

**Pontifícia Universidade Católica de Minas Gerais**

**Laboratório de Experimentação de Software**

**Pesquisa: Características de Repositórios populares**

Ian Asenjo Dominguez Cunha

Jully Ketely Alves da Silva

Warley Leandro dos Anjos

## 1 INTRODUÇÃO

O objetivo do relatório é analisar as características dos relatórios mais populares no GitHub. Essa análise foi proposta como atividade na disciplina de Laboratório de Experimentação de Software. A intenção é identificar padrões e tendências nesses relatórios, considerando a relevância que o GitHub tem para desenvolvedores de software e projetos de código aberto.

**Observação:** A análise da base de dados foi feita no dia 05 de Março, ou seja, resultados podem estar desatualizados até a data atual.

Foram analisados os 1000 repositórios mais populares do Github, com o objetivo de analisar características e responder algumas questões, através de resultados analíticos. A linguagem utilizada foi Python para mineração, e GraphQL para requisições de API.

O sistema gera um conjunto de dados, que são armazenados em um arquivo .csv (planilha). Com base, nesta planilha, o script gera os resultados e os gráficos para prosseguir com a análise e consequente chegar a um resultado coerente. O script está armazenado no repositório: <https://github.com/jullysilva/Laboratorio06>

### 1.1 Hipóteses

#### **RQ 01. Sistemas populares são maduros/antigos?**

**Hipótese:** Sim, pois estes sistemas antigos tem, na maioria das vezes, o maior período de contribuições e evolução do repositório.

#### **RQ 02. Sistemas populares recebem muita contribuição externa?**

**Hipótese:** Não, pois sistemas populares costumam receber um padrão e performance de código, e alguns colaboradores ficam receosos em expor suas opiniões na comunidade open source.

#### **RQ 03. Sistemas populares lançam releases com frequência?**

**Hipótese:** Sim, sistemas populares tendem a manter seus repositórios atualizados sempre que possível para manter.

**RQ 04. Sistemas populares são atualizados com frequência?**

**Hipótese:** Sim. Pois, são vários contribuidores ajudando na evolução e desenvolvimento do sistema, principalmente em repositórios *open source*.

**RQ 05. Sistemas populares são escritos nas [linguagens mais populares](#) [Links to an external site](#)?**

**Hipótese:** De acordo com o site da *octoverse* o top 5 das linguagens são JavaScript, Python, Java, TypeScript e C#. Acredito que a linguagem mais popular seja o C# por estar mais tempo no mercado e ser bastante usado.

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

**Hipótese:** Sim. Pensando na quantidade de contribuições feitas ao repositório, que é *open source* e popular.

## 2 METODOLOGIA

A metodologia usada para a análise dos resultados envolveu o desenvolvimento de um script em Python que utilizou uma consulta em GraphQL para obter dados sobre os repositórios mais populares do Github. Esses dados foram exportados para um arquivo csv. Em seguida, foram analisados 1.000 repositórios na data 05/03/2023, utilizando outro script em Python que usou as bibliotecas pandas para calcular as medianas das métricas relevantes para cada questão de pesquisa e matplotlib para plotar os gráficos correspondentes. O código-fonte do script para a geração do Csv, cálculo das métricas das RQs, o arquivo csv e outros itens relacionados podem ser encontrados no repositório: <https://github.com/jullysilva/Laboratorio06>.

### 3 RESULTADOS OBTIDOS

#### RQ 01. Sistemas populares são maduros/antigos?

**Hipótese:** Sim, pois estes sistemas antigos tem, na maioria das vezes, o maior período de contribuições e evolução do repositório.

**Resultado:** A mediana da idade do repositório é 2728.5 dias, ou seja, por volta de 7 anos e 5 meses e 1 semana.

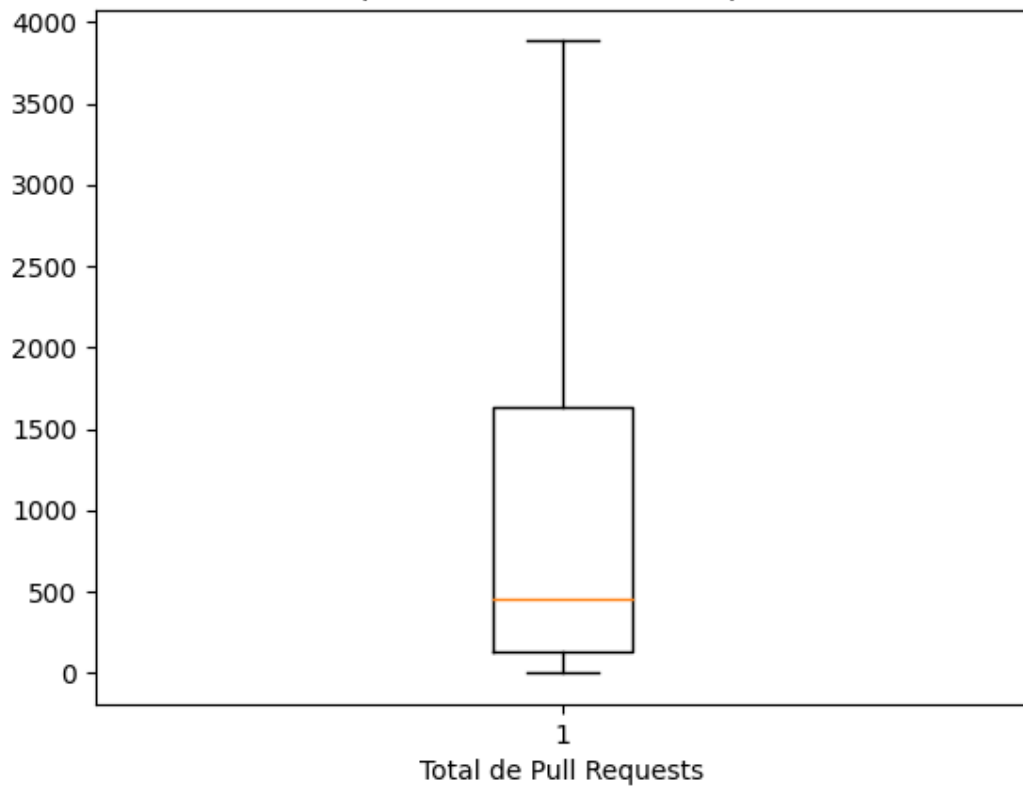


#### RQ 02. Sistemas populares recebem muita contribuição externa?

**Hipótese:** Não, pois sistemas populares costumam receber um padrão e performance de código, e alguns colaboradores ficam receosos em expor suas opiniões na comunidade open source.

**Resultado:** A mediana da total de pull requests que os repositórios aceitam é 459. Concluimos que, não, os repositórios não costumam aceitar/receber tanta contribuição externa. A mediana está bem abaixo da média dos "pull requests" feitas em repositórios *open source*.

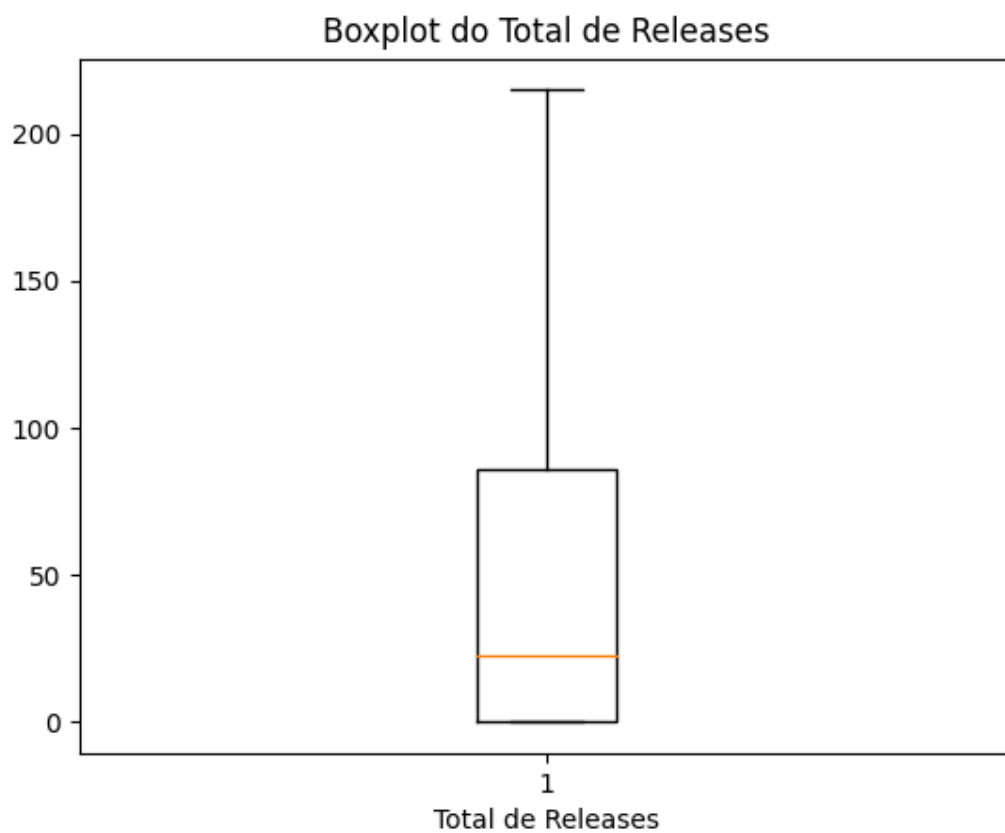
Boxplot do Total de Pull Requests



### RQ 03. Sistemas populares lançam releases com frequência?

**Hipótese:** Sim, sistemas populares tendem a manter seus repositórios atualizados sempre que possível para manter.

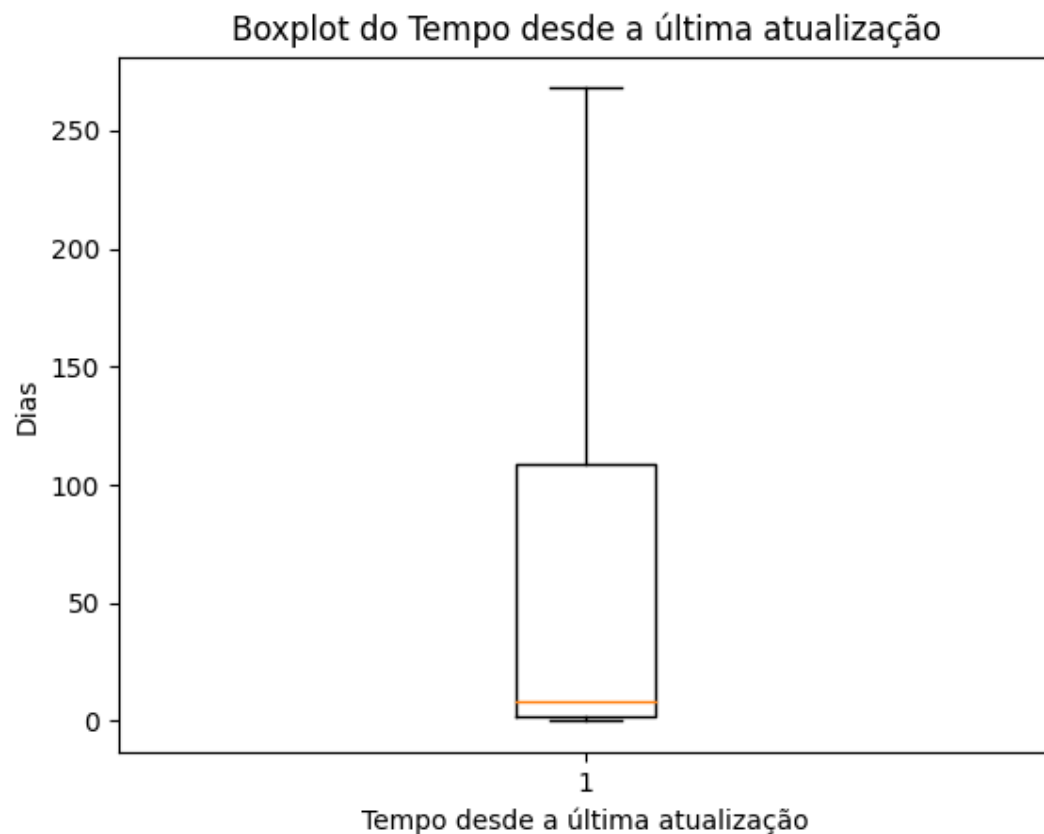
**Resultado:** A mediana da total de Releases é 22.5. A média de releases é 73,64 portanto não, os repositórios populares não costumam lançar releases com frequência e isto pode ser um problema, ainda por cima por ser popular e *open source*.



#### RQ 04. Sistemas populares são atualizados com frequência?

**Hipótese:** Sim. Pois, são vários contribuidores ajudando na evolução e desenvolvimento do sistema, principalmente em repositórios *open source*.

**Resultado:** A mediana do tempo de atualização é de 8 dias. Podemos concluir que, apesar da quantidade de contribuições, os repositórios populares costumam ser atualizados a cada 8 dias, e isso pode ser um problema, dependendo do objetivo do projeto e como funciona o processo interno dele, devido a quantidade de contribuidores e por ser um sistema *open source*.

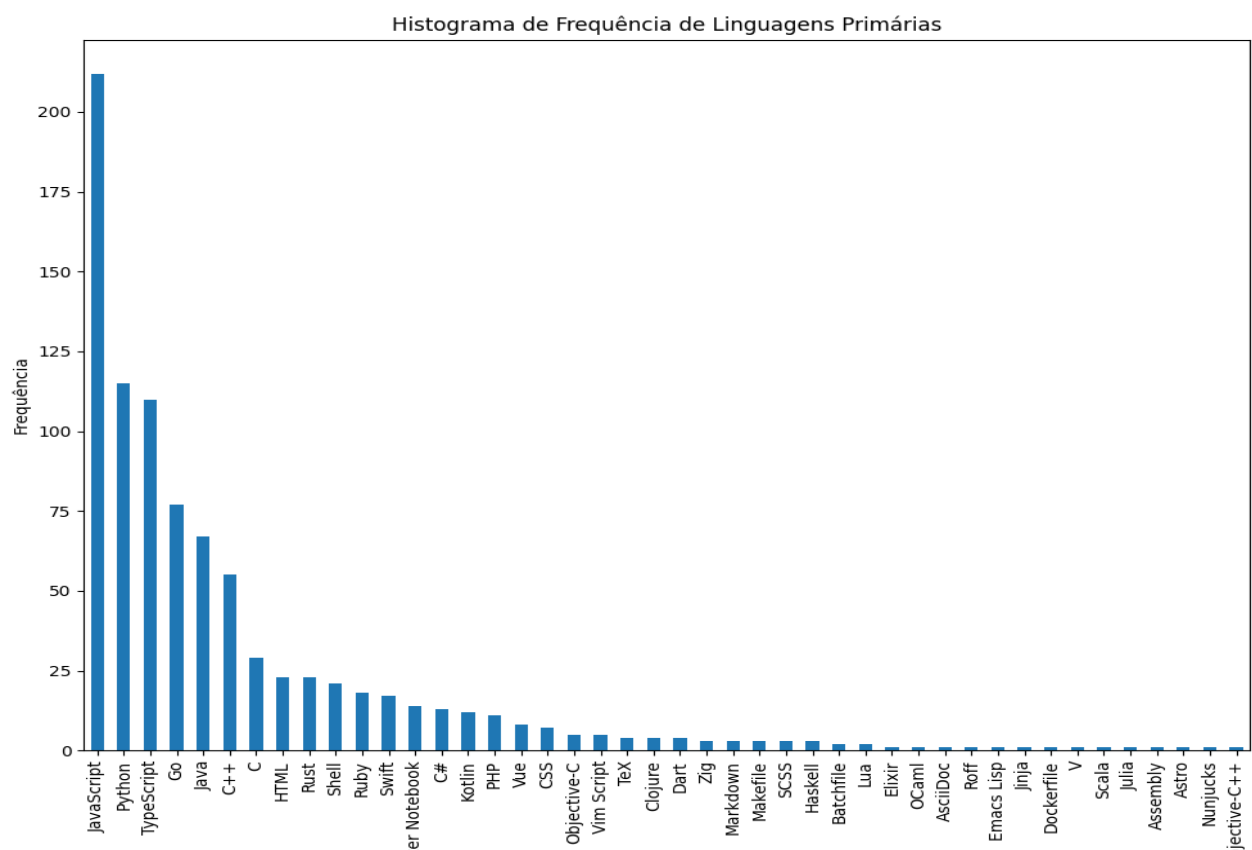




**RQ 05. Sistemas populares são escritos nas [linguagens mais populares](#) [Links to an external site?](#)**

**Hipótese:** De acordo com o site da *octoverse* o top 5 das linguagens são JavaScript, Python, Java, TypeScript e C#. Acredito que a linguagem mais popular seja o C# por está mais tempo no mercado e ser bastante usado.

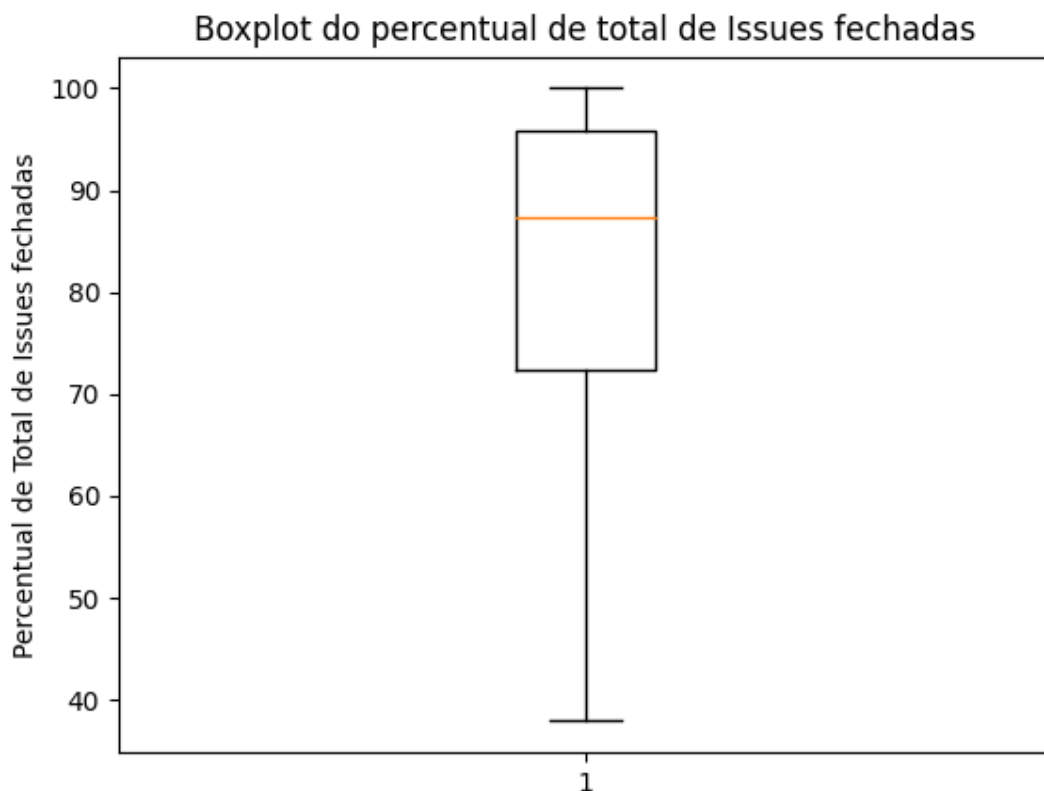
**Resultado:** O Top 5 de acordo com o resultado da mineração: JavaScript; Python; TypeScript; Go; e Java. Os repositórios mais populares estão sendo feitos em linguagem JavaScript e Python. E interessante é Python se tornou uma linguagem popular há pouco tempo e isso mostra o engajamento dos desenvolvedores em relação a esta linguagem voltada para Data Science entre outros.



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

**Hipótese:** Sim. Pensando na quantidade de contribuições feitas ao repositório, que é *open source* e popular.

**Resultado:** A mediana do total de Issues fechadas é 87.32%. Portanto, sim, com base no gráfico podemos ver que a razão de issues fechados em relação ao total de issues ultrapassar a média. Isso diz, que repositórios populares possuem muitas issues fechadas.



## 4 ANÁLISE DOS RESULTADOS

Nessa etapa serão apresentadas as análises dos resultados obtidos para cada pergunta de pesquisa durante o estudo em relação às hipóteses previamente formuladas.

### RQ 01.

**Análise:** Com base na mediana da idade dos repositórios, que é de cerca de 7 anos e 5 meses, pode-se avaliar que os repositórios mais populares são, em geral, maduros.

### RQ 02.

**Análise:** Com base na mediana da quantidade de pull requests aceitas pelos repositórios, que é de cerca de 459, pode-se avaliar que, em geral, os repositórios não costumam receber muitas contribuições externas.

### RQ 03.

**Análise:** Com base na mediana da quantidade de releases dos repositórios, que é de cerca de 22.5, pode-se avaliar que, em geral, os repositórios populares não costumam lançar releases com frequência.

### RQ 04.

**Análise:** Com base na mediana do tempo de atualização dos repositórios, que é de cerca de 8 dias, pode-se avaliar que, em geral, os repositórios populares costumam ser atualizados com uma frequência relativamente alta.

### RQ 05.

**Análise:** Com base nos resultados da mineração, pode-se avaliar que as linguagens de programação mais populares nos repositórios de código aberto são JavaScript, Python, TypeScript, Go e Java.

**RQ 06.**

**Análise:** Com base na mediana do total de issues fechadas, que é de cerca de 87.32%, pode-se avaliar que, em geral, os repositórios populares possuem muitas issues fechadas.