

RELATÓRIO

Caracterizando a Atividade de Code Review no GitHub

Characterizing the Code Review Activity on GitHub

Lorrayne Oliveira [Pontifícia Universidade Católica de Minas Gerais | lorrayne.marayze@gmail.com]
Pedro Pires [Pontifícia Universidade Católica de Minas Gerais | pedro.pires@gmail.com]

Resumo. Este trabalho apresenta um estudo empírico sobre a prática de code review em repositórios populares do GitHub. A atividade de revisão de código, fundamental para assegurar a qualidade e manutenibilidade do software, é analisada sob diferentes dimensões, considerando métricas associadas ao tamanho, tempo de análise, descrição e interações de pull requests (PRs). O objetivo central é compreender como essas variáveis influenciam o resultado das revisões (merge ou rejeição) e o número de revisões realizadas. Para isso, foi construído um dataset contendo PRs dos 200 repositórios mais populares do GitHub, avaliando somente aqueles que passaram por processos de revisão humana e tiveram tempo mínimo de análise superior a uma hora. As correlações entre as variáveis foram examinadas por meio de análises estatísticas de Pearson, de modo a identificar padrões significativos no comportamento das revisões de código. Os resultados obtidos fornecem evidências quantitativas sobre fatores que impactam o sucesso das contribuições, oferecendo insights sobre as dinâmicas colaborativas de revisão em projetos open-source.

Abstract. This work presents an empirical study on the practice of code review in popular GitHub repositories. The code review activity, essential for ensuring software quality and maintainability, is analyzed through multiple dimensions, considering metrics related to the size, review time, description, and interactions of pull requests (PRs). The main goal is to understand how these variables influence the review outcomes (merge or rejection) and the number of reviews performed. A dataset was built containing PRs from the 200 most popular GitHub repositories, considering only those with human reviews and analysis time greater than one hour. Correlations among variables were examined using the Pearson statistical test to identify significant patterns in review behavior. The results provide quantitative evidence about factors influencing contribution success, offering insights into collaborative review dynamics in open-source projects.

Palavras-chave: Code Review; Pull Requests; GitHub; Revisão de Código; Engenharia de Software Colaborativa

Keywords: Code Review; Pull Requests; GitHub; Code Quality; Collaborative Software Engineering

1 Introdução

A revisão de código (code review) é uma das práticas mais consolidadas em processos ágeis de desenvolvimento de software. Ela consiste na inspeção do código por revisores antes de sua integração ao repositório principal, com o intuito de detectar defeitos, melhorar a legibilidade e assegurar a aderência aos padrões de qualidade. No contexto de plataformas open-source como o GitHub, essa atividade é operacionalizada por meio de pull requests (PRs), que representam contribuições submetidas por desenvolvedores e avaliadas por colaboradores do projeto.

Neste trabalho, busca-se caracterizar empiricamente o comportamento das atividades de code review em repositórios populares do GitHub. Especificamente, este estudo visa compreender a forma com que os fatores, como o tamanho do PR, o tempo de análise, a extensão da descrição e o volume de interações influenciam o resultado final da revisão (merge ou rejeição) e o número de revisões realizadas.

A partir da análise, pretende-se identificar padrões recorrentes e variáveis críticas que impactam o sucesso de revisões, contribuindo para a melhoria de práticas de colaboração e qualidade de código em ambientes distribuídos de desenvolvimento.

1.1 Hipóteses Informais

Com base nas observações do ecossistema de desenvolvimento open-source, foram elaboradas as seguintes hipóteses informais para orientar a investigação:

IH01: Pull requests menores, com menos arquivos e li-

nhas alteradas, têm maior probabilidade de serem aprovados.

IH02: PRs analisados em tempo moderado (entre 1h e 48h) tendem a ser mais aceitos, enquanto análises muito longas reduzem a taxa de merge.

IH03: PRs com descrições mais completas aumentam a chance de aprovação, por facilitar a compreensão do revisor.

IH04: Um número moderado de interações (comentários e participantes) está positivamente associado à aceitação do PR, enquanto muitas interações podem indicar divergências.

IH05: PRs menores demandam menos ciclos de revisão, reduzindo o esforço necessário do revisor.

IH06: O tempo de análise está positivamente correlacionado com o número de revisões. Revisões mais longas tendem a envolver mais ciclos de feedback.

IH07: Descrições detalhadas diminuem o número de revisões necessárias, pois fornecem contexto adequado desde o início.

IH08: PRs com mais interações (comentários, revisores e discussões) tendem a passar por mais revisões até sua aprovação final.

1.2 Objetivos

O objetivo principal deste trabalho é caracterizar empiricamente a atividade de *code review* em repositórios populares do GitHub, analisando fatores que influenciam o resultado e a dinâmica das revisões de código. A partir de métricas quantitativas extraídas de *pull requests* (PRs), busca-se responder às seguintes questões de pesquisa:

- **RQ01:** Qual a relação entre o tamanho dos *pull requests* e o feedback final das revisões?
- **RQ02:** Qual a relação entre o tempo de análise dos *pull requests* e o feedback final das revisões?
- **RQ03:** Qual a relação entre a descrição dos *pull requests* e o feedback final das revisões?
- **RQ04:** Qual a relação entre as interações nos *pull requests* e o feedback final das revisões?
- **RQ05:** Qual a relação entre o tamanho dos *pull requests* e o número de revisões realizadas?
- **RQ06:** Qual a relação entre o tempo de análise dos *pull requests* e o número de revisões realizadas?
- **RQ07:** Qual a relação entre a descrição dos *pull requests* e o número de revisões realizadas?
- **RQ08:** Qual a relação entre as interações nos *pull requests* e o número de revisões realizadas?

Como objetivo secundário, pretende-se validar ou refutar as hipóteses informais elaboradas, contribuindo para o entendimento dos fatores que afetam o sucesso e a eficiência do processo de revisão de código em projetos *open-source*.

2 Metodologia

2.1 Coleta de Dados

A coleta de dados foi realizada através da API GraphQL do GitHub, utilizando um script em Python desenvolvido especificamente para esta análise. O processo de coleta contemplou informações relacionadas às atividades de revisão de código, englobando tanto características dos *pull requests* (PRs) quanto atributos do processo de *code review*. As etapas seguiram o seguinte fluxo:

Autenticação: Utilização de token de acesso pessoal do GitHub para autenticar as requisições à API e permitir a coleta de grandes volumes de dados.

Consulta e Paginação: Implementação de consultas em GraphQL para coletar dados dos 200 repositórios mais populares do GitHub, considerando o número de estrelas como critério de popularidade. Foi utilizado um mecanismo de paginação para garantir a coleta completa de todos os 200 PRs por repositório.

Filtragem dos PRs: Seleção apenas dos PRs com status *MERGED* ou *CLOSED*, que possuíam ao menos uma revisão (*review count* > 0) e tempo total de análise superior a uma hora, a fim de descartar revisões automáticas realizadas por *bots* ou pipelines de CI/CD.

Extração de Métricas de Revisão: Para cada PR foram coletadas informações sobre tamanho (arquivos e linhas modificadas), tempo de análise (diferença entre *createdAt* e *closedAt*), descrição (tamanho do texto em *markdown*), interações (número de comentários e participantes) e número total de revisões.

Tratamento de Erros: Implementação de mecanismos de controle de taxa (*rate limiting*) e de repetição automática de requisições em caso de falhas, assegurando consistência e integridade dos dados coletados.

2.2 Métricas Coletadas

Para cada *pull request* analisado, foram coletadas métricas relacionadas tanto ao processo de revisão quanto às características estruturais do PR. As métricas foram agrupadas em

quatro dimensões principais, conforme descrito a seguir:

Tamanho: número de arquivos modificados e total de linhas adicionadas/removidas. **Tempo de Análise:** intervalo de tempo entre a criação e o fechamento do PR. **Descrição:** número de caracteres no corpo da descrição do PR, em formato *markdown*. **Interações:** número total de participantes e comentários na discussão. **Revisões:** quantidade de revisões formais registradas no campo *review count*. **Status Final:** estado final do PR, podendo ser *MERGED* (aceito) ou *CLOSED* (rejeitado).

Essas métricas permitem analisar as correlações entre características dos PRs e o resultado das revisões, respondendo às questões de pesquisa propostas no trabalho.

2.3 Exportação e Análise

Os dados processados foram exportados para um arquivo JSON (*repositorios_github_dados.json*) com o objetivo de facilitar as análises estatísticas e permitir a replicação dos resultados. A análise foi conduzida a partir de duas abordagens complementares: medidas de tendência central e análise de correlação.

A etapa de tendência central envolveu o uso de medidas como a mediana e a média para resumir a distribuição dos valores obtidos em cada métrica (tamanho, tempo de análise, descrição, interações e revisões). Esse procedimento possibilita observar o comportamento geral dos *pull requests* (PRs) sem a influência de valores extremos.

Na segunda etapa, foi aplicada a **correlação de Pearson**, a fim de identificar o grau e a direção das relações lineares entre as variáveis analisadas. A escolha do coeficiente de Pearson, em detrimento de outras técnicas como a correlação de Spearman, deve-se a duas razões principais:

- Natureza contínua e quantitativa das variáveis: As métricas utilizadas (número de linhas alteradas, tempo de análise em horas, tamanho das descrições, número de comentários e revisões) são essencialmente contínuas e quantitativas, o que torna o coeficiente de Pearson mais apropriado, já que ele mede relações lineares em dados intervalares ou racionais.
- Análise de força linear: Diferentemente do coeficiente de Spearman, que captura relações monotônicas (lineares ou não), o objetivo neste estudo é quantificar especificamente o grau de associação linear entre variáveis, permitindo avaliar se aumentos em uma métrica (por exemplo, tamanho do PR) estão diretamente associados a aumentos ou reduções em outra (como tempo de análise ou número de revisões).

2.4 Limitações

Este estudo apresenta algumas limitações que devem ser consideradas para uma interpretação adequada dos resultados.

Primeiramente, devido a restrições de processamento computacional, memória e tempo de execução das consultas à API do GitHub, não foram coletados todos os *pull requests* de cada um dos 200 repositórios analisados. Em vez disso, foram amostrados 200 PRs por repositório, totalizando 40.000 registros no conjunto de dados. Esse recorte, pode não refletir integralmente o comportamento completo de cada projeto, especialmente em repositórios com volume muito maior de contribuições.

Outra limitação refere-se à ausência de informações qualitativas sobre o contexto dos PRs. O tempo total de análise, por exemplo, pode incluir períodos de inatividade do revisor ou do autor, não refletindo necessariamente o esforço real de revisão necessário. De forma semelhante, o tamanho da descrição foi avaliado apenas em número de caracteres, sem análise semântica da clareza, completude ou relevância do conteúdo.

3 Resultados

3.1 RQ01: Qual a relação entre o tamanho dos PRs e o feedback final das revisões?

No gráfico abaixo, é possível observar que as categorias de tamanho “muito pequeno” e “pequeno” concentram o maior número de pull requests aceitos, enquanto conforme o tamanho aumenta, a quantidade de merges diminui gradualmente.

Os dados mostram que nas faixas de 0 a 10 e de 11 a 100 linhas modificadas, o número de pull requests aprovados é significativamente maior do que o número de fechados. Isso indica que alterações menores são mais comuns e têm mais chance de serem aceitas. Já nos pull requests médios (101 a 500 linhas) e grandes (501 a 1000 linhas), o número de merges cai consideravelmente, mostrando que revisões mais extensas são menos prováveis de serem aprovadas. Para os pull requests muito grandes, acima de 1000 linhas modificadas, o número de merges continua baixo, o que reforça essa tendência.

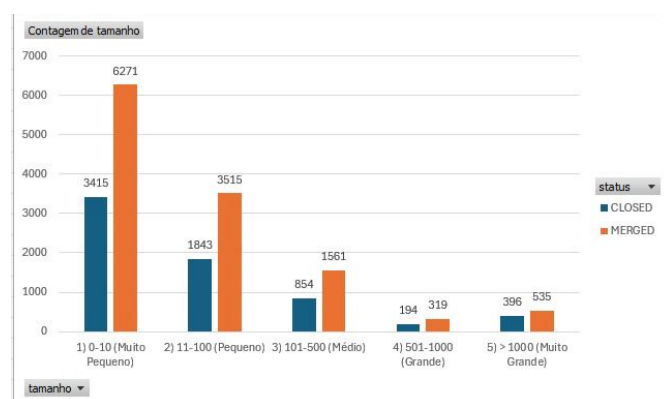


Figura 1. Tamanho dos PRs x Feedback Final

Esses resultados confirmam a hipótese inicial (IH01), que propunha que pull requests menores têm maior probabilidade de aprovação. É possível entender isso porque revisões pequenas são mais rápidas de analisar, têm menor risco de causar conflitos e exigem menos esforço de revisão. À medida que o tamanho do pull request aumenta, a complexidade também cresce, tornando o processo de análise mais demorado e sujeito a rejeições.

Com base nessas observações, podemos concluir que existe uma relação negativa entre o tamanho do pull request e o sucesso da revisão. Ou seja, quanto maior o volume de modificações, menor a chance de o pull request ser aceito.

3.2 RQ02: Qual a relação entre o tempo de análise dos PRs e o feedback final das revisões?

Os pull requests foram agrupados em quatro intervalos de tempo: de 0 a 24 horas (muito rápidos ou triviais), de 24 horas a 1 semana (pico de engajamento), de 1 semana a 1 mês (moderado ou de longa duração) e acima de 1 mês (estagnado ou muito longo).

Os resultados mostram que a grande maioria dos pull requests foi analisada e concluída em até 24 horas, com 8.612 merges e 3.099 fechamentos. Essa faixa de tempo concentra o maior número de aprovações, indicando que revisões rápidas tendem a ser mais bem-sucedidas. No intervalo de 24 horas a 1 semana, o número de merges diminui para 2.247 e os fechamentos para 1.312, o que ainda representa uma boa taxa de aprovação, embora menor que na faixa anterior. Já nas revisões com duração de 1 semana a 1 mês, observa-se um equilíbrio entre merges (791) e fechamentos (812), sugerindo que revisões mais longas não garantem necessariamente melhores resultados. Por fim, os pull requests com duração superior a 1 mês apresentam uma inversão clara, com 1.479 fechamentos contra apenas 551 merges, o que mostra que revisões muito demoradas têm uma alta chance de serem rejeitadas ou abandonadas.

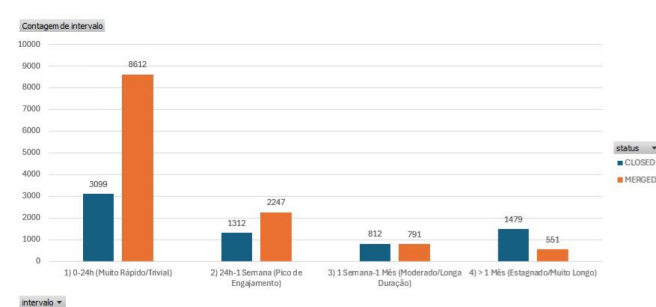


Figura 2. Tempo de Análise x Feedback Final

Esses resultados estão alinhados com a hipótese IH02, que previa que revisões com duração moderada (entre 1 hora e 48 horas) teriam maior probabilidade de aprovação, enquanto tempos muito curtos ou longos reduziram essa taxa. No gráfico, é possível perceber que a maioria dos merges ocorre em revisões rápidas, mas não excessivamente demoradas. Quando o tempo aumenta para além de uma semana, o número de merges cai progressivamente, o que sugere que revisões longas perdem engajamento dos revisores e dos autores, podendo gerar acúmulo de comentários e atrasos na decisão final.

Revisões rápidas e objetivas apresentam melhores resultados, enquanto revisões prolongadas tendem a ser rejeitadas com mais frequência. Essa relação linear entre o aumento do tempo e a diminuição dos merges justifica o uso do coeficiente de correlação de Pearson, já que ele permite quantificar o grau de dependência linear entre o tempo de análise e o sucesso do pull request.

3.3 RQ03: Qual a relação entre a descrição dos PRs e o feedback final das revisões?

A RQ03 busca compreender como o tamanho da descrição dos pull requests se relaciona com o resultado final das revisões. As descrições no gráfico foram agrupadas em quatro faixas: curtas (0 a 5.000 caracteres), moderadas (5.001 a 15.000), longas (15.001 a 30.000) e muito longas (acima de 30.000 caracteres).

O gráfico mostra que a maioria dos pull requests apresenta descrições curtas, com 12.150 merges e 6.628 fechamentos. Esse comportamento indica que a maior parte das revisões aprovadas está concentrada em PRs que possuem descrições curtas.

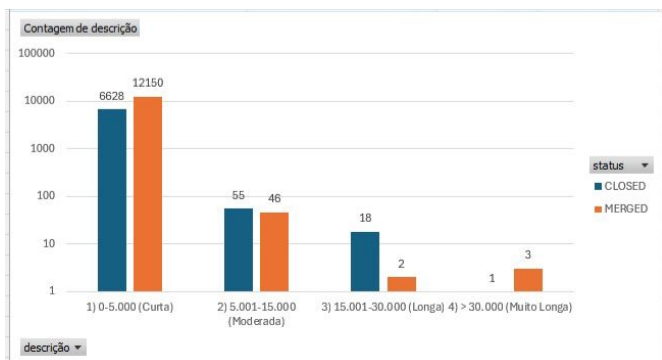


Figura 3. Descrição x Feedback Final

Nas categorias de descrições moderadas, há 55 fechamentos e 46 merges, sugerindo uma pequena redução na taxa de aprovação, mas sem uma diferença expressiva em relação às curtas. Já nas descrições longas e muito longas, o número de ocorrências é bastante reduzido, o que limita a análise quantitativa e dificulta tirar conclusões estatísticas robustas. Ainda assim, observa-se que pull requests com descrições muito extensas são menos comuns e, em alguns casos, menos aceitos.

Os resultados confirmam parcialmente a hipótese IH03, que previa que descrições mais completas e detalhadas favoreceriam a aprovação do pull request por facilitar a compreensão do revisor. No entanto, os dados mostram que a maioria dos PRs aceitos possui descrições curtas, o que pode estar relacionado ao fato de muitos deles envolverem pequenas alterações, que não exigem explicações extensas. Assim, a clareza e a objetividade da descrição parecem exercer um papel mais relevante do que o comprimento do texto em si.

3.4 RQ04: Qual a relação entre as interações nos PRs e o feedback final das revisões?

A análise da RQ04 tem como objetivo verificar a relação entre o número de participantes e comentários nos pull requests e o resultado final da revisão, considerando os status merged e closed. O gráfico mostra a distribuição desses dois fatores (participantes e comentários) para cada tipo de resultado, evidenciando a variação e a densidade dos dados.

De modo geral, observa-se que a maior parte dos pull requests apresenta baixa participação e poucos comentários, concentrando-se nas faixas inferiores do gráfico. Tanto os

PRs merged quanto os closed tendem a ter entre um e cinco participantes e um número reduzido de comentários. Casos com mais de 50 participantes ou mais de 100 comentários são raros e podem representar situações excepcionais, como revisões de grande impacto ou com discussões prolongadas.

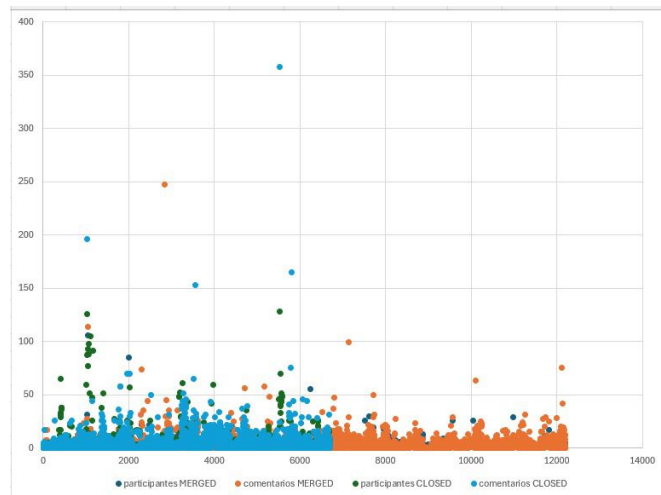


Figura 4. Interações x Feedback Final

Ao comparar os resultados entre PRs merged e closed, nota-se que a dispersão dos dados é semelhante. Os pull requests aprovados apresentam uma quantidade levemente maior de interações, mas sem diferença significativa. Isso indica que um número maior de comentários ou de participantes não garante a aprovação do PR. Em alguns casos, revisões com muitos comentários podem sinalizar divergências ou necessidade de ajustes, o que aumenta a chance de fechamento sem merge.

Esses resultados confirmam parcialmente a hipótese IH04, que sugeria que revisões com maior engajamento — representado por mais comentários e participantes — teriam maior probabilidade de aprovação. Os dados mostram que o engajamento está presente em revisões aceitas, mas não é um fator determinante. O volume de interações pode estar mais relacionado à complexidade ou ao tamanho do pull request do que ao seu resultado final.

3.5 RQ05: Qual a relação entre o tamanho dos PRs e o número de revisões realizadas?

Observando os dados apresentados, é possível perceber que a hipótese se confirma de forma bastante clara. Os pull requests com baixo ou padrão de revisões, que representam de 0 a 10 revisões, concentram a maior quantidade de casos em todos os tamanhos. Dentro dessa categoria, os pull requests muito pequenos aparecem com 9679 casos, seguidos pelos pequenos com 5322 casos, médios com 2360 casos, grandes com 497 casos e muito grandes com 909 casos. Isso demonstra que quanto menor o pull request, maior tende a ser a quantidade que passa por poucas revisões.

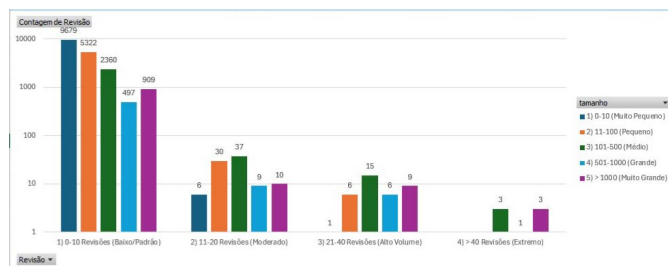


Figura 5. Tamanho x Número de Revisões

Quando analisamos as outras categorias de quantidade de revisões, o padrão se mantém. Na categoria moderada, que vai de 11 a 20 revisões, os números caem bastante para todos os tamanhos, mas ainda assim os menores aparecem com mais frequência. Os muito pequenos têm 6 casos, os pequenos 30 casos, os médios 37 casos, os grandes 9 casos e os muito grandes 10 casos. Aqui já começamos a ver uma mudança interessante, pois os pull requests médios aparecem com um número ligeiramente maior que os pequenos. Na categoria de alto volume, de 21 a 40 revisões, os números são ainda menores. Os muito pequenos aparecem com apenas 1 caso, os pequenos com 6 casos, os médios com 15 casos, os grandes com 6 casos e os muito grandes com 9 casos. Nessa faixa, os pull requests médios se destacam com a maior contagem, o que pode indicar que eles acabam exigindo mais ciclos de revisão em alguns casos específicos.

Por fim, na categoria extrema, com mais de 40 revisões, os números são mínimos. Não há registro de pull requests muito pequenos ou grandes nessa categoria, apenas 3 casos de médios, 1 caso de pequenos e 3 casos de muito grandes. Isso mostra que situações com tantas revisões são raras independentemente do tamanho.

De forma geral, os dados sustentam a hipótese de que pull requests menores realmente demandam menos ciclos de revisão. A grande maioria dos pull requests menores se concentra na faixa de baixa quantidade de revisões, e conforme o tamanho aumenta, a distribuição tende a se espalhar um pouco mais pelas outras categorias. Isso faz sentido porque pull requests menores são mais fáceis de revisar e entender, então os revisores conseguem analisar e aprovar mais rapidamente, sem necessidade de muitos ciclos de feedback. Já os pull requests maiores podem ter mais complexidade e mais chances de problemas, o que acaba gerando mais discussões e mais rodadas de revisão até a aprovação final.

3.6 RQ06: Qual a relação entre o tempo de análise dos PRs e o número de revisões realizadas?

A análise do gráfico mostra a relação entre o tempo de análise dos pull requests e a quantidade de revisões realizadas. A hipótese IH06 sugeria que o tempo de análise está positivamente correlacionado com o número de revisões, ou seja, revisões mais longas tendem a envolver mais ciclos de feedback.

A grande maioria dos pull requests se concentra na categoria de baixo ou padrão de revisões, que vai de 0 a 10 revisões. Nessa categoria, os pull requests analisados de forma muito rápida ou trivial aparecem com 11681 casos, os analisados em um pico de engajamento têm 3514 casos, os de moderada ou longa duração têm 1565 casos, e os estagnados

ou muito longos têm 2007 casos. Isso mostra que independentemente do tempo de análise, a maioria dos pull requests passa por poucas revisões.

Quando avançamos para a categoria moderada, de 11 a 20 revisões, os números caem bastante em todos os intervalos de tempo. Os pull requests muito rápidos aparecem com 25 casos, os de pico de engajamento com 31 casos, os moderados com 21 casos e os estagnados com 15 casos. Aqui já começamos a perceber que a distribuição fica mais equilibrada entre os diferentes tempos de análise, mas ainda assim os números são bem menores comparados à primeira categoria.

Na categoria de alto volume, que vai de 21 a 40 revisões, os números são ainda menores. Os pull requests muito rápidos têm 5 casos, os de pico de engajamento têm 11 casos, os moderados têm 14 casos e os estagnados têm 7 casos. Nessa faixa, é interessante notar que os pull requests com tempo moderado de análise aparecem com a maior contagem, seguidos pelos de pico de engajamento. Isso sugere que quando há muitas revisões, o tempo de análise tende a se estender um pouco mais.

Por fim, na categoria extrema, com mais de 40 revisões, os números são mínimos. Não há casos de pull requests muito rápidos nessa categoria, aparecem 3 casos de pico de engajamento, 3 casos de moderados e apenas 1 caso de estagnados. Isso mostra que situações com tantas revisões são extremamente raras, mas quando acontecem, podem ocorrer em diferentes faixas de tempo.

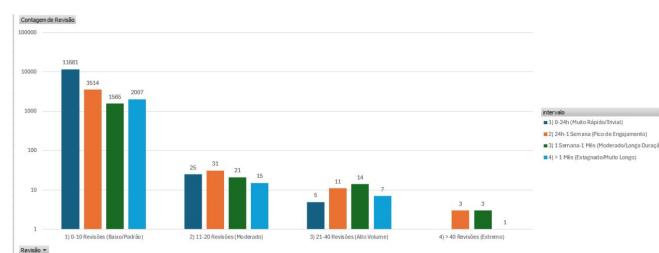


Figura 6. Tempo de Análise x Número de Revisões

De forma geral, os dados sustentam parcialmente a hipótese de que o tempo de análise está correlacionado com o número de revisões. Embora a maioria dos pull requests se concentre na faixa de poucas revisões independentemente do tempo, quando olhamos para as categorias com mais revisões, percebemos que os tempos de análise tendem a ser mais variados e por vezes mais longos. Pull requests que passam por mais ciclos de revisão naturalmente tendem a levar mais tempo para serem concluídos, pois cada rodada de feedback adiciona tempo ao processo. No entanto, é importante notar que existem muitos pull requests que são analisados rapidamente e ainda assim passam por poucas revisões, o que mostra que a eficiência do processo também depende de outros fatores além do número de ciclos de revisão.

3.7 RQ07: Qual a relação entre a descrição dos PRs e o número de revisões realizadas?

A análise do gráfico evidencia a relação entre o tipo de descrição dos pull requests e o número de revisões realizadas.

A hipótese IH07 sugeria que descrições mais detalhadas

reduziriam a quantidade de revisões necessárias, fornecendo contexto adequado desde o início.

Os dados mostram um padrão predominante de descrições curtas. Na categoria de baixo ou padrão de revisões (0 a 10 revisões), os pull requests com descrições curtas representam 18.643 casos, seguidos por 100 casos de descrição moderada, 20 casos de descrição longa e apenas 4 casos de descrição muito longa. Essa distribuição indica que a maioria dos pull requests que passam por poucas revisões possui descrições curtas, o que aparentemente contraria a hipótese inicial.

Na categoria moderada (11 a 20 revisões), os números são significativamente menores: 91 casos de descrições curtas, apenas 1 caso de descrição moderada e nenhum registro de descrições longas ou muito longas. Isso mostra que, mesmo com descrições curtas, alguns pull requests necessitam de mais ciclos de revisão, embora em menor quantidade comparado à primeira categoria.

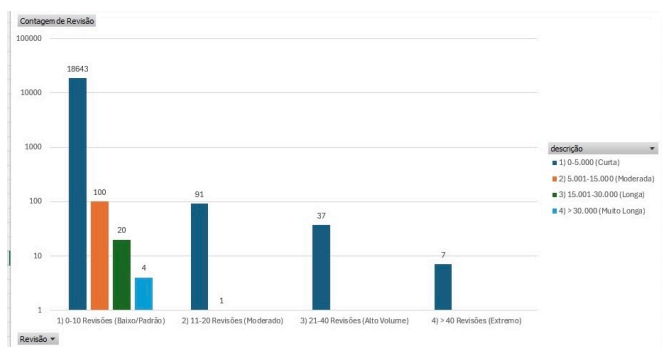


Figura 7. Descrição x Número de Revisões

Na categoria de alto volume (21 a 40 revisões), há 37 casos de pull requests com descrição curta, sem registros de outras categorias de descrição. Esse padrão sugere que a ausência de contexto detalhado pode contribuir para um aumento nas rodadas de revisão.

Na categoria extrema (mais de 40 revisões), existem apenas 7 casos de pull requests com descrição curta, novamente sem registros de descrições mais longas. Apesar do número reduzido, isso reforça a tendência de que pull requests com muitas revisões geralmente possuem descrições mais simples.

Considerando o conjunto de dados, o cenário observado difere do esperado pela hipótese. A maioria dos pull requests apresenta descrições curtas, independentemente do número de revisões. Entretanto, a proporção de pull requests com descrições curtas que passam por poucas revisões é muito maior do que aqueles que necessitam de muitas revisões. Isso indica que descrições curtas funcionam adequadamente na maioria dos casos, mas, em situações de maior complexidade ou problemas no código, mesmo elas podem levar a ciclos adicionais de revisão. A quantidade muito reduzida de pull requests com descrições longas impede conclusões definitivas sobre o impacto de descrições detalhadas, podendo indicar que elas são utilizadas apenas em casos específicos ou que a cultura do projeto privilegia descrições mais concisas.

3.8 RQ08: Qual a relação entre as interações nos PRs e o número de revisões realizadas?

A análise dos gráficos abaixo mostra a relação entre as interações nos pull requests e a quantidade de revisões realizadas. A hipótese IH08 sugeria que pull requests com mais interações, como comentários, revisores e discussões, tendem a passar por mais revisões até sua aprovação final.

O primeiro gráfico apresenta a relação entre o número de participantes e as revisões. É possível perceber uma concentração muito grande de pontos na parte inferior esquerda do gráfico, principalmente entre 0 e 20 revisões e com até 40 participantes. A linha de tendência mostra uma inclinação positiva muito suave, com a equação $y = 0,086x + 0,2536$ e um coeficiente de determinação $R^2 = 0,1211$. Esse valor de R^2 indica que a correlação entre participantes e revisões é fraca, ou seja, o número de participantes explica apenas cerca de 12% da variação no número de revisões. Ainda assim, a linha de tendência positiva sugere que existe alguma relação entre mais participantes e mais revisões, mesmo que não seja muito forte.

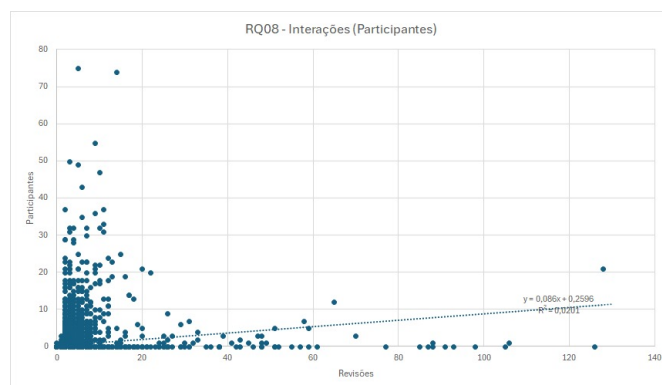


Figura 8. Interações (Participantes) x Número de Revisões

O segundo gráfico mostra a relação entre comentários e revisões. Assim como no primeiro gráfico, há uma concentração massiva de pontos na região inferior esquerda, especialmente entre 0 e 50 revisões e com até 50 comentários. A linha de tendência também apresenta inclinação positiva, com a equação $y = 0,0641x + 0,3488$ e um coeficiente $R^2 = 0,0975$. Esse valor de R^2 é ainda menor que o anterior, indicando que a correlação entre comentários e revisões é ainda mais fraca, explicando apenas cerca de 10% da variação. Mesmo assim, a tendência positiva mostra que existe alguma relação entre mais comentários e mais revisões.

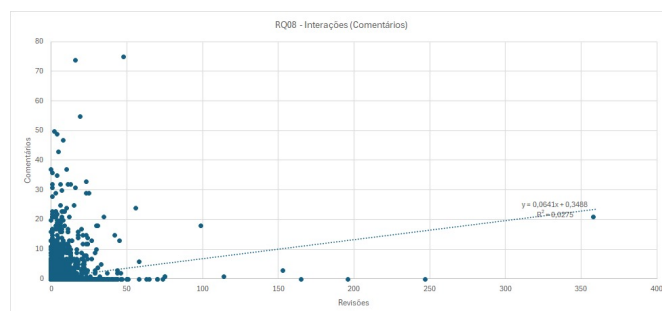


Figura 9. Interações (Revisões) x Número de Revisões

Ambos os gráficos revelam alguns pontos outliers interessantes. Existem casos de pull requests com poucas revisões mas muitos participantes ou comentários, e também casos de pull requests com muitas revisões mas poucos participantes ou comentários. Esses casos atípicos mostram que a relação entre interações e revisões não é direta e pode ser influenciada por outros fatores não considerados nos gráficos.

A dispersão dos dados é muito grande em ambos os gráficos. A maioria dos pull requests tem poucas revisões e poucas interações, o que é esperado para projetos que funcionam bem. Porém, conforme o número de revisões aumenta, parece haver uma tendência de aumento nas interações, mesmo que essa tendência não seja muito forte. Isso faz sentido porque pull requests que passam por mais ciclos de revisão naturalmente geram mais discussões, mais participantes precisam se envolver para resolver os problemas, e mais comentários são necessários para comunicar as mudanças e ajustes solicitados.

Os dados sustentam parcialmente a hipótese IH08. Existe sim uma correlação positiva entre interações e número de revisões, mas essa correlação é fraca. Isso significa que embora pull requests com mais interações tendam a ter mais revisões, essa não é uma regra absoluta. Muitos outros fatores podem influenciar o número de revisões necessárias, como a complexidade do código, a clareza da implementação, a experiência do desenvolvedor, e até questões específicas do projeto. As interações são apenas um dos elementos que podem indicar que um pull request está passando por um processo de revisão mais elaborado, mas não são o único fator determinante.

4 Conclusão

Este trabalho apresentou um estudo empírico sobre a prática de code review em repositórios populares do GitHub, analisando como diferentes fatores influenciam o resultado das revisões e o número de ciclos necessários até a aprovação ou rejeição dos pull requests.

Os resultados demonstraram que o tamanho dos pull requests exerce influência direta tanto no resultado final quanto no número de revisões. Pull requests menores apresentaram maior taxa de aprovação e demandaram menos ciclos de revisão, confirmando as hipóteses IH01 e IH05. A análise de correlação de Pearson revelou uma forte correlação positiva entre o número de arquivos modificados e linhas adicionadas ($r = 0,704$), bem como uma correlação moderada entre arquivos e linhas removidas ($r = 0,543$). A correlação entre linhas adicionadas e removidas ($r = 0,518$) sugere que pull requests com muitas adições frequentemente também envolvem remoções significativas.

A análise das interações mostrou correlações fracas. A correlação entre participantes e revisões foi de **0,208**, enquanto entre comentários e revisões foi de apenas **0,026**. Os valores de R^2 encontrados, **0,1211** para participantes e **0,0975** para comentários, indicam que essas variáveis explicam apenas uma pequena parte da variação no número de revisões. Isso confirma parcialmente as hipóteses IH04 e IH08, mostrando que outros fatores, como a complexidade técnica do código e a experiência dos desenvolvedores, podem exercer

influência mais significativa no processo de revisão.

De maneira geral, os resultados fornecem evidências quantitativas sobre os fatores que impactam o sucesso das contribuições em projetos open-source. Pull requests menores e analisados rapidamente tendem a ter maior probabilidade de aprovação e demandam menos esforço de revisão. As correlações encontradas revelaram que o tamanho do pull request possui forte correlação interna entre suas métricas, enquanto as interações apresentam correlações mais fracas com o número de revisões, sugerindo que múltiplos fatores influenciam o processo de code review em ambientes colaborativos.

Referências

- Gonzalez, G., Zhang, X., Yu, Y., Wang, T., Rastogi, A., and Wang, H. (2019). Improving the pull requests review process using learning-to-rank algorithms. *Proceedings of the 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 1–10. DOI: 10.1109/ESEM.2019.00006.
- LinearB (2022). Why estimated review time improves pull requests and reduces cycle time. *LinearB Blog*.
- Swarmia (2022). Why small pull requests are better. *Swarmia Blog*.
- Zhang, X., Yu, Y., Wang, T., Rastogi, A., and Wang, H. (2022). Pull request latency explained: An empirical overview. *Empirical Software Engineering*, 27(4):126. DOI: 10.1007/s10664-022-10143-4.