

Caracterizando a atividade de code review no github

Alfredo Luis Vieira, Vinicius Salles

15 de outubro de 2025

1 Introdução

A prática de revisão de código (code review) é um pilar fundamental nos processos de desenvolvimento de software modernos, especialmente em metodologias ágeis. Ela consiste na inspeção do código-fonte por outros membros da equipe antes de sua integração à base principal do projeto, garantindo a qualidade do que é produzido e prevenindo a introdução de novos defeitos.

No ecossistema de desenvolvimento de software de código aberto (open source), plataformizado em grande parte pelo GitHub, a atividade de revisão de código se materializa por meio das Pull Requests (PRs). Desenvolvedores submetem suas contribuições através de PRs, que são então avaliadas e debatidas por outros colaboradores do projeto. Ao final desse processo iterativo, a contribuição pode ser aceita (merged) e integrada ao código principal, ou rejeitada (closed).

Este estudo tem como objetivo principal analisar e caracterizar a atividade de code review em repositórios populares do GitHub. Buscamos identificar as variáveis que influenciam tanto o resultado final de uma Pull Request (seja ela aceita ou rejeitada) quanto a intensidade do processo de revisão, medido pelo número de revisões que uma PR recebe. A análise será conduzida sob a perspectiva de desenvolvedores que submetem suas contribuições, investigando como fatores como o tamanho da contribuição, o tempo de análise, a qualidade da descrição e o nível de interação podem se correlacionar com o desfecho do processo de revisão.

2 Metodologia

A construção do dataset foi realizada através da funcionalidade de busca avançada do GitHub. Primeiramente, elaboramos uma consulta estruturada com filtros para selecionar Pull Requests (PRs) que atendiam aos critérios da pesquisa, como status (merged ou closed) e popularidade do repositório, esta estratégia se difere da utilização da API GraphQL do Github, nesse caso, foi utilizado uma query de busca com alguns filtros (status merged ou closed, e que possuía uma revisão), para que a pesquisa seja realizada. Devido a massa de dados ser muito grande, foi necessário deixar uma máquina executando a busca particionada durante o período da noite. Tentamos utilizar a API GraphQL do github, mas ela impossibilitava buscas mais específicas nas prs, como prs que passaram por revisão, por isso, optamos por utilizar a estratégia da query de pesquisa.

Com os dados obtidos, foram implementados scripts para analisar estes dados, estes scripts foram implementados utilizando python como linguagem de programação, esta análise esta voltada a dados acerca das reviews da pr, como comentários, descrição e etc.

Foi utilizado a biblioteca Pandas do python para realizar a criação de gráficos para melhor visualização dos dados obtidos.

Questões de pesquisa

Este trabalho de pesquisa tem como objetivo analisar o processo de revisão de código em projetos de software, utilizando um dataset de Pull Requests (PRs). A investigação busca correlações entre características dos PRs como tamanho, tempo de análise, qualidade da descrição e número de interações, e dois resultados principais. O primeiro é o feedback final da revisão (status do PR), e o segundo é a quantidade de revisões que um PR recebe, a fim de entender melhor a dinâmica das revisões de código.

As questões de pesquisa são:

Feedback Final das Revisões (Status do PR):

- **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?

Número de Revisões:

- **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?

3 Métricas utilizadas

Para investigar as questões de pesquisa propostas, o trabalho define um conjunto de métricas específicas que quantificam cada uma das dimensões de análise. Essas métricas são essenciais para a coleta de dados e para a subsequente análise de correlação, permitindo que os conceitos abstratos de "tamanho", "tempo", "descrição" e "interação" sejam medidos de forma consistente em todo o dataset de Pull Requests.

As métricas definidas são:

- **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.

4 Resultados obtidos

4.1 Qual a relação entre o tamanho dos PRs e o feedback final das revisões?

Nesta questão tentamos encontrar alguma correlação entre o tamanho dos PRs (em LOCs) e o feedback final, no boxplot abaixo podemos observar uma diferença entre PRs com status de merged e PRs com status de closed, podemos observar que PRs com status merged, possuem mais linhas de código. Esta visualização pode ser justificada por diversos motivos, sendo eles, que PRs maiores podem representar features completas, mudanças significativas, ou correções de bugs complexos, o que tende a ter uma aceitação em repositórios. Enquanto PRs fechadas, possuem menos linhas de código, o que pode indicar correções mais simples, modificações mais futeis que não agregam muito ao sistema, ou incrementos de baixa qualidade, o que pode levar a rejeição nos repositórios. Falando em numeros, podemos dizer que PRs merged são 61.9% maiores que PRs fechadas, e a mediana de PRs aceitos (68 linhas) é superior à de PRs rejeitados (42 linhas).

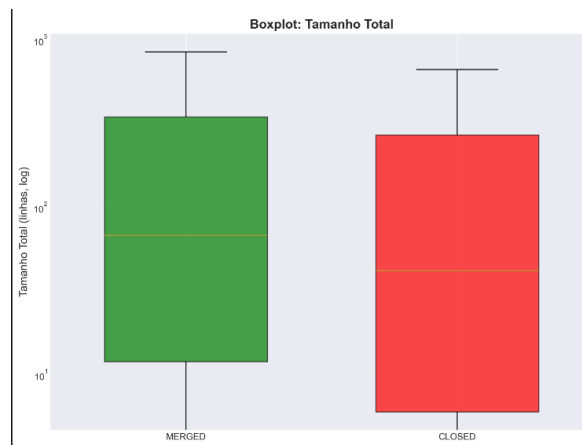


Figura 1: Tamanho x Feedback final

4.2 Qual a relação entre o tempo de análise dos PRs e o feedback final das revisões?

Ao analisar o tempo de análise dos PRs e o feedback final deles, podemos observar uma correlação a partir do afágico observado abaixo. podemos inferir que PRs que foram fechadas demoram mais tempo em sua análise do que PRs que foram mergeadas. em termos de numeros, podemos identificar que PRs fechadas possuem sua mediana em 4,6 dias A MAIS do que a mediana de PRs mergeadas. Este fenômeno pode ser explicado pelo fato de PRs merged possuem mais qualidade, e isto diminui o tempo de correção, seja pela ausência de código excessivamente complexo, e que PRs que foram fechadas, passaram por tentativas de correção, o que gasta mais tempo de análise e etc, podemos dizer que esta análise gira entorno da qualidade das PRs, o que influencia diretamente no tempo gasto em análise.

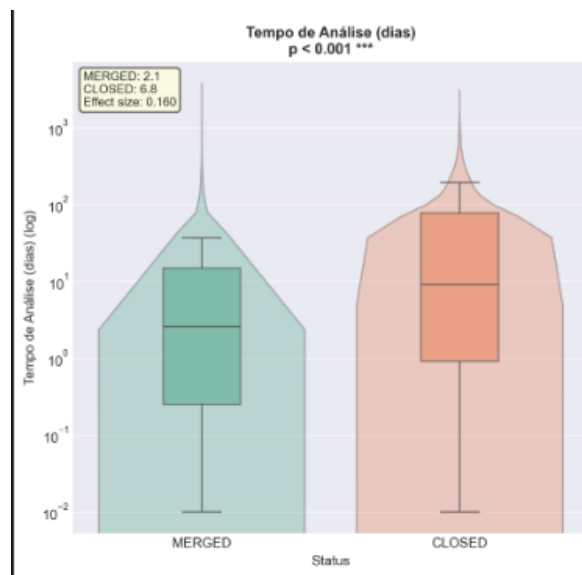


Figura 2: Tempo de análise x Feedback final

4.3 Qual a relação entre a descrição dos PRs e o feedback final das revisões?

Ao analisar o boxplot abaixo, podemos observar uma correlação direta entre o tamanho da descrição de uma PR e seu feedback final, podemos identificar que PRs com descrição maiores possuem maiores níveis de merged, em numeros, podemos dizer que PRs merged possuem descrição 20% maiores que PRs closed, isso pode ser justificado por alguns fatos, como uma boa documentação pode facilitar

o entendimento da PR, o que ela propõe, como ela está estruturada e etc, o que consequentemente, aumenta as chances da PR ser mergeada junto ao repositório.

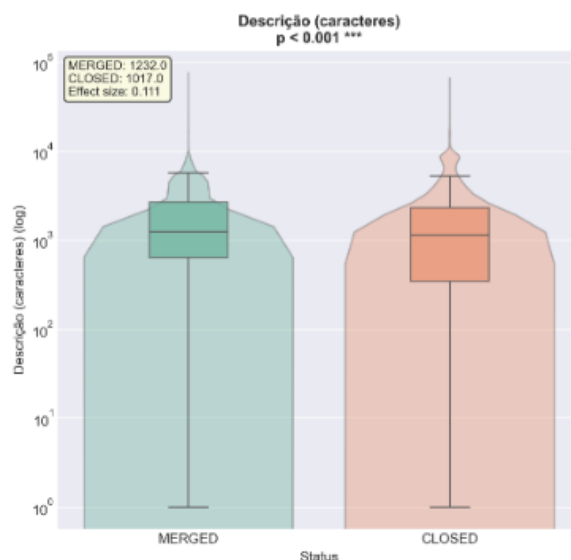


Figura 3: Descrição x Feedback final

4.4 Qual a relação entre as interações nos PRs e o feedback final das revisões?

Para responder esta questão, consideramos comentários e participantes como interações em PRs, visto que os comentários são as principais formas de interagir com membros relacionados as PRs, pontuando questionamentos, considerações e etc. A partir da análise representada pelo boxplot abaixo, podemos identificar que PRs closed possuem um nível de comentários e participantes ligeiramente maior que PRs merged, esta conclusão pode se basear que PRs fechadas tendem a gerar mais debate, controvérsia e feedback negativo que precisa ser documentado nos comentários antes de a proposta ser finalmente fechada, já o numero de participantes pode ser justificado pela justificativa de que PRs fechadas tendem a possuir análises mais complexas, devido a natureza do código, na maioria dos casos, estas análises podem envolver um numero maior de participantes que irão interagir na validação

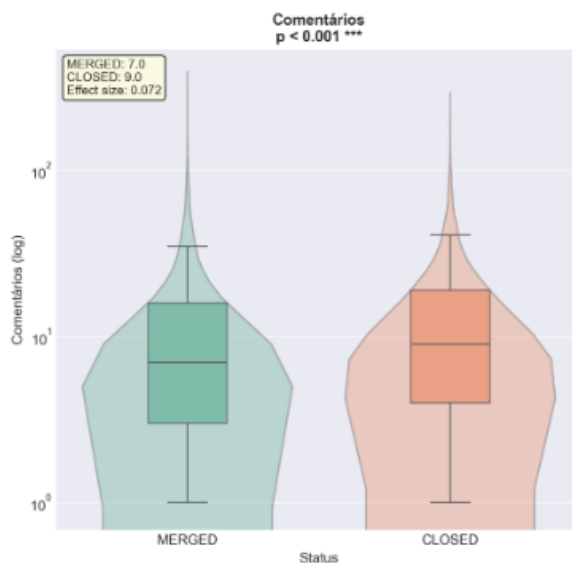


Figura 4: Comentários x Feedback final

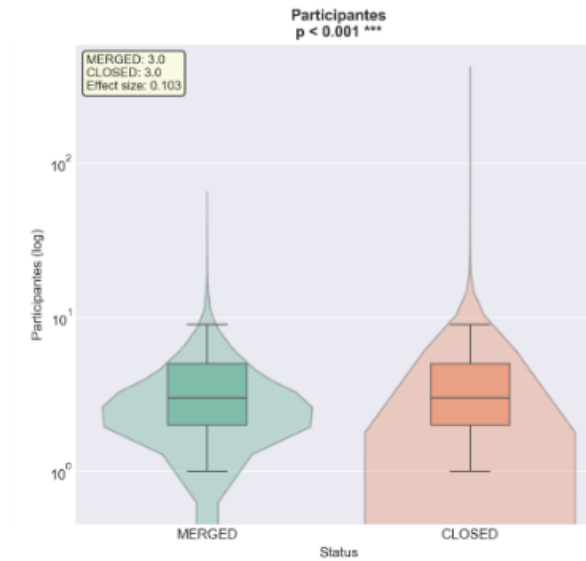


Figura 5: Participantes x Feedback final

4.5 Qual a relação entre o tamanho dos PRs e o número de revisões realizadas?

Ao analisar o número de revisões e os respectivos status das PRs podemos identificar que PRs maiores possuem um número maior de revisões, isso pode ser explicado pela complexidade elevada que PRs muito grandes podem possuir, para uma boa análise, deve-se revisar o código com atenção, utilizando técnicas de code-review, isso faz com que alterações maiores necessitam de mais revisões, também deve-se atentar ao fato de que revisões maiores introduzem um maior risco ao sistema, o que reforça a necessidade de revisar mais vezes, também existe a sobrecarga cognitiva que os revisores podem passar para revisar grandes quantidades de código, o que os induz a dividir este trabalho. Segue abaixo um gráfico exibindo visualmente estas informações:

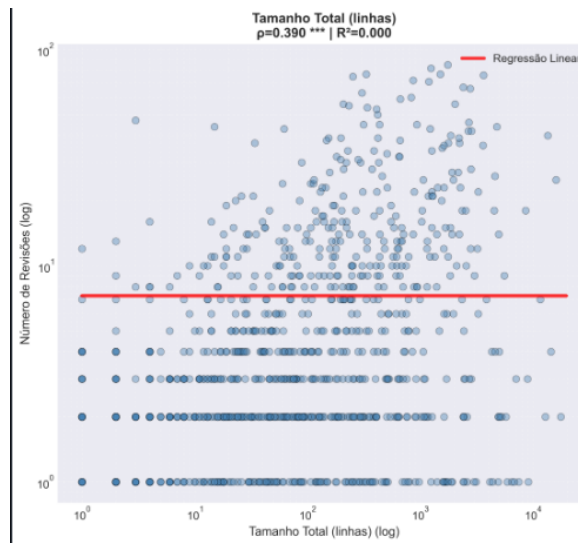


Figura 6: Tamanho x numero de revisões

4.6 Qual a relação entre o tempo de análise dos PRs e o número de revisões realizadas?

Após a análise do dataset, podemos identificar uma tendência de que o número de revisões impacta no tempo de análise, pois revisões podem gastar mais tempo, visto que análises demandam tempo e para estas validações aumentam o tempo gasto, segue abaixo um gráfico exibindo esta correlação

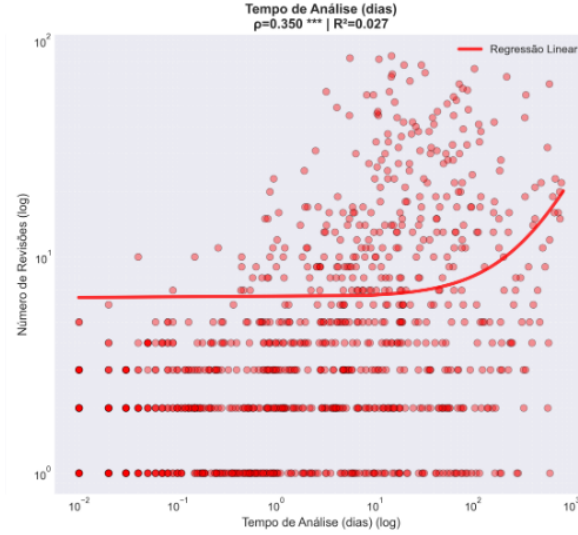


Figura 7: Tempo de análise x numero de revisões

4.7 Qual a relação entre a descrição dos PRs e o número de revisões realizadas?

Após a análises podemos identificar que descrições marginalmente mais longas estão associadas a um leve aumento no número de revisões, e não a uma diminuição como hipotetizado inicialmente. A conclusão principal é que a complexidade do PR é o fator subjacente que influencia tanto o tamanho da descrição quanto a quantidade de revisões necessárias. Ou seja, PRs mais complexos naturalmente exigem descrições mais detalhadas para explicar as mudanças e, por sua natureza intrincada, também demandam um maior número de ciclos de revisão. Portanto, a hipótese de que uma descrição detalhada reduziria as revisões foi rejeitada, sugerindo que o foco deve ser na qualidade e clareza da descrição, e não apenas em sua extensão.

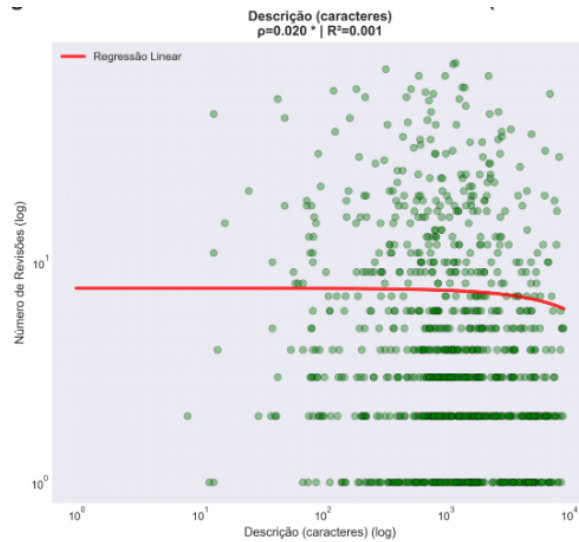


Figura 8: Descrição x numero de revisões

4.8 Qual a relação entre as interações nos PRs e o número de revisões realizadas?

Ao analisar a relação entre interações nos PRs e no numero de revisões, encontramos uma correlação positiva forte entre o número de participantes e o de revisões ($= 0.5395$) e uma correlação positiva moderada entre o número de comentários e o de revisões ($= 0.4588$). Esta conclusão pode ser justificada pelo fato de que um maior número de participantes envolvidos em um PR introduz múltiplas perspectivas, o que naturalmente leva a mais questionamentos e ciclos de feedback. Da mesma forma, um volume elevado de comentários indica discussões ativas e a necessidade de realizar ajustes no código, resultando em mais revisões. Assim, as interações se mostram como um indicador significativo do esforço de revisão, onde PRs com maior debate e engajamento tendem a passar por um processo de validação mais iterativo e extenso

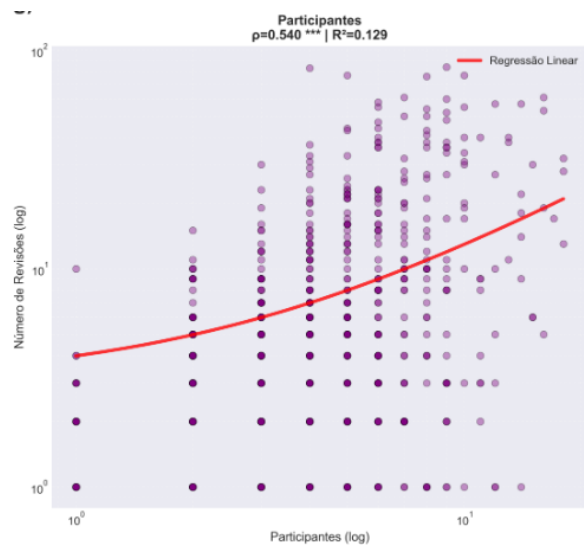


Figura 9: Participantes x numero de revisões

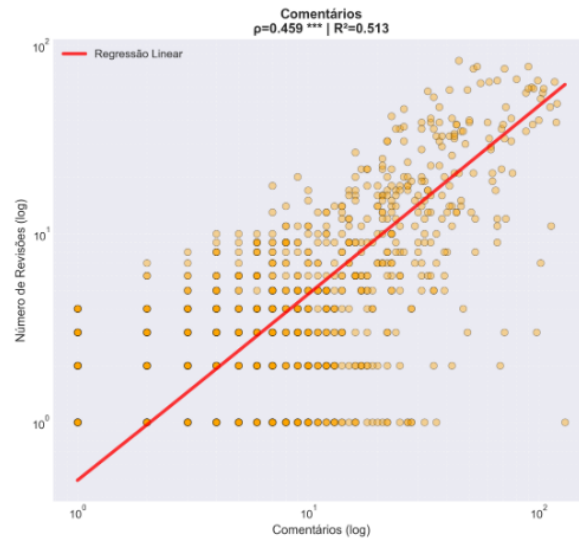


Figura 10: Comentários x numero de revisões

5 Considerações gerais

Para realizar as análises deste estudo, utilizamos o coeficiente de correlação de Spearman, que nos oferece diversas vantagens, sendo elas, o fato de que a violação da normalidade em todas as variáveis desqualifica o uso do Pearson, que pressupõe uma distribuição normal. Além disso, a presença massiva de outliers, indicada pela forte assimetria e alta curtose, tornaria o Pearson extremamente sensível e propenso a distorções. O Spearman supera essa limitação ao trabalhar com ranks (posições) dos dados, garantindo robustez contra extremos. Essa metodologia também é mais eficaz para capturar relações monotônicas não lineares, que são frequentes em dados de software, e é intrinsecamente mais apropriada para variáveis discretas de contagem como as de repositórios GitHub. Em suma, o Spearman oferece uma estimativa de correlação mais confiável e estatisticamente apropriada.

Segue abaixo um gráfico exibindo um resumo das correlações trabalhadas neste artigo:

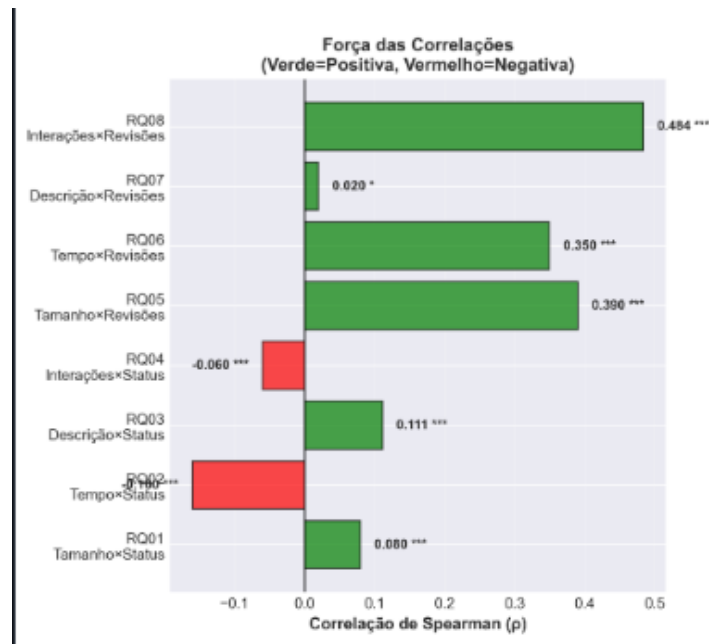


Figura 11:

6 Considerações finais

Com base na análise das oito questões de pesquisa, este estudo alcançou seu objetivo de caracterizar a atividade de code review no GitHub, identificando variáveis cruciais que influenciam tanto o desfecho das Pull Requests (PRs) quanto a intensidade do processo de revisão. As evidências apontam que qualidade e engajamento são fatores determinantes.

Observamos que PRs mescladas (merged) tendem a ser maiores e a ter descrições significativamente mais longas, sugerindo que contribuições substanciais, quando bem documentadas, são bem recebidas. Por outro lado, PRs que são fechadas (closed) consomem mais tempo de análise e geram um maior volume de comentários e participantes, indicando que a dificuldade em chegar a um consenso ou o desalinhamento com o projeto levam ao prolongamento do debate e, por fim, à rejeição.

Quanto à intensidade da revisão, há uma correlação clara: PRs maiores e mais longas requerem mais ciclos de revisão, uma medida de mitigação de risco e da sobrecarga cognitiva do revisor. Além disso, o aumento nas interações (participantes e comentários) também se mostrou um forte preditor de mais revisões. Em resumo, o sucesso de uma PR está ligado ao seu tamanho e qualidade inicial (descrição e menor tempo de análise), enquanto a rejeição e o aumento nas revisões são catalisados pela complexidade e pela falta de consenso, atraindo mais stakeholders do projeto para a discussão.