

Relatório – Características de Repositórios Populares no GitHub

Nomes: Pedro franco e Gabriel Pongelupe

Introdução

Este trabalho teve como objetivo analisar os 1000 repositórios mais populares do GitHub, a partir de suas métricas básicas de desenvolvimento. A ideia foi levantar hipóteses sobre a maturidade, frequência de contribuição externa, práticas de releases, frequência de atualização, linguagens utilizadas e tratamento de issues.

Hipóteses iniciais:

- **RQ01:** Espera-se que sistemas populares sejam maduros/antigos, pois o tempo tende a consolidar sua base de usuários.
- **RQ02:** Repositórios populares devem receber muita contribuição externa, visto que são amplamente utilizados.
- **RQ03:** Supõe-se que sistemas populares lançam releases com frequência para atender suas comunidades.
- **RQ04:** Repositórios populares provavelmente são atualizados constantemente, pela relevância e demanda de manutenção.
- **RQ05:** Espera-se que estejam escritos principalmente nas linguagens mais populares (ex.: Python, JavaScript, TypeScript).
- **RQ06:** Sistemas populares devem ter boa taxa de issues fechadas, indicando manutenção ativa.

Metodologia

- Foi utilizada a **API GraphQL do GitHub**.
- A coleta envolveu **100 páginas de 10 repositórios cada**, totalizando **1000 repositórios**.
- Para cada repositório, foram extraídas as métricas: idade, número de PRs aceitos, releases, tempo desde a última atualização, linguagem primária e issues (abertas e

fechadas).

- Após a coleta, os dados foram processados e salvos em CSV.
 - A análise foi conduzida a partir de **medianas**, conforme orientado no enunciado.
-

Resultados Obtidos

RQ01 – Idade dos repositórios

- **Mediana:** 3050,5 dias (cerca de 8,3 anos).
- **Média:** 2948 dias (~8 anos).
- **Hipótese confirmada:** a maioria dos sistemas populares é madura e antiga, possuindo longa trajetória no GitHub.

RQ02 – Contribuições externas (Pull Requests aceitas)

- **Mediana:** 702 PRs.
- **Média:** 3592 PRs.
- **Hipótese confirmada parcialmente:** muitos repositórios recebem bastante contribuição externa, mas a grande discrepância entre mediana e média indica que poucos projetos muito grandes (como React, TensorFlow, Kubernetes) puxam a média para cima.

RQ03 – Frequência de releases

- **Mediana:** 35 releases.
- **Média:** 109 releases.
- **Repositórios sem releases:** 308 (30,8%).
- **Hipótese refutada parcialmente:** embora muitos sistemas populares lancem releases, há uma quantidade significativa de projetos que não seguem práticas formais de versionamento (talvez utilizem outras estratégias, como *rolling release* ou tags informais).

RQ04 – Atualização dos repositórios

- **Mediana:** 0 dias desde a última atualização.
- **100% dos repositórios atualizados nos últimos 30 dias.**
- **Hipótese confirmada fortemente:** os repositórios populares estão em constante evolução e mantêm alta frequência de atualização.

RQ05 – Linguagens utilizadas

Top 5 linguagens primárias:

1. Python – 189 repositórios
 2. TypeScript – 156
 3. JavaScript – 129
 4. Unknown – 104
 5. Go – 73
- **Hipótese confirmada:** as linguagens mais populares do mercado também são predominantes nos repositórios populares.
 - A presença de “Unknown” (104) pode indicar projetos multi-linguagem ou onde o GitHub não identifica uma linguagem dominante.

RQ06 – Issues fechadas

- **Percentual mediano:** 85,7%
- **Percentual médio:** 76,3%
- **100% fechadas:** 14 repositórios
- **Hipótese confirmada:** a maioria dos sistemas populares demonstra boa taxa de resolução de issues, sugerindo gestão ativa das comunidades.

Discussão

Os resultados apontam que **repositórios populares do GitHub apresentam maturidade, alta frequência de atualização e boa manutenção de issues**. Apesar disso, a **quantidade de releases** não é homogênea, o que sugere que nem todos os projetos adotam versionamento formal.

As contribuições externas são significativas, mas há uma concentração em poucos projetos gigantes, o que distorce a média. Em relação às linguagens, os resultados confirmam o esperado: Python, JavaScript e TypeScript lideram, o que reforça a tendência atual do mercado.

Conclusão

- **Sistemas populares são maduros** (idade média de ~8 anos).
- **Recebem contribuições externas**, embora concentradas em alguns projetos grandes.
- **Lançam releases com frequência variável**, e cerca de 30% não publicam releases formais.
- **São extremamente ativos**, com atualizações praticamente diárias.
- **Utilizam majoritariamente linguagens populares**, como Python, TypeScript e JavaScript.
- **Possuem alto percentual de issues fechadas**, o que indica boa gestão comunitária.

Esses achados reforçam que a **popularidade no GitHub está associada à maturidade, comunidade ativa, alta manutenção e uso de linguagens mainstream**.