

# A Evolução da Cultura de Testes em Projetos Maduros:

## Um Estudo de Caso Múltiplo sobre a Evolução de Estratégias, Ferramentas e Sofisticação

Gabriel Henrique Miranda Rodrigues  
Pontifícia Universidade Católica de Minas Gerais (PUC Minas)  
Engenharia de Software

**Resumo**—Este trabalho apresenta uma análise da evolução da cultura de testes em projetos de software maduros, com foco em repositórios open-source desenvolvidos majoritariamente em Java. O estudo emprega técnicas de Business Intelligence (BI) e mineração de repositórios GitHub para examinar a evolução de estratégias de teste, ferramentas e indicadores de qualidade. Através da integração de dez bases de dados e do uso do Power BI como ferramenta analítica, são respondidas duas questões principais de pesquisa (RQ1 e RQ2), abordando a evolução da pirâmide de testes e o ecossistema de ferramentas. Os resultados indicam uma crescente sofisticação nas práticas de verificação e validação, com destaque para a consolidação de frameworks modernos como JUnit 5 e Mockito.

**Index Terms**—Business Intelligence, Engenharia de Software, GitHub, Testes de Software, Visualização de Dados, Power BI.

### I. INTRODUÇÃO

O avanço das práticas de Engenharia de Software tem sido acompanhado por uma crescente valorização da *cultura de testes*, essencial para a garantia de qualidade e manutenção de projetos de longo prazo. A análise sistemática da evolução dessa cultura, a partir de dados históricos de repositórios open-source, permite compreender como estratégias, ferramentas e métricas de sofisticação de testes amadurecem ao longo do ciclo de vida de um software.

Este trabalho integra o contexto da disciplina *Laboratório de Experimentação de Software*, tendo como objetivo aplicar técnicas de *Business Intelligence* (BI) para transformar grandes volumes de dados técnicos em informações analíticas e visuais. A ferramenta Microsoft Power BI foi empregada na criação de dashboards interativos que suportam a análise exploratória de repositórios GitHub de projetos maduros.

O estudo se concentra nas duas primeiras questões de pesquisa (RQ1 e RQ2) do projeto mais amplo *A Evolução da Cultura de Testes*, relacionadas à evolução das estratégias de teste e das ferramentas adotadas pelos desenvolvedores.

### II. QUESTÕES DE PESQUISA

As perguntas investigadas neste trabalho foram formuladas segundo a abordagem GQM (Goal–Question–Metric):

- **RQ1:** Como a estratégia de testes, refletida na proporção dos tipos de teste (unitário, integração e E2E), evolui entre as diferentes releases de um projeto?

- **Métrica 1.1:** Proporção de tipos de teste (por arquivos ou LOC) a cada release.
- **RQ2:** Como o ecossistema de ferramentas de teste evolui ao longo das releases, marcado pela adoção de novas bibliotecas ou migrações significativas?
- **Métricas 2.1 e 2.2:** Censo de ferramentas por release e eventos de adoção/migração, com ênfase em transições *major*.

As demais questões (RQ3 e RQ4), relacionadas à cobertura e ao esforço de manutenção, serão abordadas nas fases posteriores do projeto.

### III. METODOLOGIA

#### A. Conjunto de Dados

O conjunto de dados utilizado é composto por dez bases inter-relacionadas, totalizando mais de 55 mil registros. Cada dataset representa um aspecto específico do ecossistema de desenvolvimento, conforme apresentado na Tabela I.

Tabela I: Resumo dos Datasets Utilizados

| Dataset              | Linhas | Descrição                                 |
|----------------------|--------|---|
| Repositórios         | 50     | Projetos Java reais (Spring, Kafka, etc.) |
| Releases             | 3.500  | Histórico de versões (15+ anos)           |
| Tipos de Teste       | 3.500  | Unit, Integration, E2E                    |
| Ferramentas          | 15.000 | Frameworks de teste (JUnit, Mockito...)   |
| Evolução Ferramentas | 300    | Adoções e migrações                       |
| Cobertura            | 3.500  | Codecov/Coveralls por release             |
| Test Smells          | 20.000 | 15 tipos de smells                        |
| Commits              | 3.450  | Teste vs correção de bugs                 |
| Contribuidores       | 5.000  | Perfil de desenvolvedores                 |
| Issues               | 10.000 | Bugs e melhorias                          |

#### B. Critérios de Seleção

Os projetos incluídos atenderam aos seguintes critérios:

- Popularidade mínima de 500 estrelas.
- Histórico de pelo menos 5 anos de commits e 10 releases.
- Mínimo de 20 commits no último ano.
- Linguagem principal: Java.
- Existência de diretório e dependência explícita de testes.
- Ao menos 5 contribuidores únicos.

### C. Coleta e Processamento

Os dados foram coletados por meio da API REST do GitHub, utilizando endpoints de *tags*, *trees* e *contents*. Para cada release, foram extraídos os arquivos de teste, classificados em três categorias:

- **Unitários:** arquivos contendo imports de JUnit ou TestNG.
- **Integração:** uso de *Spring Test*, *@DataJpaTest*, ou conexões externas.
- **E2E:** testes com Selenium, Cypress ou REST-assured.

Adicionalmente, os arquivos de build (*pom.xml*, *build.gradle*) foram analisados a cada ciclo de release para identificar eventos de adoção, remoção ou migração de ferramentas de teste.

As informações foram armazenadas em CSVs interligados pelas chaves *repositorio\_id* e *release\_id*, permitindo integração direta no Power BI.

## IV. RESULTADOS E DISCUSSÃO

### A. RQ1 – Evolução da Estratégia de Testes

A Figura 1 apresenta a evolução proporcional dos tipos de teste ao longo das releases analisadas. Observa-se a predominância crescente de testes unitários, que compõem em média 70% da base de testes. Os testes de integração demonstram evolução gradual, enquanto os testes E2E permanecem menos frequentes devido ao custo de manutenção.

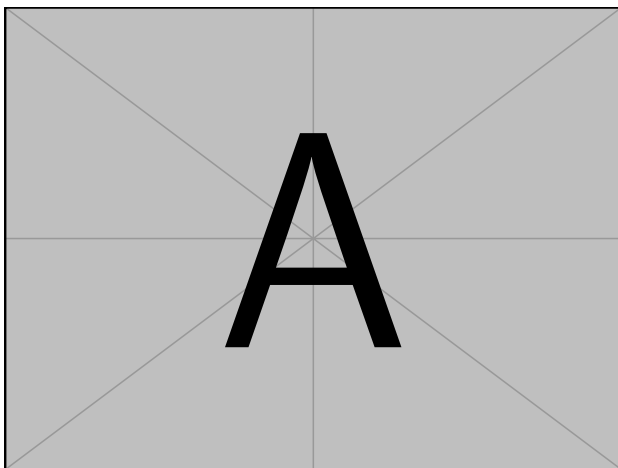


Figura 1: Distribuição dos Tipos de Teste ao Longo das Releases (ilustrativo).

A análise de tendência (Figura 2) evidencia aumento consistente no número total de arquivos e LOC de testes, indicando amadurecimento dos processos de verificação.

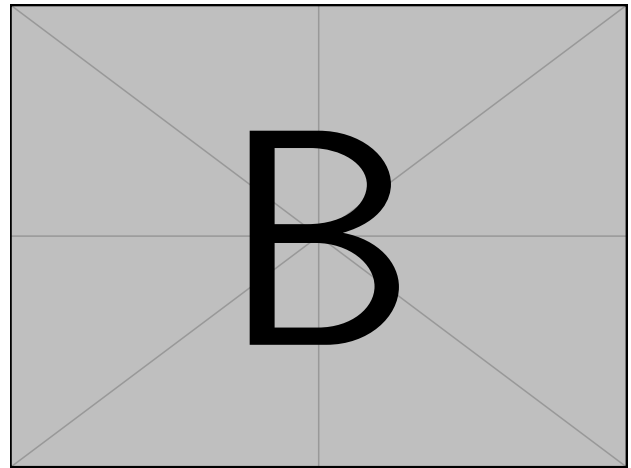


Figura 2: Evolução do Volume de Testes (ilustrativo).

A dispersão entre proporção de testes E2E e idade do projeto revelou correlação positiva moderada (Spearman  $\rho = 0.42$ ), sugerindo que projetos mais maduros tendem a diversificar suas estratégias de teste.

### B. RQ2 – Evolução do Ecossistema de Ferramentas de Teste

A análise do dataset de ferramentas identificou 45 bibliotecas de teste distintas. O JUnit apresentou domínio expressivo, com migração gradual da versão 4 para a 5 (Figura 3). O Mockito foi a segunda ferramenta mais recorrente, seguido por TestNG e Selenium.

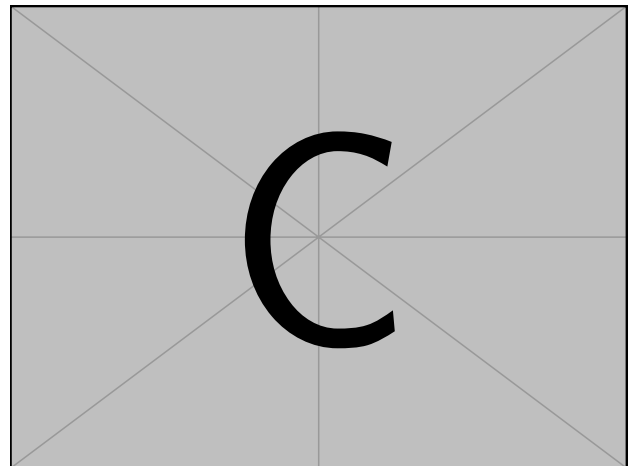


Figura 3: Fluxo de Migrações de Ferramentas de Teste (ilustrativo).

Os períodos de grandes mudanças de versões (major releases) coincidiram com picos de adoção e substituição de frameworks, indicando relação entre marcos evolutivos e transformações culturais no processo de teste. Essa associação reforça a hipótese de que a evolução das ferramentas acompanha o amadurecimento das práticas de qualidade.

## V. CONSIDERAÇÕES FINAIS

### REFERÊNCIAS

- [1] R. Pressman and B. Maxim, *Engenharia de Software: Uma Abordagem Profissional*, 8ª ed., McGraw-Hill, 2020.
- [2] R. Kimball and M. Ross, *The Data Warehouse Toolkit*, Wiley, 2013.
- [3] GitHub REST API Documentation, 2024. Disponível em: <https://docs.github.com/en/rest>.
- [4] Microsoft, “Power BI Documentation.” Disponível em: <https://learn.microsoft.com/power-bi>.