

1. .

- A. **Hipótese:** Neste item, é questionado se os repositórios mais populares do GitHub (com as maiores quantidades de estrelas) são maduros/antigos. A hipótese levantada por mim é de que sim, os repositórios mais populares são também antigos. Acredito que apenas a qualidade de um projeto não basta para que este se torne amplamente conhecido, utilizado e aprovado. Acredito que o tempo também é crucial por uma série de motivos, como por exemplo: é ao longo do tempo que problemas e vulnerabilidades vão sendo descobertas e corrigidas; o tempo muitas vezes filtra pessoas mal-intencionadas e também leva-se tempo para que a informação sobre a existência de determinado projeto se espalhe pela comunidade. Todos esses aspectos são verdadeiros para vários outros aspectos da atividade humana, e acredito que estes se mantêm válidos para projetos open-source.
- B. **Metodologia:** Para responder a esta questão, foi realizada uma consulta à API do GitHub usando GraphQL, através de um script escrito em Python. Neste primeiro momento, foram analisados os 100 repositórios mais populares da plataforma, ou seja, com as maiores quantidades de estrelas atribuídas. Para a filtragem desta consulta, foi utilizado o campo “createdAt”.
- C. **Resultados obtidos:** O resultado encontrado mostrou que a mediana da “idade” dos 100 repositórios mais populares é de 3247,5 dias, ou seja, aproximadamente **8 anos e 11 meses**. O conceito de maturidade de repositórios é bastante relativo, mas, na minha opinião, pode-se dizer que quase 9 anos de projeto representa um elevado grau de maturidade.
- D. **Discussão sobre resultados esperados e obtidos:** O resultado obtido confirma o que era esperado.

2. .

- A. **Hipótese:** Este item questiona se projetos populares recebem muita contribuição externa. Mais uma vez, acredito que sim. No contexto específico de projetos de código aberto, entendo que um grande fator determinante de seu sucesso é a quantidade de colaboradores, vide projetos amplamente utilizados e respeitados como o Linux e o Bitcoin. Além disso, é muito lógico pensar que a popularidade de um projeto e a quantidade de colaboradores são duas variáveis diretamente proporcionais, ou seja, projetos com muitos colaboradores são mais divulgados e comentados, da mesma forma que a própria popularidade também traz novos colaboradores, o que faz dessa relação uma espiral retroalimentada.
- B. **Metodologia:** Para responder a esta questão, foi realizada uma consulta à API do GitHub usando GraphQL, através de um script escrito em Python. Neste primeiro momento, foram analisados os 100 repositórios mais populares da plataforma, ou seja, com as maiores quantidades de estrelas atribuídas. Para a filtragem desta consulta, foi utilizado o campo “pullRequests”, com o modificador “states: MERGED”, juntamente ao subcampo “totalCount”.
- C. **Resultados obtidos:** O resultado encontrado foi uma **mediana de 1.606 pull requests aceitos**. Trata-se de um valor bastante expressivo.

D. Discussão sobre resultados esperados e obtidos: O resultado obtido confirma o que era esperado.

3. .

A. Hipótese: Acredito que projetos populares lançam releases com frequência. Isso porque o uso amplo de determinada tecnologia leva inevitavelmente à descoberta de falhas e pontos de melhoria.

B. Metodologia: Erro não solucionado de timeout =/

C. Resultados obtidos: Erro não solucionado de timeout =/

D. Discussão sobre resultados esperados e obtidos: Erro não solucionado de timeout =/

4. .

A. Hipótese: Desta vez, acredito que não existe uma relação direta entre a popularidade de um repositório e a proximidade da última atualização deste. Não porque nego essa relação, mas sim porque acredito que a métrica usada para avaliar popularidade pode ser falha neste aspecto. Na minha opinião, o número de estrelas de um projeto não é uma métrica com um bom atributo de sensibilidade, ou seja, ela não responde em um tempo tão satisfatório as alterações negativas de popularidade. Assim, projetos que foram sendo abandonados com o tempo podem continuar com muitas estrelas pelos mais diversos motivos, como simplesmente porque os usuários se esqueceram de retirar a estrela, não ligaram para isso, perderam suas contas, abandonam o GitHub, etc. Acredito que repositórios que já foram muito populares no passado e não o são mais podem continuar com muitas estrelas e, assim, aqueles considerados entre os mais populares podem ter um tempo muito grande desde a última atualização.

B. Metodologia: Para responder a esta questão, foi realizada uma consulta à API do GitHub usando GraphQL, através de um script escrito em Python. Neste primeiro momento, foram analisados os 100 repositórios mais populares da plataforma, ou seja, com as maiores quantidades de estrelas atribuídas. Para a filtragem desta consulta, foi utilizado o campo “latestRelease”, com o sub-campo “publishedAt” dos repositórios mais populares.

C. Resultados obtidos: O resultado encontrado foi uma **mediana de 20 dias desde o último lançamento de uma release**. Releases com 20 dias de lançamento podem ser consideradas como consideravelmente recentes.

D. Discussão sobre resultados esperados e obtidos: Os resultados obtidos refutam a hipótese inicialmente levantada.

5.

A. Hipótese: Acredito que sim, uma vez que projetos que utilizam linguagens populares são projetos que provavelmente terão mais usuários e mais colaboradores, o que aumenta a probabilidade de que o projeto se torne popular.

B. Metodologia: Para responder a esta questão, foi realizada uma consulta à API do GitHub usando GraphQL, através de um script escrito em Python. Neste primeiro momento, foram analisados os 100 repositórios mais populares da plataforma, ou seja, com as maiores quantidades de estrelas atribuídas. Para a filtragem desta consulta, foi utilizado o campo “primaryLanguage”, com o sub-campo “name” dos repositórios mais populares.

- C. Resultados obtidos:** Descobriu-se que **80% dos 100 repositórios mais populares do GitHub** apresentam como linguagem principal alguma das **10 linguagens mais populares do GitHub**.
- D. Discussão sobre resultados esperados e obtidos:** O resultado obtido confirma o que era esperado.

6. .

- A. Hipótese:** Acredito que sim, uma vez que projetos populares tipicamente são aqueles que apresentam muitos colaboradores, o que contribui positivamente para o percentual de issues fechadas.
- B. Metodologia:** Para responder a esta questão, foi realizada uma consulta à API do GitHub usando GraphQL, através de um script escrito em Python. Neste primeiro momento, foram analisados os 100 repositórios mais populares da plataforma, ou seja, com as maiores quantidades de estrelas atribuídas. Para a filtragem desta consulta, foi utilizado o campo “issues” com o modificador “states: CLOSED”, juntamente ao subcampo “totalCount”; além do campo “issues” com o subcampo “totalCount”.
- C. Resultados obtidos:** A mediana da razão entre issues fechadas e issues totais para os 100 repositórios mais populares do GitHub foi de **0,9085 (90,85%)**. Trata-se de um valor bastante expressivo.
- D. Discussão sobre resultados esperados e obtidos:** O resultado obtido confirma o que era esperado.

Prints dos Resultados:

```
Loop numero 99
{'name': 'opencv', 'stargazerCount': 80566, 'url': 'https://github.com/opencv/opencv', 'createdAt': '2012-07-19T09:40:17Z'}
Loop numero 100
{'name': 'gin', 'stargazerCount': 80296, 'url': 'https://github.com/gin-gonic/gin', 'createdAt': '2014-06-16T23:57:25Z'}
Lista das idades: [3704, 4144, 3871, 2473, 3252, 3175, 2893, 2909, 2152, 3884, 4283, 2865, 3546, 4218, 3135, 2519, 3387, 4911, 4108, 2558, 5648, 3939, 701, 4948, 3632, 3451, 5211, 2541, 2428, 3558, 2475, 907, 4487, 2300, 3876, 5220, 3053, 1822, 598, 3832, 2633, 3146, 3688, 802, 2641, 1859, 4326, 2115, 3905, 4120, 5253, 3732, 833, 3832, 5441, 3134, 3895, 2709, 2467, 5356, 851, 1572, 3208, 2743, 3801, 3832, 4058, 3584, 3002, 1870, 3620, 4079, 3841, 3343, 2283, 3600, 2837, 2043, 3243, 2860, 3107, 4032, 2731, 3255, 2688, 1557, 50, 4675, 5418, 1600, 2180, 5171, 3007, 4874, 3275, 2260, 772, 706, 4593, 3895]
Mediana das idades: 3247.5 dias
○ .venveduardobandeirang@Mac code %
```

```
99º Repositório mais popular do GitHub:
{'name': 'opencv', 'stargazerCount': 80568, 'url': 'https://github.com/opencv/opencv', 'createdAt': '2012-07-19T09:40:17Z', 'pullRequests': {'totalCount': 12711}}
100º Repositório mais popular do GitHub:
{'name': 'gin', 'stargazerCount': 80298, 'url': 'https://github.com/gin-gonic/gin', 'createdAt': '2014-06-16T23:57:25Z', 'pullRequests': {'totalCount': 1046}}
Lista das quantidades de pull requests aceitos: [24319, 6787, 661, 140, 1872, 415, 3314, 199, 1068, 543, 11340, 182, 2393, 1136, 2911, 302, 28995, 10, 367, 374, 3288, 281, 2575, 8842, 40813, 31084, 1573, 13, 122, 317, 1128, 1869, 811, 14574, 3392, 1639, 20775, 238, 1980, 1, 821, 466, 886, 158, 109, 6, 21462, 5061, 60659, 282, 491, 9724, 897, 934, 13665, 1949, 14532, 765, 12326, 61134, 13154, 2505, 17, 4022, 3694, 19844, 36231, 14947, 149, 3558, 2641, 433, 1450, 654, 6490, 481, 5133, 4789, 34, 311, 5998, 14571, 99, 10720, 2725, 497, 16, 9516, 725, 3308, 149, 14893, 5457, 315, 433, 2533, 470, 868, 12711, 1046]
Mediana dos pull requests aceitos: 1606.0
○ .venveduardobandeirang@Mac code %
```

```
98º Repositório mais popular do GitHub:
{'name': 'NextChat', 'stargazerCount': 80748, 'primaryLanguage': {'name': 'TypeScript'}}
99º Repositório mais popular do GitHub:
{'name': 'opencv', 'stargazerCount': 80597, 'primaryLanguage': {'name': 'C++'}}
100º Repositório mais popular do GitHub:
{'name': 'gin', 'stargazerCount': 80331, 'primaryLanguage': {'name': 'Go'}}
80.0% dos repositórios mais populares têm como linguagem principal alguma das 10 linguagens mais populares do GitHub
```

Ln 66, Col 28 Spaces: 4 UTF-8 LF {} Python 3.13.2 (.venv: venv)

```
99º Repositório mais popular do GitHub:
{'name': 'opencv', 'stargazerCount': 80577, 'latestRelease': {'publishedAt': '2025-01-09T10:25:23Z'}}
100º Repositório mais popular do GitHub:
{'name': 'gin', 'stargazerCount': 80317, 'latestRelease': {'publishedAt': '2024-05-07T03:48:51Z'}}
Lista das quantidades de dias desde a última release: [771, 71, 542, 418, 113, 2, 360, 483, 1, 6, 8, 1156, 4, 1206, 1, 41, 1, 1, 67, 2, 17, 2, 2294, 3, 39, 1, 71, 73, 15, 1039, 37, 2, 15, 2, 20, 19, 9, 2, 3, 184, 4, 371, 60, 1139, 1, 20, 1699, 16, 17, 0, 4, 171, 25, 30, 0, 1621, 15, 17, 96, 37, 284]
Quantidade de repositórios populares que lançaram releases: 60
Mediana da quantidade de dias desde a última release: 20.0
```

Ln 59, Col 54 Spaces: 2 UTF-8 LF {} Python 3.13.2 (.venv: venv)

```
100º Repositório mais popular do GitHub:
{'name': 'gin', 'stargazerCount': 80331, 'issuesTotais': {'totalCount': 2288}, 'issuesFechadas': {'totalCount': 1647}}
ISSUES TOTAIS: 2288
ISSUES FECHADAS: 1647
RAZÃO: 0.7198426573426573
Mediana das razões entre issues fechadas e issues totais dos 100 repositórios mais populares do GitHub: 0.9085364069712194
```

Ln 64, Col 32 Spaces: 4 UTF-8 LF {} Python 3.13.2 (.venv: venv)