

Relatório Final: Sistemas Populares e suas Características

Grupo: Gustavo Riegert, Hitalo Porto, Pedro Motta

Data: 24/08/2024

1. Introdução

Este relatório busca analisar características comuns a sistemas populares no GitHub, respondendo a questões de pesquisa através de métricas específicas. Com base nas questões levantadas, foram feitas hipóteses informais sobre os resultados esperados.

2. Metodologia

Para responder às questões de pesquisa, foram coletados dados de repositórios populares no GitHub. As métricas utilizadas incluem idade do repositório, número de pull requests aceitas, número de releases, tempo até a última atualização, linguagem de programação primária e a razão entre issues fechadas e o total de issues. A análise foi realizada com base em valores medianos e contagem por categoria.

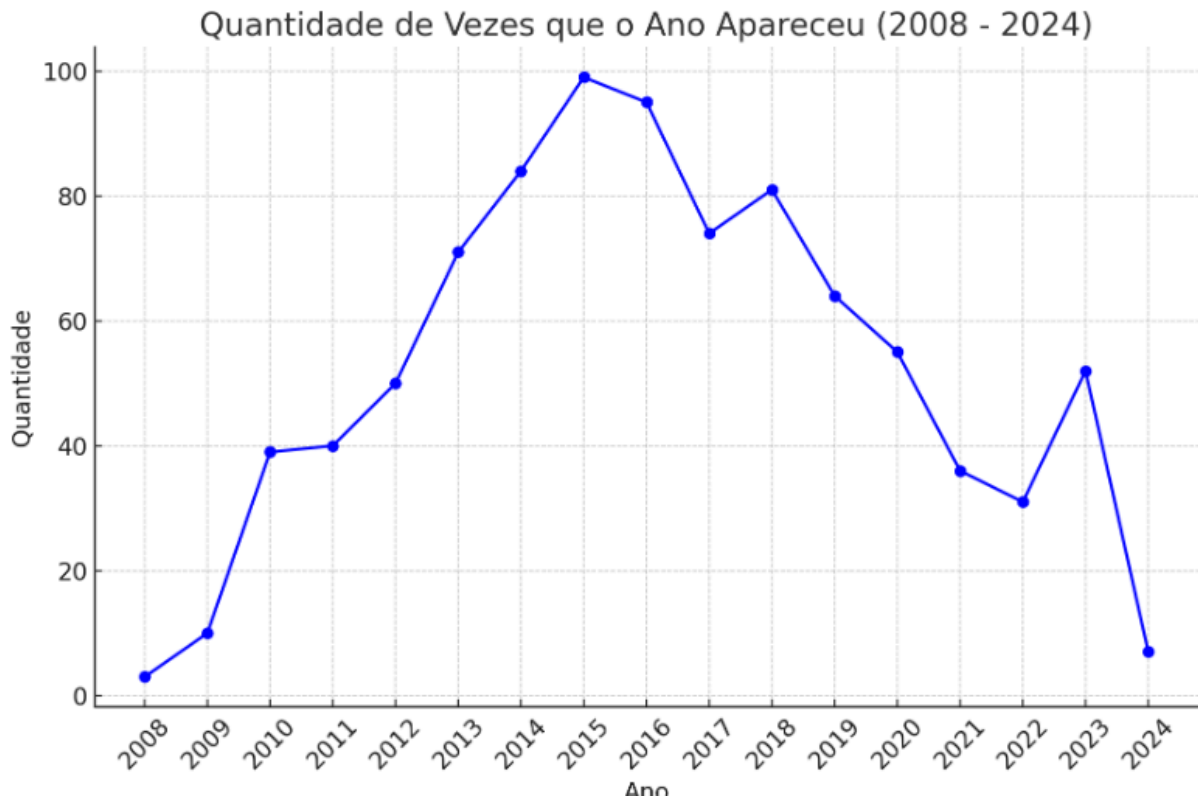
3. Resultados

RQ 01: Os sistemas populares são maduros/antigos?

Métrica: Idade do repositório

Hipótese: Espera-se que sistemas populares tenham uma idade considerável, já que a maturidade pode estar relacionada à popularidade.

Resultados: A idade mediana dos repositórios é de 9 anos, o que sugere que muitos repositórios populares têm uma história relativamente longa.



RQ 02: Os sistemas populares recebem muita contribuição externa?

Métrica: Total de pull requests aceitas

Hipótese: Espera-se que sistemas populares recebam muitas contribuições externas por serem mais conhecidos

Resultados: A mediana do total de pull requests aceitas é de 21, indicando um nível moderado de contribuição externa.

RQ 03: Os sistemas populares lançam releases com frequência?

Métrica: Total de releases

Hipótese: Supõe-se que sistemas populares tenham um número considerável de releases, indicando manutenção e desenvolvimento contínuos devido a grande maioria serem sistemas antigos.

Resultados: O número mediano de releases é 29, o que sugere que muitos repositórios populares têm um histórico ativo de lançamentos.

RQ 04: Os sistemas populares são atualizados com frequência?

Métrica: Tempo até a última atualização

Hipótese: Espera-se que sistemas populares sejam frequentemente atualizados, indicando que estão ativamente mantidos.

Resultados: A mediana do tempo desde a última atualização é de 1 dia, indicando que os repositórios populares são atualizados frequentemente.

RQ 05: Os sistemas populares são escritos nas linguagens mais populares?

Métrica: Linguagem primária

Hipótese: Linguagens populares como Python, JavaScript e Java devem ser predominantes em repositórios populares.

Resultados: As linguagens mais frequentes são Python (147 repositórios), JavaScript (143 repositórios), e TypeScript (122 repositórios).

RQ 06: Os sistemas populares possuem um alto percentual de issues fechadas?

Métrica: Razão entre número de issues fechadas pelo total de issues

Hipótese Informal: Supõe-se que sistemas populares possuam um alto percentual de issues fechadas, refletindo uma boa manutenção e gerenciamento de problemas.

Resultados: A razão mediana de issues fechadas é de 0.5, sugerindo que aproximadamente metade das issues abertas são resolvidas.

RQ 07: Sistemas escritos em linguagens mais populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência?

Métrica: Divisão dos resultados das RQ 02, 03 e 04 por linguagem.

Hipótese informal: Espera-se que em repositórios mais populares a contribuição seja maior por serem mais conhecidos por utilizarem linguagens mais conhecidas pelos usuários e consequentemente gerando uma maior frequência de atualização

Resultado: Ao dividir os resultados por linguagem, observamos que as linguagens populares como Python e JavaScript tendem a ter mais contribuições externas e maior frequência de releases e atualizações.

4. Discussão

Os resultados obtidos em grande parte confirmam as hipóteses iniciais. A mediana de 9 anos para a idade dos repositórios sugere que os repositórios populares geralmente são maduros. O número significativo de pull requests aceitas e a alta frequência de releases e atualizações corroboram a ideia de que repositórios populares têm um nível elevado de manutenção e suporte da comunidade.

No entanto, a razão de issues fechadas sendo mediana a 0.5 pode indicar desafios na resolução de problemas, possivelmente devido ao alto volume de issues ou à complexidade dos projetos.

A análise por linguagem sugere que as linguagens mais populares tendem a atrair mais contribuições e têm maior atividade de desenvolvimento, o que é consistente com a hipótese 7.

5. Conclusão

Este estudo confirmou que os repositórios populares têm características de maturidade, alta contribuição externa, e são frequentemente atualizados. Além disso, a popularidade das linguagens de programação desempenha um papel importante na atividade do repositório.