

Caracterizando a Atividade de Code Review no GitHub

**Bruno Gomes Ferreira, João Pedro Mairinque de Azevedo,
Matheus Vieira dos Santos, Marcio Lucas Machado Pereira**

¹ Instituto de Ciências Exatas e Informática (ICEI)
Pontifícia Universidade Católica de Minas Gerais
Engenharia de Software
Belo Horizonte – MG – Brazil

1. Introdução

A prática de Code Review é uma constante nos processos de desenvolvimento ágeis. Envolve a interação entre desenvolvedores e revisores para inspecionar o código antes de integrá-lo à base principal. Isso garante a qualidade do código, respeito dos padrões pré-estabelecidos pelo projeto e evita a inclusão de defeitos. No contexto de sistemas open source, especialmente aqueles hospedados no GitHub, as atividades de Code Review ocorrem por meio de Pull Requests (PRs). Antes de um código ser integrado à branch principal, é necessário criar uma solicitação de pull, que será avaliada e discutida por um colaborador do projeto. Ao final desse processo, a solicitação de merge pode ser aprovada ou rejeitada pelo revisor. [Chen et al. 2021] Além disso, práticas como Continuous Integration (CI) e Continuous Delivery (CD) são amplamente adotadas na Engenharia de Software, elas orientam os desenvolvedores a automatizar testes, builds e deployment com o auxílio de ferramentas específicas. Essa abordagem contribui para a eficiência e qualidade do processo de desenvolvimento.

Ao conduzir uma revisão de código, é essencial garantir que o código atenda a padrões elevados de qualidade, manutenibilidade e funcionalidade. [Github 2024] De acordo com a documentação oficial do GitHub para Code Review, os desenvolvedores ao revisar um código devem verificar se o design geral do código está sendo mantido, verificando se a arquitetura está alinhada com os objetivos do projeto em termos de usabilidade e experiência do usuário. Além disso, deve-se atentar a existência de fatores essenciais como testes unitários do que está sendo proposto, nomenclatura clara das variáveis e uma complexidade coerente com o que está sendo proposto.

Para a caracterização do Code Review traçou-se duas áreas de estudo nas quais o grupo criou questões de pesquisa: Em relação ao feedback final das revisões, ou seja, o status da PR que foi aberta, se ela foi integrada a base principal de código ou não. E também perguntas de pesquisa em relação ao número de revisões feitas em uma PR.

Em relação ao feedback final das revisões, traçou-se as seguintes perguntas de pesquisa:

- RQ 01. Qual a relação entre o tamanho dos PRs e o feedback final das revisões?
- RQ 02. Qual a relação entre o tempo de análise dos PRs e o feedback final das revisões?
- RQ 03. Qual a relação entre a descrição dos PRs e o feedback final das revisões?
- RQ 04. Qual a relação entre as interações nos PRs e o feedback final das revisões?

Para os autores, como hipóteses informais, os repositórios com PRs grandes tendem a ter um feedback negativo visto que serão revisões mais criteriosas, pois normalmente aqueles que irão revisar são pessoas que tem um conhecimento maior do código. O tempo de revisão é proporcional ao critério imposto para revisar aquela PR, Pull Requests mais críticas terão um tempo de revisão maior. A descrição da PR é um fator que facilita a revisão, por isso tende a ter melhores feedbacks. PRs que tem uma maior interação costumam ter um melhor feedback final das revisões já que tem uma maior discussão entre os desenvolvedores

Para as questões relacionadas ao número de revisões, foram traçadas as seguintes perguntas:

- RQ 05. Qual a relação entre o tamanho dos PRs e o número de revisões realizadas?
- RQ 06. Qual a relação entre o tempo de análise dos PRs e o número de revisões realizadas?
- RQ 07. Qual a relação entre a descrição dos PRs e o número de revisões realizadas?
- RQ 08. Qual a relação entre as interações nos PRs e o número de revisões realizadas?

PRs grandes tendem a ter uma quantidade menor de revisões pois será revisado principalmente por quem tem uma maior familiaridade com o projeto. O tempo de análise de uma PR é inversamente proporcional ao número de revisões, pois PRs que demandam maior tempo de revisão normalmente tem um menor engajamento. PRs com descrição detalhada tendem a atrair mais revisões, pois o revisor terá um contexto do motivo daquela PR ser feita. Finalmente, PRs com maior número de interações terão uma quantidade maior de revisões

Com o intuito de responder as perguntas de pesquisa, foram utilizadas métricas de qualidade para caracterizar os Code Reviews realizados na plataforma do GitHub. Para este trabalho, foram selecionadas as seguintes métricas de qualidade:

- Tamanho: número de arquivos; total de linhas adicionadas e removidas.
- Tempo de Análise: intervalo entre a criação do PR e a última atividade (fechamento ou merge).
- Descrição: número de caracteres do corpo de descrição do PR (na versão markdown).
- Interações: número de participantes; número de comentários.

Essas métricas serão correlacionadas com as perguntas de pesquisa realizadas inicialmente a fim de traçar as características dos Pull Requests

2. Metodologia

Com o objetivo de obter características de qualidade de Pull Requests, os autores seguiram a metodologia de pesquisas quantitativas onde através dos valores numéricos obtidos, será possível determinar as respostas das perguntas de pesquisa. Para a obtenção dos dados necessários foi utilizado as seguintes ferramentas: Python e GraphQL para obter os 200 repositórios mais populares. Para obter as métricas dos repositórios. O passo a passo da estratégia utilizada pode ser resumida pelo diagrama de fluxo (Figura 1)

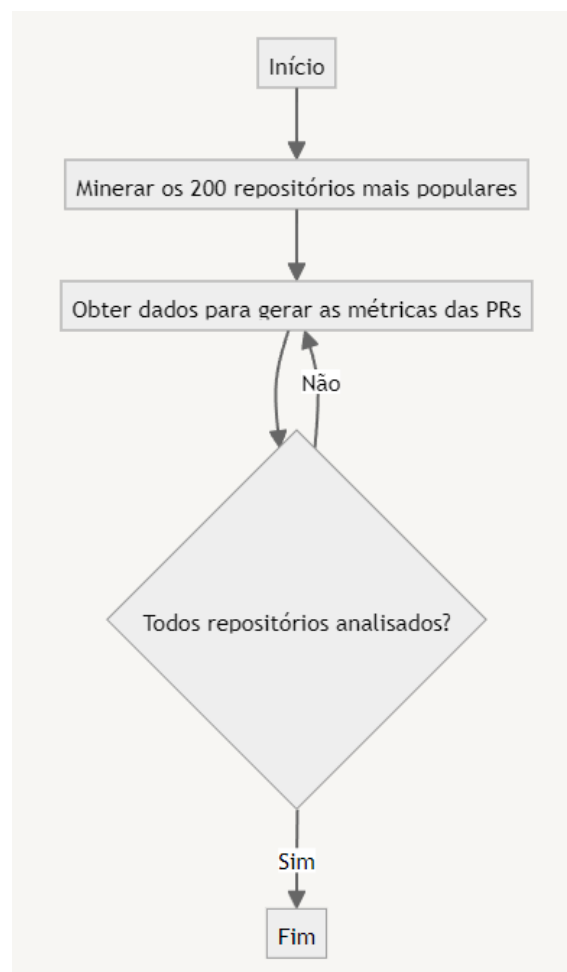


Figure 1. Diagrama de fluxo para obtenção dos resultados

Inicialmente, foram minerados os 200 repositórios mais populares no Github através do GraphQL. A linguagem de programação Python que é utilizada para desenvolver o script que fez a requisição da query na api do Github. Nessa query foi gerado um dataset em CSV contendo valores das métricas necessárias para a análise (tamanho, tempo de análise, descrição, interações) e suas relações com o status final do PR e o número total de revisões realizadas.

A requisição foi realizada de forma que nos assegurássemos que todos os 200 repositórios foram analisados, e que a obtenção de dados se repetisse até que a aquisição do objetivo fosse visível.

References

- Chen, T., Zhang, Y., Chen, S., Wang, T., and Wu, Y. (2021). Let's supercharge the workflows: An empirical study of github actions. In *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, page 01–10.
- Github (2024). **What to look for in a code review.** <https://google.github.io/eng-practices/review/reviewer/looking-for.html>
Acesso em: 17/04/2024.