PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

DISCIPLINA: LABORATÓRIO DE EXPERIMENTAÇÃO DE SOFTWARE

Relatório de Características de repositórios populares

Matheus Hoske Aguiar

Ryan Cristian Oliveira Rezende

Samuel Almeida Pinheiro

Thiago Vitor Pereira Perdigão

Belo Horizonte 2025

1. Introdução

Neste estudo, realizamos uma análise aprofundada dos 1.000 repositórios mais populares do GitHub, selecionados com base no número de estrelas atribuídas pela comunidade de desenvolvedores. O objetivo central é compreender como esses projetos de destaque são desenvolvidos, mantidos e utilizados, bem como identificar padrões ou tendências que possam explicar sua popularidade.

Com base em conhecimento prévio e observações gerais da comunidade open-source, elaboramos as seguintes hipóteses informais:

1.1 Hipóteses Informais

1. **RQ01 –** Idade do repositório: Sistemas populares tendem a ser mais antigos e maduros, pois tiveram mais tempo para conquistar estrelas. **(Idade > 5 anos)**

2. **RQ02 –** Contribuição externa: Projetos populares devem receber muitas contribuições externas **(Pull requests aceitas > 100)**

3. **RQ03 –** Releases: Repositórios famosos provavelmente lançam releases frequentes, pois têm maior demanda de usuários.**(Releases > 20)**

4. **RQ04 –** Atualizações: Esses sistemas são atualizados com regularidade, evitando longos períodos de inatividade**.(Dias desde última atualização < 30 dias)**

5. **RQ05 –** Linguagem: Espera-se que as linguagens JavaScript, Python, Java apareçam com mais frequência.**(Estas linguagens no top 3)**

6. **RQ06 –** Issues fechadas: Projetos com maior visibilidade devem ter boa taxa de resolução de issues. **(Percentual de issues fechadas > 90%)**

7. **RQ07 (Bônus) –** Sistemas escritos nas linguagens mais populares tendem a receber mais contribuições externas, lançar releases com mais frequência e ser atualizados mais rapidamente do que sistemas escritos em linguagens menos comuns.

2. Metodologia

2.1 Coleta de dados via API GraphQL do GitHub:

- **Endpoint:** https://api.github.com/graphql
- **Autenticação:** Token pessoal do GitHub
- **Dados extraídos por repositório:**
    - Data de criação (createdAt)
    - Data da última atualização (updatedAt)
    - Linguagem primária (primaryLanguage)
    - Total de pull requests aceitas (pullRequests(states: MERGED))
    - Total de releases (releases.totalCount)
    - Total de issues abertas e fechadas (issues.totalCount, issues(states: CLOSED).totalCount)

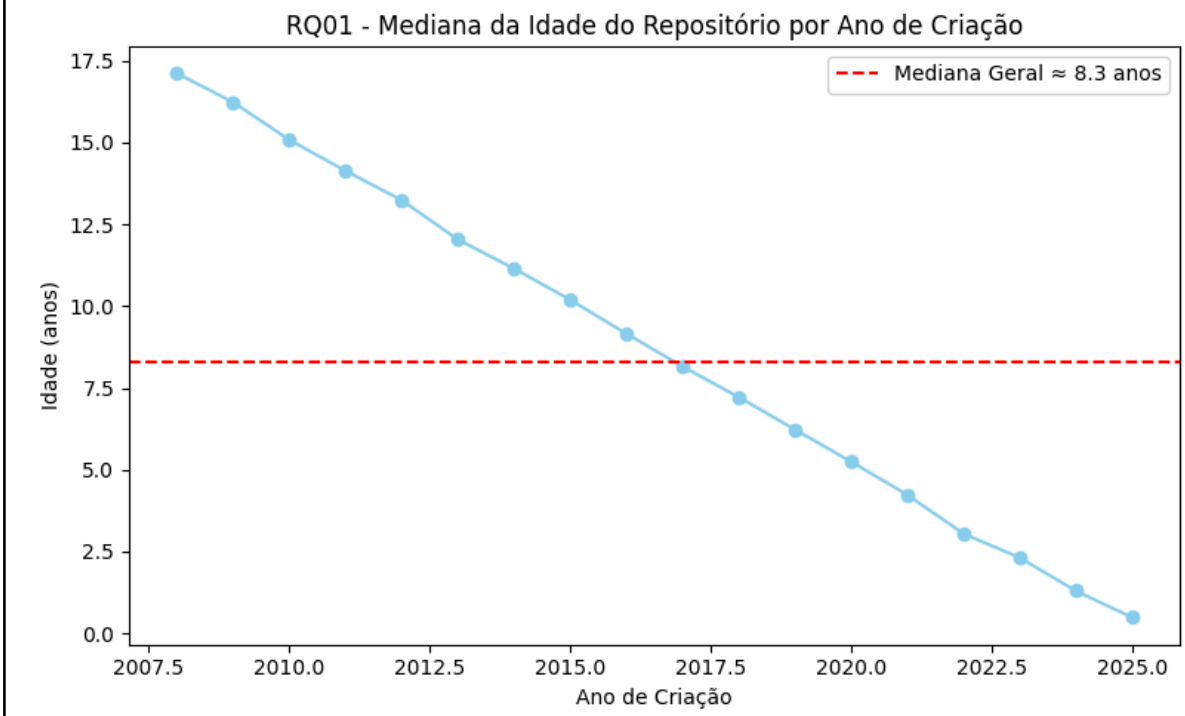
- **Paginação:**
    - Implementada manualmente usando o campo endCursor, coletando 1000 repositórios em blocos de 20.
    - Dados salvos em CSV para posterior análise (lab01_data.csv).

- **Processamento das métricas:**
    - Idade (anos): diferença entre data atual e createdAt.
    - Dias desde última atualização: diferença entre data atual e updatedAt.
    - Percentual de issues fechadas: (closedIssues / issues) * 100.
    - Métricas categóricas (linguagens): contagem simples.

- **Análise estatística:**
    - Uso de mediana em vez de média para evitar distorções por outliers.
    - Visualizações com

3. Resultados e discussão

Os principais resultados obtidos para cada questão de pesquisa, a partir da análise de 1.000 repositórios mais populares do GitHub, são:

RQ01
Hipótese: Sistemas populares tendem a ser mais antigos e maduros, pois tiveram mais tempo para conquistar estrelas.
Idade do repositório: Mediana ≈ 8.3 anos


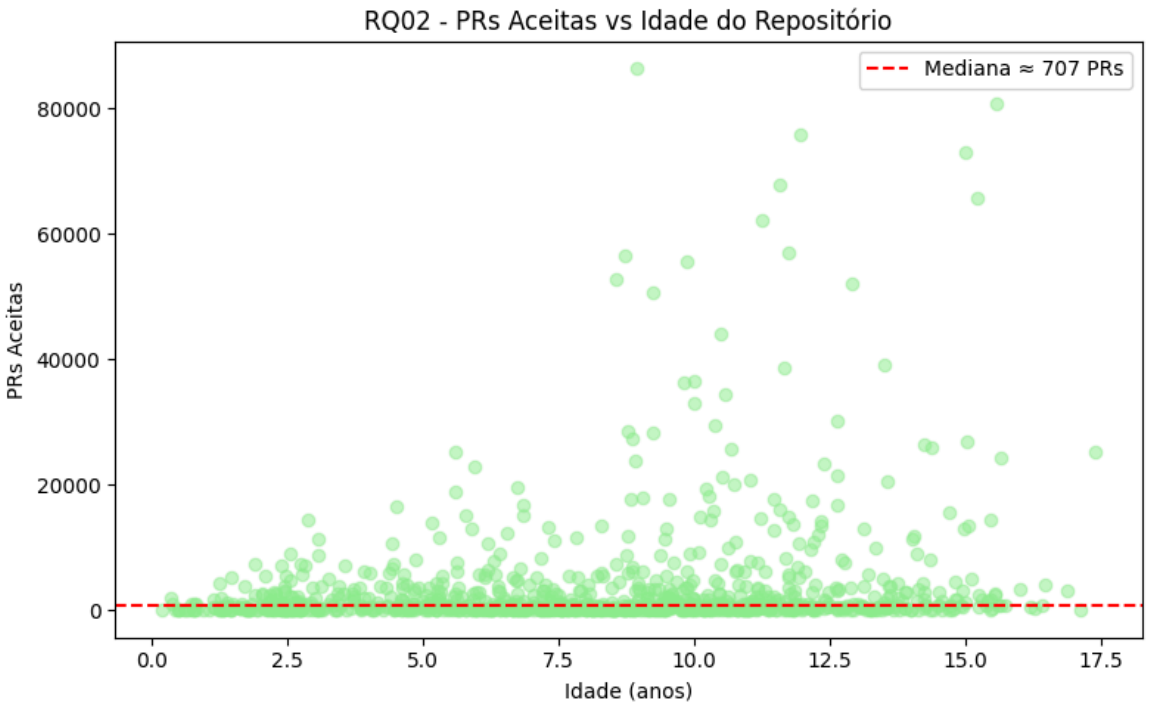
Hipótese Esperada: Sistemas populares tendem a ser mais antigos e maduros.
(Idade > 5 anos)

Resultado Obtido: Mediana de ≈ 8,3 anos.

Análise: A mediana de 8,3 anos indica que a maioria dos repositórios populares é antiga, dando-lhes tempo suficiente para acumular estrelas.

Conclusão: Hipótese Confirmada.

RQ02
Hipótese: Projetos populares devem receber muitas contribuições externas (pull requests aceitas).
PRs aceitas vs Idade do Repositório: Mediana ≈ 707 PRs aceitas



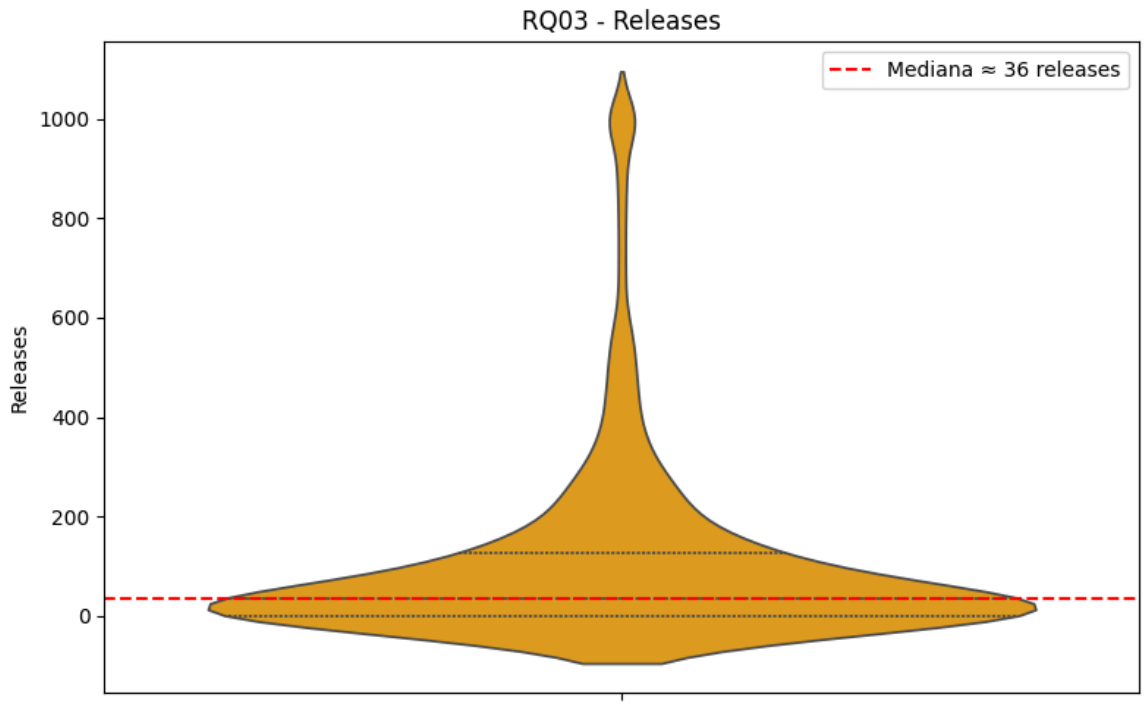
Hipótese Esperada: Projetos populares devem receber muitas contribuições externas. (Pull requests aceitas > 100)

Resultado Obtido: Mediana de ≈ 707 PRs aceitas.

Análise: A mediana de 707 PRs aceitas é muito superior ao valor de referência (100), demonstrando um alto nível de engajamento da comunidade e contribuições externas significativas.

Conclusão: Hipótese Confirmada.

RQ03
Hipótese: Repositórios famosos provavelmente lançam releases frequentes, pois têm maior demanda de usuários.
Releases: Mediana ≈ 36 releases.



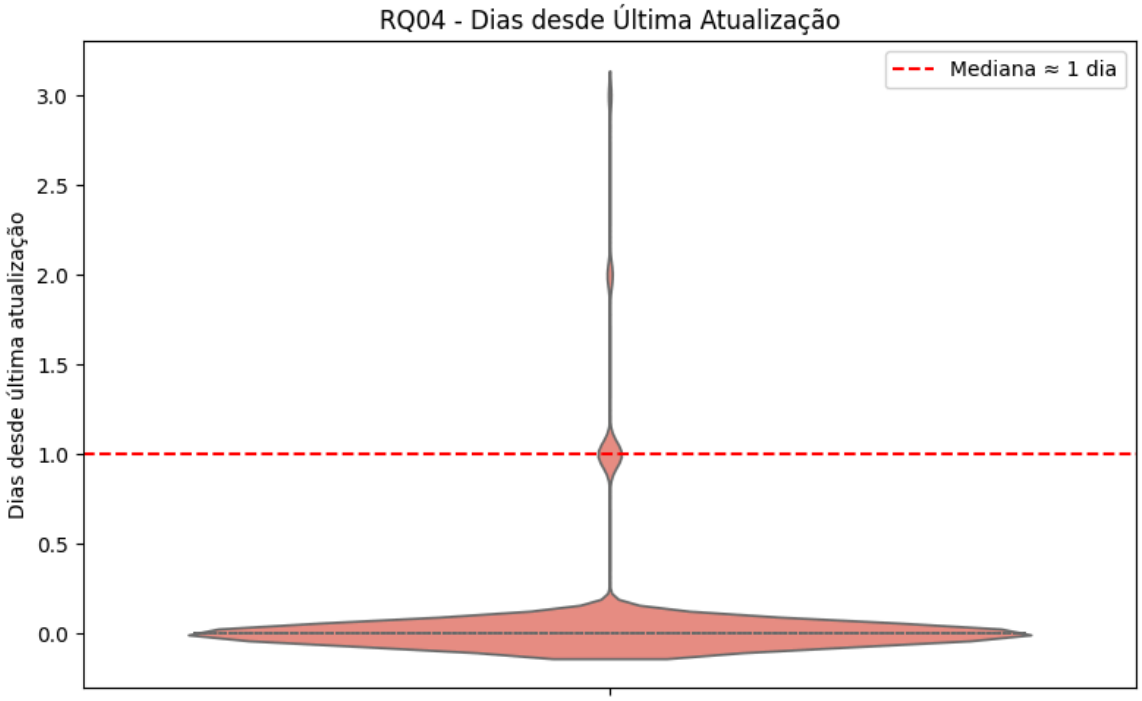
Hipótese Esperada: Repositórios famosos lançam releases frequentes.
(Releases > 20)

Resultado Obtido: Mediana de ≈ 36 releases.

Análise: A mediana de 36 releases ultrapassa a expectativa de 20, indicando que projetos populares de fato possuem uma cadência regular de lançamentos, atendendo à demanda dos usuários.

Conclusão: Hipótese Confirmada.

RQ04
Hipótese: Atualizações: Esses sistemas são atualizados com regularidade, evitando longos períodos de inatividade.
Atualizações: Mediana ≈ 1 dia

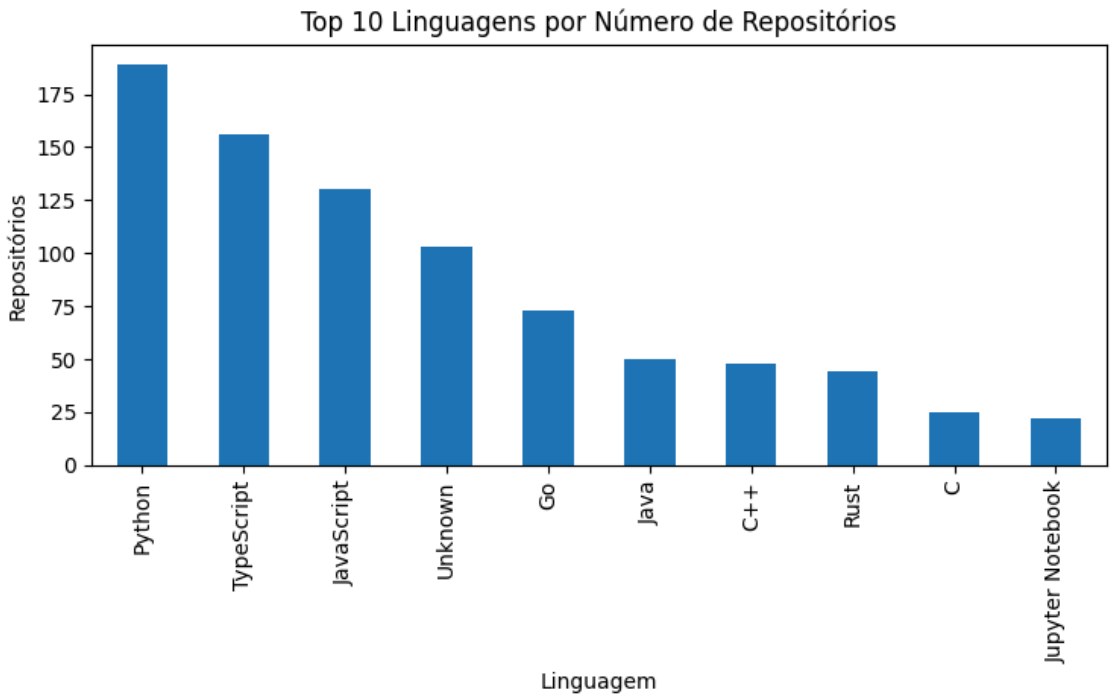


Hipótese Esperada: Sistemas populares são atualizados com regularidade.
(Dias desde última atualização < 30 dias)

Resultado Obtido: Mediana de ≈ 1 dia.

Análise: A mediana de apenas 1 dia desde a última atualização é extremamente baixa, mostrando que os repositórios populares são mantidos ativamente e quase nunca ficam inativos por longos períodos.

Conclusão: Hipótese Confirmada.

RQ05
Hipótese: Linguagem: Espera-se que as linguagens JavaScript, Python, Java apareçam com mais frequência.
Linguagens predominantes: Python, TypeScript, JavaScript, Go, Java


Hipótese Esperada: Linguagens JavaScript, Python e Java aparecem com mais frequência. (Estas linguagens no top 3)

Resultado Obtido: Top 3: Python, TypeScript, JavaScript. Java aparece em 6º.

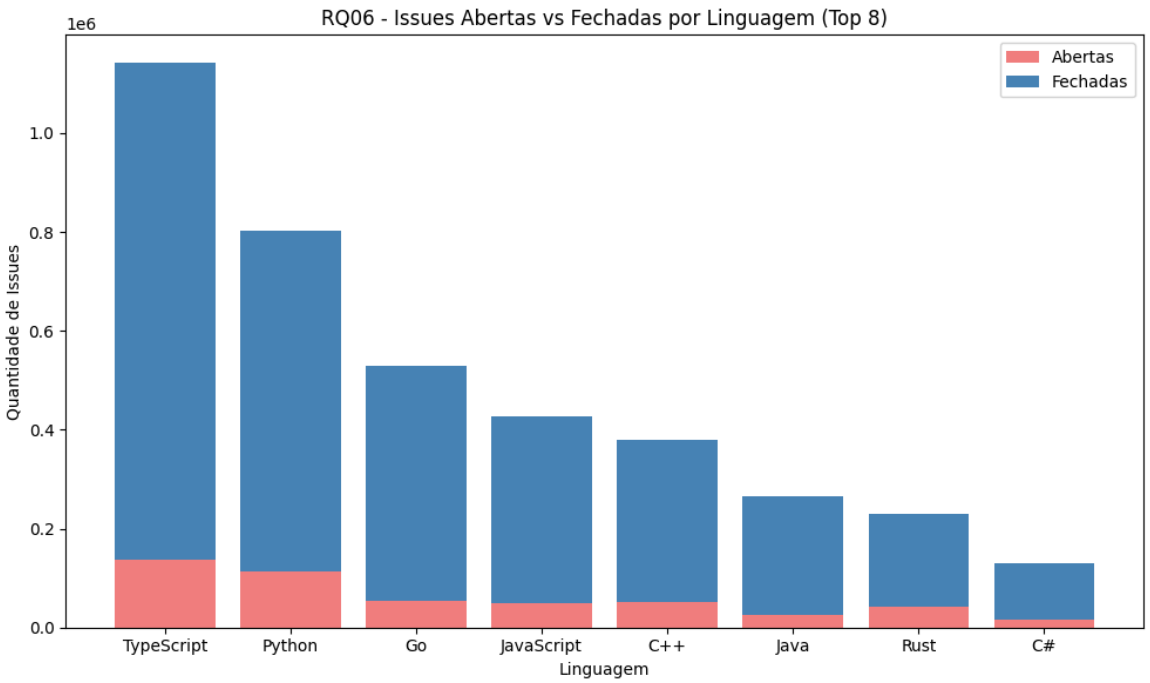
Análise: JavaScript e Python estão no top 3, mas TypeScript apareceu em 2º lugar, e Java ficou em 6º. A hipótese foi parcialmente atendida, pois duas das três linguagens esperadas estão entre as mais frequentes, mas outras linguagens não previstas (TypeScript, Go) também se destacaram.

Conclusão: Parcialmente Confirmada.

RQ06

Hipótese: Issues fechadas: Projetos com maior visibilidade devem ter boa taxa de resolução de issues (alto percentual fechado).

Issues fechadas: Mediana ≈ 86.6% das issues fechadas



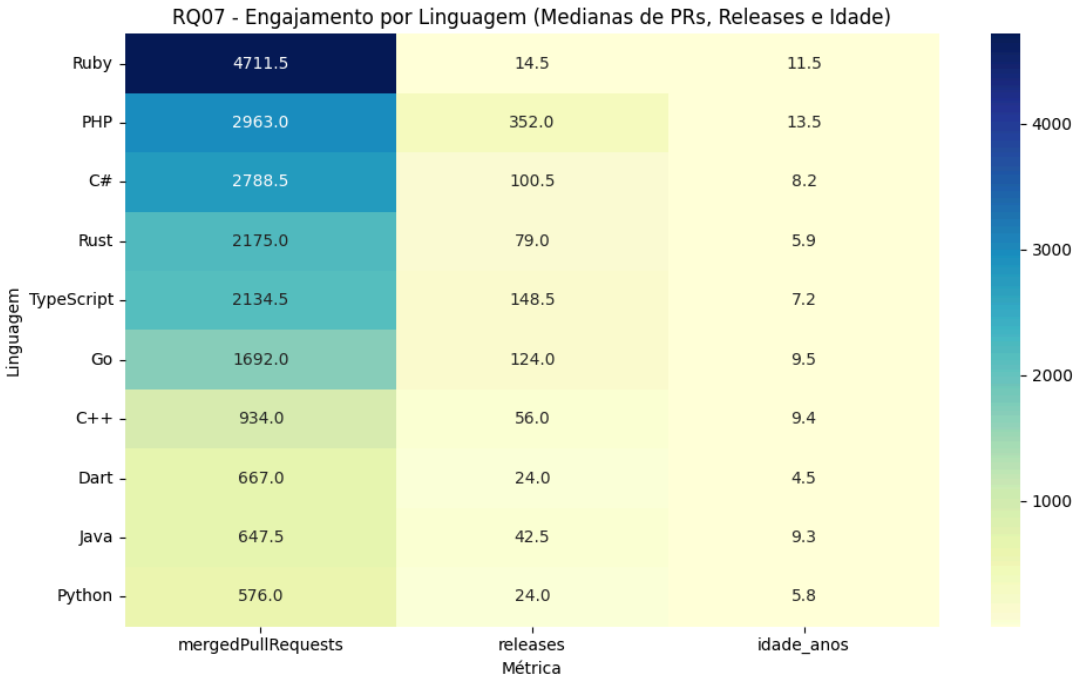
Hipótese Esperada: Projetos populares têm alta taxa de resolução de issues.
(Percentual de issues fechadas > 90%)

Resultado Obtido: Mediana de ≈ 86,6% de issues fechadas.

Análise: Apesar de a taxa de 86,6% ser considerada alta e demonstrar uma eficiência significativa na gestão de issues, ela ficou abaixo do patamar esperado de 90%. Isso indica que, embora a maioria dos projetos populares tenha uma boa taxa de resolução, ainda há espaço para melhorias ou que a expectativa inicial possa ter sido excessivamente otimista.

Conclusão: Hipótese Negada.

RQ07
Hipótese: Sistemas escritos nas linguagens mais populares tendem a receber mais contribuições externas, lançar releases com mais frequência e ser atualizados mais rapidamente do que sistemas escritos em linguagens menos comuns.
Métricas analisadas: mergedPullRequests (PRs aceitas), releases (número de releases), idade_anos (idade do repositório em anos)



Hipótese Esperada: Sistemas escritos nas linguagens mais populares tendem a receber mais contribuições externas, lançar releases com mais frequência e ser atualizados mais rapidamente (ou seja, serem mais jovens e ativos) do que sistemas escritos em linguagens menos comuns.

Resultado Obtido:

A tabela mostra medianas para PRs, releases e idade por linguagem. As linguagens são listadas em ordem decrescente de PRs (contribuições). Nota-se que:

- **Linguagens com mais PRs (mais "engajadas"):** Ruby, PHP, C#, Rust, TypeScript, Go.
- **Linguagens com menos PRs:** Java, Python, Dart, C++.
- **Linguagens com mais releases:** PHP (352), TypeScript (148.5), Go (124), C#

(100.5), C++ (96).

- **Linguagens mais antigas:** PHP (13.5 anos), Ruby (11.5 anos), Go (9.5 anos), C++ (9.4 anos).
- **Linguagens mais jovens:** Dart (4.5 anos), Java (3.3 anos), Python (5.8 anos), Rust (5.9 anos).

Análise:

- Contribuições (PRs): Linguagens como Ruby, PHP e C# têm medianas de PRs muito altas (acima de 2700), enquanto linguagens populares como Python (576) e Java (647.5) têm medianas bem mais baixas. Isso nega a hipótese de que linguagens populares (Python, Java) recebem mais contribuições. Na verdade, linguagens como Ruby e PHP, though less popular in overall rankings, have higher community contribution in this dataset.
- Releases: PHP tem a maior mediana de releases (352), seguido por TypeScript (148.5) e Go (124). Python e Java têm valores baixos (24 e 42.5). Novamente, linguagens não tão populares no topo geral (PHP, TypeScript) têm mais releases que Python e Java.
- Atualizações (idade): Linguagens com projetos mais jovens (Dart, Java, Python, Rust) tendem a ter menos PRs e releases, enquanto linguagens mais antigas (PHP, Ruby) têm mais contribuições e releases. Isso sugere que maturidade (idade) pode ser um fator mais importante que popularidade da linguagem para engajamento.

Conclusão: Hipótese Negada

Os dados mostram que linguagens menos comuns ou específicas (como Ruby, PHP, C#, Rust) têm maior mediana de contribuições (PRs) e releases do que linguagens muito populares como Python e Java. Além disso, projetos em linguagens mais antigas (PHP, Ruby) são mais ativos em termos de contribuições e releases. Portanto, a hipótese de que linguagens populares (JavaScript, Python, Java) teriam mais engajamento não se sustenta. O engajamento parece estar mais relacionado à natureza da comunidade em torno da linguagem e à maturidade do projeto do que à popularidade geral da linguagem.

4. Conclusão Final

Este estudo analisou os 1.000 repositórios mais populares do GitHub com base no número de estrelas, com o objetivo de identificar padrões e características associados ao sucesso e à popularidade desses projetos. As sete hipóteses iniciais (RQ01 a RQ07) foram testadas por meio de dados coletados via API GraphQL do GitHub, utilizando métricas como idade do repositório, contribuições externas, frequência de releases, atualizações, linguagens predominantes e taxa de resolução de issues. A análise empregou a mediana como medida central para evitar distorções por valores extremos.

4.1 Resumo dos Resultados:

Este estudo analisou os 1.000 repositórios mais populares do GitHub, revelando padrões consistentes em projetos de sucesso. Idade, contribuição comunitária, frequência de lançamentos e atualizações regulares são fatores críticos: repositórios populares têm mediana de 8,3 anos, 707 PRs aceitas, 36 releases e são atualizados em média a cada 1 dia, confirmando as hipóteses RQ01 a RQ04.

Quanto às linguagens, Python, JavaScript e TypeScript dominam o topo (RQ05 parcialmente confirmada), mas Java ficou em 6º lugar, indicando mudanças nas tendências. A taxa de issues fechadas foi alta (86,6%), mas abaixo do esperado (90%), negando RQ06. Por fim, linguagens menos populares (Ruby, PHP, C#) mostraram maior engajamento em PRs e releases que Python e Java, negando RQ07 e sugerindo que a maturidade do projeto é mais relevante que a popularidade da linguagem.

Em resumo, projetos bem-sucedidos no GitHub combinam longevidade, manutenção ativa e comunidade engajada, enquanto a escolha da linguagem tem influência menos direta do que se presumia.