

Relatório de Análise de Repositórios Populares do GitHub

1. Introdução

Este relatório tem como objetivo realizar uma análise detalhada de repositórios populares no GitHub, utilizando dados extraídos por meio da API GraphQL. Para tanto, foram aplicadas métricas específicas com o intuito de avaliar aspectos essenciais como a maturidade, manutenção e relevância desses projetos. A pesquisa leva em consideração fatores como o tempo de existência dos repositórios, o volume de contribuições, a frequência de atualizações e o engajamento da comunidade. A análise abrange dois conjuntos de dados, um composto por 100 repositórios e outro com 1000, com a finalidade de identificar características comuns e tendências entre os projetos. Entre os aspectos investigados, destacam-se a idade dos repositórios, a quantidade de contribuições, o número de releases, a frequência de atualizações, a popularidade das linguagens utilizadas e o volume de issues fechadas.

2. Metodologia

Os dados foram obtidos a partir de consultas GraphQL ao GitHub, processados com JavaScript e analisados estatisticamente. As métricas aplicadas foram:

- **Percentual de repositórios com mais de 5 anos**
- **Percentual de repositórios com mais de 70% Pull Requests aceitos**
- **Percentual de repositórios com mais de 10 releases**
- **Tempo médio de atualização dos repositórios**
- **Percentual de repositórios que utilizam linguagens populares**
- **Percentual de repositórios com mais de 30 issues fechadas**

Os repositórios analisados pertencem a categorias diversas e foram filtrados considerando sua popularidade (número de estrelas > 10000) .

3. Resultados e Análise

Experimento com 100 Repositórios

Os resultados obtidos para o primeiro experimento com 100 repositórios populares foram:

- 87,00% dos repositórios são antigos (mais de 5 anos);
- 36,00% tem mais de 70% de PR's aceitos;
- 53,00% possuem 10 ou mais releases;
- 100,00% tiveram atualização em menos de um ano;

- 65,00% utilizam uma linguagem popular;
- 87,00% possuem mais de 30 issues fechadas.

Os dados indicam que a maioria dos repositórios populares possuem uma longa história, um alto número de contribuições e um bom gerenciamento de issues. No entanto, nem todos possuem uma alta quantidade de releases, o que pode estar relacionado à natureza do projeto.

Experimento com 1000 Repositórios Os resultados obtidos para o segundo experimento com 1000 repositórios populares foram:

- 76,30% dos repositórios são antigos (mais de 5 anos);
- 50,00% tem mais de 70% de pull requests (PR's) aceitos;
- 60,20% possuem 10 ou mais releases;
- 100,00% tiveram atualização em menos de um ano;
- 65,40% utilizam uma linguagem popular;
- 92,40% possuem mais de 30 issues fechadas.

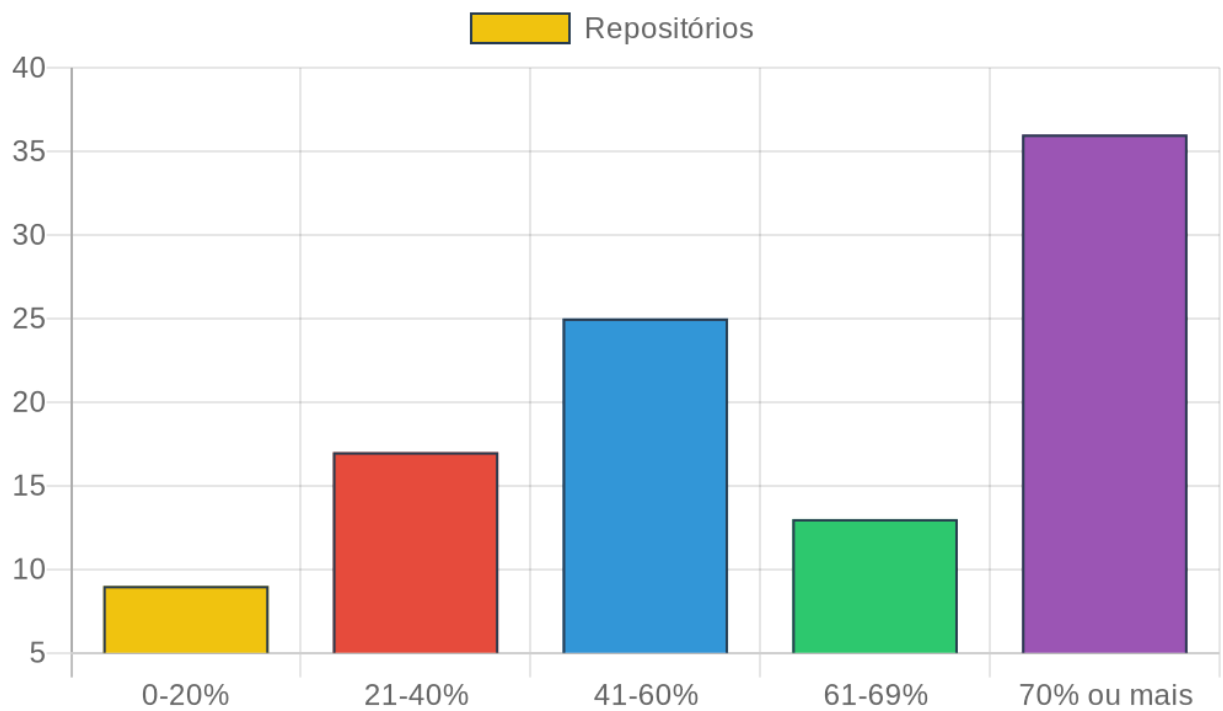
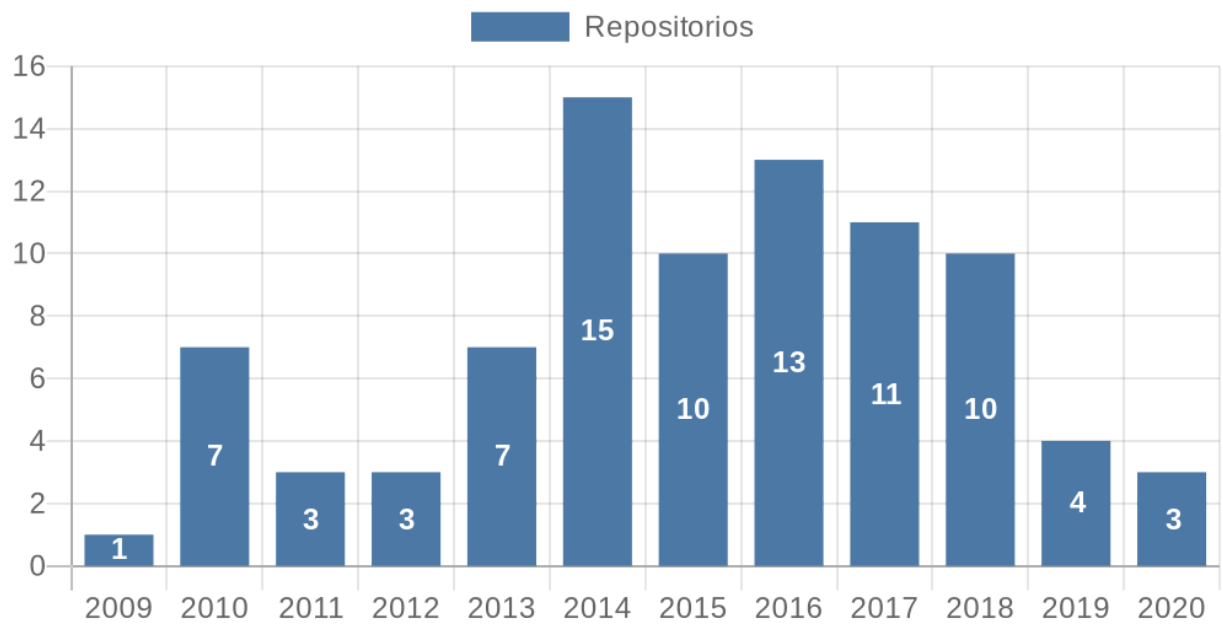
O volume maior de repositórios analisados, a proporção de repositórios antigos diminuiu.

Métrica	100 Repositórios	1000 Repositórios	Diferença (%)
% de repositórios antigos (> 5 anos)	87,00%	76,30%	-10,70%
% com mais de 70% de PR aceitos	36,00%	50,02%	+20,02%
% com 10 ou mais releases	53,00%	60,20%	+7,20%
% atualizados no último ano	100,00%	100,00%	0,00%
% que utilizam uma linguagem popular	65,00%	65,40%	+0,40%
% com mais de 30 issues fechadas	87,00%	92,40%	+5,40%

Visualização dos dados

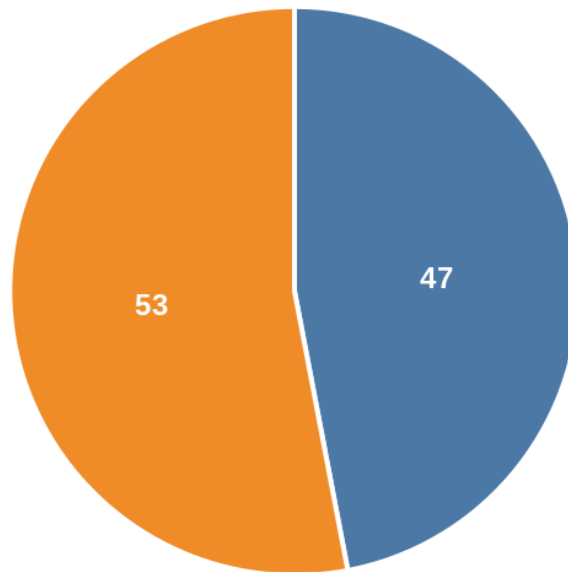
Para 100 repositórios:

Repositórios com mais de 5 anos

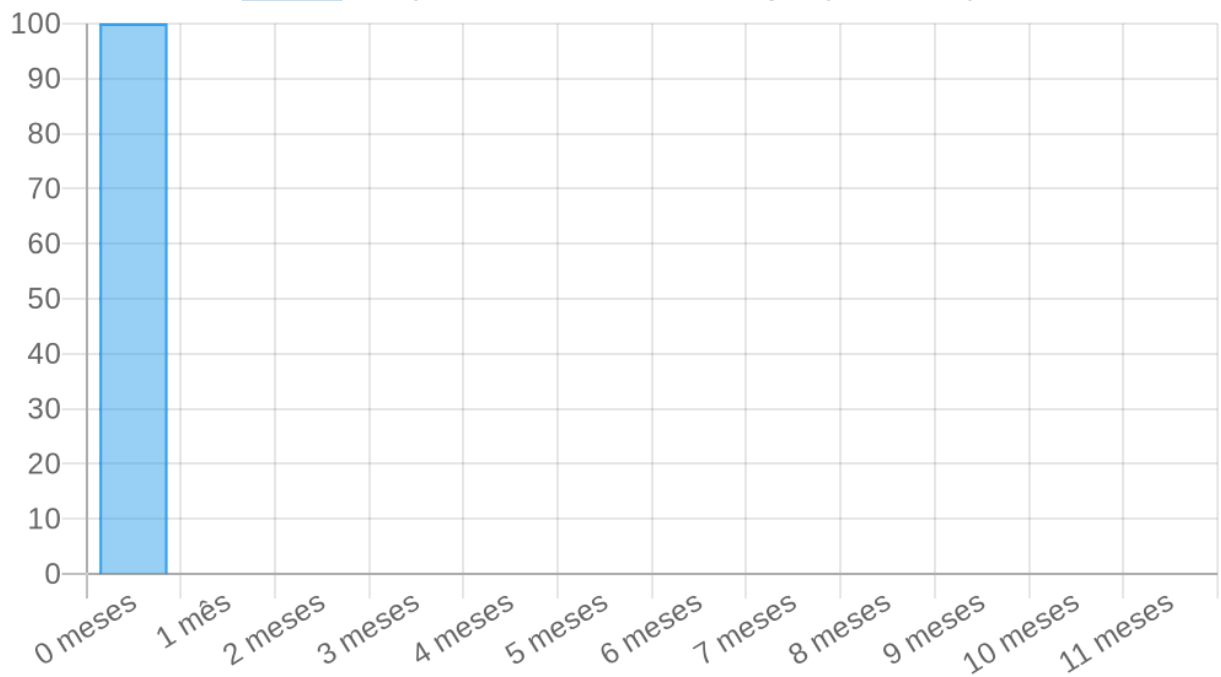


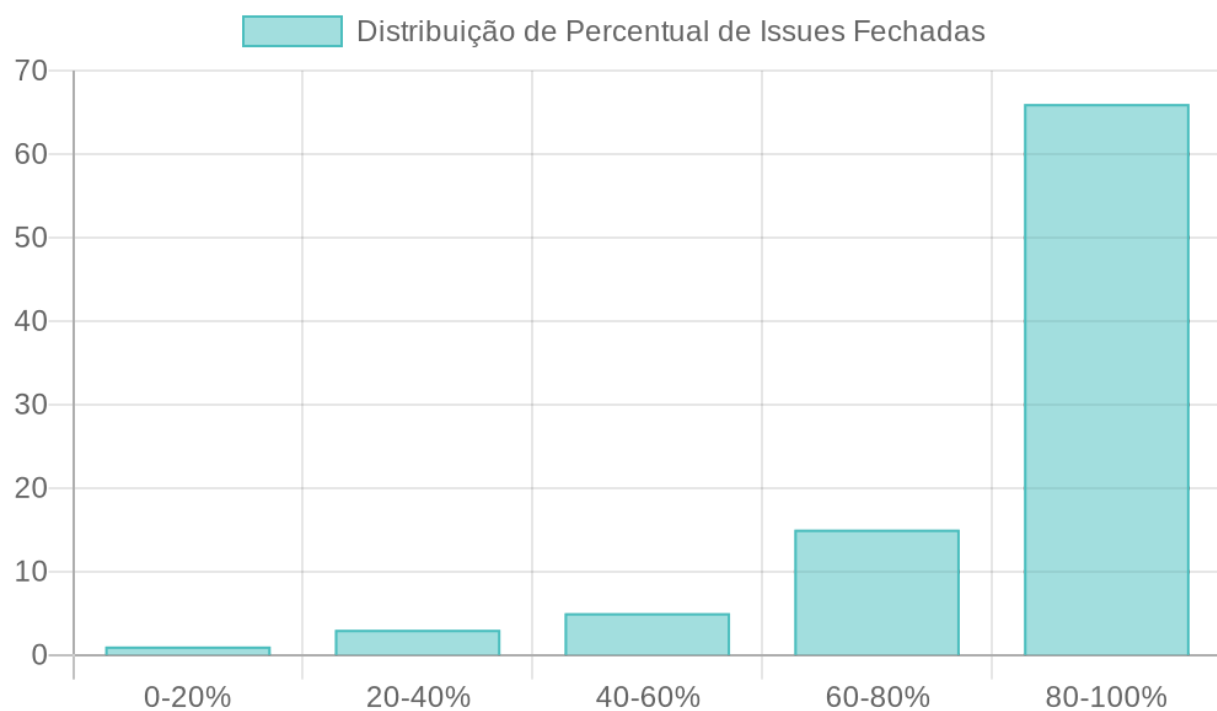
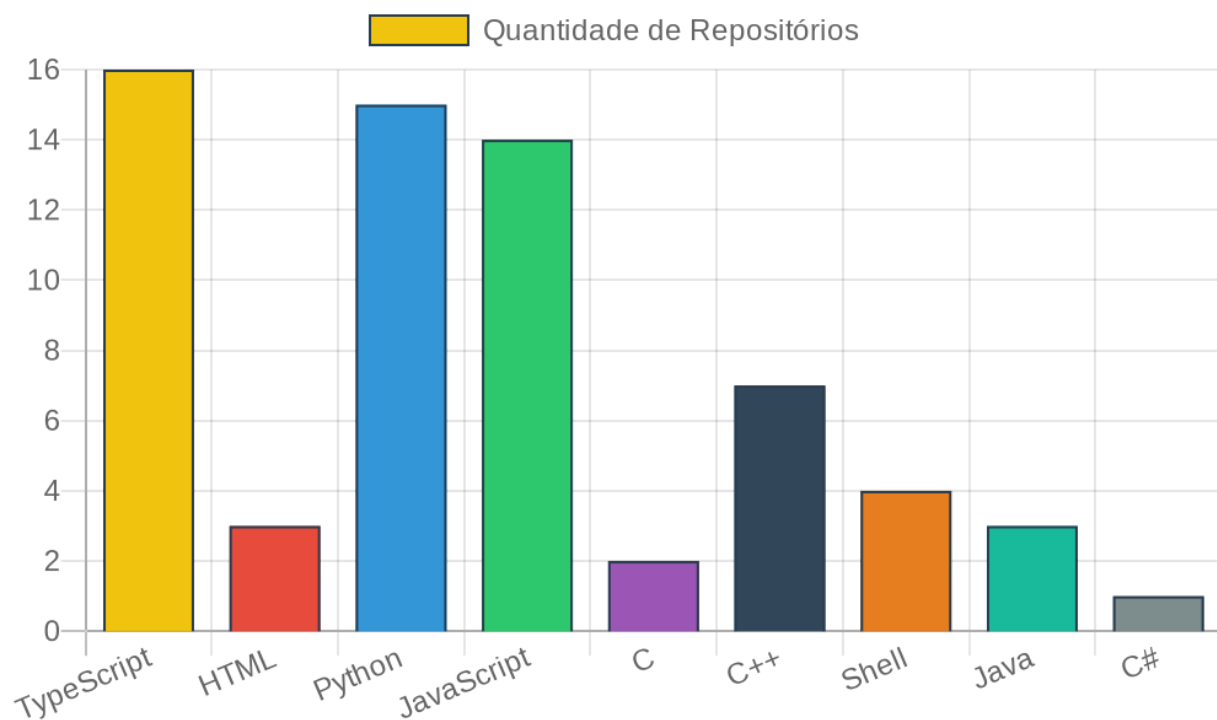
Número de repositórios com mais de 10 releases

0-9 10-ou-mais



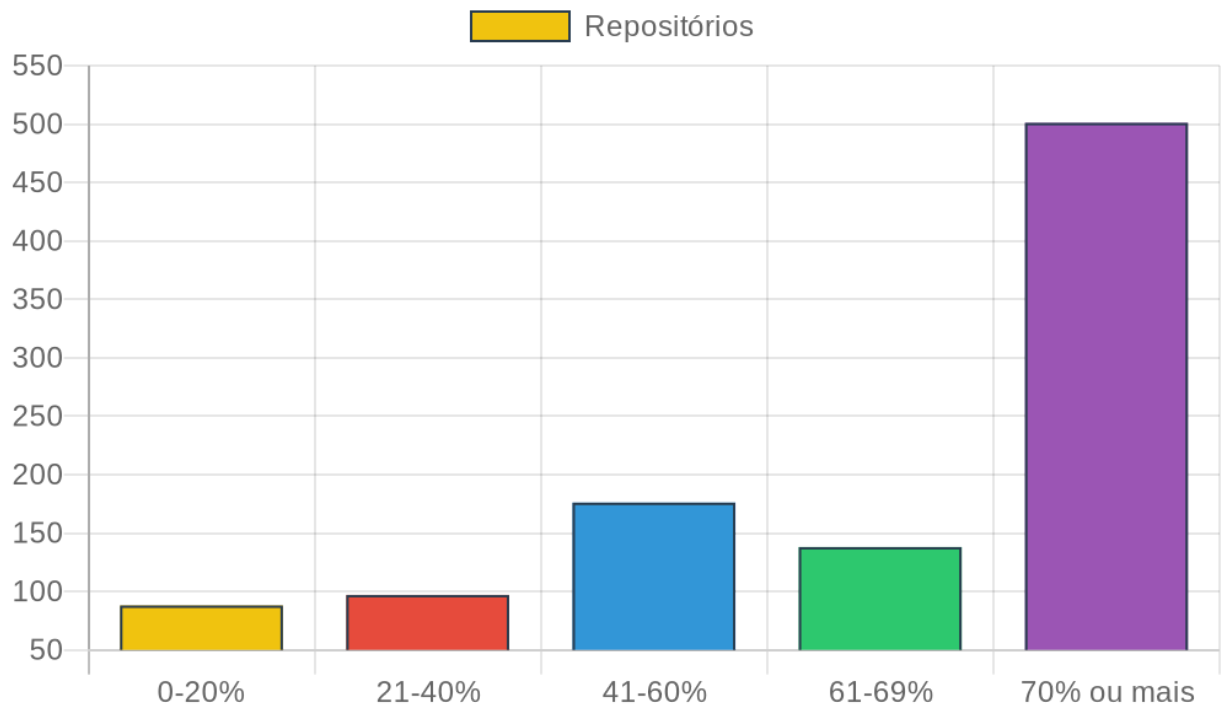
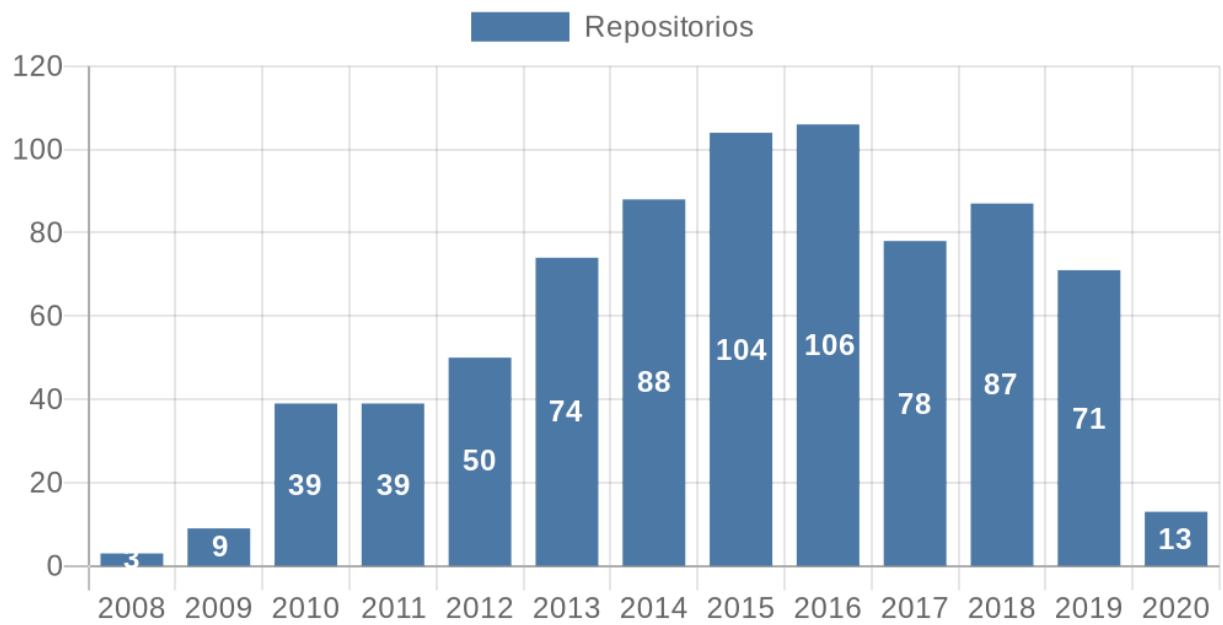
Tempo desde a última atualização (em meses)





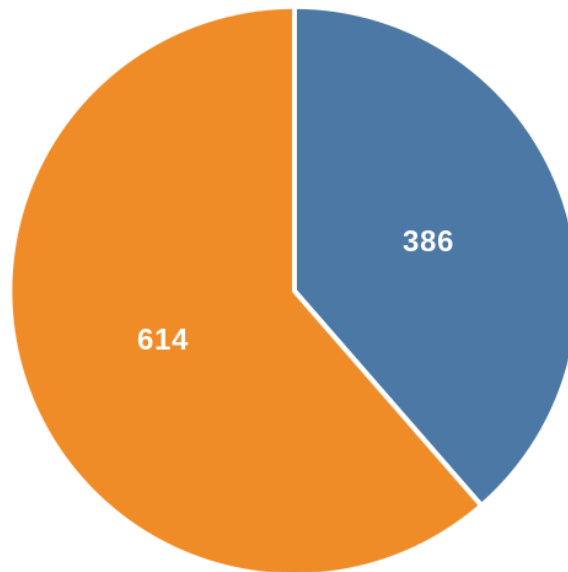
Para 1000 repositórios:

Repositórios com mais de 5 anos

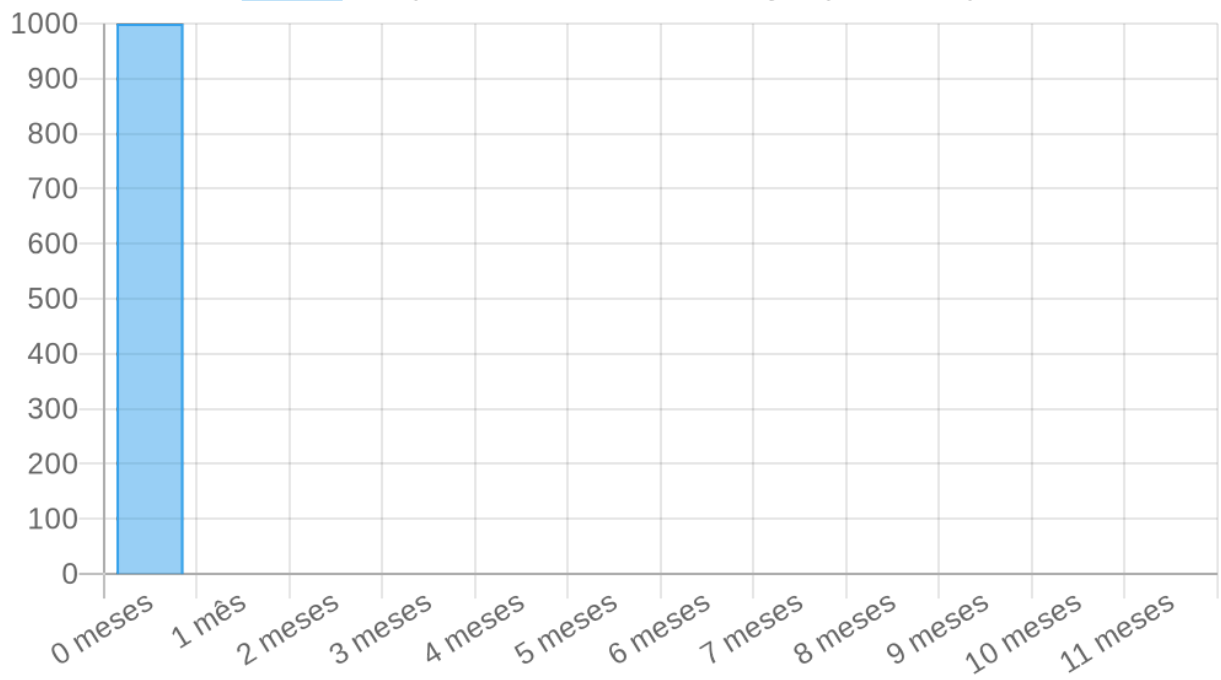


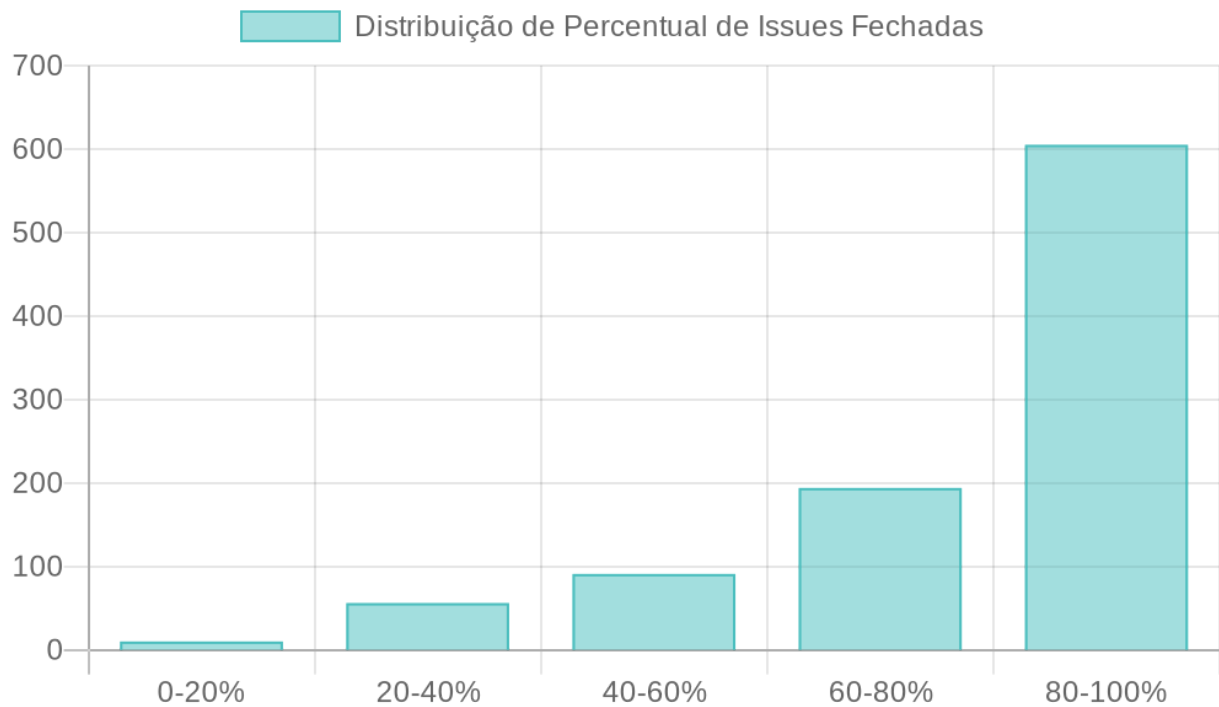
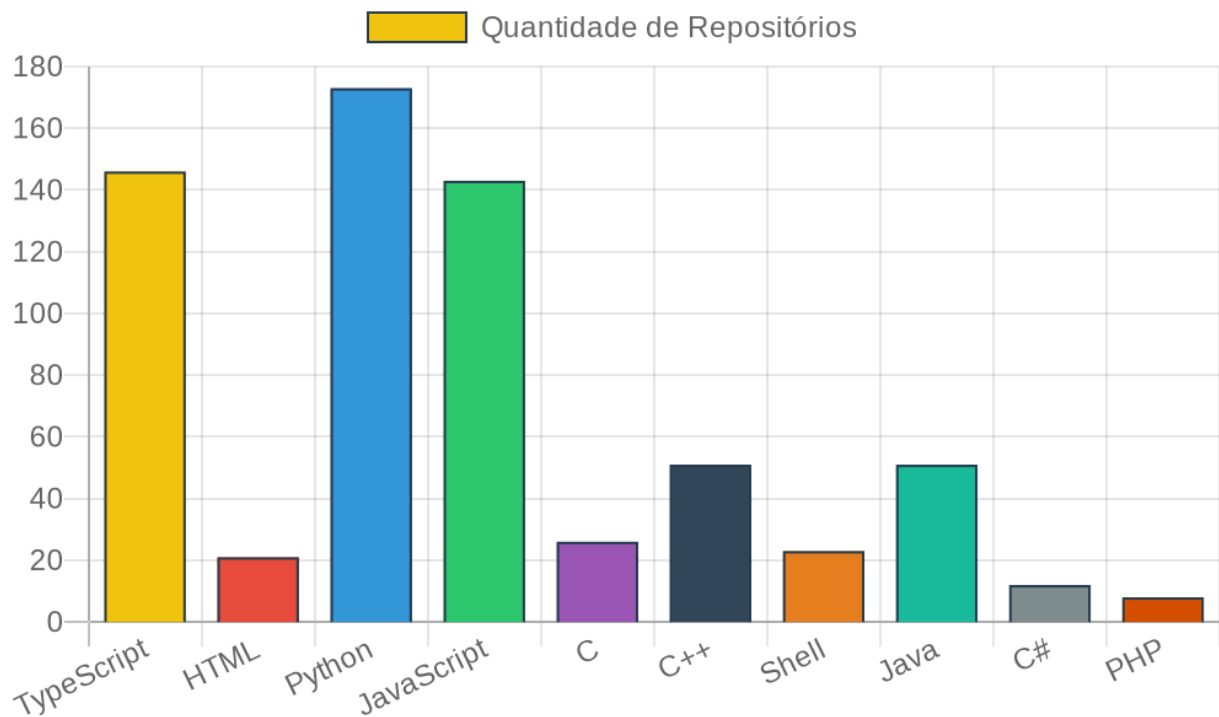
Número de repositórios com mais de 10 releases

0-9 10-ou-mais



Tempo desde a última atualização (em meses)





Análise Técnica das Diferenças

1. Idade dos Repositórios

Observa-se uma redução na proporção de repositórios antigos ao expandir a amostra de 100 para 1000. Isso indica que, entre os repositórios populares, existe uma maior diversidade etária, e repositórios mais recentes conseguem ganhar popularidade, desafiando a suposição de que apenas projetos antigos acumulam relevância.

2. Porcentagem de PR aceitos

O aumento sugere que, ao analisar um conjunto maior, encontramos mais repositórios populares com maior volume de contribuições. Isso pode ser explicado observando que repositórios populares precisam trazer melhorias contínuas para os utilizadores, ocasionando o aumento de pr criados e aceitos.

3. Número de Releases

O aumento de 7,20% no percentual de repositórios com 10 ou mais releases ao expandir a amostra indica que, entre um conjunto maior de projetos, há mais repositórios com um ciclo de desenvolvimento mais ativo e estruturado, refletindo um modelo de versionamento bem estabelecido.

4. Frequência de Atualizações

A porcentagem igual de repositórios atualizados no último ano sugere que a maioria dos repositórios populares continuam sendo mantidos ativamente, independentemente do tamanho da amostra.

5. Uso de Linguagens Populares

Aumento de 0,40% indica que, independente do tamanho das amostras, as linguagens que definimos como populares continuam sendo amplamente utilizadas. Isso pode ocorrer porque as linguagens populares facilitam a atração de contribuidores e usuários, garantindo maior engajamento.

6. Issues Fechadas

O aumento na porcentagem analisada sugere que repositórios populares são mais ativos em relação a contribuição no código, uma vez que as issues fechadas podem tanto ter sido descartadas quanto margeadas. Isso pode ser observado se analisarmos esses números juntamente com os dados do requisito 2, onde observamos um maior número de PR aceitos quando aumentamos a amostra. PR esses que foram aceitos e posteriormente podem ter sido margeados.

4. Conclusão

A análise comparativa mostra que, ao expandir a amostra, encontramos um ecossistema de repositórios populares mais diversificado. Projetos mais novos e com menos contribuições ainda conseguem se destacar, enquanto a adoção de linguagens populares se torna mais evidente. Além disso, observa-se um aumento na proporção de repositórios com um ciclo de lançamento bem estruturado, embora a quantidade de issues fechadas varie conforme a cultura e organização do projeto.

5. Referências

- GitHub GraphQL API: <https://docs.github.com/graphql>
 - Documentação sobre colaboração open-source
-