

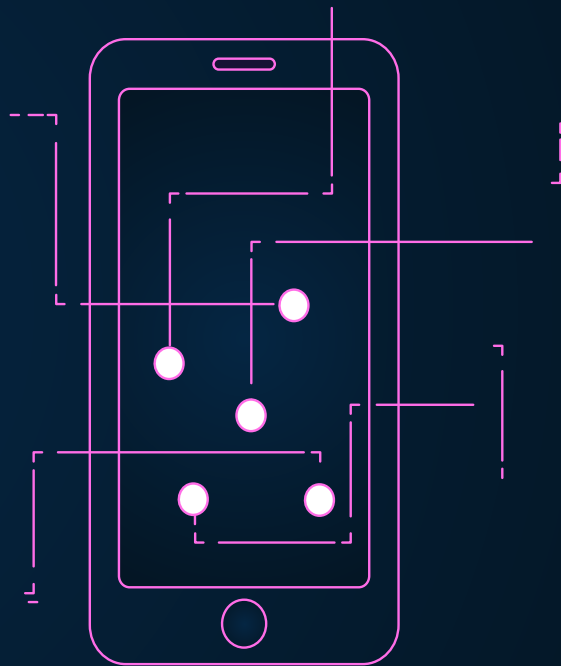


Características de Repositorios Populares

Gabriel Faria, João Victor Salim, Lucas Garcia, Maisa Pires e Miguel Vieira

INTRODUÇÃO

- **Escopo** 1000 repositórios mais estrelados do GitHub (snapshot agosto/2025)
- **Objetivo** medir maturidade, contribuição externa, releases, atualização, linguagens e manutenção
- **Abordagem** coleta via API GraphQL e análise em Python/Pandas com visualizações
- **Valor** gerar insights práticos para escolher tecnologias e melhorar manutenção



HIPÓTESES

RQ01

"Repositórios populares terão idade média superior a 5 anos, pois precisam de tempo para acumular estrelas e comunidade."

RQ02

"Repositórios com mais estrelas terão mediana de PRs aceitas acima de 500, indicando alta atividade de contribuidores externos."

RQ03

"A mediana de releases será maior que 20, sugerindo atualizações regulares (pelo menos 2-3 por ano)."

HIPÓTESES

RQ04

"Pelo menos 80% dos repositórios terão sido atualizados nos últimos 6 meses."

RQ05

"JavaScript, Python e Java representarão mais de 60% das linguagens primárias."

RQ06

"O percentual médio de issues fechadas será superior a 70%, indicando boa manutenção."

HIPÓTESES

RQ07

"Linguagens mais populares apresentarão métricas superiores de contribuição, releases e atualização devido ao maior engajamento da comunidade."

METODOLOGÍA

AMOSTRA:

- 1.000 repositórios mais estrelados do GitHub (>10.000 stars)
- Snapshot: agosto/2025

FONTE E COLETA:

- API GraphQL do GitHub com autenticação por token
- Paginação em lotes de 20 repositórios por requisição
- Tratamento de erros e delays entre chamadas
- Saída dupla: repos_1000.json (bruto) e repos_1000.csv (análise)

MÉTRICAS EXTRAÍDAS:

- Nome/owner, estrelas e linguagem primária
- Datas: criação e última atualização
- Pull requests aceitas (merged), releases
- Issues abertas e fechadas



METODOLOGÍA

CÁLCULOS:

- Idade do repositório
- Dias desde a última atualização
- Percentual de issues fechadas = $\text{fechadas} / \text{total} \times 100$

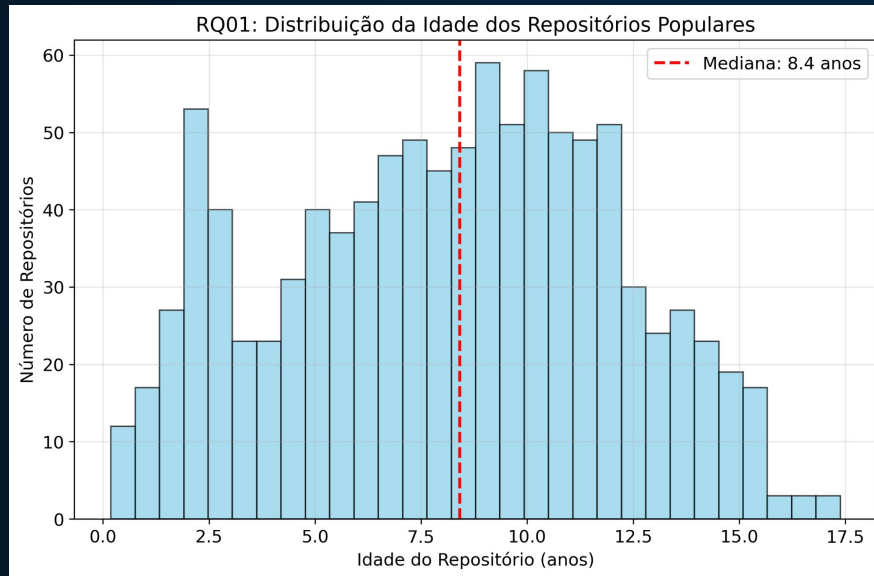
FERRAMENTAS:

- Python + Pandas para análise
- Matplotlib/Seaborn para gráficos
- Consultas GraphQL para a coleta



RESULTADOS

RQ01: Sistemas populares são maduros/antigos?



Mediana: 8.4 anos

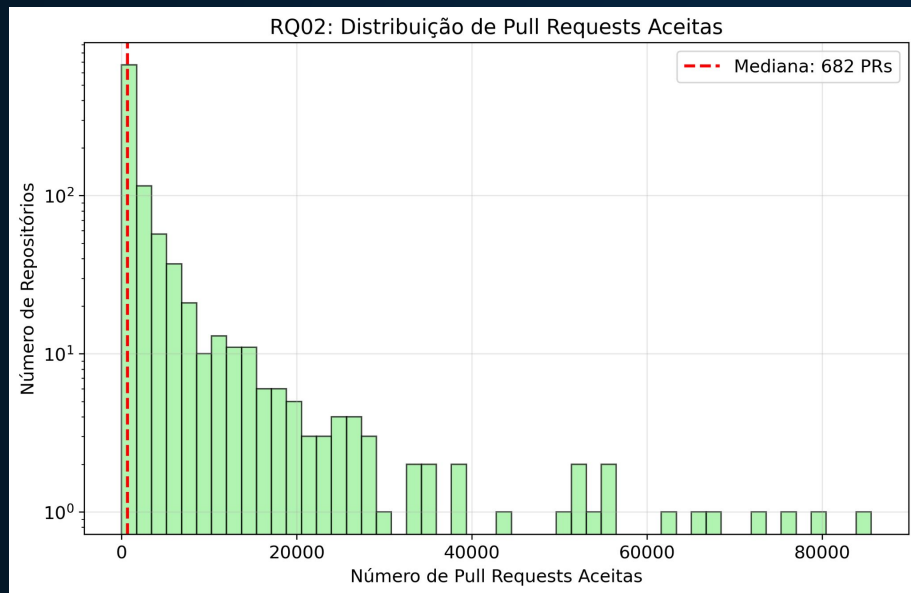
Média: 8.1 anos

Máximo: 0.2 anos

Mínimo: 17.4 anos

RESULTADOS

RQ02: Sistemas populares recebem muita contribuição externa?



Mediana: 682 PRs

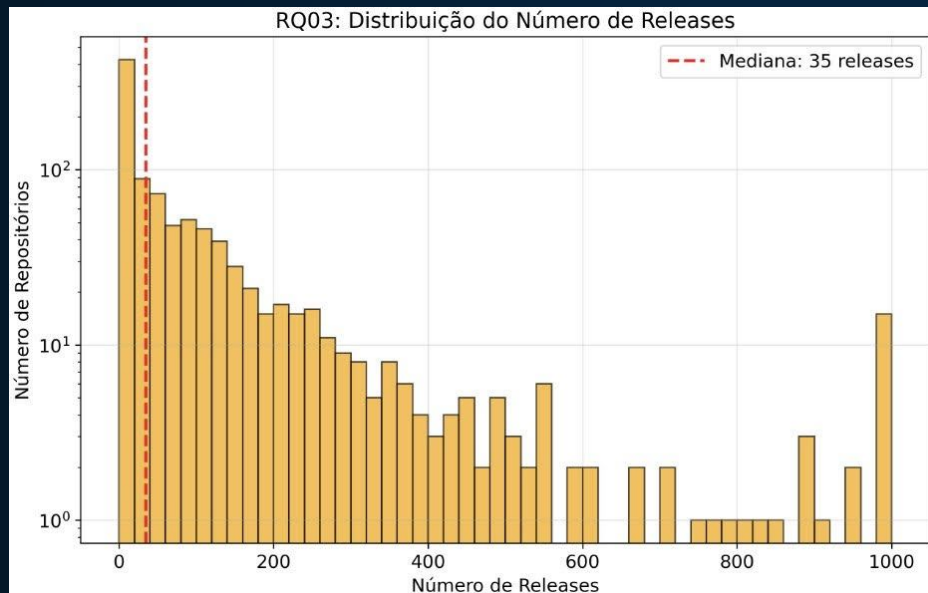
Média: 3.568 PRs

Máximo: 0 PRs

Mínimo: 85.587 PRs

RESULTADOS

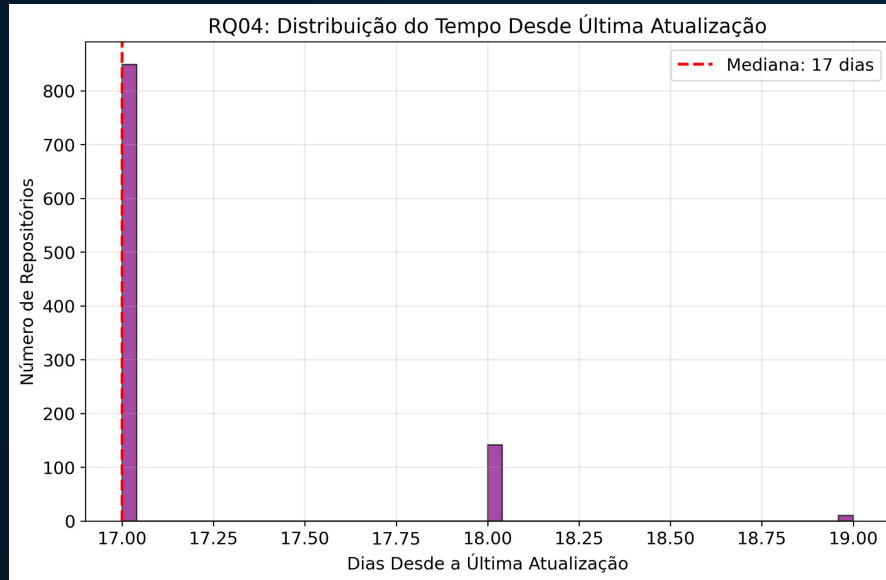
RQ03: Sistemas populares recebem muita contribuição externa?



Mediana: 35 releases
Média: 108 releases
Mínimo: 0 releases
Máximo: 1.000 releases

RESULTADOS

RQ04: Sistemas populares são atualizados com frequência?



Mediana: 17 dias

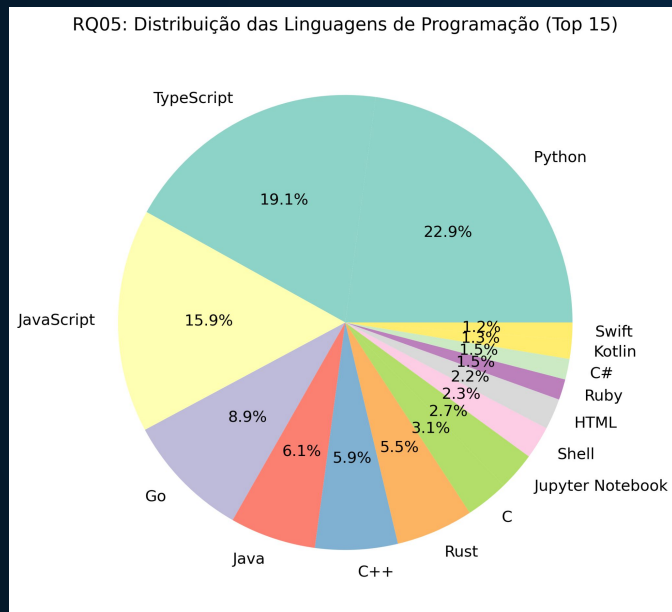
Média: 17 dias

Mínimo: 17 dias

Máximo: 19 dias

RESULTADOS

RQ05: Sistemas populares são escritos nas linguagens mais populares?



Python: 187 repositórios (18.7%)

TypeScript: 156 repositórios (15.6%)

JavaScript: 130 repositórios (13.0%)

Go: 73 repositórios (7.3%)

Java: 50 repositórios (5.0%)

C++: 48 repositórios (4.8%)

Rust: 45 repositórios (4.5%)

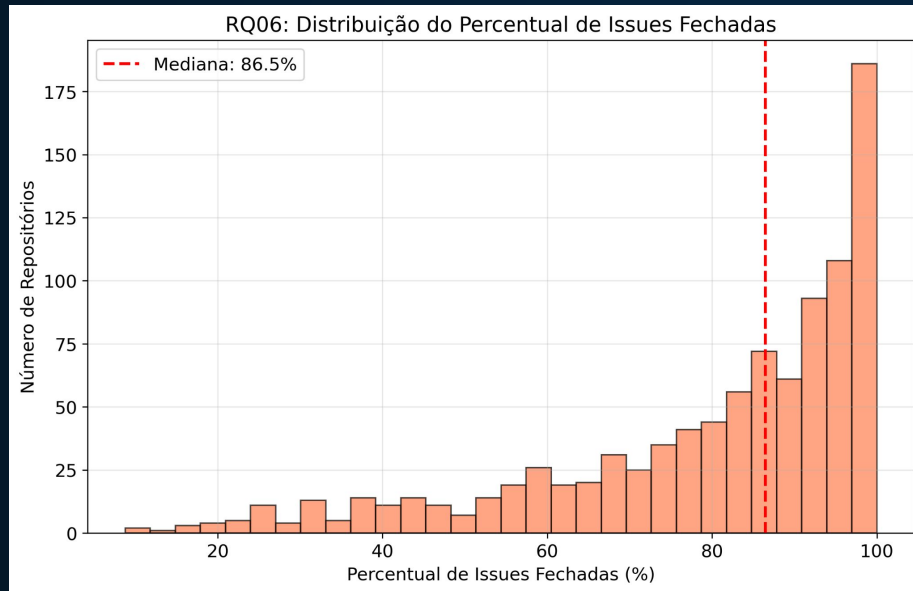
C: 25 repositórios (2.5%)

Jupyter Notebook: 22 repositórios (2.2%)

Shell: 19 repositórios (1.9%)

RESULTADOS

RQ06: Sistemas populares são maduros/antigos?



Mediana: 86.5%

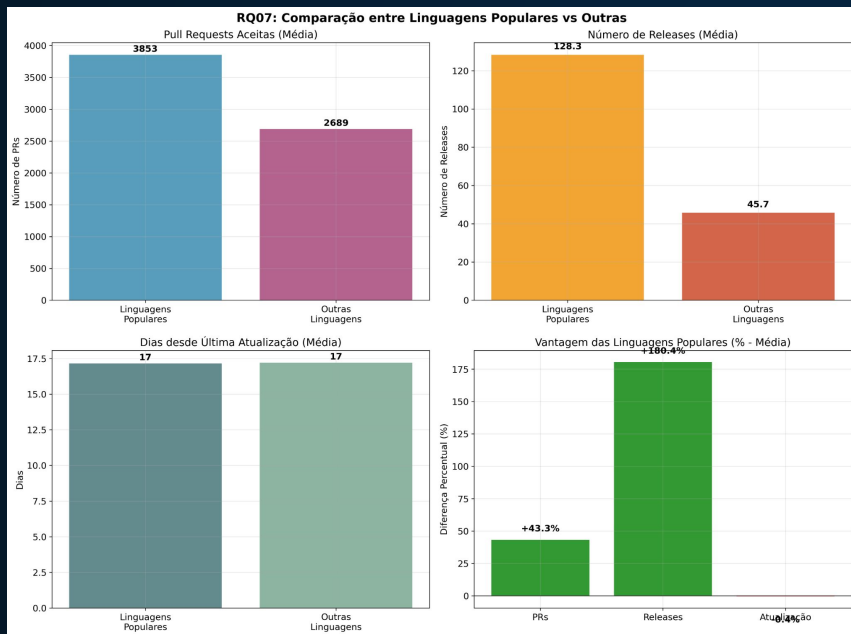
Média: 79.9%

Mínimo: 8.8%

Máximo: 100.0%

RESULTADOS

RQ07: Linguagens populares vs. outras linguagens



Pull Requests Aceitas:

Linguagens Populares (Média):

3.853 PRs

Outras Linguagens (Média): 2.689

PRs

Diferença: +43.3% (linguagens populares recebem mais PRs em média)

Número de Releases:

Linguagens Populares (Média):

128.3 releases

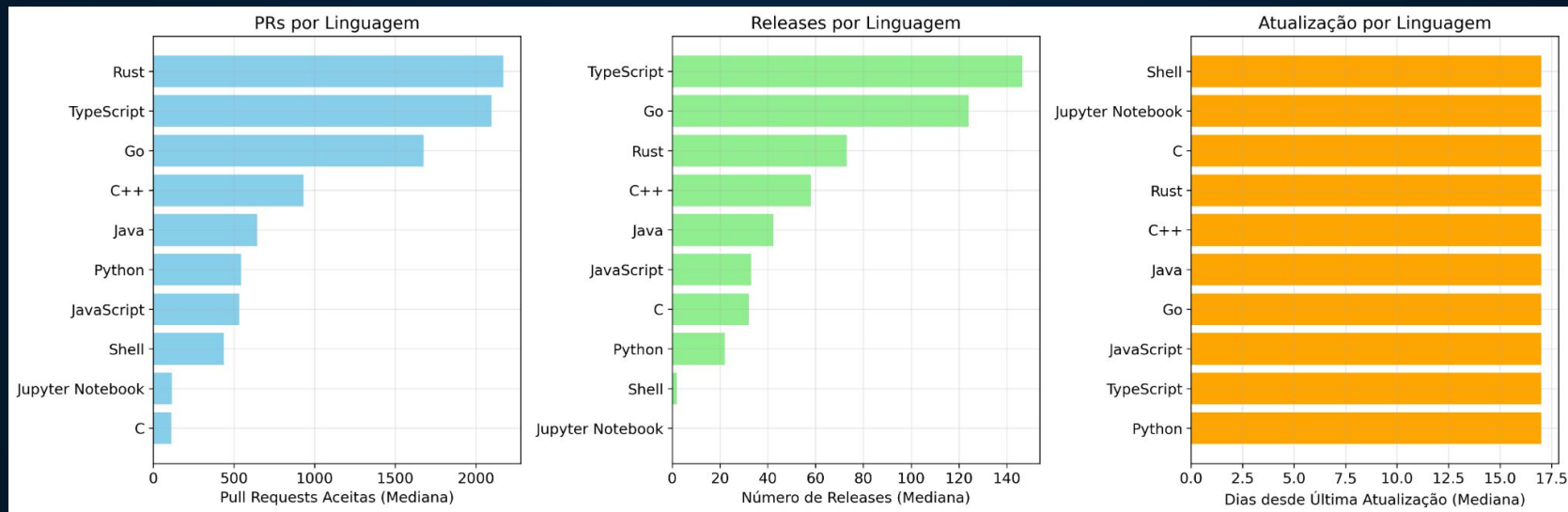
Outras Linguagens (Média): 45.7

releases

Diferença: +180.4% (linguagens populares lançam mais releases em média)

RESULTADOS

RQ07: Linguagens populares s. outras linguagens



RESULTADOS

Métricas por Linguagem

Rust e TypeScript lideram em PRs aceitas

TypeScript e Go lideram em número de releases

Tempo de atualização semelhante entre todas (mediana de 17 dias)

Interpretação: linguagens populares recebem mais contribuições externas e lançam mais releases, enquanto a frequência de atualização se mantém alta em todos os projetos.

DISCUSSÃO

5.1 Confirmação das hipóteses

- **Confirmadas** RQ01 maturidade alta 8,4 anos mediana; RQ02 muita contribuição externa 682 PRs mediana; RQ04 atualização frequente 17 dias mediana; RQ05 linguagens dominantes Python TypeScript JavaScript; RQ06 alta taxa de issues fechadas 86,5; RQ07 linguagens populares com métricas superiores
- **Parcialmente confirmada** RQ03 frequência de releases mediana 35 com alta variabilidade entre projetos

5.2 Descobertas inesperadas

- **TypeScript** superou JavaScript refletindo adoção de tipagem estática
- **Rust** em destaque com alta média de contribuições por repositório
- **Atualização uniforme** mediana ~17 dias em todas as linguagens
- **Alta variabilidade** diferenças grandes entre medianas e médias

DISCUSSÃO

5.3 Implicações

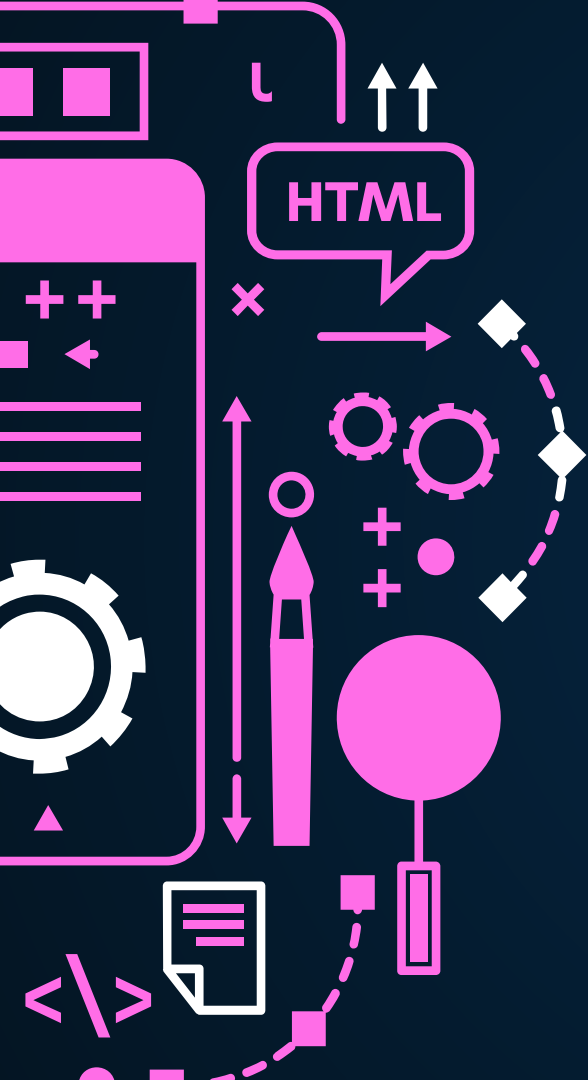
- **Para desenvolvedores** linguagens populares aumentam chance de engajamento e suporte da comunidade
- **Para empresas** apostar em stacks estabelecidas Python e TypeScript facilita contribuições e manutenção
- **Para pesquisa** maturidade do projeto é fator central na popularidade

5.4 Limitações

- **Viés de seleção** apenas repositórios com mais de 10.000 stars
Recorte temporal snapshot de agosto de 2025
- **Cobertura de métricas** não inclui outras dimensões de qualidade não observáveis pela API

CONCLUSÃO

Repositórios populares do GitHub se destacam por maturidade (8+ anos), manutenção ativa (atualizações a cada 17 dias), alta resolução de issues (86,5%) e forte engajamento da comunidade, com Python, TypeScript e JavaScript entre as linguagens líderes.



Obrigado!

Alguma Pergunta?

Bibliografía

- GITHUB. GitHub GraphQL API Documentation. Disponível em: <https://docs.github.com/en/graphql>. Acesso em: ago. 2025.
- GITHUB. About stars. Disponível em: <https://docs.github.com/en/repositories/managing-your-repositorys-settings-and-features/customizing-your-repository/about-stars>. Acesso em: ago. 2025.
- MCKINNEY, W. Python for Data Analysis. O'Reilly Media, 2017. (referência clássica do pandas)