

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA
UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE
Bacharelado em Engenharia de Software**

**Belle Nerissa Aguiar Elizeu
Felipe Caldas Liduario
Letícia Amanda Franco Gonçalves**

RELATÓRIO: CARACTERÍSTICAS DE REPOSITÓRIOS POPULARES

Belo Horizonte
2023

**Belle Nerissa Aguiar Elizeu
Felipe Caldas Liduario
Letícia Amanda Franco Gonçalves**

RELATÓRIO: CARACTERÍSTICAS DE REPOSITÓRIOS POPULARES

Relatório apresentado na Sprint 1 da disciplina
Laboratório de Experimentação de Software.

Professor: José Laerte Pires Xavier

Belo Horizonte

2023

1 INTRODUÇÃO

O relatório apresentado visa analisar as características de relatórios populares do GitHub. Essa atividade foi proposta na disciplina de Laboratório de Experimentação de Software.

1.1 Hipóteses

RQ01. Sistemas populares são maduros/antigos?

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

Hipótese: Sim, sistemas populares são mais antigos, pois tiveram mais tempo no GitHub para receber contribuições e para serem vistos por mais pessoas.

RQ02. Sistemas populares recebem muita contribuição externa?

Métrica: total de pull requests aceitas.

Hipótese: Sim, considerando que esses sistemas são normalmente escritos em linguagens muito conhecidas, existem inúmeras pessoas que entendem e que contribuem para a evolução desse sistema.

RQ03. Sistemas populares lançam releases com frequência?

Métrica: total de releases.

Hipótese: Sim, devido ao grande número de pessoas contribuindo com os sistemas mais populares, estes lançam releases com uma frequência maior se comparado a alguns sistemas mais desconhecidos.

RQ04. Sistemas populares são atualizados com frequência?

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

Hipótese: Sim, devido à abundância de contribuições que esses sistemas recebem, as atualizações são feitas com uma frequência maior se compararmos com sistemas menos populares.

RQ05. Sistemas populares são escritos nas linguagens mais populares?

Métrica: linguagem primária de cada um desses repositórios

Hipótese: Sim, quanto mais popular e conhecida a linguagem for, mais pessoas conheceram o sistema e conseguirão interagir com pull requests e avaliações.

RQ06. Sistemas populares possuem um alto percentual de issues fechadas?

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

Hipótese: Sim, por se tratar de um sistema popular, existem mais pessoas trabalhando nele, portanto a chance de uma pessoa fechar um issue aberta nesse sistema é grande quando comparamos a um sistema não popular onde poucas pessoas realizam contribuições.

RQ 07: Sistemas escritos em linguagens mais populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência?

Métrica: Divisão dos resultados obtidos nas RQs 02, 03 e 04 por linguagem e análise de como esses valores se comportam de acordo com as linguagem de cada repositório.

Hipótese: Linguagens de programação populares possuem alto engajamento da sua comunidade, ou seja, repositórios escritos em linguagens populares podem obter mais contribuições dessas pessoas. Como referência para esta hipótese, assumimos que para cada linguagem a quantidade de pull requests aceita deve ser pelo menos metade dos pull requests totais, a quantidade de releases por ano deve ser pelo menos 10 e a atualização deve ser pelo menos mensal.

2 METODOLOGIA

Para realizar a análise das hipóteses, foi desenvolvido um script em Python com uma consulta em GraphQL que realiza consultas dos repositórios mais populares do Github e faz a exportação dos dados em um arquivo csv. O código-fonte do script e o arquivo csv está contido no repositório: https://github.com/octaviolage/ES_lab_VI_01/tree/main/sprint_01.

Foram analisados 1000 repositórios na data 04/03/2023 e calculadas as medianas através de outro script em python que utiliza das bibliotecas pandas, para o cálculo das respostas das RQs de acordo com as métricas solicitadas, e matplotlib para a plotagem dos gráficos.

3 RESULTADOS OBTIDOS

RQ01: A mediana de idade dos repositórios é de 2733,5 dias, que correspondem a cerca de 7 anos e 5 meses e meio. O gráfico abaixo apresenta um gráfico do tipo boxplot com a idade dos repositórios em dias.

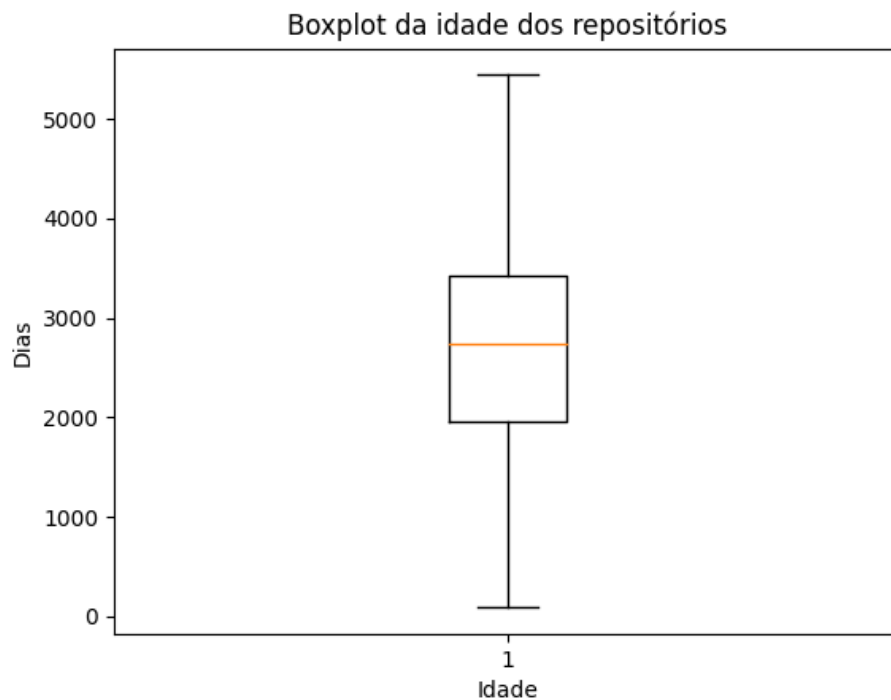


Figura 1 - Boxplot das idades dos repositórios

RQ02: A mediana do total de pull requests aceitas dos repositórios é de 466,5.

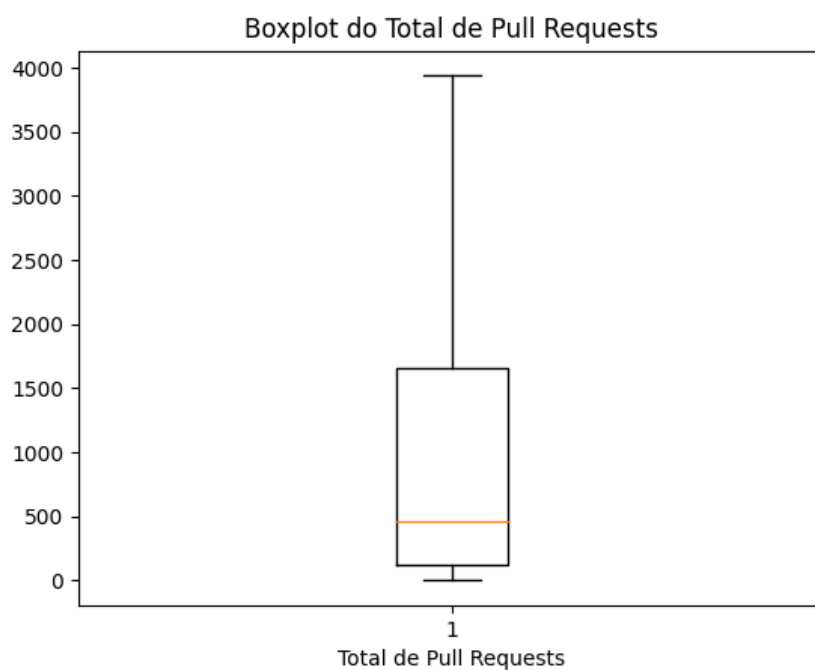


Figura 2 - Boxplot do total de pull requests aceitas dos repositórios

RQ03: A mediana de releases é 24,0. A mediana de quantidade de releases por ano de acordo com a mediana da idade dos repositórios é $24,0 / 7,41 \approx 3,23886$.

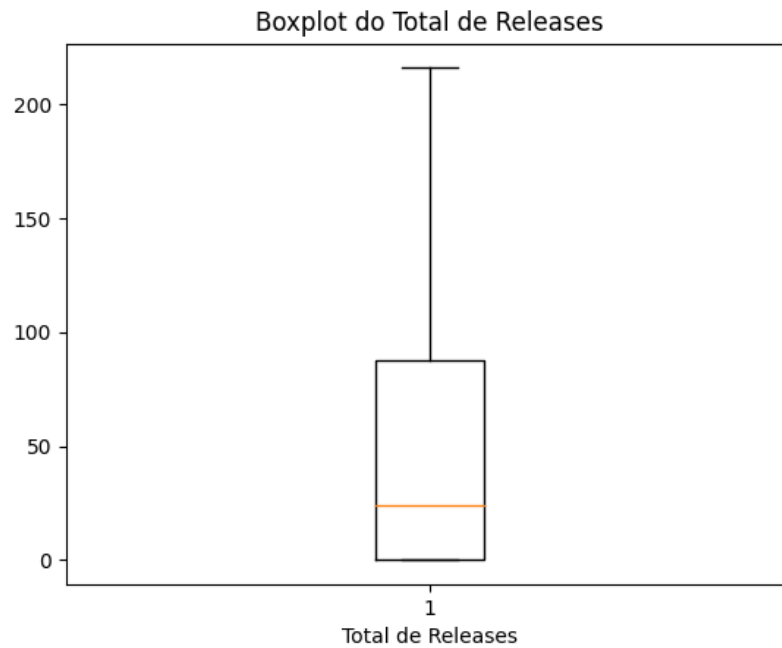


Figura 3 - Boxplot do Total de Releases dos repositórios

RQ04: A mediana do tempo desde a última atualização é de 8 dias. Portanto, os repositórios são atualizados frequentemente.



Figura 4 - Boxplot do Total de Tempo desde a última atualização dos repositórios

RQ05: As 5 linguagens que mais apareceram nos repositórios populares foram: Javascript, Python, Typescript, Go, Java.

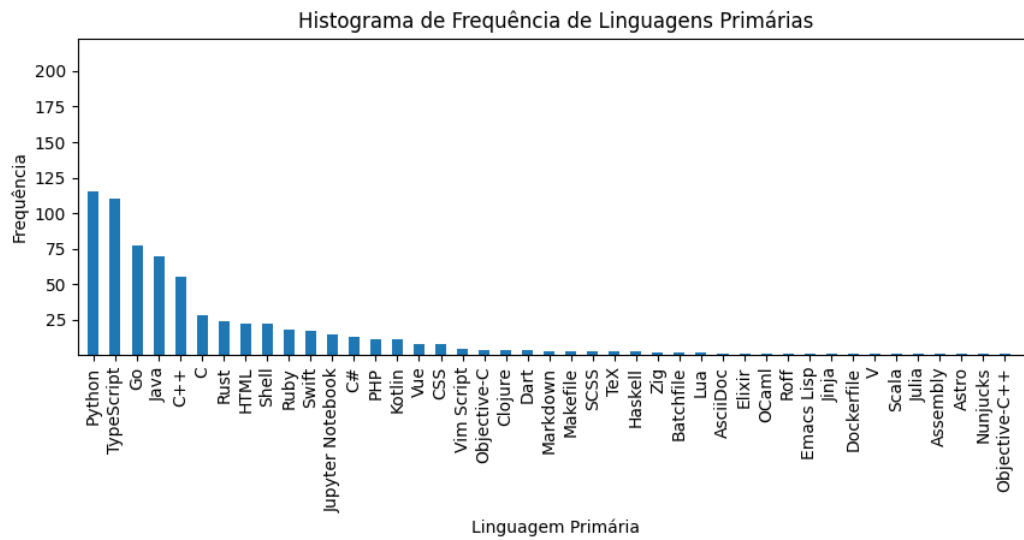


Figura 5 - Histograma com as linguagens mais populares

RQ06: A mediana da razão entre issues fechadas e o total é de $\approx 87\%$.

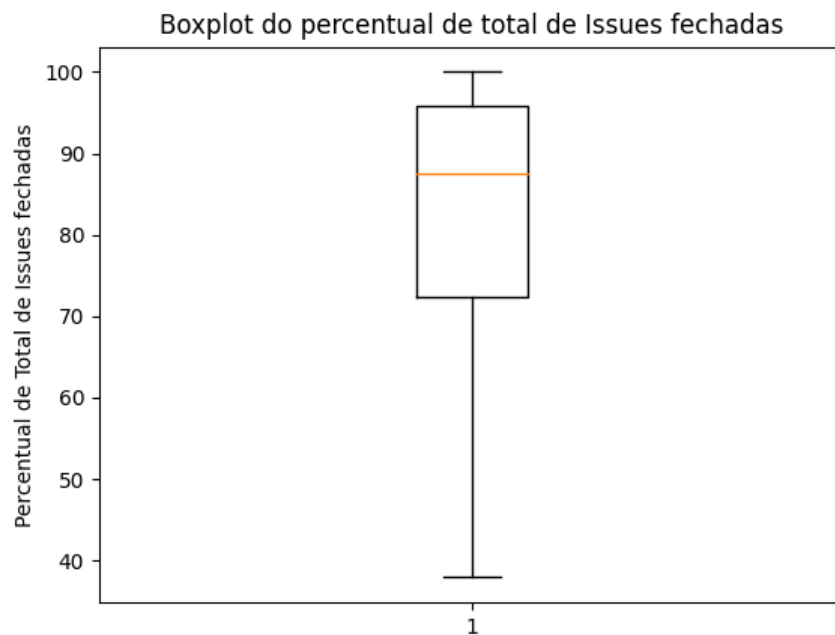


Figura 6 - Boxplot do percentual do total de Issues Fechadas do repositórios mais populares

RQ07: Resultados obtidos para cada uma das cinco principais linguagens dos repositórios analisados:

TypeScript

Mediana de **pull requests aceitas**: 1631,5

Mediana de **releases**: 106,5

Mediana de **atualizações**: 2 dias

Python

Mediana de **pull requests aceitas**: 530,0

Mediana de **releases**: 8,0

Mediana de **atualizações**: 6 dias

Go

Mediana de **pull requests aceitas**: 905,0

Mediana de **releases**: 78,0

Mediana de **atualizações**: 3 dias

Java

Mediana de **pull requests aceitas**: 398,5

Mediana de **releases**: 26,0

Mediana de **atualizações**: 5 dias

C++

Mediana de **pull requests aceitas**: 1087,0

Mediana de **releases**: 40,0

Mediana de **atualizações**: 2 dias

4 ANÁLISE DOS RESULTADOS

RQ01:

RQ02:

RQ03:

RQ04:

RQ05:

RQ06:

RQ07:

5 REFERÊNCIAS