Relatório de Métricas e Qualidade de Repositórios Java

# 1. Introdução

Este relatório apresenta uma análise exploratória sobre a relação entre métricas de qualidade de código e características de repositórios Java hospedados no GitHub. O objetivo é responder quatro questões principais (RQ01–RQ04), considerando popularidade, maturidade, atividade e tamanho dos projetos.  
  
Hipóteses iniciais (informais):  
- Repositórios mais populares (mais estrelas) teriam código de melhor qualidade (menos acoplado e mais coeso);  
- Repositórios mais antigos apresentariam maior complexidade, devido ao acúmulo de funcionalidades;  
- Repositórios mais ativos (com muitos releases) poderiam ter métricas de qualidade piores por evoluírem rápido;  
- Repositórios maiores (mais linhas de código) tenderiam a ter maior acoplamento e menor coesão.

# 2. Metodologia

Foram coletadas métricas de qualidade usando a ferramenta CK (Coupling Between Objects – CBO, Depth of Inheritance Tree – DIT, Lack of Cohesion of Methods – LCOM) e métricas de processo via GitHub (número de estrelas, idade do projeto em anos, número de releases e tamanho em linhas de código – LOC). Esses dados foram reunidos em um CSV unificado e analisados com Python, Pandas e Matplotlib. Foram gerados gráficos de distribuição (ECDF, boxplots) e correlação (scatterplots e hexbin).

# 3. Resultados e Discussão

## RQ01. Popularidade e Qualidade

Para avaliar a popularidade, utilizamos o número de estrelas (stars) de cada repositório. Ao comparar com métricas de qualidade (CBO, DIT, LCOM), observamos:  
- O scatter entre CBO e stars mostra uma tendência levemente negativa, sugerindo que projetos mais populares tendem a ter menor acoplamento;  
- O mesmo ocorre com DIT: repositórios populares em geral evitam heranças profundas;  
- A relação com LCOM é menos clara, mas a maioria dos projetos populares mantém valores razoáveis de coesão.  
  
Em resumo, repositórios com mais estrelas não apresentaram código mais complexo, o que indica que popularidade pode estar associada a boas práticas.

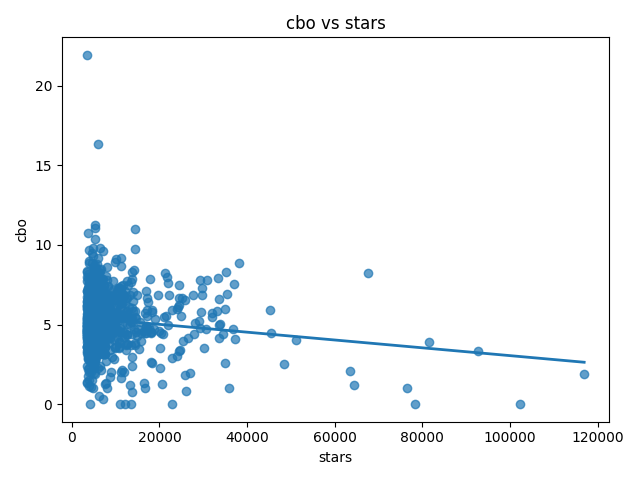


Figura: Relação entre CBO e popularidade (stars).

## RQ02. Maturidade e Qualidade

A maturidade foi medida pela idade em anos do repositório. Os resultados mostraram que:  
- O scatter entre CBO e idade não indica aumento de acoplamento com o tempo;  
- O scatter entre DIT e idade mostra uma leve tendência de aumento, sugerindo que projetos mais antigos podem acumular hierarquias de herança um pouco mais profundas;  
- Para LCOM, não há relação clara, embora alguns projetos antigos apareçam como outliers com baixa coesão.  
  
Assim, maturidade parece ter impacto limitado na qualidade interna, com exceção da herança, que tende a crescer com o tempo.

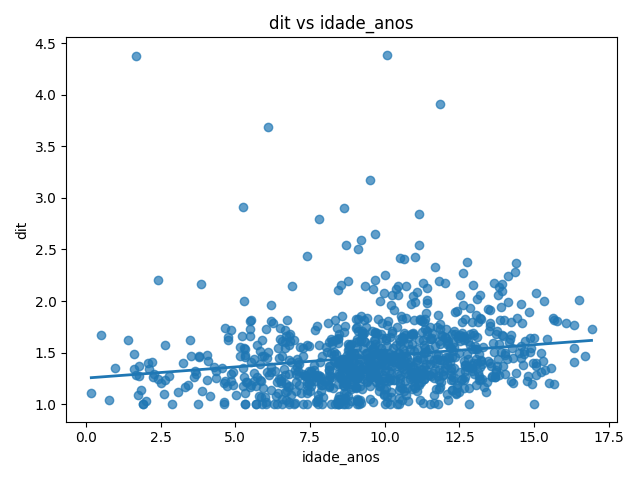


Figura: Relação entre DIT e idade do repositório.

## RQ03. Atividade e Qualidade

Atividade foi medida pelo número de releases. As análises indicaram que:  
- O scatter entre CBO e releases aponta uma tendência levemente positiva: projetos mais ativos podem acumular mais dependências;  
- DIT também cresce levemente com releases, sugerindo que a evolução contínua pode levar a hierarquias um pouco mais profundas;  
- LCOM apresenta grande dispersão: alguns projetos muito ativos apresentam baixa coesão em parte de suas classes.  
  
Concluímos que a atividade está associada a um aumento modesto da complexidade estrutural.

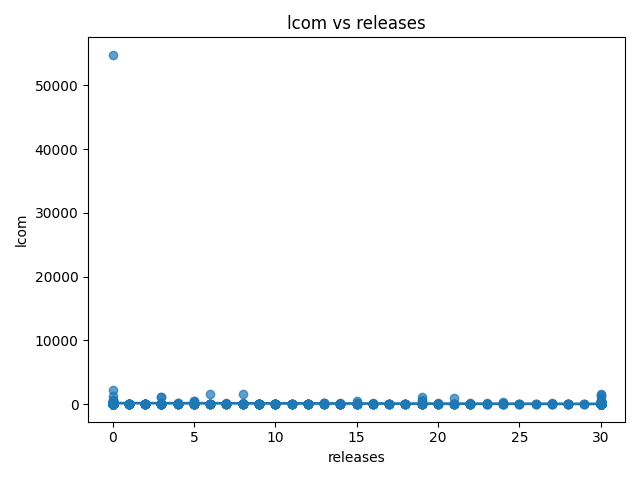


Figura: Relação entre LCOM e número de releases.

## RQ04. Tamanho e Qualidade

O tamanho dos repositórios foi medido pelo número de linhas de código (LOC). Aqui as correlações foram mais claras:  
- O scatter entre CBO e LOC mostra uma relação positiva: quanto maior o sistema, maior o acoplamento;  
- O scatter entre DIT e LOC indica que sistemas maiores tendem a ter hierarquias mais profundas;  
- LCOM também cresce em alguns casos, mostrando que classes grandes podem ser menos coesas.  
  
Portanto, o tamanho está diretamente relacionado ao aumento de complexidade interna, confirmando a hipótese inicial.

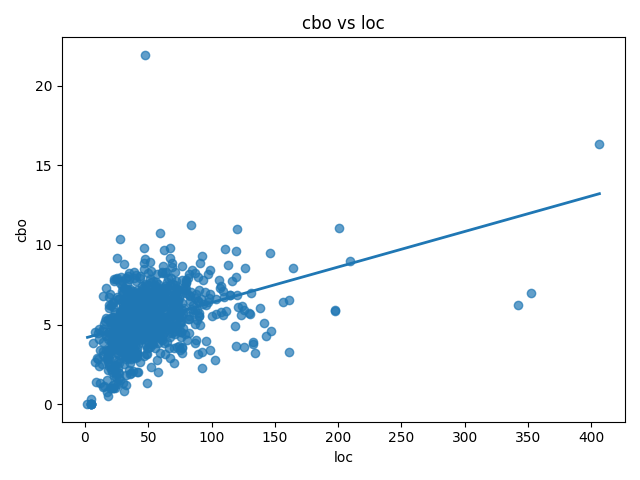


Figura: Relação entre CBO e tamanho do repositório (LOC).

# 4. Conclusão

De forma geral, os resultados mostraram que:  
- Popularidade está associada a projetos com métricas de qualidade razoáveis (baixo CBO e DIT);  
- Maturidade impacta pouco, exceto por heranças mais profundas em projetos antigos;  
- Atividade aumenta levemente o acoplamento e profundidade, mas não de forma drástica;  
- Tamanho se mostrou o fator mais fortemente ligado a piora na qualidade, com mais LOC levando a maior acoplamento e menor coesão.  
  
Esses achados reforçam que a complexidade tende a crescer com o tamanho e evolução contínua do software, mas popularidade e maturidade, por si só, não implicam em perda de qualidade.