

# Análise de Repositórios Populares no GitHub:

## Características e Padrões

### Grupo 01:

Nataniel Geraldo Mendes Peixoto  
Nelson de Campos Nolasco  
Rubia Coelho de Matos

---

### Introdução

Repositórios populares no GitHub são frequentemente utilizados como referência para entender padrões de desenvolvimento de software e colaboração. Neste trabalho, investigamos características dos repositórios mais populares, levantando as seguintes hipóteses informais:

1. Repositórios populares tendem a ser maduros, com muitos anos de desenvolvimento.
2. Eles recebem contribuições consistentes de desenvolvedores externos.
3. Repositórios populares lançam atualizações e releases com alta frequência.
4. São escritos principalmente nas linguagens de programação mais amplamente utilizadas.
5. Possuem uma alta taxa de issues resolvidas, indicando boa manutenção.

Nosso objetivo é analisar um conjunto de 1000 repositórios e validar essas hipóteses com base em métricas extraídas da API GraphQL do GitHub.

---

### Metodologia

1. **Coleta de Dados:** Utilizamos a API GraphQL do GitHub para coletar dados detalhados sobre 1000 repositórios populares, incluindo informações sobre estrelas, contribuições (pull requests), releases, issues e linguagens de programação.
  2. **Análise de Métricas:**
    - Calculamos a idade dos repositórios e o tempo desde a última atualização.
    - Analisamos o número médio e mediano de pull requests aceitas.
    - Verificamos a distribuição das linguagens de programação mais utilizadas.
    - Calculamos o percentual de issues fechadas.
    - Agrupamos os repositórios por linguagem para identificar padrões específicos.
  3. **Visualizações:** Geramos gráficos para representar visualmente algumas métricas, como a distribuição de pull requests aceitas e as linguagens mais utilizadas.
-

## Resultados

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

- **Idade média dos repositórios:** 7.98 anos
- **Mediana da idade:** 8.29 anos

Os repositórios populares tendem a ser antigos, com idade média e mediana próximas de 8 anos, o que corrobora a hipótese de que sistemas populares possuem maturidade.

---

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

- **Média de PRs aceitas:** 3219.35
- **Mediana de PRs aceitas:** 613.50

A média elevada é influenciada por alguns repositórios com um número muito alto de pull requests aceitas. A mediana, significativamente menor, sugere que a maioria dos repositórios tem contribuições externas mais moderadas.

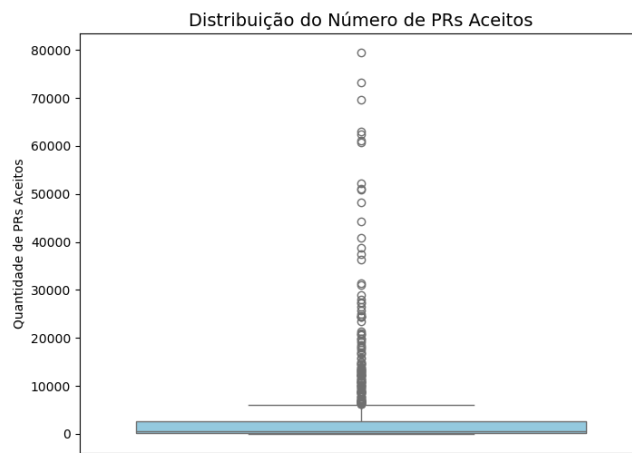


Gráfico salvo como 'qtd\_PRs\_aceitos.png'

---

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

- **Média de releases:** 101.65
- **Mediana de releases:** 32.50

Embora a média de releases seja alta, a mediana indica que muitos repositórios possuem um número mais modesto de releases, o que mostra diversidade na frequência de lançamentos.

---

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

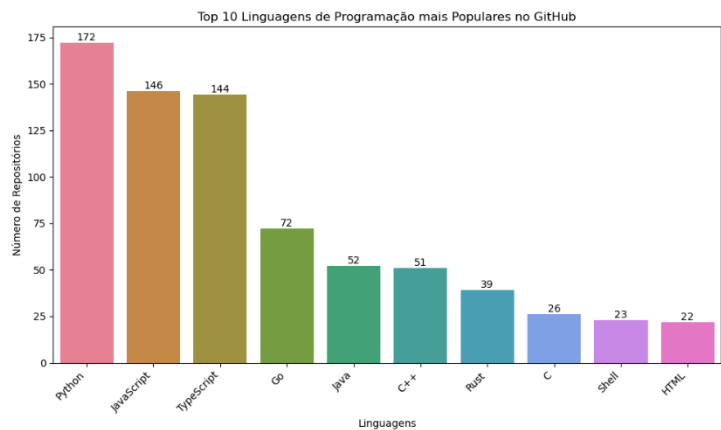
- **Média de dias desde última atualização:** 0.15 (cerca de 3h e 36 min)
- **Mediana de dias desde última atualização:** 0.07 (cerca de 1h e 41 min)

Os repositórios populares são extremamente ativos, com atualizações frequentes (praticamente diárias), o que reflete alta manutenção.

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

**Top 10 linguagens mais usadas nos repositórios populares:**

1. Python (172 repositórios)
2. JavaScript (146 repositórios)
3. TypeScript (144 repositórios)
4. Go (72 repositórios)
5. Java (52 repositórios)
6. C++ (51 repositórios)
7. Rust (39 repositórios)
8. C (26 repositórios)
9. Shell (23 repositórios)
10. HTML (22 repositórios)



As linguagens mais amplamente utilizadas, como Python e JavaScript, dominam a lista, confirmando a hipótese.

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

- **Média do percentual de issues fechadas:** 76.31%
- **Mediana do percentual de issues fechadas:** 85.78%

A maioria dos repositórios apresenta uma boa manutenção, com uma alta taxa de fechamento de issues.

#### RQ 07. Análise por linguagem das principais métricas:

Linguagem	Média de PRs Aceitas	Média de Releases	Média de Dias Desde Última Atualização
C	1152.50	58.73	0.10
C++	6348.00	134.65	0.13
Go	5518.69	148.64	0.10
HTML	821.77	15.45	0.24
Java	2831.96	62.31	0.27
JavaScript	1699.48	99.36	0.21
Python	3320.84	67.22	0.10
Rust	5388.85	153.44	0.08
Shell	743.22	46.91	0.14
TypeScript	4512.38	240.68	0.10

Linguagens como **TypeScript**, **Rust** e **Go** apresentam as maiores médias de releases e pull requests aceitas, enquanto linguagens como **HTML** e **Shell** têm menores valores, indicando diferentes perfis de uso.

---

## Discussão

### Hipóteses iniciais e validação:

1. **Hipótese: Repositórios populares são antigos.**
  - Validada. A idade média (7.98 anos) indica maturidade.
2. **Hipótese: Eles recebem muita contribuição externa.**
  - Parcialmente validada. A média de PRs aceitas (3219.35) é alta, mas a mediana (613.50) sugere que apenas alguns repositórios recebem contribuições externas em larga escala.
3. **Hipótese: Sistemas populares lançam releases com frequência.**
  - Parcialmente validada. Embora a média (101.65 releases) seja alta, a mediana (32.50 releases) mostra que muitos repositórios possuem uma frequência de lançamentos mais moderada.
4. **Hipótese: São escritos nas linguagens mais populares.**
  - Validada. Python, JavaScript e TypeScript dominam a lista.
5. **Hipótese: Possuem uma alta taxa de fechamento de issues.**
  - Validada. A média do percentual de issues fechadas (76.31%) é alta, com uma mediana ainda maior (85.78%).

### Interpretação dos resultados:

- A alta frequência de atualizações e a maturidade dos sistemas reforçam a ideia de que repositórios populares são bem mantidos.
- O domínio de linguagens como Python e JavaScript reflete tendências de mercado e comunidade.
- A discrepância entre médias e medianas em várias métricas destaca a diversidade nos tipos de repositórios populares (alguns extremamente ativos e outros com menor atividade).

---

## Conclusão

Os resultados suportam a maioria das hipóteses levantadas, embora haja variações significativas entre os repositórios. Essa análise oferece uma visão abrangente sobre as

características dos sistemas populares no GitHub, servindo como base para estudos futuros sobre padrões de desenvolvimento e colaboração em software open source.