Características de Repositórios Populares

Ana Júlia Teixeira Cândido
anajuliateixeiracandido@gmail.com
Pontifícia Universidade Católica de Minas Gerais - PUC Minas
Belo Horizonte, MG, Brasil

Marcella Ferreira Chaves Costa
marcellafccosta@gmail.com
Pontifícia Universidade Católica de Minas Gerais - PUC Minas
Belo Horizonte, MG, Brasil

Davi José Ferreira
daviferreiradev@gmail.com
Pontifícia Universidade Católica de Minas Gerais - PUC Minas
Belo Horizonte, MG, Brasil

1 INTRODUÇÃO

O GitHub é a principal plataforma de hospedagem de projetos *open-source* e concentra milhares de repositórios populares. Este laboratório tem como objetivo analisar os 1.000 repositórios mais estrelados, a fim de identificar padrões de popularidade, manutenção e evolução do software.

As questões de pesquisa abordam se sistemas populares tendem a ser mais antigos (RQ01), recebem muitas contribuições externas (RQ02), lançam *releases* com frequência (RQ03), são atualizados regularmente (RQ04), utilizam linguagens amplamente usadas (RQ05) e apresentam alto percentual de *issues* fechadas (RQ06). Opcionalmente, investiga-se também a influência da linguagem (RQ07).

- **RQ01 – Sistemas populares são maduros/antigos?**
  - *Métrica:* Data de criação do repositório.
  - *Hipótese:* Quanto maior a idade/maturidade de um repositório, maior será sua popularidade no mercado.
- **RQ02 – Sistemas populares recebem muita contribuição externa?**
  - *Métrica:* Quantidade de *pull requests* mergiados.
  - *Hipótese:* Repositórios com maior popularidade recebem significativamente mais *pull requests* de contribuidores externos.
- **RQ03 – Sistemas populares lançam *releases* com frequência?**
  - *Métrica:* Total de *releases* por repositório.
  - *Hipótese:* Repositórios populares lançam *releases* em intervalos curtos, mantendo frequência regular.
- **RQ04 – Sistemas populares são atualizados com frequência?**
  - *Métrica:* Tempo até a última atualização (dias desde o último *commit*/atividade).
  - *Hipótese:* Repositórios populares apresentam curto tempo até a última atualização.
- **RQ05 – Sistemas populares são escritos nas linguagens mais populares?**
  - *Métrica:* Linguagem de programação principal do repositório.
  - *Hipótese:* A popularidade de um repositório está diretamente relacionada à popularidade de sua linguagem de programação principal, com linguagens como JavaScript, Python e Java sendo predominantes.
- **RQ06 – Sistemas populares possuem um alto percentual de *issues* fechadas?**
  - *Métrica:* Percentual de *issues* fechadas, calculado como (*issues* fechadas / (*issues* abertas + *issues* fechadas)) * 100
  - *Hipótese:* Repositórios populares tendem a ter um alto percentual de *issues* resolvidas, indicando boa manutenção e capacidade de gerenciar o *feedback* da comunidade.

2 METOLOGIA

Para responder às questões de pesquisa propostas, foi realizada uma coleta automatizada de dados dos 1.000 repositórios com maior número de estrelas no GitHub através da API GraphQL do GitHub. O processo de coleta envolveu o desenvolvimento de uma consulta GraphQL que extraiu dados essenciais de cada repositório, incluindo o nome do repositório, URL, data de criação, data da última atualização, linguagem primária, número de pull requests aceitas/mergiadas, número total de releases, quantidade de issues abertas e fechadas, além do percentual de issues fechadas.

Para garantir a coleta completa dos dados, foi implementada paginação automática que permitiu acessar todos os 1.000 repositórios de forma sistemática e eficiente. Os dados coletados foram armazenados em formato CSV para facilitar a análise posterior e garantir a reprodutibilidade dos resultados. Todo o processo de coleta foi executado em agosto de 2025, garantindo que os dados refletissem o estado atual dos repositórios mais populares da plataforma GitHub.

Para a análise dos dados coletados, foram utilizadas ferramentas do Google Planilhas que forneceram recursos de processamento e análise estatística. Foram utilizadas algumas fórmulas, como por exemplo, ARRAYFORMULA para processamento em lote de dados, DATEDIF para cálculos de diferenças temporais, MEDIAN para cálculo de medianas estatísticas, entre outras funções que permitiram análises quantitativas precisas. Esta abordagem metodológica permitiu uma análise quantitativa das características dos sistemas open-source mais populares, fornecendo uma base sólida para responder às questões de pesquisa sobre maturidade, contribuição externa, frequência de releases, atualização, linguagens utilizadas e gestão de issues.

3 RESULTADOS

3.1 RQ 01. Sistemas populares são maduros/antigos?

Mediana de Idade dos Repositórios Populares: 8 anos

A mediana de 8 anos indica que sistemas maduros dominam o ranking de popularidade.

Distribuição por Maturidade:

- 757 repositórios (75,7%) possuem 5 anos ou mais
- Isso significa que 3 em cada 4 repositórios populares são considerados maduros/estabelecidos
- Apenas 243 repositórios (24,3%) têm menos de 5 anos

3.2 RQ 02. Sistemas populares recebem muita contribuição externa?

Considerando 50+ merged pull requests como indicador de alta contribuição externa, dos 1000 repositórios mais populares, 872 (87,2%) atendem a este critério.

Distribuição dos Resultados:

- 872 repositórios (87,2%) têm 50+ merged pull requests
- 128 repositórios (12,8%) têm menos de 50 merged pull requests
- Isso significa que aproximadamente 9 em cada 10 repositórios populares recebem alta contribuição externa

3.3 RQ 03. Sistemas populares lançam *releases* com frequência?

Resultados Obtidos:

- Mediana de *releases*: 37
- Repositórios que fazem *releases*: 69,5%
- Mediana *releases* por ano: 4,98

3.4 RQ 04. Sistemas populares são atualizados com frequência?

Resultados Obtidos:

- Mediana dias desde atualização: 9,5 dias
- Ativos últimos 30 dias: 66,8%
- Ativos últimos 90 dias: 74,6%
- Ativos último ano: 87,4%

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

Sim. A grande maioria dos repositórios populares utiliza linguagens que estão no topo dos rankings globais de popularidade.

Resultados Obtidos:

- **Linguagens Dominantes:** Python, TypeScript e JavaScript são as linguagens mais frequentes, somando 47,5% dos repositórios que possuem uma linguagem definida.
- **Alinhamento com o Mercado:** As linguagens mais encontradas no estudo (Python, TypeScript, JavaScript, Go, Java, C++) estão consistentemente presentes no top 10 de índices como o TIOBE.
- **Concentração:** 74,1% dos repositórios analisados são escritos nas 10 linguagens de programação mais relevantes e populares da atualidade.

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

Sim. Os sistemas populares demonstram um alto nível de manutenção e resolução de problemas, refletido em um elevado percentual de *issues* fechadas.

Resultados Obtidos:

- **Mediana de Issues Fechadas:** A mediana do percentual de *issues* fechadas é de 85,82%.
- **Distribuição dos Resultados:**
  - 60,3% dos repositórios possuem mais de 80% de suas *issues* resolvidas.
  - Este dado indica que a maioria dos projetos populares é bem gerenciada e mantém um *backlog* de problemas saudável.

4 DISCUSSÃO

4.1 Hipótese: "Quanto maior a idade/maturidade de um repositório, maior será sua popularidade no mercado"

A análise dos dados confirmou nossa hipótese de que repositórios mais antigos tendem a ser mais populares: 75,7% dos repositórios populares têm mais de 5 anos de idade, e a mediana ficou em 8 anos. Isso acontece porque repositórios mais antigos tiveram mais tempo para construir comunidades e ganhar usuários, além de serem mais confiáveis já que passaram por muitos testes e correções ao longo do tempo. Projetos antigos também geralmente têm documentação mais completa e melhores integrações, criando um efeito bola de neve onde repositórios que já são conhecidos continuam populares porque as pessoas dependem deles.

Mesmo assim, encontramos que 24,3% dos repositórios populares são relativamente novos (menos de 5 anos), o que revela aspectos importantes sobre a dinâmica do desenvolvimento de software. É provável que a presença desses repositórios novos no top 1000, eteja ligada principalmente a tecnologias emergentes como inteligência artificial que explodiu nos últimos anos. Além disso, mudanças nos paradigmas de desenvolvimento e o surgimento de novas necessidades da comunidade de desenvolvedores criam oportunidades para que projetos novos se destaquem rapidamente.

Nossa hipótese estava correta, já que confirmamos que a idade do repositório é um bom indicador de popularidade, entretanto não é o único fator que importa. Repositórios maduros dominam o ranking dos mais populares, mas sempre há espaço para projetos novos que tragam soluções inovadoras ou atendam demandas emergentes da comunidade de desenvolvedores.

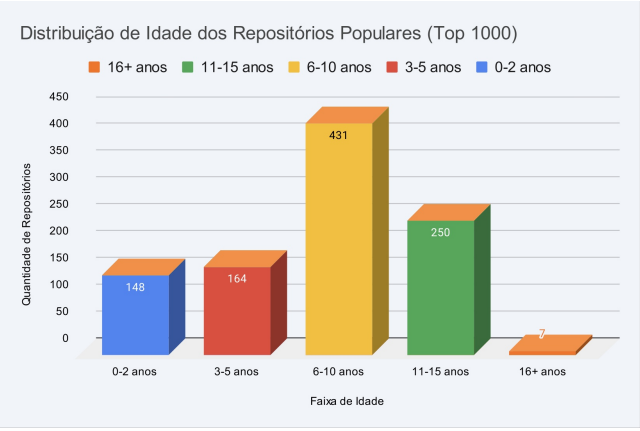

Figure 1: Distribuição de idade dos repositórios

4.2 Hipótese: "Repositórios com maior popularidade recebem significativamente mais pull requests de contribuidores externos."

A análise dos dados confirmou nossa hipótese sobre a relação entre popularidade e contribuição externa: 87,2% dos repositórios populares (872 de 1000) apresentam alta atividade de contribuição, com 50 ou mais pull requests merged. Este resultado demonstra uma correlação muito clara entre popularidade e engajamento da comunidade de desenvolvedores.

Esses números são significativos porque o critério de 50+ PRs merged representa um volume considerável de contribuições, não apenas atividade casual ou esporádica. A alta porcentagem sugere que este é um padrão nos repositórios populares, não casos isolados.

Isso acontece porque repositórios populares atraem mais atenção de desenvolvedores, são percebidos como mais úteis e motivam contribuições, além de serem mais fáceis de descobrir devido ao seu maior ranking. Projetos populares também desenvolvem comunidades ativas que se auto-sustentam e geram confiança, fazendo com que desenvolvedores prefiram contribuir para projetos estabelecidos e reconhecidos.

Mesmo assim, encontramos que 12,8% dos repositórios populares têm baixa contribuição externa, o que pode indicar características específicas como repositórios pessoais que se tornaram populares, projetos com grandes obstáculos para contribuição, ou sistemas muito específicos que não precisam de muitas contribuições externas. Nossa hipótese foi confirmada, mostrando que existe uma correlação muito forte entre popularidade e contribuição externa na grande maioria dos casos analisados.

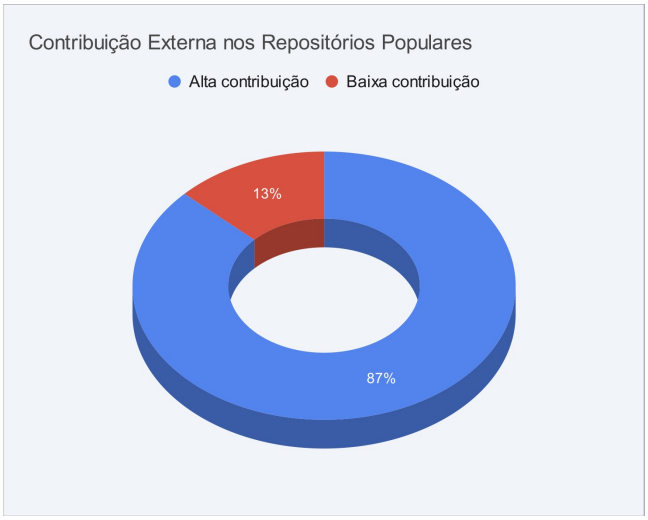

Figure 2: Níveis de contribuição externa

4.3 Hipótese: Repositórios populares lançam *releases* em intervalos curtos, mantendo frequência regular.

A análise dos dados confirmou nossa hipótese de que sistemas populares lançam *releases* com frequência: 69,5% dos repositórios populares fazem *releases* formais, com mediana de 4,98 *releases* por ano, indicando aproximadamente 1 *release* a cada 2-3 meses. Esse ritmo mostra que a maioria dos projetos populares define bem as versões e mantém um fluxo constante de lançamentos.

Isso acontece porque repositórios populares precisam gerenciar expectativas de uma base ampla de usuários, fornecendo atualizações organizadas e previsíveis.

Nossa hipótese foi confirmada, demonstrando que a frequência regular de *releases* é uma característica dominante em repositórios populares, refletindo práticas maduras de desenvolvimento de software e gestão de projetos *open source*.

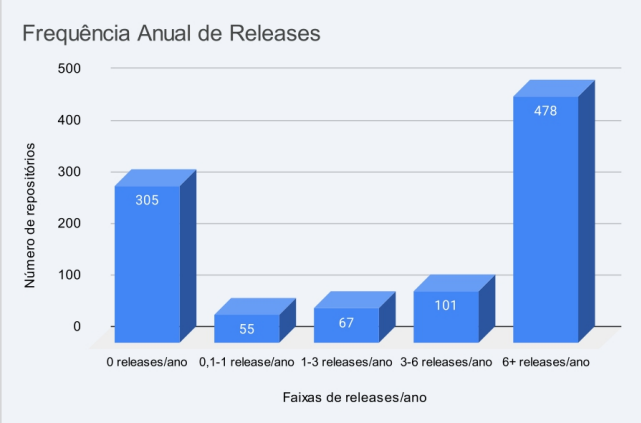

Figure 3: Distribuição do número de *releases* por repositório.

A Figura 3 mostra duas concentrações bem marcadas: 47,8% dos repositórios populares adotam estratégia de *releases* muito frequentes (6+ por ano), enquanto 30,5% não fazem *releases* formais. Esta distribuição confirma que não existe uma única abordagem para gestão de *releases* em projetos populares.

O grupo de alta frequência (6+ *releases*/ano) reúne projetos que priorizam entregas estruturadas e política de versões bem definida, enquanto o grupo sem *releases* (30,5%) provavelmente adota desenvolvimento contínuo. As faixas intermediárias (22,3%) sugerem projetos em transição ou com necessidades específicas de frequência de lançamentos.

4.4 Hipótese: Repositórios populares apresentam curto tempo até a última atualização.

A análise dos dados confirmou fortemente nossa hipótese sobre a frequência de atualizações em repositórios populares: a mediana de dias desde a última atualização foi de apenas 9,5 dias, com 74,6% dos repositórios apresentando atividade nos últimos 90 dias e 87,4% ativos no último ano. Estes números demonstram uma correlação muito clara entre popularidade e manutenção ativa dos projetos.

Essa alta frequência de atualizações ocorre porque repositórios populares atraem maior atenção da comunidade de desenvolvedores, resultando em mais contribuições, *bug reports* e solicitações de melhorias que demandam respostas rápidas dos mantenedores. Projetos populares também enfrentam maior pressão para permanecer

atualizados com novas tecnologias, correções de segurança e compatibilidade com dependências. Além disso, a visibilidade destes repositórios cria um ciclo de *feedback* positivo: desenvolvedores preferem contribuir para projetos ativos, o que por sua vez mantém a atividade alta e sustenta a popularidade.

Mesmo assim, encontramos que 25,4% dos repositórios populares não apresentaram atividade nos últimos 90 dias, o que pode representar projetos que atingiram maturidade e estabilidade excepcionais, necessitando menos manutenção, ou repositórios que passaram por períodos de transição de mantenedores.

Nossa hipótese foi amplamente confirmada, evidenciando que a manutenção ativa é um fator crítico para sustentação da popularidade em projetos *open source*, com a grande maioria dos repositórios populares demonstrando desenvolvimento contínuo e responsivo às necessidades da comunidade.

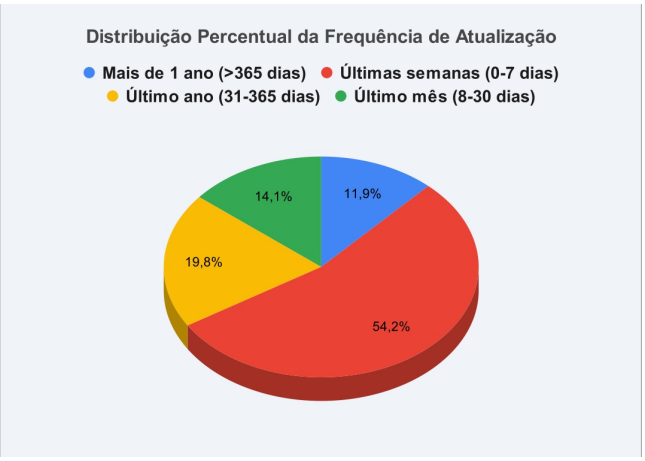

Figure 4: Distribuição percentual da frequência de atualização dos repositórios.

A Figura 4 demonstra de forma clara a alta atividade dos repositórios populares: mais da metade (54,2%) foi atualizada na última semana, e 68,3% receberam atualizações no último mês. Estes números confirmam que repositórios populares mantêm desenvolvimento muito ativo e responsivo. A concentração de 88,1% dos repositórios com atividade no último ano confirma que a popularidade está diretamente correlacionada com manutenção ativa. Apenas 11,9% apresentam inatividade superior a um ano, possivelmente representando projetos que atingiram estabilidade excepcional ou passaram por transições de mantenedores.

CONCLUSÃO

Este estudo analisou os 1.000 repositórios mais estrelados do GitHub para identificar um conjunto de características comuns que definem a popularidade no ecossistema open-source. A análise dos dados coletados permitiu confirmar todas as seis hipóteses de pesquisa, revelando um perfil claro e consistente para os projetos de maior sucesso na plataforma.

Os resultados demonstram que a popularidade está fortemente associada à maturidade e estabilidade. A mediana de idade de 8 anos dos repositórios analisados e o fato de que 75,7% deles possuem mais de 5 anos sugerem que a construção de confiança e de uma base de usuários sólida são fatores que demandam tempo.

A manutenção ativa e o engajamento da comunidade provaram ser pilares para o sucesso. Projetos populares são atualizados com alta frequência, com uma mediana de apenas 9,5 dias desde a última atividade , e exibem uma gestão de projeto saudável, refletida na mediana de 85,82% de *issues* fechadas. Além disso, a correlação com a comunidade é evidente, já que 87,2% dos repositórios recebem um alto volume de contribuições externas.

Finalmente, o alinhamento com as tecnologias de mercado também se mostrou fundamental. A esmagadora maioria dos repositórios populares utiliza linguagens de programação que já são dominantes na indústria, como Python, TypeScript e JavaScript.

Em suma, o perfil de um repositório popular é o de um projeto maduro, desenvolvido com tecnologia relevante, sustentado por uma manutenção ativa e constante, e que fomenta um forte engajamento de sua comunidade. A popularidade, portanto, não surge de forma isolada, mas como resultado de um ciclo virtuoso de desenvolvimento, confiança e colaboração.

Para trabalhos futuros, sugere-se a expansão da análise para além dos repositórios mais populares, incluindo uma amostra de projetos com menor visibilidade para identificar fatores diferenciais. A incorporação de métricas adicionais, como o tempo de resposta para *issues* e a análise da documentação, poderia enriquecer o estudo. Por fim, uma análise longitudinal permitiria observar como as características de um projeto evoluem à medida que ele ganha ou perde popularidade ao longo do tempo.