadas?

Hipótese: Sim, pois como são populares e recebem muitos acessos, sempre haverá novas issues e, por consequência, fechá-las

Questão 7 - Sistemas escritos em linguagens mais populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência?

Hipótese: Sim, as linguagens populares são mais procuradas e por esse motivo, mais pessoas desejam conhecer e contribuir com os códigos. Desse modo, acabam lançando mais releases.

2 METODOLOGIA

A metodologia utilizada para responder às questões foi fundamentada ppor um script em GraphQL no qual foram feitas alterações e melhorias para atender a cada tipo de pergunta. Após as mudanças, é necessário adequar a forma que a API retorna as respostas como, por exemplo, a busca por repositórios antigos e novos, retorna a data de criação, no entanto, para a melhor visualização de dados e para tornar o resultado legível, é preciso converter a data para total de anos (ou dias, se for o caso).

Outra melhoria que fez-se necessária, foi criar somente uma query que consiga retornar os dados para responder a todas as questões. O motivo desta escolha leva em conta o custo para realizar 6 querys que buscam o conteúdo específico em 1000 repositórios multiplicado pelo número de vezes que a API seria chamada. Teria um alto valor de custo e tempo para aguardar todas as buscas nos repositórios.

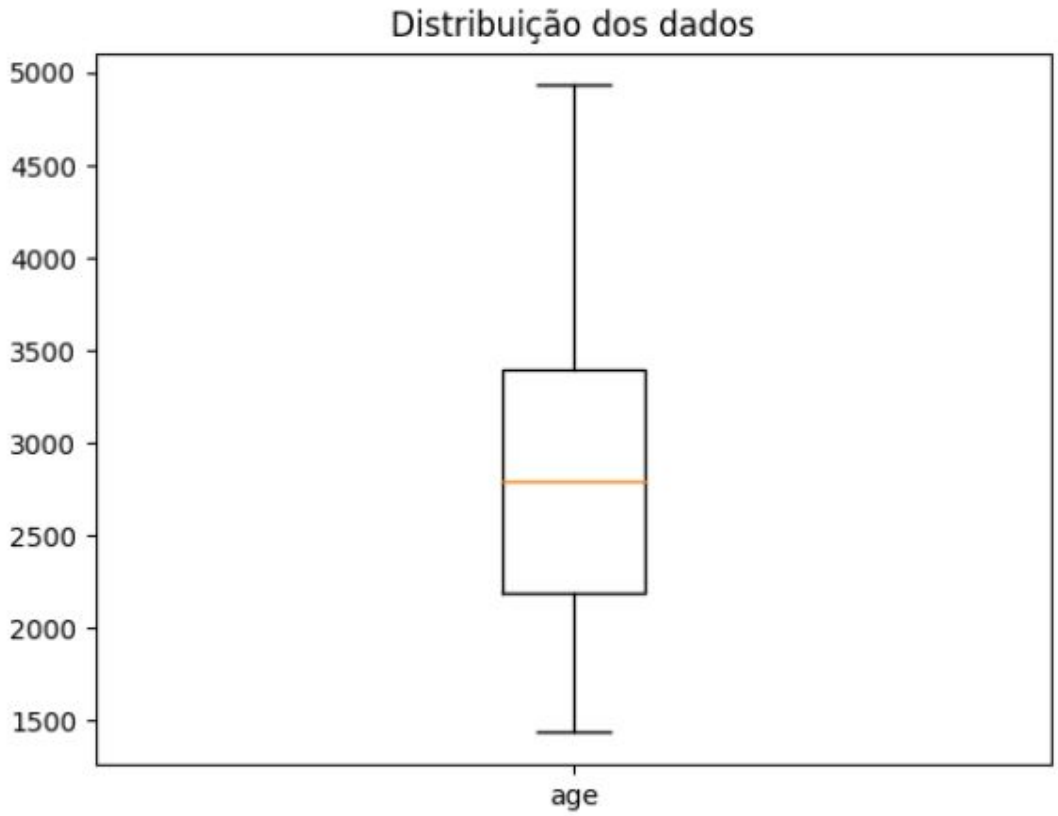
Figura 1 – Representação da query utilizada como base para as questões.

```
import requests //biblioteca para requisição

headers = {"Authorization": "Bearer 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"} // header com token do git hub url =
'https://api.github.com/graphql'

query = """
query { search( type: #escolherTipo, query: "stars:>100" first: 100 ) { edges{ node{
... on Repository { #parâmetros } } }, } } """

response = requests.post(url, headers=headers, json={"query": query}) // requisição para o graphQL data = response.json()

for i, repo in enumerate(data["data"]["search"]["edges"]): // imprimindo resultados no terminal

... // print com as variaveis e/ou parâmetros
```

3 RESULTADOS OBTIDOS

3.1 Questão 1:

Figura 2 – Idade do repositório.



3.2 Questão 2:

3.3 Questão 3:

A partir dos dados coletados pela API, a maior parte dos repositórios não possuem nenhuma release lançada. Dados: 75% dos repositórios continham a informação de que haviam 0 releases nos projetos.

Figura 3 – Pull Requests.

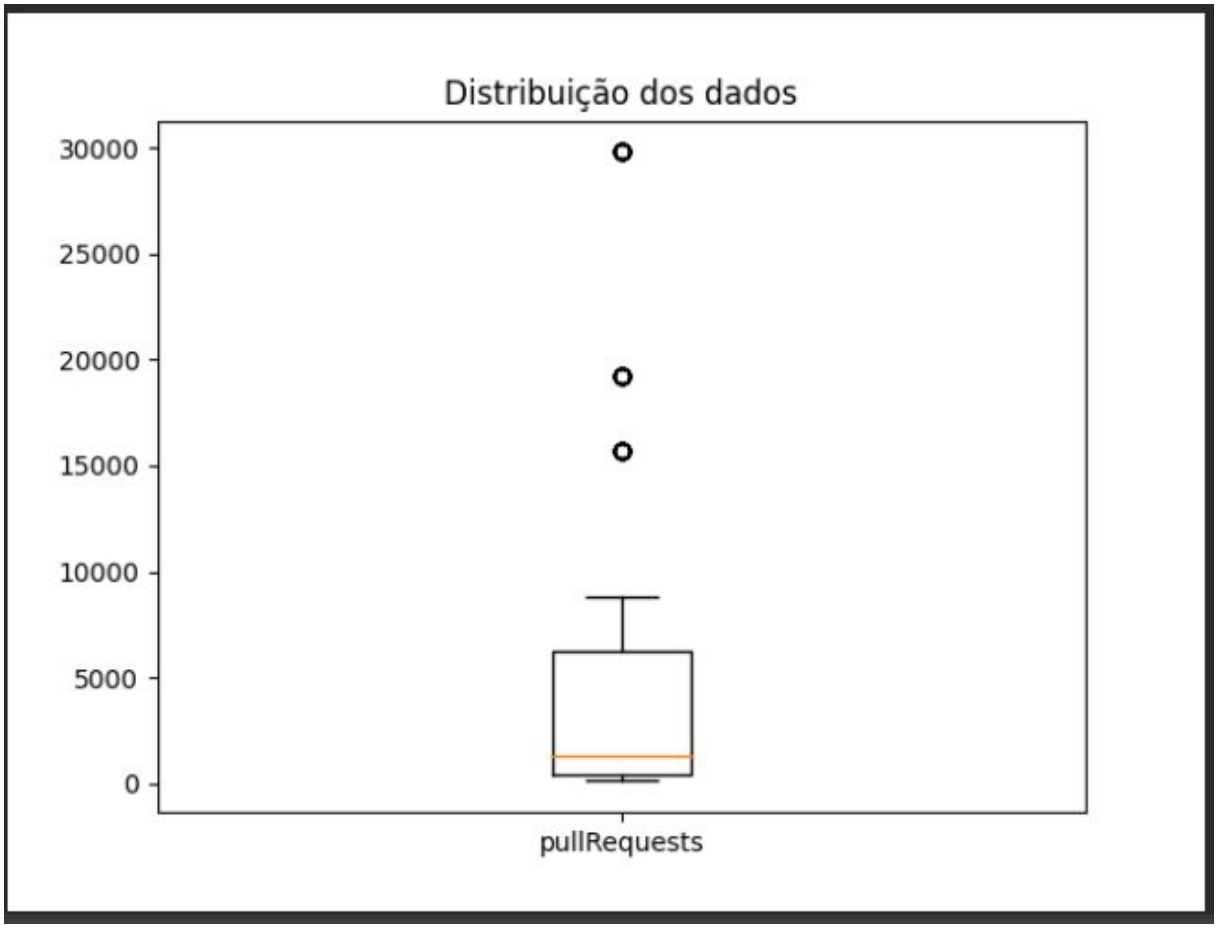
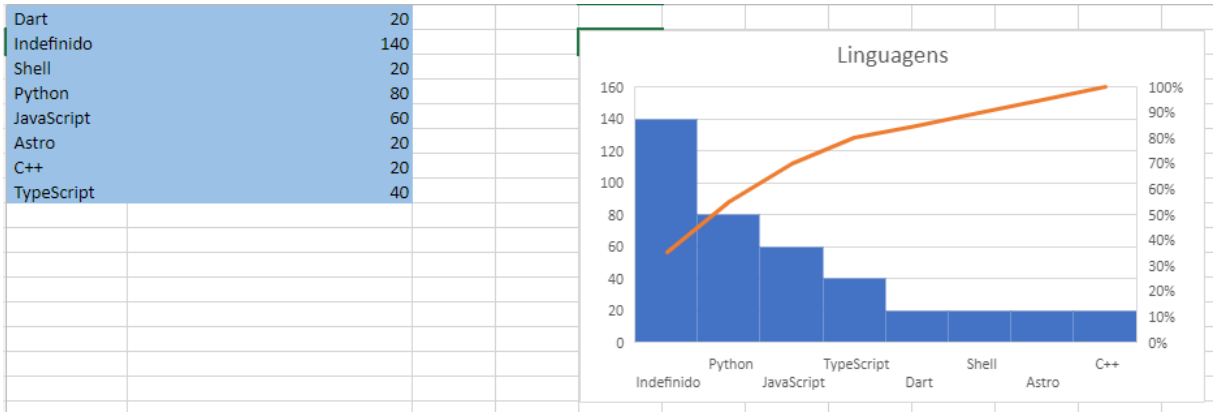


Figura 4 – Linguagens Populares.



3.4 Questão 4:

3.5 Questão 5:

3.6 Questão 6: Linguagens populares

O resultado da razão entre número de issues fechadas pelo total de issues, para saber se sistemas populares possuem um alto percentual de issues fechadas, foi que na maior parte dos casos, a diferença não passou de 1,0, ou seja, as issues são constantemente fechadas.

Informações complementares:

Soma das closed issues: 3436520

Média de closed issues: 791

Desvio padrão: 16306,2455

Soma do total de issues: 3762440

Média do total de Issues: 907

Desvio padrão: 18569,3105

Figura 5 – Total de issues.

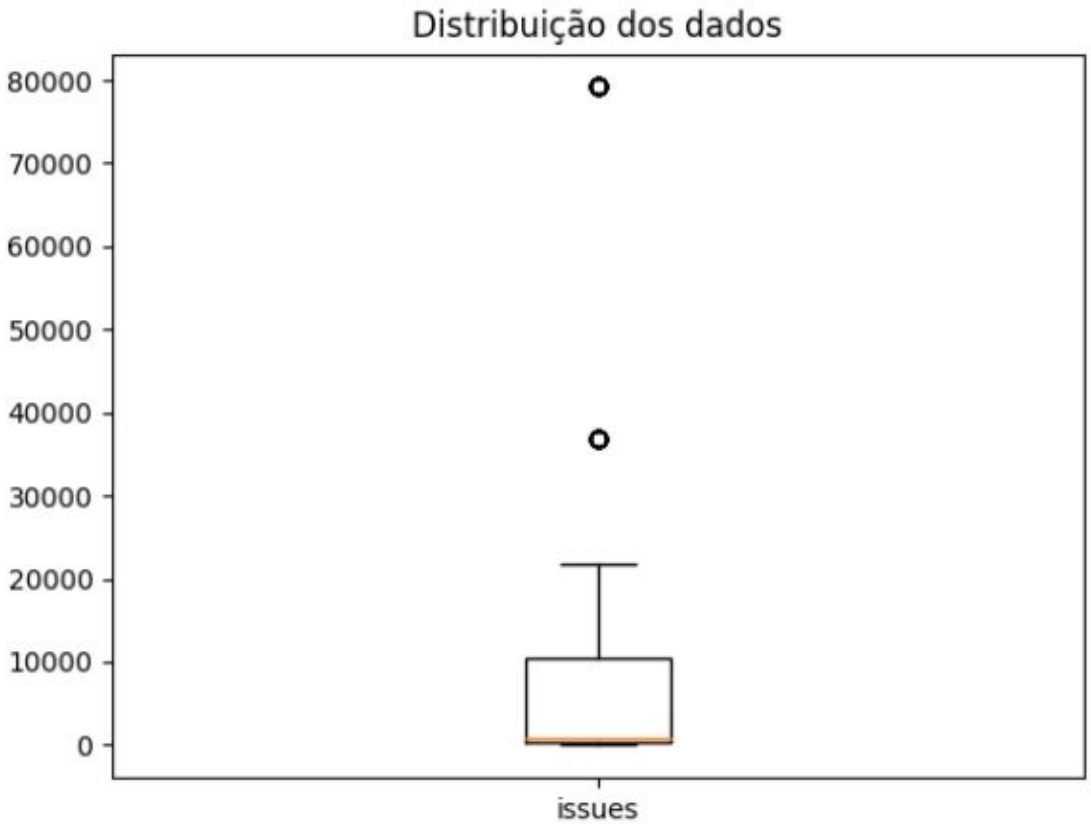
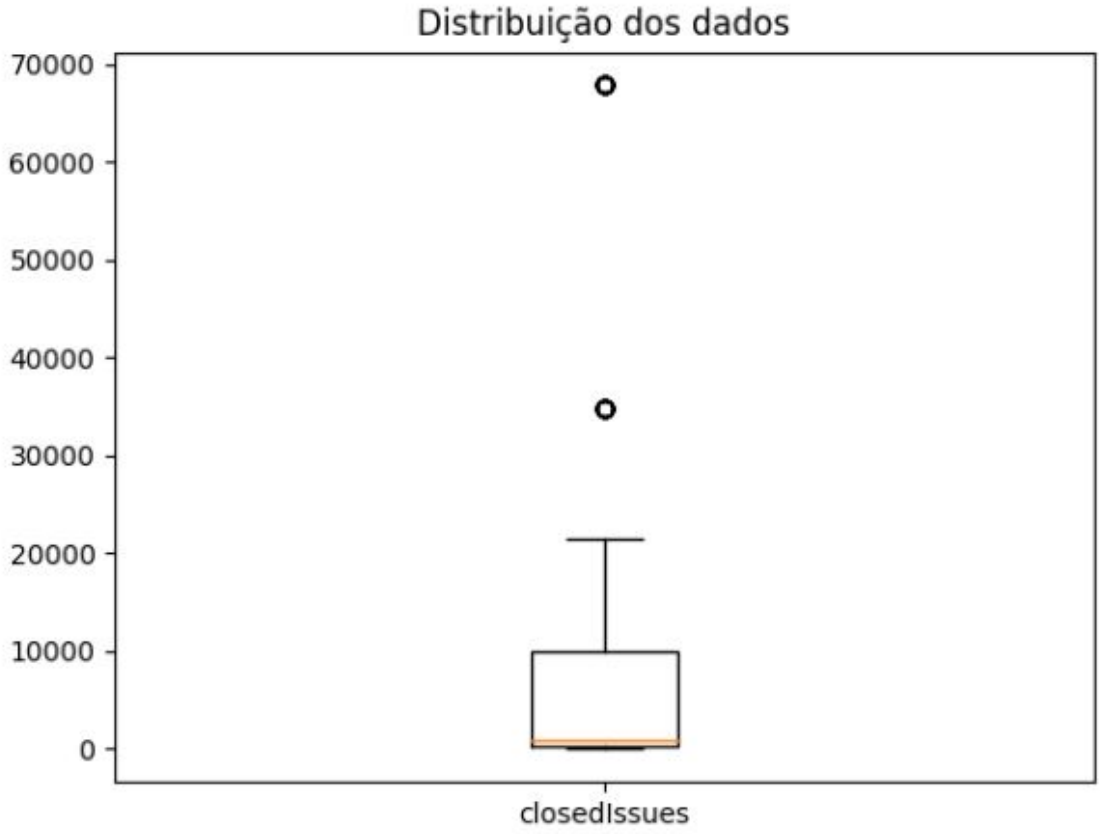


Figura 6 – Closed issues.



4 DISCUSSÃO

Questão 1: O resultado não saiu como esperado, pois a hipótese dizia que os repositórios populares tendem a ser mais novos. No entanto, a média de idade que foi obtida através de um cálculo a partir da data de criação, retornou 8. Ou seja, oito anos para um repositório é considerado muito tempo, pois as linguagens podem se tornar depreciadas e/ou modificadas. Se um desenvolvedor encontra um repositório antigo, pode ser que existam funções, métodos e conceitos que não são utilizados.

Questão 2: O resultado saiu como o esperado, sendo que a média de Pull Requests feitas foi de 1286, o que pode ser considerado um número alto de solicitações para a maioria dos projetos de código aberto, mas pode ser relativamente baixo para grandes projetos populares com muitos contribuidores ativos.

Questão 3:

Questão 4:

Questão 5: O resultado saiu como o esperado, sendo que os dois repositórios com maior frequencia nas respostas foram os que utilizam as linguagens 'Python' - com 20% de frequencia - e 'JavaScript' - com 15% de frequencia. Que são, atualmente, as linguagens de programação mais populares. No entanto, também obtivemos uma margem de repositórios que não contém uma linguagem definida, que foi a maior parte, com 25% de frequencia. Pelo fato de a API não ter retornado certas linguagens, não é possível ter certeza de que eles utilizam javascript ou python, por exemplo, então concluímos que a hipótese estava correta - ignorando o fato destas linguagens não identificadas.

Questão 6: Essa métrica tratava sobre sistemas populares e alto percentual de issues fechadas. E a hipotese criada fez sentido com o resultado obtido pela busca da query pois conforme foi dito na hipótese, o percentual de closed issues seria alto pelo fato de receberem muitos acessos. E com isso, o resultado mostrou que nenhum repositório apresentou uma razão entre closed e total maior do que 1,0. Ou seja, os repositórios fecham a maior parte das issues.