|  |
| --- |
| **PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**  **CURSO: ENGENHARIA DE SOFTWARE – CAMPUS LOURDES**  ANDRÉ RODRIGUES, KLAUS TELES, LUCAS TABOSA  **LABORATÓRIO DE EXPERIMENTAÇÃO DE SOFTWARE** |
| **Características de repositórios populares** |
| **Belo Horizonte**  **2024** |

**LISTA DE FIGURAS**

Gráfio Box Plot. Fonte: Elaborado pelo autor. 9

Gráfio Box Plot. Fonte: Elaborado pelo autor. 9

Gráfio Box Plot. Fonte: Elaborado pelo autor. 10

Gráfio Box Plot. Fonte: Elaborado pelo autor. 10

Gráfio Box Plot. Fonte: Elaborado pelo autor. 11

Gráfio Box Plot. Fonte: Elaborado pelo autor. 11

Gráfio Box Plot. Fonte: Elaborado pelo autor. 12

Gráfio Box Plot. Fonte: Elaborado pelo autor. 12

Gráfio Box Plot. Fonte: Elaborado pelo autor. 13

SUMÁRIO

[1 INTRODUÇÃO 4](#_Toc175920710)

[1.2.1 Objetivo geral](#_Toc175920711) 5

[1.2.2 Objetivos específicos](#_Toc175920712) 5

[1.3 Justificativa 6](#_Toc175920713)

2 [METODOLOGIA](#_Toc175920715) 7

3 [RESULTADOS](#_Toc175920716) 8

4 [CONSIDERAÇÕES FINAIS 20](#_Toc175920717)

[Referências 21](#_Toc175920718)

[APÊNDICE A – A maneira certa de fazer 22](#_Toc175920719)

[ANEXO A – Como apresentar 23](#_Toc175920720)

1 INTRODUÇÃO

Este estudo está inserido na área de análise de repositórios de software e suas características, com foco nas plataformas de colaboração em código aberto, como o GitHub. A pesquisa visa investigar os fatores que contribuem para a popularidade de projetos open-source, levantando questões sobre a maturidade dos repositórios, a contribuição externa recebida, a frequência de lançamentos, a popularidade das linguagens de programação utilizadas, e o gerenciamento de issues.

A pesquisa parte de uma pergunta central: “Quais são os fatores que tornam um repositório popular no GitHub?”. Para responder a essa pergunta, o estudo propõe explorar os 1.000 repositórios com maior número de estrelas, aplicando métricas objetivas para avaliar sua atividade, evolução e engajamento da comunidade.

Este estudo propõe atender ao objetivo geral de analisar características comuns a repositórios populares no GitHub, e aos seguintes objetivos específicos:

– Avaliar se sistemas populares são maduros ou antigos.

– Investigar a quantidade de contribuição externa recebida por sistemas populares.

– Analisar a frequência de lançamentos de novos releases nesses sistemas.

– Determinar a frequência de atualizações realizadas nos repositórios.

– Verificar se sistemas populares utilizam linguagens de programação amplamente usadas.

– Avaliar o percentual de issues fechadas nesses repositórios.

– Examinar se sistemas escritos em linguagens populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência.

### 1.2.1 Objetivo geral

Este estudo é relevante porque **analisa** os fatores que explicam a popularidade dos repositórios no GitHub, uma plataforma central para o desenvolvimento colaborativo de software. Ao **avaliar** a influência de aspectos como linguagens de programação, frequência de atualizações e engajamento da comunidade, o estudo busca **verificar** como essas variáveis contribuem para o sucesso de projetos open-source.

Além disso, a pesquisa **explica** como práticas eficientes de manutenção e colaboração podem impactar a trajetória de projetos, oferecendo uma compreensão mais profunda das dinâmicas de desenvolvimento de software. Dessa forma, o estudo **esclarece** pontos cruciais para a otimização de estratégias de gestão e **compreende** o papel dessas práticas na promoção de projetos bem-sucedidos.

### 1.2.2 Objetivos específicos

**Identificar** as características comuns entre os repositórios mais populares do GitHub.

**Examinar** a quantidade de contribuições externas nos repositórios.

**Verificar** a relação entre a maturidade (idade) dos projetos e sua popularidade.

**Avaliar** a frequência de atualizações e lançamentos nos sistemas mais populares.

**Classificar** as linguagens de programação utilizadas nos repositórios populares e seu impacto no sucesso do projeto.

**Comparar** a quantidade de contribuições recebidas por sistemas escritos em linguagens populares e não populares.

**Analisar** a taxa de fechamento de issues em sistemas populares versus menos populares.

## 1.3 Justificativa

A relevância deste estudo está diretamente relacionada ao impacto crescente que os projetos open-source têm sobre a indústria de software, inovação tecnológica e a comunidade de desenvolvedores. O GitHub, sendo uma das maiores plataformas de hospedagem de código, desempenha um papel central na colaboração global e no desenvolvimento de soluções tecnológicas. Identificar os fatores que tornam um repositório popular na plataforma é essencial para compreender como as dinâmicas de colaboração, a escolha de linguagens de programação e o gerenciamento de projetos podem influenciar o sucesso de projetos open-source.

Do ponto de vista prático, este estudo oferece insights para desenvolvedores e gestores de projetos que buscam aumentar o engajamento e a contribuição externa em seus repositórios. A compreensão das características que atraem mais contribuições, como a escolha de linguagens de programação populares e a manutenção ativa, pode fornecer estratégias eficazes para o crescimento e a sustentabilidade de projetos.

Teoricamente, o estudo amplia os horizontes da pesquisa em engenharia de software ao fornecer dados concretos sobre a relação entre popularidade e práticas de desenvolvimento. A análise das métricas de repositórios populares contribui para uma melhor compreensão de como a comunidade open-source funciona e oferece subsídios para pesquisas futuras que visam aprimorar as práticas de desenvolvimento colaborativo e a gestão de projetos.

2 METODOLOGIA

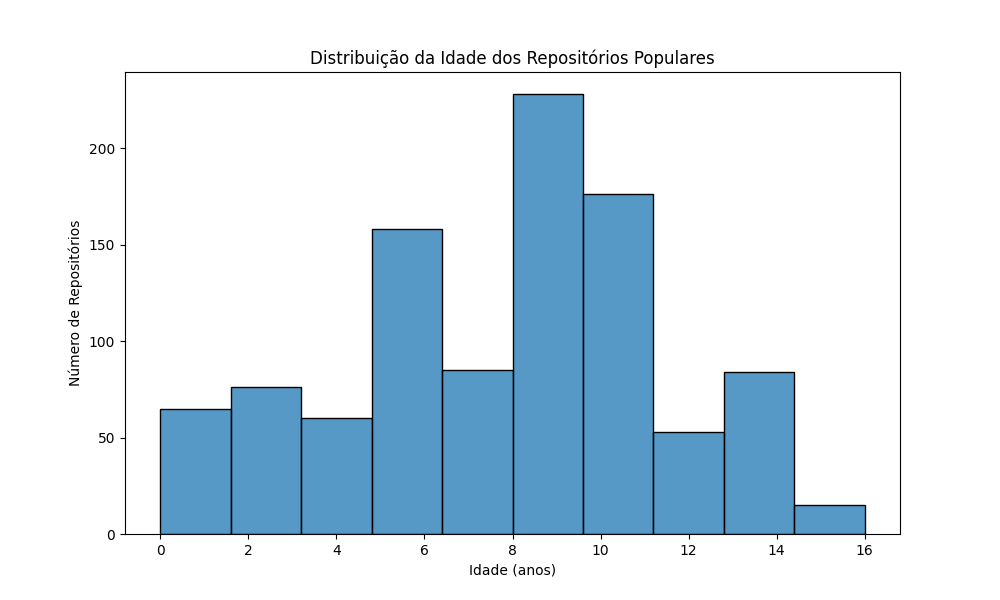
Neste capítulo, serão apresentados os métodos e procedimentos adotados para o desenvolvimento deste estudo. A metodologia utilizada segue uma abordagem quantitativa, uma vez que o foco está na análise de dados numéricos extraídos dos repositórios mais populares do GitHub. Para garantir a precisão dos resultados, foi utilizado um script automatizado que coletou dados relevantes de 1.000 repositórios, permitindo realizar uma análise detalhada das questões de pesquisa previamente estabelecidas.

A escolha da abordagem quantitativa se justifica pela necessidade de mensurar características como a frequência de contribuições, atualizações, lançamentos, entre outros aspectos que podem ser objetivamente quantificados. Esses dados foram processados e analisados de forma a responder às hipóteses formuladas, garantindo a robustez dos achados e a consistência científica do estudo.

O rigor na coleta e análise dos dados foi essencial para assegurar a validade dos resultados, uma vez que a metodologia adotada permitiu identificar padrões e relações que explicam o sucesso e popularidade dos repositórios analisados.

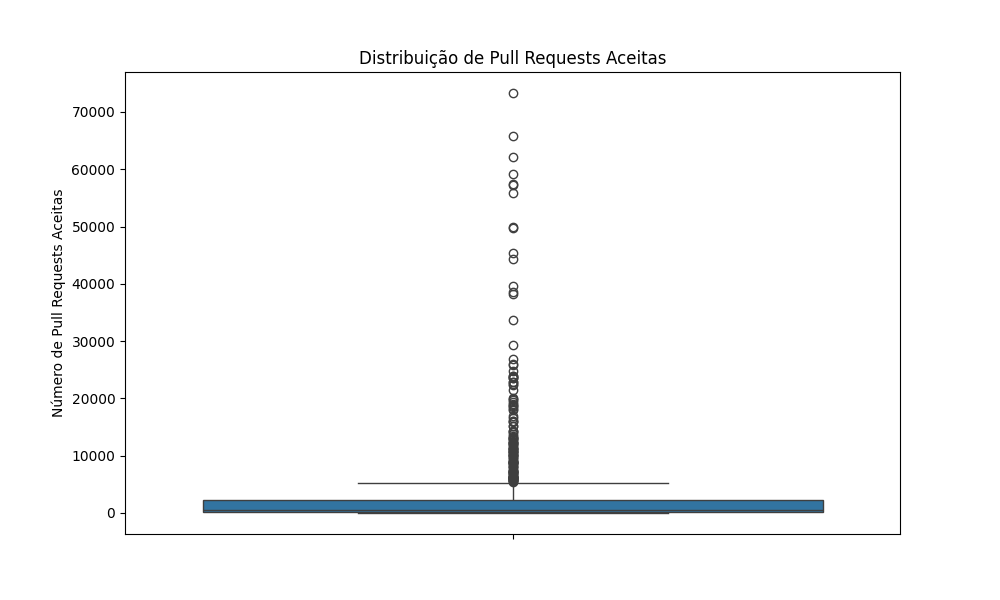
3 RESULTADOS

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



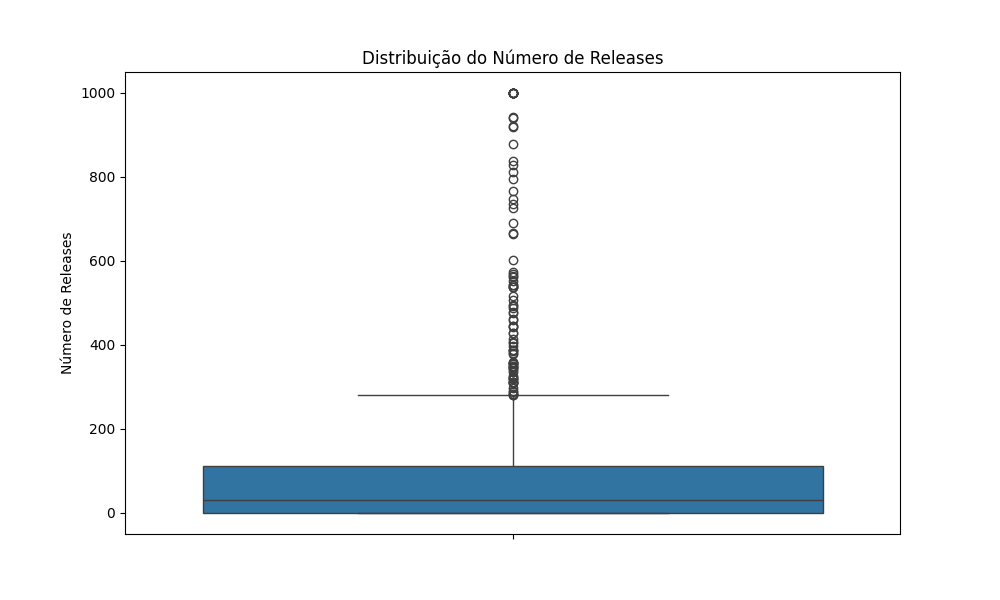
Foi verificado que a média de idade dos sistemas populares analizados segue uma distribuição normal, sendo o mais antigo com 8 anos. Isso posto, os parâmetros mais usado na indústria de software dizem que uma margem de idade de 5 até 10 anos já significa que um nível de maturidade maior será encontrado, então pode-se concluir que uma parte significativa não é considerada velha na visão do mercado mas a maioria sim. Deve-se levar em conta que a plataforma Github tem 16 anos de existência.

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

****

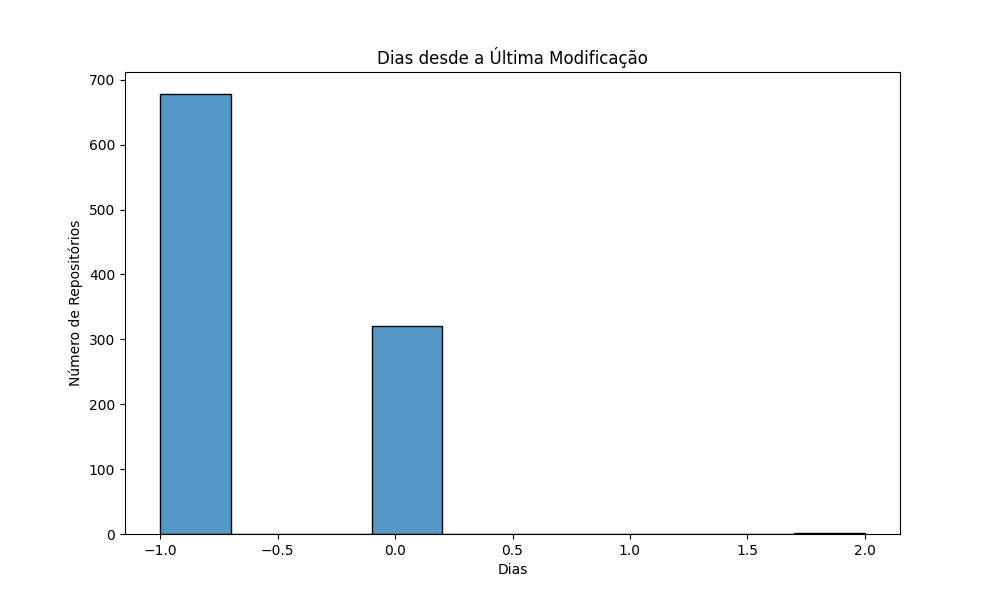
Tudo indica que sistemas populares recebem muito foco por parte dos desenvolvedores, algumas amostras de dados receberam de contribuição externa um número de pull requests na casa de dezenas de milhar, principalmente aquelas que usam linguagens muito populares.

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



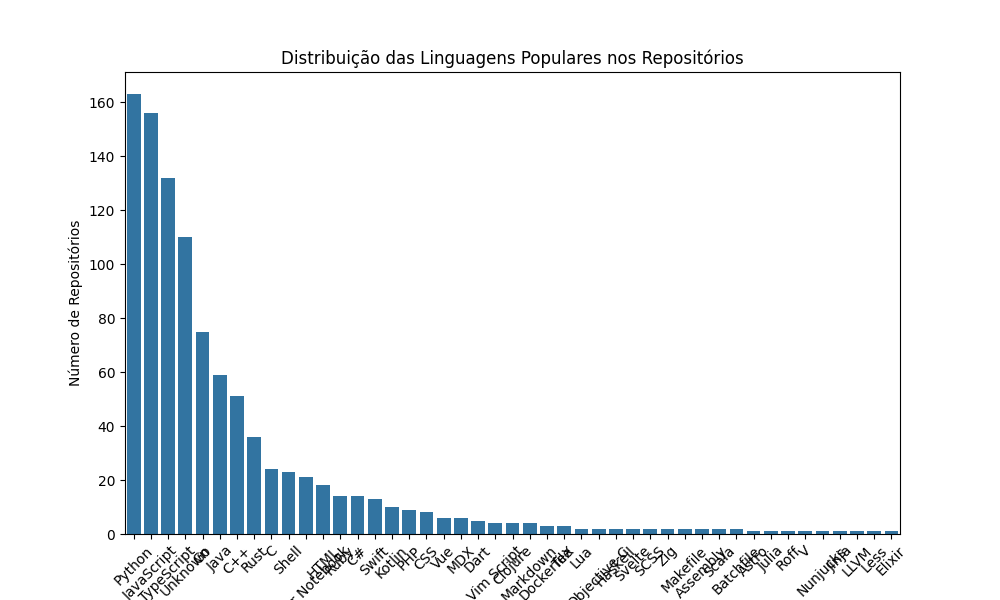
Não necessariamente, a maioria dos sistemas possui um número moderado de releases, algo normal para aplicações já consolidadas, mas há exemplos fora da curva com lançamentos muito frequentes, indicando que não existe uma correlação direta.

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



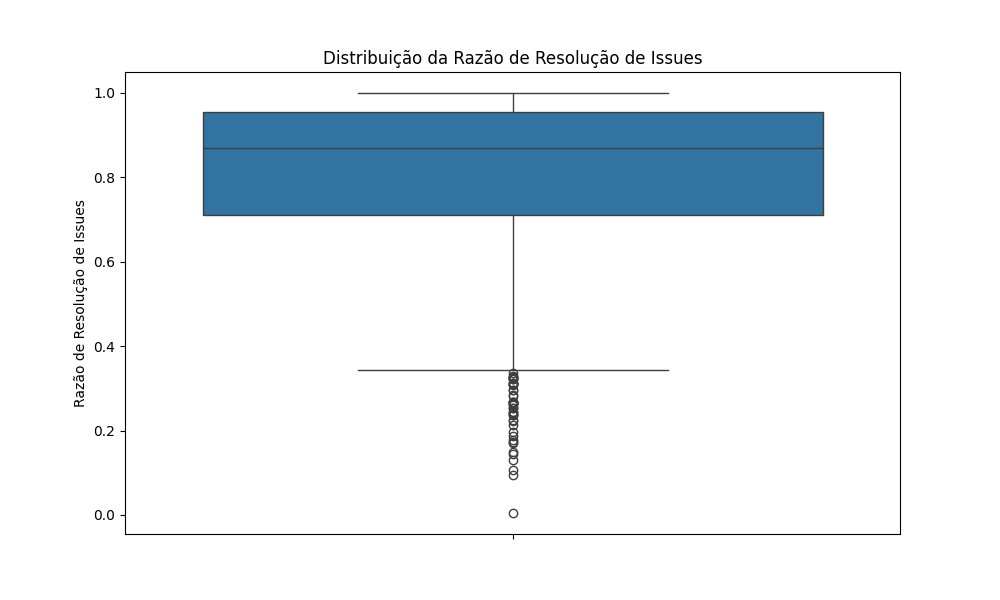
Sim, a maioria dos sistemas populares tem atualizações recentes, com a maioria dos repositórios sendo atualizados há menos de um dia.

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



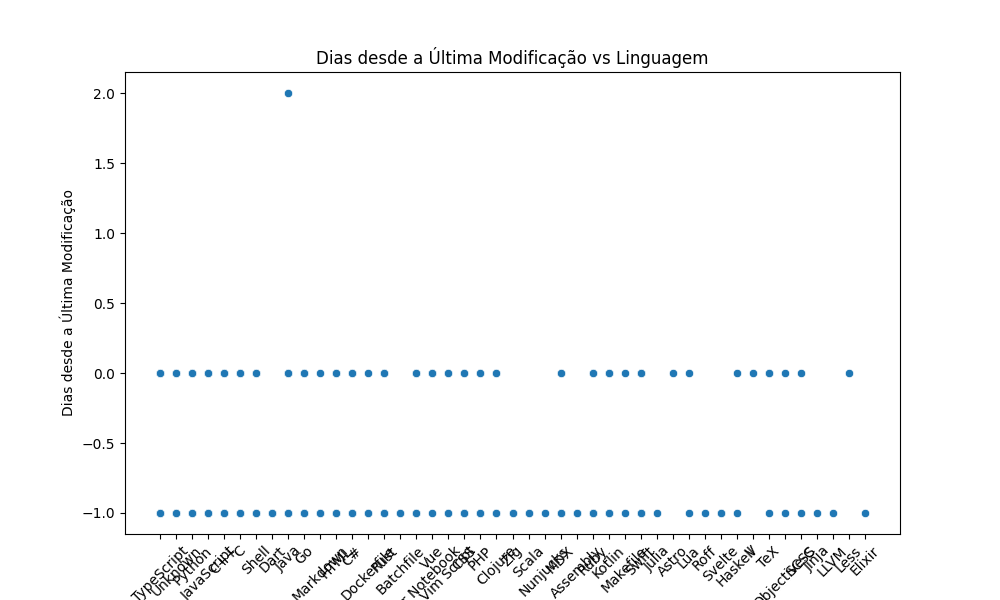
Sim, há uma forte correlação, com a maioria dos repositórios populares sendo escritos em linguagens amplamente utilizadas como Python, e Javascript que segundo o blog “somosTera” são as linguagens top 1 e top 3 respectivamente.

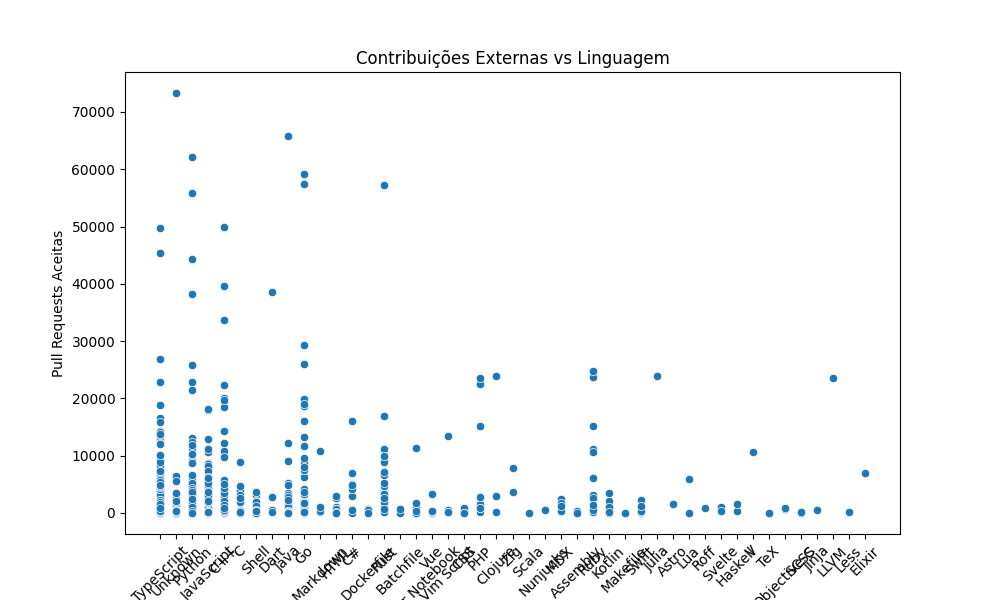
**RQ 06. Sistemas populares possuem um alto percentual de issues fechadas?**

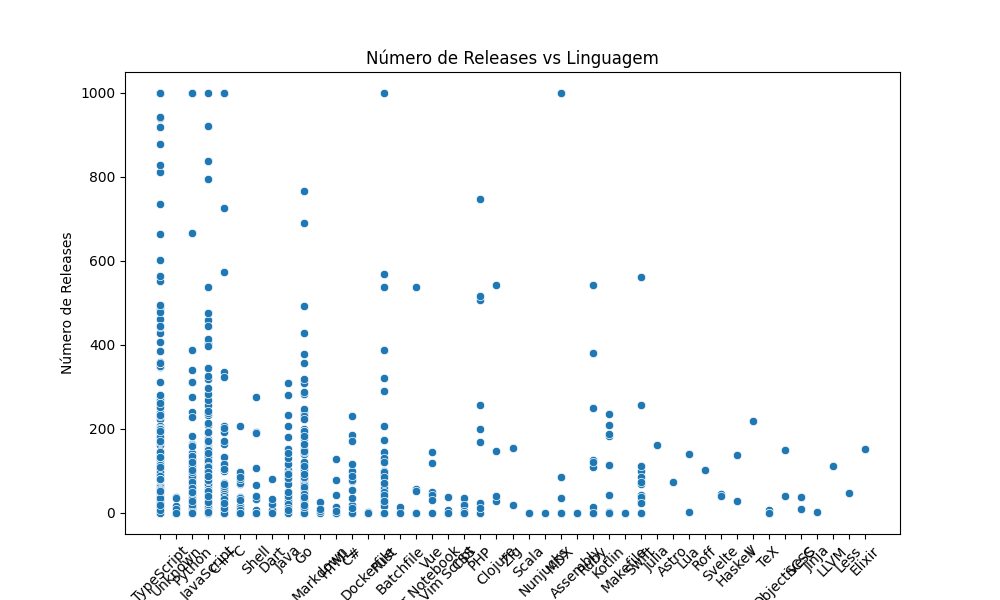


A maioria dos sistemas populares tem uma alta taxa de resolução de issues, com uma media de soluções de issues por número total de issues de 0.87, o que é um alto percentual.

**RQ 07: Sistemas escritos em linguagens mais populares recebem mais contribuição externa,lançam mais releases e são atualizados com mais 1 frequência?**







Sim, sistemas que utilizam linguagens populares como por exemplo Python e JavaScript tendem a receber mais contribuições externas, lançam mais releases e têm atualizações mais frequentes, refletindo sua alta atividade e popularidade.

4 CONSIDERAÇÕES FINAIS

Este estudo buscou analisar as características que tornam um repositório popular no GitHub, utilizando dados quantitativos coletados de 1.000 repositórios com maior número de estrelas. Os resultados obtidos permitiram responder às perguntas de pesquisa e confirmar, em sua maioria, as hipóteses inicialmente propostas.

Os repositórios populares, em sua grande maioria, são maduros, com uma média de idade que demonstra sua consolidação no mercado de software. Verificou-se que a contribuição externa é significativa, especialmente em projetos que utilizam linguagens populares como Python e JavaScript. No entanto, embora sistemas populares tenham atualizações frequentes, a quantidade de releases não segue um padrão definido, o que sugere que a popularidade não está diretamente ligada à frequência de lançamentos.

Outro ponto importante é a alta taxa de fechamento de issues nos sistemas populares, o que indica uma manutenção ativa e um grande engajamento da comunidade. Além disso, repositórios escritos em linguagens populares tendem a receber mais contribuições e serem mais frequentemente atualizados, reforçando a importância dessas linguagens no desenvolvimento colaborativo.

Os achados deste estudo fornecem uma base sólida para o entendimento das práticas que influenciam o sucesso de repositórios open-source e abrem caminho para novas pesquisas que possam explorar outros fatores não contemplados aqui, como a estrutura organizacional dos projetos ou o impacto das práticas de governança.

Propostas para trabalhos futuros incluem uma análise mais profunda sobre a influência de fatores externos, como a visibilidade em redes sociais, bem como a investigação de padrões em linguagens menos populares, mas que apresentam casos de sucesso específicos.

# Referências

Internet:

Linguagens de programação mais utilizadas. BH: Escola superior de Redes, 2024. Disponível em:<https://esr.rnp.br/desenvolvimento-de-sistemas/linguagens-de-programacao-mais-utilizadas/>. Acesso em: 4/9/2024.