

# **Relatório Final - Laboratório 03**

**Gabriel Alejandro Figueiro Galindo<sup>1</sup>, Marcelo Aguilar Araújo D’Almeida<sup>1</sup>,  
Philippe Roberto Dutra Chaves Vieira<sup>1</sup>**

<sup>1</sup>Instituto de Ciências Exatas e Informática  
Pontifícia Universidade de Minas Gerais (PUC Minas)  
Belo Horizonte – MG – Brasil

## **1. Introdução e Hipóteses**

A prática de code review envolve a interação entre desenvolvedores e revisores para inspecionar o código produzido, garantindo sua integração à base principal sem riscos de introduzir novos erros à aplicação. Em sistemas open source, especialmente nos desenvolvidos no GitHub, esse processo ocorre por meio da avaliação de contribuições submetidas via Pull Requests (PRs). Para que um código seja incorporado à branch principal, é necessário que um colaborador do projeto analise e discuta a solicitação, decidindo por aprovar ou rejeitar o merge.

Neste relatório, o objetivo é analisar a atividade de code review em repositórios populares do GitHub, identificando variáveis que influenciam a aprovação de um PR sob a perspectiva dos desenvolvedores que contribuem para os projetos selecionados. Com base nas questões de pesquisa propostas, foram formuladas as seguintes hipóteses informais:

- RQ 01: PRs grandes têm mais chances de serem rejeitados, pois são mais difíceis de revisar e podem acabar introduzindo mais problemas no repositório. PRs pequenos costumam ser mais bem recebidos e aprovados com mais facilidade.
- RQ 02: PRs que levam mais tempo para serem analisados tendem a ser rejeitados, pois o atraso pode indicar dúvidas ou problemas detectados pelo revisor.
- RQ 03: PRs com descrições mais longas e detalhadas tendem a ser aprovados, pois facilitam o entendimento do propósito da mudança e demonstram cuidado por parte do autor.
- RQ 04: PRs com muitas interações (comentários e participantes) tendem a ser rejeitados, pois isso pode indicar divergência entre os participantes, em vez de um consenso.
- RQ 05: PRs maiores provavelmente passam por mais revisões, já que o volume de código aumenta a complexidade da análise, o que pode levar a um número maior de ajustes.
- RQ 06: PRs que levam mais tempo para serem analisados tendem a ter mais revisões, pois o tempo está sendo usado para discutir ou refinar as mudanças propostas.
- RQ 07: PRs com descrições curtas tendem a passar por mais revisões, pois os revisores precisarão de mais iterações para entender o que foi feito. Por outro lado, descrições mais longas podem reduzir o número de revisões necessárias.
- RQ 08: Um maior número de interações tende a resultar em mais revisões, já que cada troca de mensagens pode levar a novas versões do código até que todas as sugestões ou objeções sejam resolvidas.

Após a coleta de dados, essas hipóteses foram analisadas e discutidas para verificar a sua validade.

**2. Metodologia**

**3. Resultados Obtidos**

**4. Conclusão**