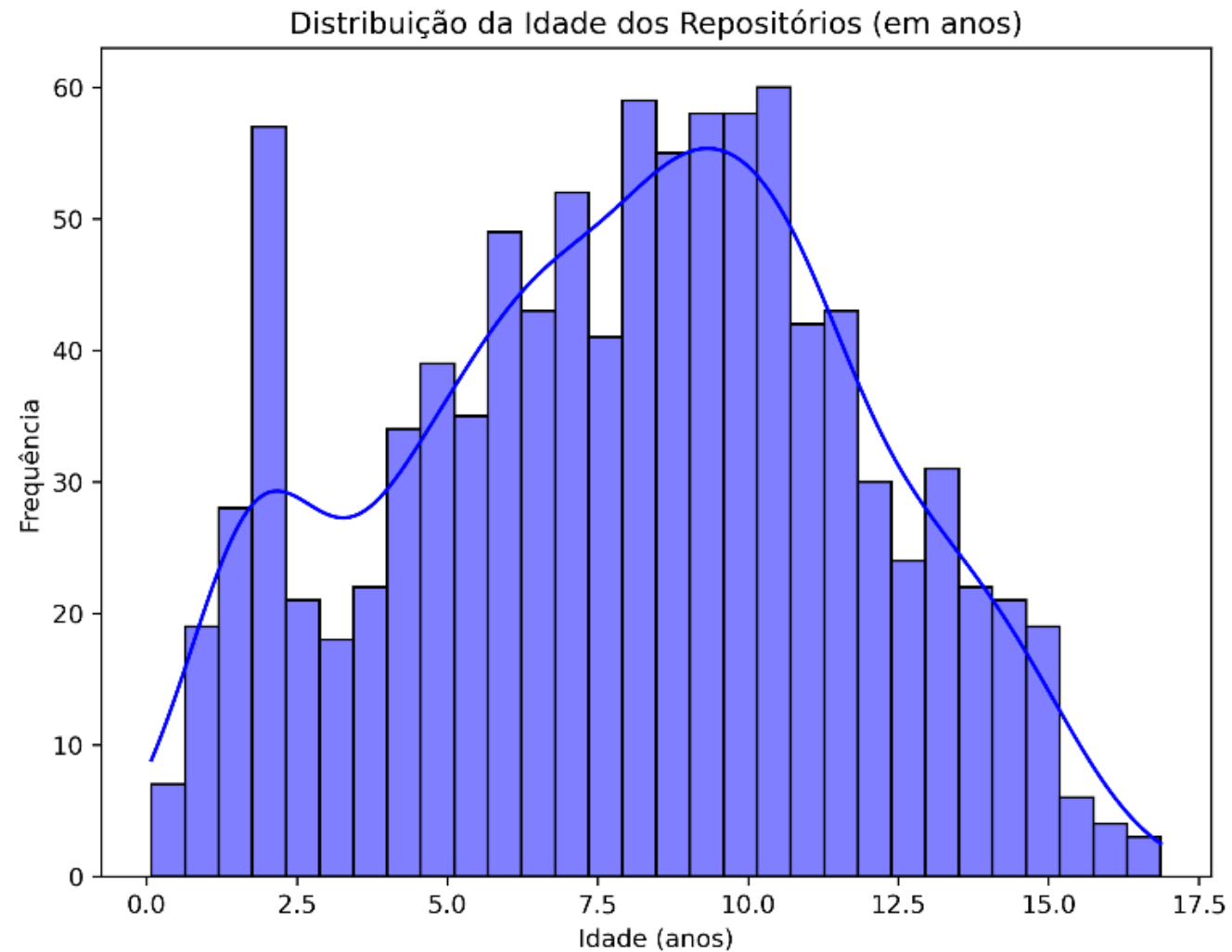
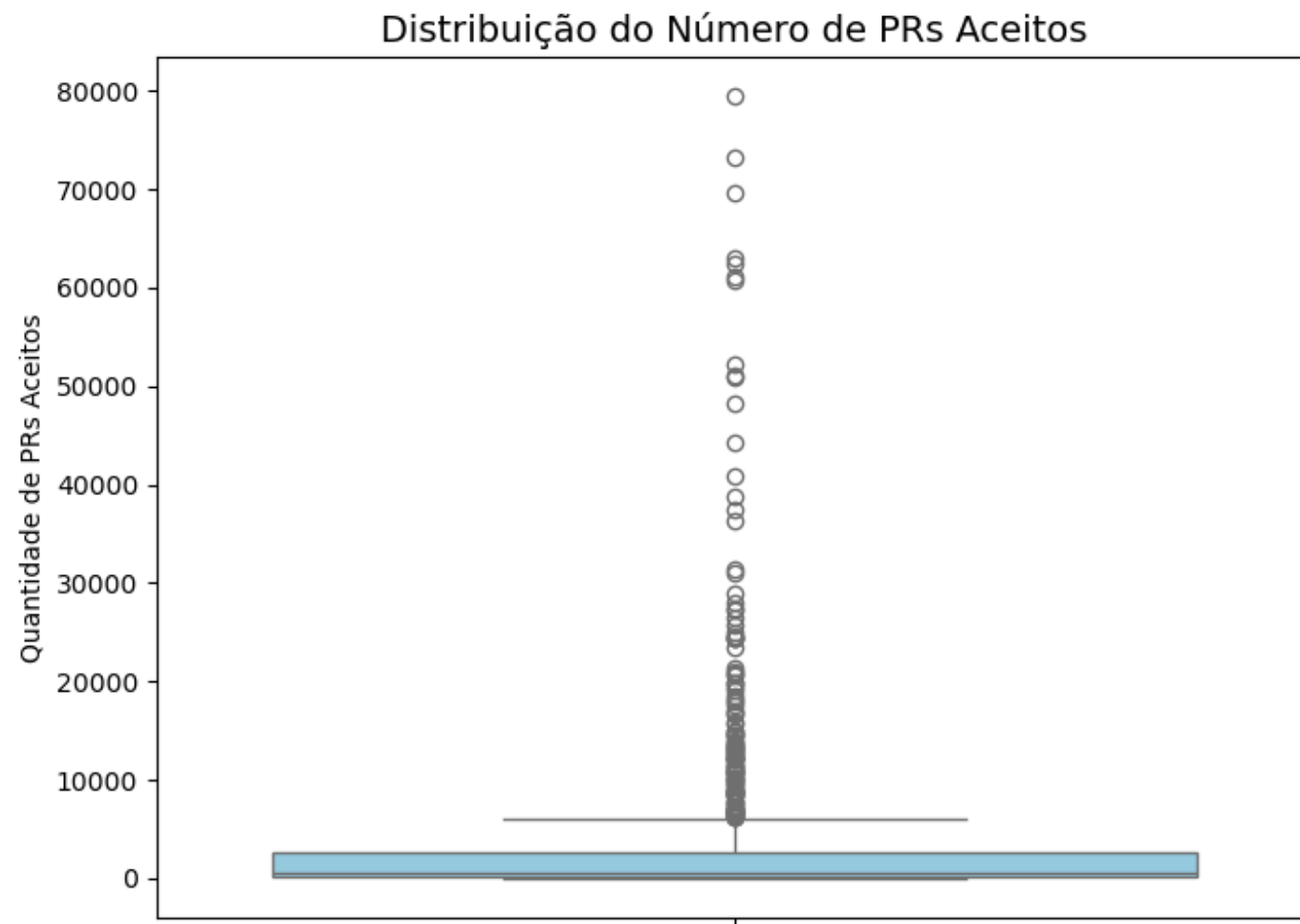


Análise de Repositórios Populares no GitHub: Características e Padrões

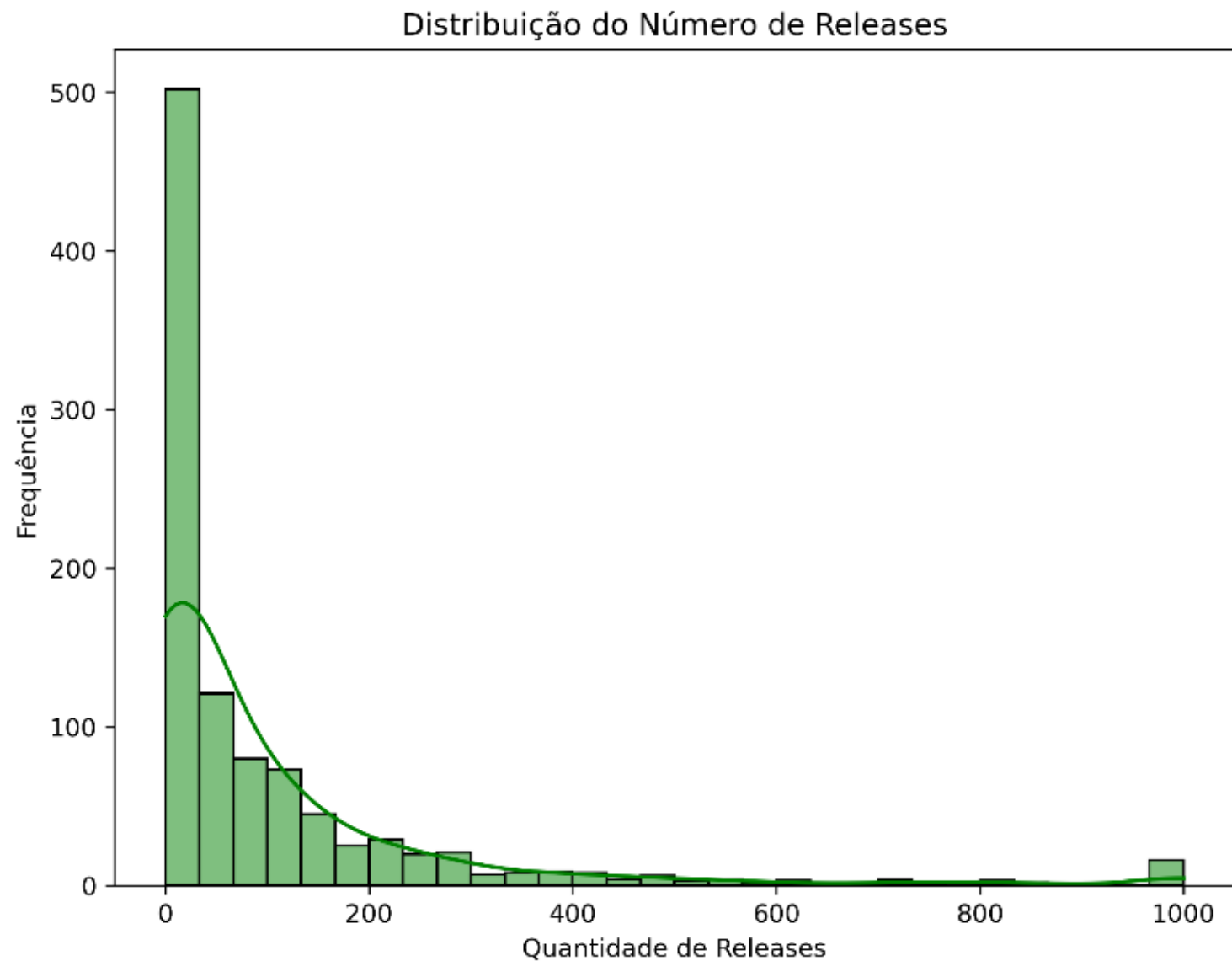
- Laboratório de medição e experimentação
- Laboratório 01 – Características de repositórios populares
- **Grupo 01:**
 - Nataniel Geraldo Mendes Peixoto
 - Nelson de Campos Nolasco
 - Rubia Coelho de Matos



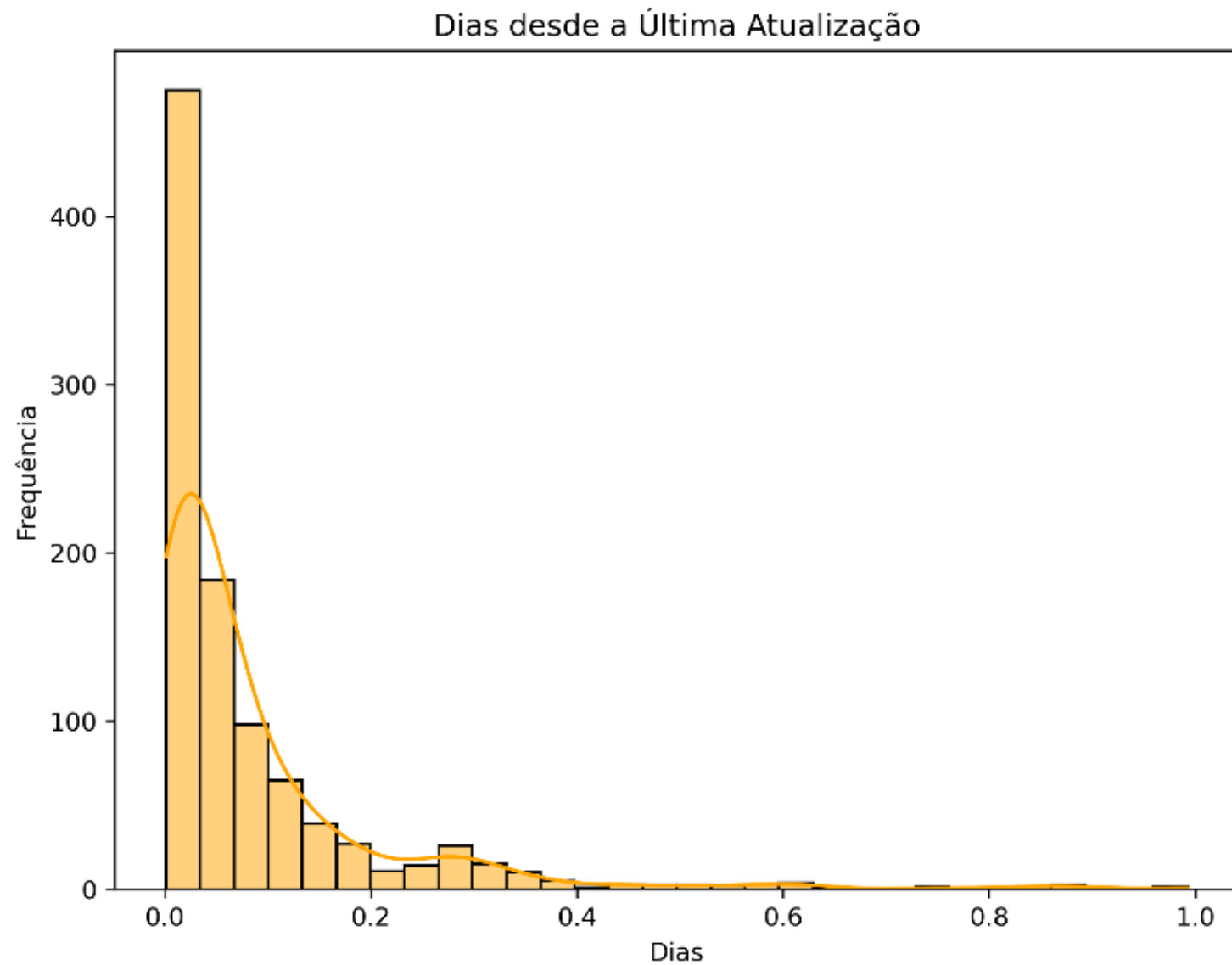
Repositórios populares tendem a ser antigos, com idade média e mediana próximas de 8 anos, o que corrobora a hipótese de que sistemas populares possuem maturidade.



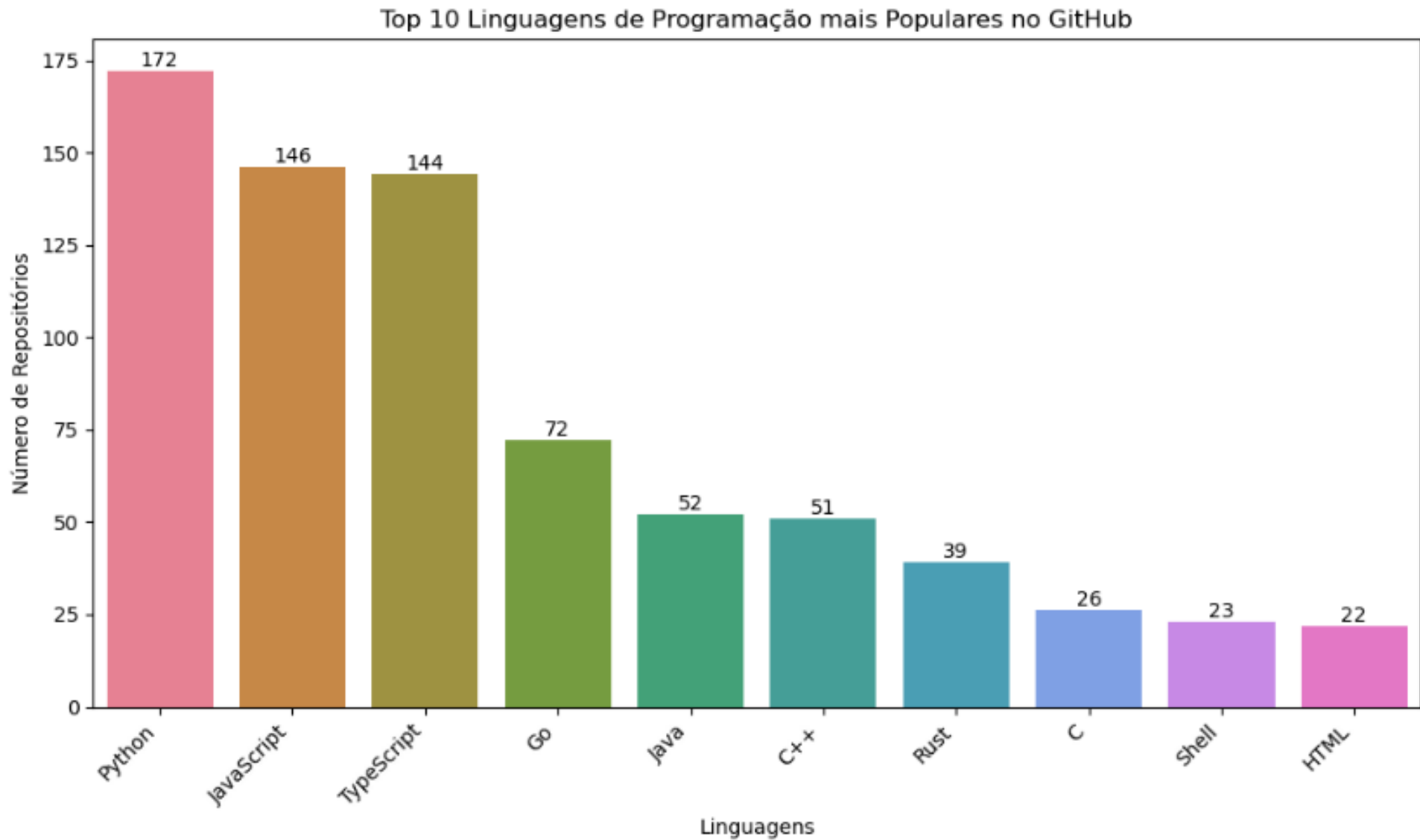
A média elevada é influenciada por alguns repositórios com um número muito alto de pull requests aceitas. A mediana, significativamente menor, sugere que a maioria dos repositórios tem contribuições externas mais moderadas.



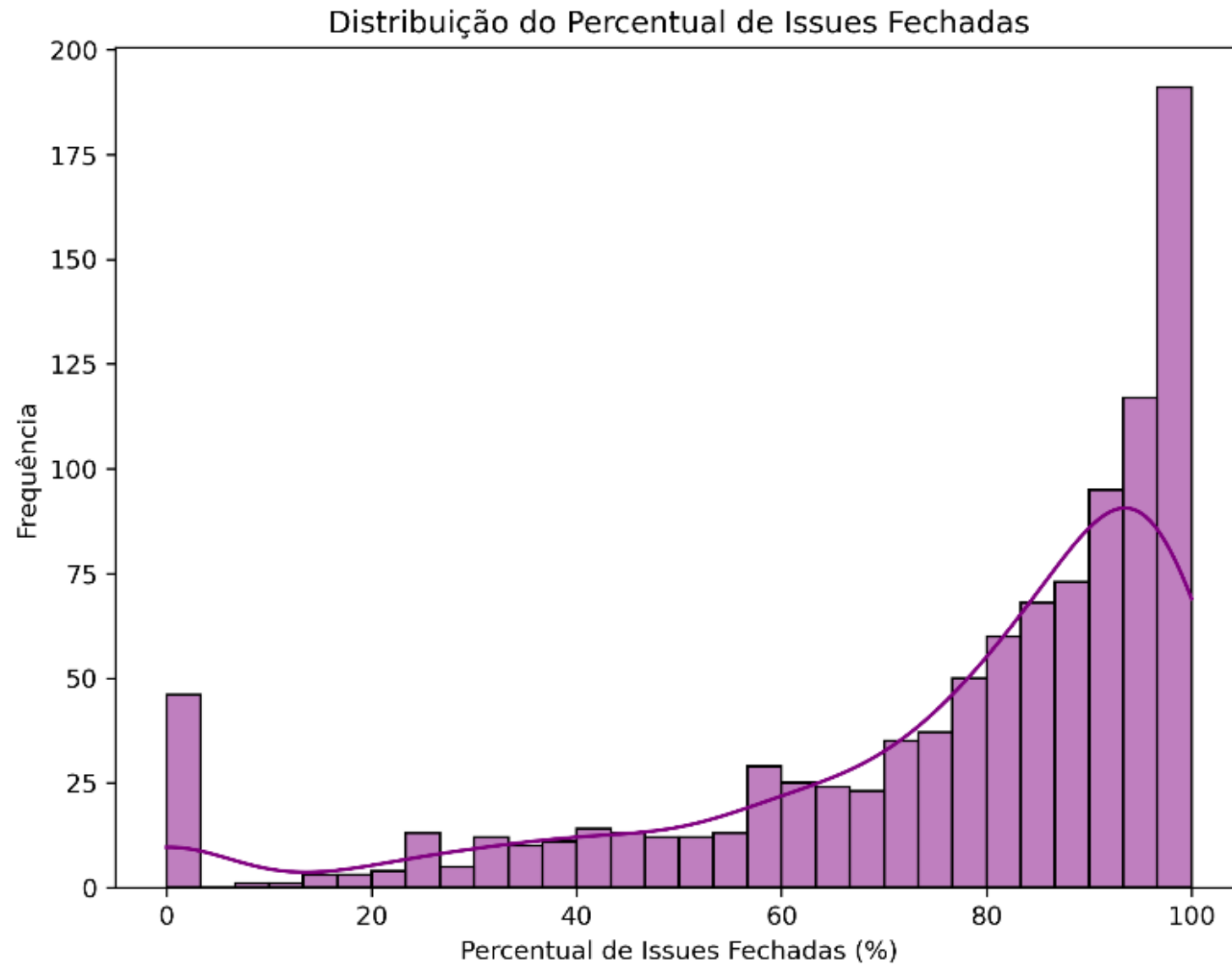
Embora a média de releases seja alta, a mediana indica que muitos repositórios possuem um número mais modesto de releases, o que mostra diversidade na frequência de lançamentos.



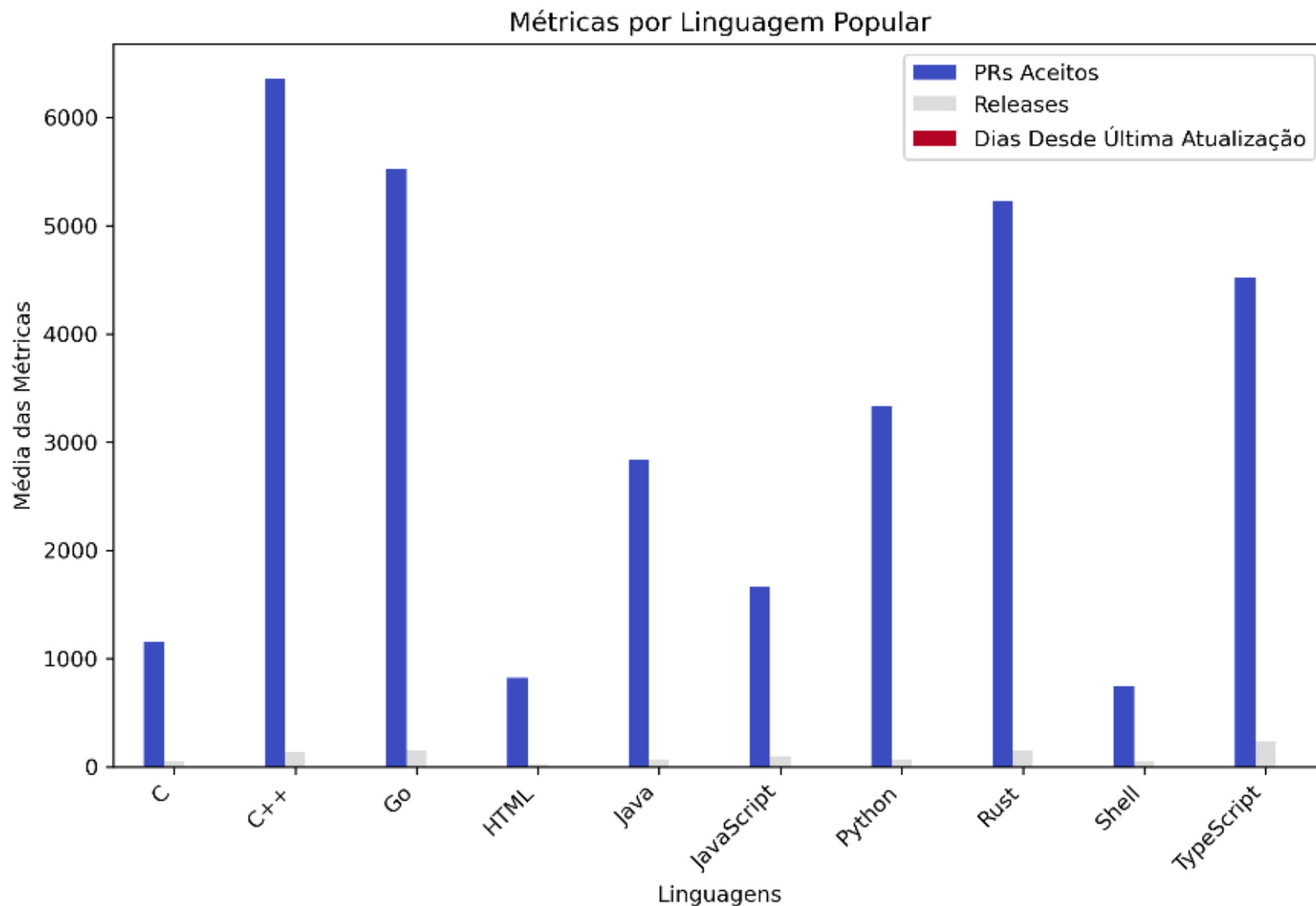
Os repositórios populares são extremamente ativos, com atualizações frequentes (praticamente diárias), o que reflete alta manutenção.



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

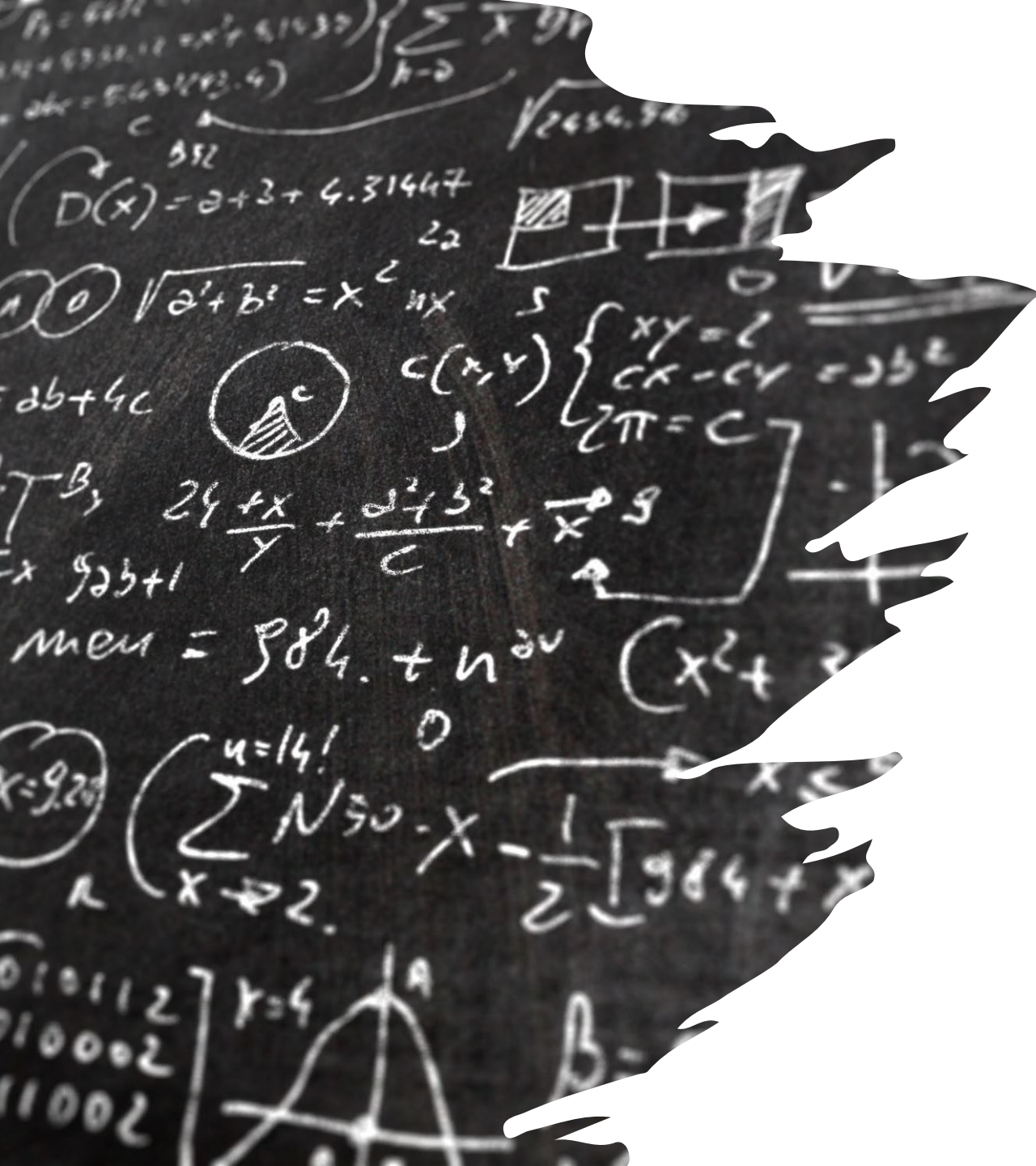


A maioria dos repositórios apresenta uma boa manutenção, com uma alta taxa de fechamento de issues.



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.



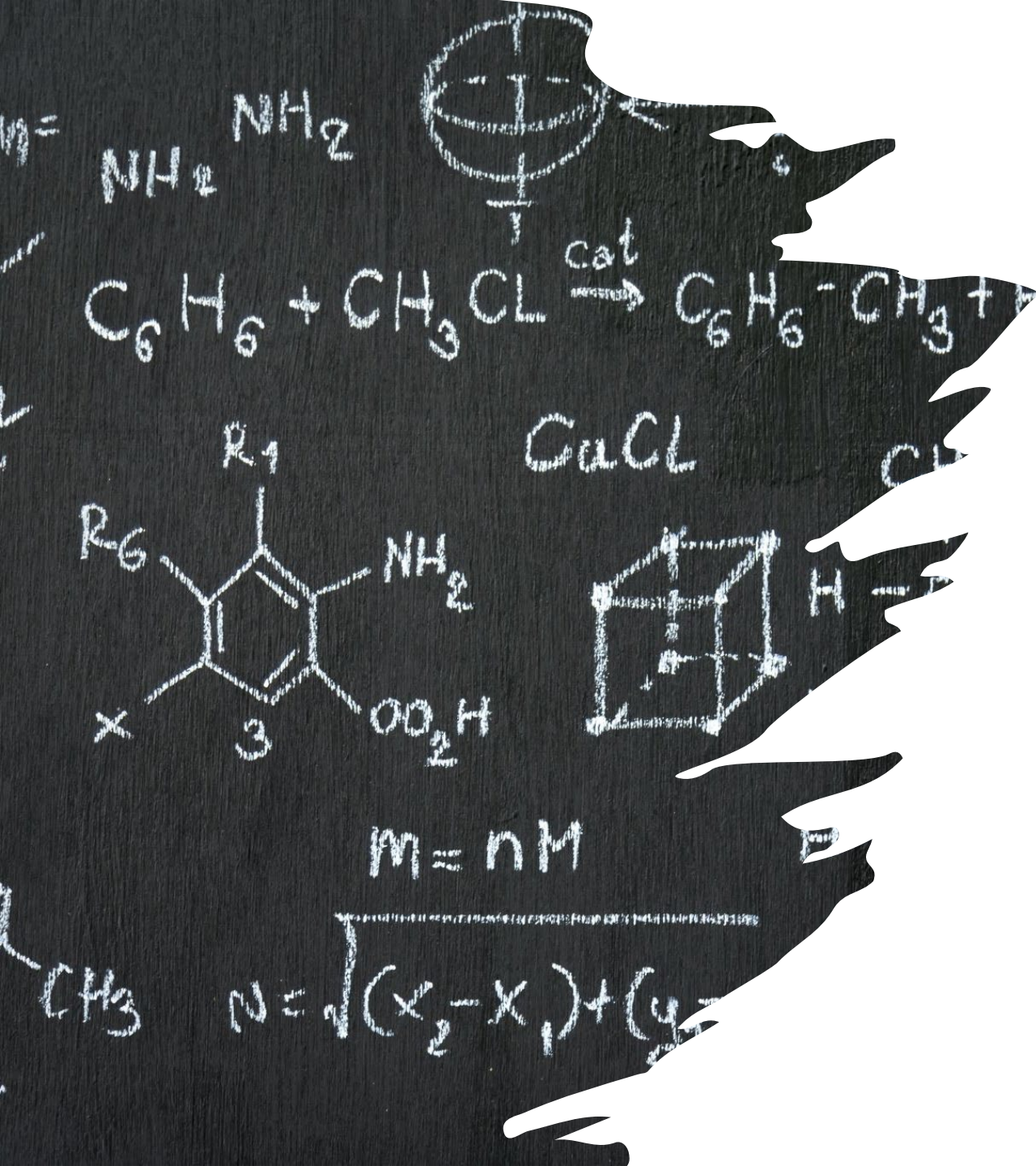


Interpretação dos resultados



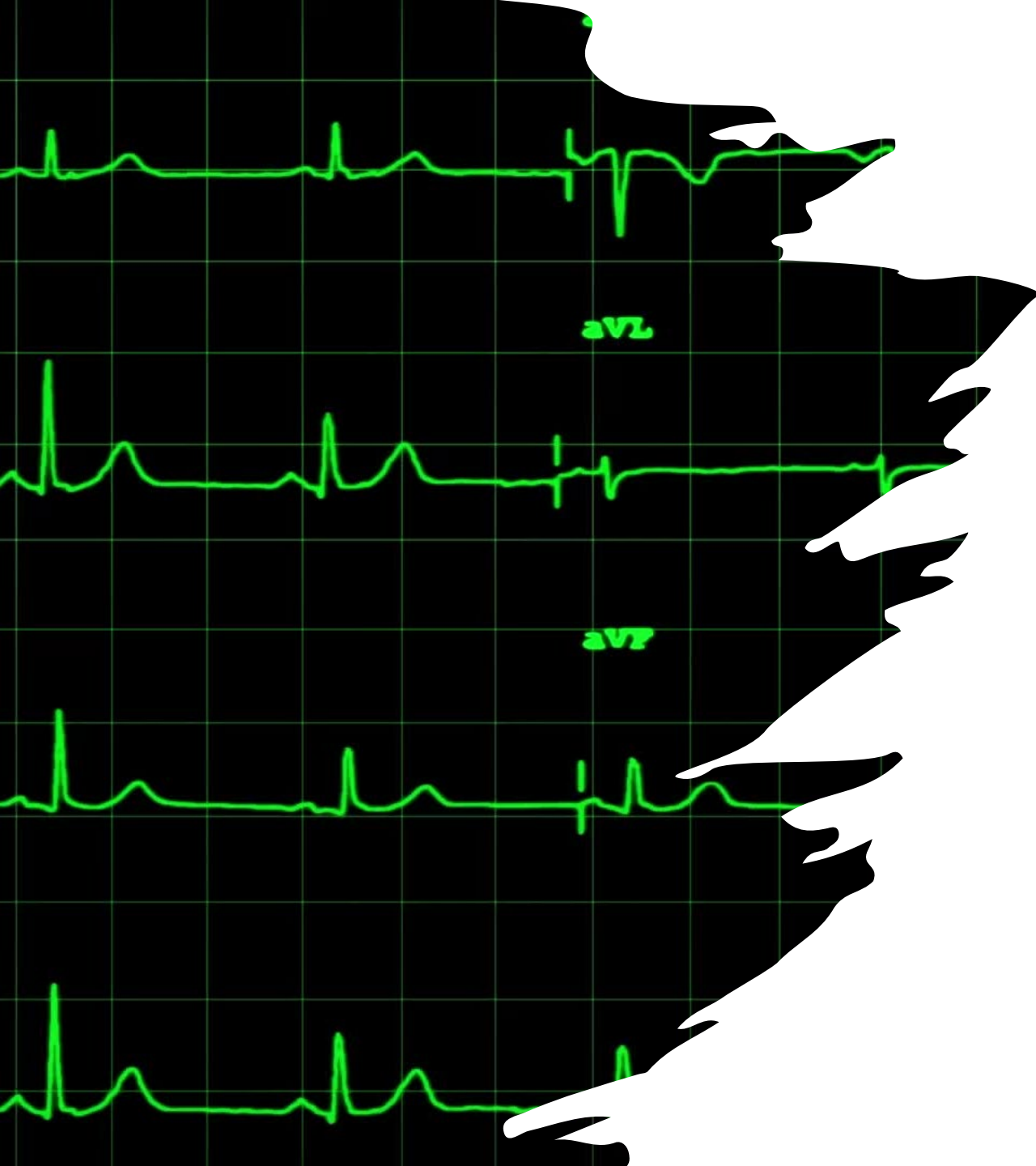
Hipótese: Repositórios populares são antigos.

- *Validada.* a idade média (7.98 anos) indica maturidade.



Hipótese: Eles recebem muita contribuição externa.

- *Parcialmente validada.* A média de PRs aceites (3219.35) é alta, mas a mediana (613.50) sugere que apenas alguns repositórios recebem contribuições externas em larga escala.



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.



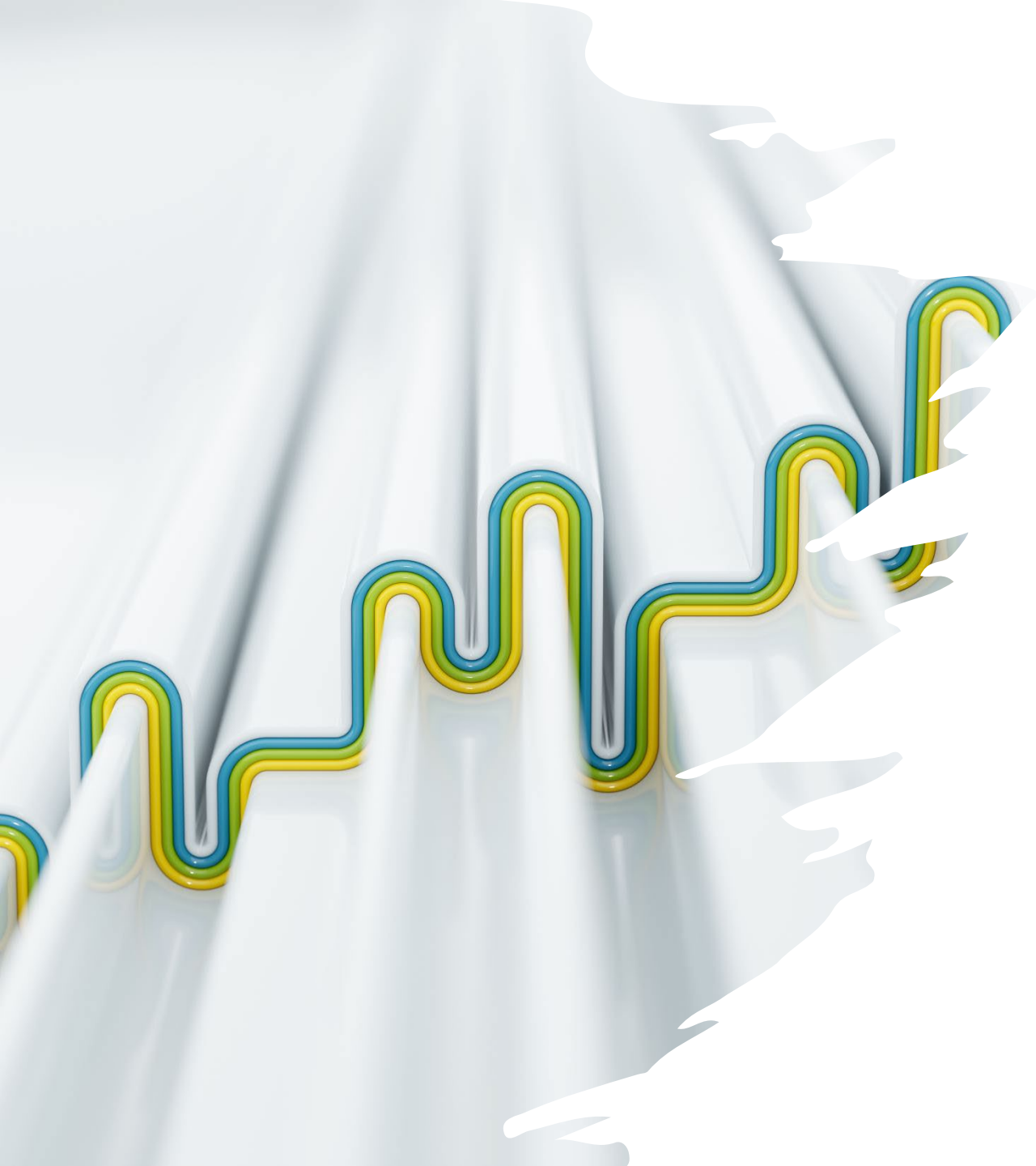
Hipótese: São escritos nas linguagens mais populares.

- *Validada*. Python, JavaScript e TypeScript dominam a lista.

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%).






- 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).





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.



Referências:

Linguagem Python e linguagem de consulta e manipulação GraphQL.



- **Referência Bibliográfica**

- KALINOWSKI, Marcos. *et all*. Engenharia de software para ciência de dados: um guia de boas práticas
- em ênfase na construção de sistemas de machine learning. 2023.
- MITCHELL, Ryan. Web Scraping com Python. 2015.
- HANASHIRO, Akira. GraphQL: a revolucionária linguagem de consulta e manipulação de dados para
- APIs. 2023
- DALE, Kyran. Visualização de dados com Python e JavaScript: raspe, limpe, explore e transforme seus
- dados. 2024.
- KNAFLIC, Cole Nussbaumer. Storytelling com dados: um guia sobre visualização de dados para
- Profissionais de negócios.





Dependências:



- pandas: para manipulação e análise de dados em Python.



- requests: faz requisições HTTP.



- matplotlib: para criação de gráficos e visualizações.



- seaborn: visualização estatística baseada no Matplotlib, oferece uma interface amigável e gráficos atraentes.



- python-dotenv: permite carregar variáveis de ambiente a partir de um arquivo .env.



- os: fornece funções para interagir com o sistema operacional.



- time: fornece várias funções para trabalhar com tempo.

Contact us!





Thank
😊 you!