

Hipótese:

Repositórios mais antigos têm maior número de estrelas acumulados mas possuem menor taxa de crescimento recente.

## 1. Introdução

Este trabalho tem como objetivo analisar as principais características de sistemas open-source populares hospedados no GitHub. Foram coletados dados de 1000 repositórios com maior número de estrelas, considerando aspectos como idade, contribuições externas, frequência de releases, atualizações recentes, linguagem primária e taxa de issues fechadas.

A motivação é entender como os projetos mais populares se comportam e quais fatores podem estar relacionados ao seu sucesso.

## 2. Hipóteses Informais

Antes da análise, levantamos as seguintes hipóteses:

- **H0:** Repositórios mais antigos têm maior número de estrelas acumulados mas possuem menor taxa de crescimento recente.
- **H1:** Existe correlação entre número de estrelas e o número de contribuidores no repositório.

## 3. Metodologia

### 1. Coleta de dados:

- Utilizamos a API GraphQL do GitHub.
- Foram coletados 1000 repositórios mais populares em número de estrelas.
- Para cada repositório, registramos:
  - Nome
  - Dono
  - Quantidade de estrelas
  - Linguagem primária
  - Idade
  - Número de releases

## 2. Análise:

- Para métricas numéricas, calculamos **mínimo, máximo, média e mediana**.
- 

## 4. Resultados

### RQ01 – Sistemas populares são maduros/antigos?

- **Métrica:** idade do repositório (anos desde a criação).
- **Mediana observada:** ~ **3051 dias**.

### RQ02 – Recebem muita contribuição externa?

- **Métrica:** número de pull requests aceitas.
- **Mediana observada:** ~ **702 pull requests aceitas**.

### RQ03 – Lançam releases com frequência?

- **Métrica:** número de releases.
- **Mediana observada:** ~ **35.5 releases** por repositório.

### RQ04 – São atualizados com frequência?

- **Métrica:** tempo desde a última atualização.
- **Mediana observada:** **0 dias** desde a última atualização.

### RQ05 – Linguagens mais utilizadas

- **Métrica:** linguagem primária.
- **Distribuição (top 5):**
  - Python - 21.07%
  - TypeScript - 17.39%
  - JavaScript - 14.49%
  - Go - 8.14%

- Java - 5.57%

## RQ06 – Percentual de issues fechadas

- **Métrica:** closed issues / total issues.
- **Mediana observada:** 85% de issues fechadas.

## Hipótese Informal 0

- Utilizado a fórmula de Pearson:  
$$r = (\Sigma (x_i - \bar{x})(y_i - \bar{y})) / \sqrt{(\Sigma (x_i - \bar{x})^2 * \Sigma (y_i - \bar{y})^2)}$$
- **A hipótese é suportada pelos dados:** projetos antigos tendem a acumular estrelas, mas o crescimento recente é mais forte em projetos novos.

## Hipótese Informal 1

- Utilizado a fórmula de Pearson:  
$$r = (\Sigma (x_i - \bar{x})(y_i - \bar{y})) / \sqrt{(\Sigma (x_i - \bar{x})^2 * \Sigma (y_i - \bar{y})^2)}$$
- A hipótese é **parcialmente confirmada**: há evidência de correlação, mas o efeito é fraco.

---

## 5. Discussão

- **RQ01:** A hipótese foi confirmada. A mediana de 3051 dias mostra que sistemas populares geralmente já possuem histórico consolidado.
- **RQ02:** Também confirmada. O número elevado de pull requests aceitas indica grande colaboração externa.
- **RQ03:** Confirmada parcialmente. Alguns sistemas realmente possuem releases frequentes (sobretudo frameworks), mas outros (como repositórios de aprendizado ou coleções de recursos) quase não lançam releases.
- **RQ04:** Confirmada. A maioria dos repositórios possui atividade recente (últimos dias/semanas).
- **RQ05:** Confirmada. JavaScript, Python e TypeScript aparecem como as linguagens mais comuns.

- **RQ06:** Confirmada parcialmente. Embora a mediana de 85% seja alta, alguns projetos apresentam grande número de issues abertas, o que pode refletir tanto alta demanda quanto dificuldades de manutenção.