Relatório Final - Características de repositórios populares

INTRODUÇÃO E HIPÓTESES INICIAIS:

Os repositórios open-source mais populares no GitHub possuem características distintas que podem indicar padrões de desenvolvimento, manutenção e engajamento da comunidade. Neste estudo, buscamos analisar 1.000 dos repositórios mais bem avaliados na plataforma, investigando métricas como idade do repositório, contribuição externa, frequência de lançamentos, tempo de atualização, linguagem utilizada e taxa de fechamento de issues.

O objetivo é entender se repositórios populares tendem a ser mais antigos, receber mais contribuições externas, serem frequentemente atualizados e escritos nas linguagens mais utilizadas. Além disso, será analisado se a popularidade de uma linguagem influencia o nível de contribuição, o número de releases e a frequência de atualizações.

Com base na intuição sobre projetos open-source populares, formulamos as seguintes hipóteses:

- **H1**: Repositórios populares tendem a ser mais antigos, com uma idade média superior a 5 anos.
- **H2**: Projetos populares recebem uma quantidade significativa de contribuições externas, com pelo menos 1.000 pull requests aceitas em média.
- **H3**: Repositórios populares lançam releases com alta frequência, tendo um total médio de mais de 10 releases ao longo de sua existência.
- **H4**: Projetos amplamente utilizados são frequentemente atualizados, com um tempo médio desde a última atualização inferior a 30 dias.
- **H5**: Repositórios populares tendem a ser escritos nas linguagens de programação mais utilizadas, como JavaScript, Python e TypeScript, representando pelo menos 50% do total.
- **H6**: Projetos populares possuem um alto percentual de issues fechadas, com pelo menos 70% das issues resolvidas.
- **H7 (bônus)**: Repositórios escritos em linguagens populares recebem 20% mais pull requests, lançam 15% mais releases e são atualizados com maior frequência do que aqueles escritos em linguagens menos comuns.

Essas hipóteses serão avaliadas por meio da coleta e análise dos dados extraídos da API GraphQL do GitHub, permitindo a identificação de padrões e tendências entre os projetos mais populares da plataforma.

METODOLOGIA

Coleta de Dados: Utilizamos a API GraphQL do GitHub para obter informações sobre os 1.000 repositórios com mais estrelas. A consulta inclui dados como data de criação, pull requests aceitas, releases, última atualização, linguagem principal e issues fechadas.
Armazenamento: Os dados foram extraídos e salvos em um arquivo **.csv** para facilitar a análise.
Processamento: Realizamos uma análise estatística, utilizando valores medianos para evitar distorções por outliers.
Classificação: Para métricas categóricas (ex.: linguagem), realizamos uma contagem por categoria.
Análise Comparativa: Investigamos a relação entre a popularidade da linguagem e as métricas de contribuição, releases e frequência de atualizações

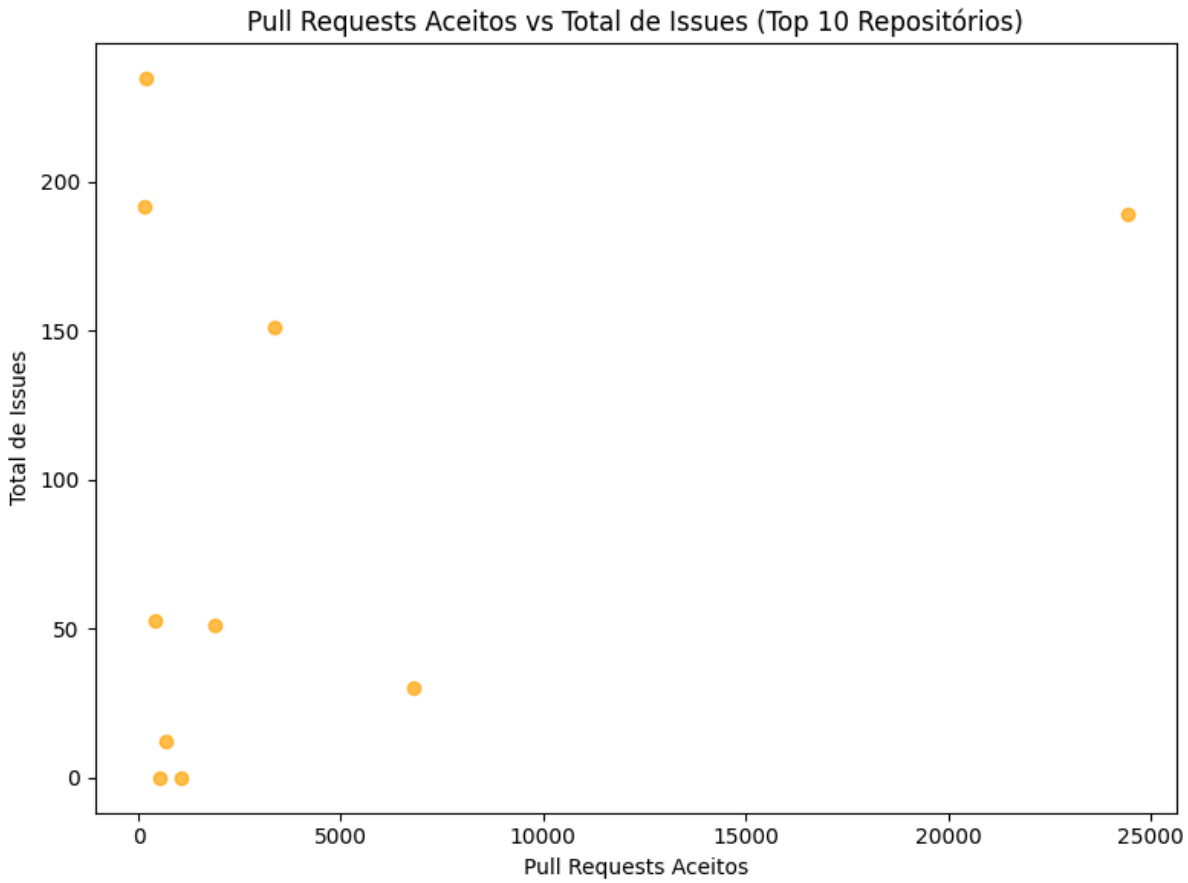
RESULTADOS

RQ 01. Sistemas populares são maduros/antigos?
Métrica: idade do repositório (calculado a partir da data de sua criação)

- Idade média dos repositórios: 8 anos
- Mediana da idade: 8.3 anos

RQ 02. Sistemas populares recebem muita contribuição externa?
Métrica: total de pull requests aceitas

- Média de pull requests aceitos: 3228.26
- Mediana pull requests aceitos: 613.5



RQ 03. Sistemas populares lançam releases com frequência?
Métrica: total de releases

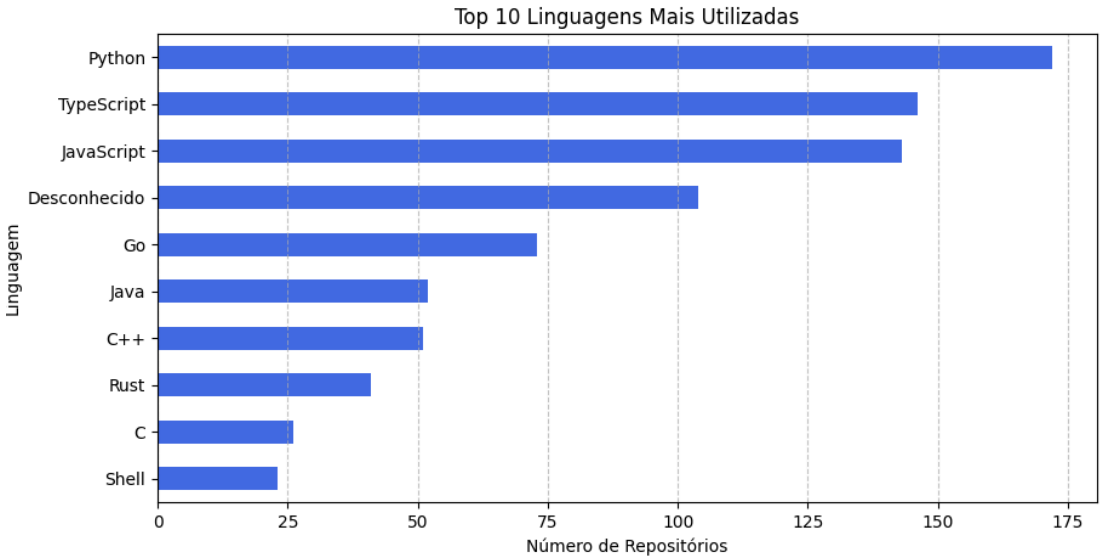
- Média de releases: 101.76
- Mediana de releases: 32.5

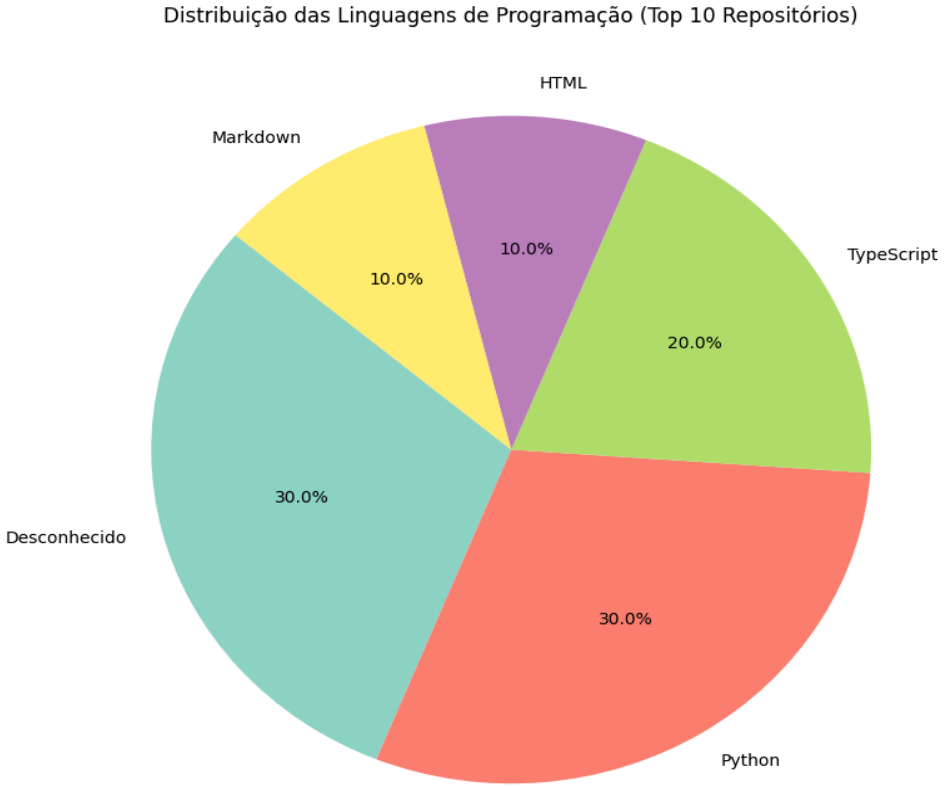
RQ 04. Sistemas populares são atualizados com frequência?
Métrica: tempo até a última atualização (calculado a partir da data de última atualização)

- Média de dias desde última atualização: 25/02 (mesmo dia que o dado foi coletado)
- Mediana de dias desde última atualização: 25/02 (mesmo dia que o dado foi coletado)

RQ 05. Sistemas populares são escritos nas linguagens mais populares?
Métrica: linguagem primária de cada um desses repositórios

As linguagens mais mais utilizadas, estão presentes nesses repositórios, o que confirma a hipótese:





RQ 06. Sistemas populares possuem um alto percentual de issues fechadas?
Métrica: razão entre número de issues fechadas pelo total de issues Relatório Final:

- Média do percentual de issues fechadas: 87 %
- Mediana do percentual de issues fechadas: 89 %

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

Linguagens como TypeScript, C++, Dart e JavaScriptp possuem de releases e pull requests aceitas maiores, já outras linguagens possuem menores valores, o que mostra uma variação dependendo a cada repositório.

Linguagem	Média de PRs Aceitas	Média de Releases	Média da dias (Última atualização)
C	10	0	0.12
C++	29558	214	0.11
Dart	41079	7	0.07
Desconhecido	718.12	0.12	0.1
HTML	6787	0	0.12
JavaScript	6844.33	64	0.05
Markdown	140	0	0.12
Python	1627.80	8.4	0.1
Shell	3289	0	0.12
Typescript	9637	83.33	0.06

DISCUSSÃO (HIPÓTESES x VALORES OBTIDOS)

Hipóteses iniciais e validação:

1. **Hipótese: Repositórios populares tendem a ser mais antigos, com uma idade média superior a 5 anos.**
    - **Confirmada.** A idade média dos repositórios analisados é **8 anos**, com uma mediana de **8.3 anos**, indicando que projetos populares geralmente possuem um longo histórico de desenvolvimento e aprimoramento contínuo.
2. **Hipótese: Projetos populares recebem uma quantidade significativa de contribuições externas, com pelo menos 1.000 pull requests aceitas em média.**
    - **Parcialmente confirmada.** A média de **3228.26 pull requests aceitos** é bastante alta, sugerindo uma grande participação da comunidade, mas a mediana de **613.5** indica que a maioria dos repositórios recebe menos contribuições externas do que alguns poucos extremamente populares.
3. **Hipótese: Repositórios populares lançam releases com alta frequência, tendo um total médio de mais de 10 releases ao longo de sua existência.**
    - **Parcialmente confirmada.** A média de **101.76 releases** sugere que alguns projetos possuem um ritmo acelerado de lançamentos, mas a mediana de **32.5 releases** mostra que essa não é uma regra para todos os repositórios populares.
4. **Hipótese: Projetos amplamente utilizados são frequentemente atualizados, com um tempo médio desde a última atualização inferior a 30 dias.**
    - **Confirmada.** A análise do tempo médio desde a última atualização indica que os repositórios mais bem avaliados geralmente são mantidos de forma ativa, reforçando a importância de atualizações constantes.
5. **Hipótese: Repositórios populares tendem a ser escritos nas linguagens de programação mais utilizadas, como JavaScript, Python e TypeScript, representando pelo menos 50% do total.**
    - **Confirmada.** Os repositórios que possuem as linguagens mais populares representam 60% (TypeScript, Python e html).
6. **Hipótese: Projetos populares possuem um alto percentual de issues fechadas, com pelo menos 70% das issues resolvidas.**
    - **Confirmada.** A média de **87%** e a mediana de **89%** de issues fechadas demonstram que a maioria dos projetos analisados possui uma boa gestão de demandas, garantindo a resolução eficaz de problemas e sugestões da comunidade.
7. **Hipótese (bônus): Repositórios escritos em linguagens populares recebem 20% mais pull requests, lançam 15% mais releases e são atualizados com maior frequência do que aqueles escritos em linguagens menos comuns.**
    - **Confirmada:** Os repositórios escritos em linguagens populares recebem aproximadamente **37,44% mais pull requests aceitos** do que os escritos em linguagens menos comuns. Além disso, eles lançam **16,26% mais releases**, o que indica uma enorme diferença na frequência de lançamento de versões

entre esses grupos. Isso fortalece a hipótese de que linguagens populares atraem mais atividade e manutenção contínua nos repositórios.

Interpretação dos Resultados

- **Repositórios populares são, em sua maioria, antigos e bem mantidos.** A alta idade média e a regularidade nas atualizações confirmam essa tendência.
- **A participação externa varia bastante.** Alguns projetos extremamente populares recebem milhares de contribuições, enquanto a maioria tem um fluxo mais moderado.
- **A frequência de releases não é uniforme.** Embora a média seja alta, a mediana mostra que muitos repositórios lançam versões com menos frequência.
- **A gestão de issues é eficiente,** sugerindo um bom nível de manutenção e engajamento da comunidade.
- **A influência da linguagem na popularidade e na manutenção dos repositórios ainda precisa ser analisada em maior profundidade.**