

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**  
**ENGENHARIA DE SOFTWARE - CAMPUS LOURDES**  
**LABORATÓRIO DE EXPERIMENTAÇÃO DE SOFTWARE**

Ana Carolina Corrêa, Caio Elias,  
Henrique Diniz e Maria Clara Santos

**Um Estudo das Características de Qualidade de Sistemas Java**

**26 de setembro de 2024**

**Belo Horizonte**

## LISTA DE FIGURAS

Figura 01 - Distribuição da popularidade dos repositórios -----	7
Figura 02 - Distribuição da maturidade por qualidade-----	8
Figura 03 - Distribuição da produtividade por qualidade-----	9
Figura 04 - Distribuição de tamanho por qualidade -----	10

# SUMÁRIO

<b>1. Introdução.....</b>	<b>4</b>
<b>2. Hipóteses.....</b>	<b>4</b>
• Popularidade e qualidade (RQ 01):.....	4
• Maturidade e qualidade (RQ 02):.....	4
• Atividade e qualidade (RQ 03):.....	5
• Tamanho e qualidade (RQ 04):.....	5
<b>3. Metodologia.....</b>	<b>5</b>
<b>4. Resultados.....</b>	<b>5</b>
Figura 01 - Distribuição da popularidade dos repositórios.....	5
Figura 02 - Distribuição da maturidade por qualidade.....	6
Figura 03 - Distribuição da produtividade por qualidade.....	6
Figura 04 - Distribuição de tamanho por qualidade.....	6
<b>5. Discussão.....</b>	<b>6</b>
<b>6. Conclusão.....</b>	<b>6</b>

## 1. Introdução

No cenário atual de desenvolvimento de software open-source, a popularidade de um repositório no GitHub pode ser um indicador de seu impacto e relevância na comunidade. Entretanto, além da popularidade medida por estrelas, é fundamental entender se essa relevância está relacionada à qualidade do código produzido. Analisar repositórios populares e identificar padrões de qualidade permite extrair informações valiosas sobre como as práticas de desenvolvimento influenciam diretamente o sucesso de projetos.

Partindo desse ponto, este estudo de repositórios populares desenvolvidos em Java, tem como objetivo explorar a relação entre as suas características de qualidade interna e seus aspectos do processo de desenvolvimento. Para isso, será investigada a correlação entre fatores como popularidade, tamanho, atividade e maturidade dos repositórios e métricas de qualidade de software, como acoplamento, profundidade da árvore de herança e coesão dos métodos.

## 2. Hipóteses

### • Popularidade e qualidade (RQ 01):

Repositórios populares tendem a ter mais qualidade, caracterizada por um baixo acoplamento entre objetos (CBO), uma árvore de herança com profundidade moderada (DIT) e alta coesão de métodos (LCOM). Isso se deve à maior visibilidade desses projetos e à propensão para atrair desenvolvedores experientes que contribuem com boas práticas de codificação.

### • Maturidade e qualidade (RQ 02):

Repositórios mais antigos (com mais de cinco anos de idade) apresentam melhores métricas de qualidade, devido ao tempo disponível para refinamento do código e adoção de práticas de desenvolvimento mais sólidas. Repositórios maduros têm uma estrutura de código mais estável e bem organizada, refletindo um processo de desenvolvimento evolutivo.

### • Atividade e qualidade (RQ 03):

Repositórios com alta atividade de desenvolvimento, caracterizada por um número significativo de releases e contribuições constantes, apresentam melhor qualidade de código. Isso pode indicar um processo de revisão e atualização contínuo, o que contribui para a redução de problemas de coesão e acoplamento.

- **Tamanho e qualidade (RQ 04):**

Repositórios menores em termos de linhas de código (LOC) apresentam melhores métricas de qualidade, pois são mais fáceis de gerenciar e manter. Entretanto, repositórios grandes podem ter qualidade igualmente alta se forem bem modularizados, sugerindo uma relação complexa entre tamanho e qualidade de código.

### **3. Metodologia**

Para realizar a análise dos repositórios populares no GitHub, seguimos uma abordagem sistemática que envolveu a coleta e processamento de dados utilizando a API GraphQL do GitHub. Abaixo, descrevemos as etapas principais do processo:

### **4. Resultados**

A análise dos 1000 repositórios populares do GitHub gerou insights valiosos sobre as características desses projetos. A seguir, apresentamos os resultados obtidos para cada uma das questões de pesquisa formuladas.

**RQ 01: Repositórios populares tendem a apresentar melhores características de qualidade?**

**Figura 01 - Distribuição da popularidade dos repositórios**

**RQ 02. Repositórios mais maduros possuem uma qualidade de código superior?**

**Figura 02 - Distribuição da maturidade por qualidade**

**RQ 03. Repositórios com alta atividade de desenvolvimento apresentam melhores métricas de qualidade?**

**Figura 03 - Distribuição da produtividade por qualidade**

**RQ 04. O tamanho dos repositórios afeta a qualidade do código?**

**Figura 04 - Distribuição de tamanho por qualidade**

## **5. Discussão**

Os resultados obtidos

## **6. Conclusão**