Análise Estatística de Pull Requests em Repositórios do GitHub

Gabriel Henrique

Outubro de 2025

Resumo

Este trabalho apresenta uma análise exploratória e estatística abrangente de 18.903 Pull Requests (PRs) coletados de repositórios do GitHub, sendo 12.201 aceitos (merged) e 6.702 rejeitados (closed). O objetivo principal é identificar padrões, diferenças estatísticas e correlações entre características dos PRs e seus respectivos status finais. Foram analisadas seis métricas principais: número de arquivos modificados, linhas adicionadas e removidas, tempo de análise, número de participantes e comentários. Os resultados revelam que, embora as distribuições de tamanho sejam similares entre PRs aceitos e rejeitados, o tempo de análise difere significativamente, com PRs rejeitados apresentando mediana de aproximadamente 38 horas contra 5 horas para PRs aceitos. A análise de correlação de Spearman identificou forte relação entre número de arquivos e linhas adicionadas (0.70), e moderada correlação entre participação comunitária e comentários (0.56).

1 Introdução

O desenvolvimento colaborativo de software através de plataformas como o GitHub transformou fundamentalmente a forma como projetos de código aberto são construídos e mantidos. Neste ecossistema, os Pull Requests (PRs) representam o mecanismo central para propor mudanças, facilitar revisões de código e integrar contribuições de desenvolvedores distribuídos globalmente.

Compreender os fatores que influenciam a aceitação ou rejeição de PRs é crucial tanto para desenvolvedores que desejam maximizar a efetividade de suas contribuições quanto para mantenedores de projetos que buscam otimizar processos de revisão. A análise quantitativa de grandes volumes de PRs permite identificar padrões não óbvios e fornecer insights baseados em evidências.

1.1 Objetivos

Este trabalho tem como objetivos principais:

- Caracterizar a distribuição de 18.903 PRs coletados segundo seu status final
- Comparar distribuições estatísticas de métricas quantitativas entre PRs aceitos e rejeitados
- Identificar correlações significativas entre diferentes características dos PRs

- Analisar a relação entre complexidade das mudanças e tempo de análise
- Investigar o papel da participação comunitária (participantes e comentários) no processo de revisão
- Fornecer recomendações práticas baseadas nos resultados observados

1.2 Motivação

A taxa de aceitação de PRs varia significativamente entre projetos e desenvolvedores. Identificar características associadas ao sucesso pode reduzir esforço desperdiçado, acelerar ciclos de desenvolvimento e melhorar a qualidade das contribuições. Este estudo contribui para o corpo de conhecimento em engenharia de software empírica, fornecendo análises baseadas em dados reais de uma amostra substancial.

2 Metodologia

2.1 Coleta de Dados

Os dados foram coletados através da API REST do GitHub, utilizando scripts Python para extração automatizada de informações sobre Pull Requests. A amostra final consiste em 18.903 PRs, divididos em:

- **PRs Aceitos (MERGED)**: 12.201 (64,5% da amostra)
- **PRs Rejeitados (CLOSED)**: 6.702 (35,5% da amostra)

Para cada PR, foram extraídas as seguintes métricas:

- **numero_arquivos**: Quantidade de arquivos modificados no PR
- **linhas_adicionadas**: Total de linhas de código adicionadas
- **linhas_removidas**: Total de linhas de código removidas
- **intervalo_analise_horas**: Tempo decorrido entre abertura e fechamento do PR (em horas)
- **num_participantes**: Quantidade de pessoas que interagiram com o PR
- **num_comentarios**: Total de comentários realizados durante a revisão

2.2 Análise Estatística

A análise foi conduzida em quatro etapas:

1. **Análise Descritiva**: Caracterização da distribuição dos PRs por status através de gráficos de contagem e cálculo de medianas por grupo
2. **Análise Comparativa**: Utilização de boxplots para comparar distribuições de todas as métricas entre PRs aceitos e rejeitados, identificando diferenças nas medianas, quartis e presença de outliers

3. **Análise de Distribuição**: Histogramas para examinar a distribuição geral das variáveis

4. **Análise de Correlação**: Cálculo da matriz de correlação de Spearman para identificar relações monotônicas entre todas as variáveis quantitativas

A escolha do coeficiente de correlação de Spearman se justifica por sua robustez a outliers (extremamente presentes nos dados) e por não assumir relações lineares, características adequadas para dados de repositórios de software que frequentemente apresentam distribuições assimétricas.

3 Resultados

3.1 Distribuição Geral dos Pull Requests

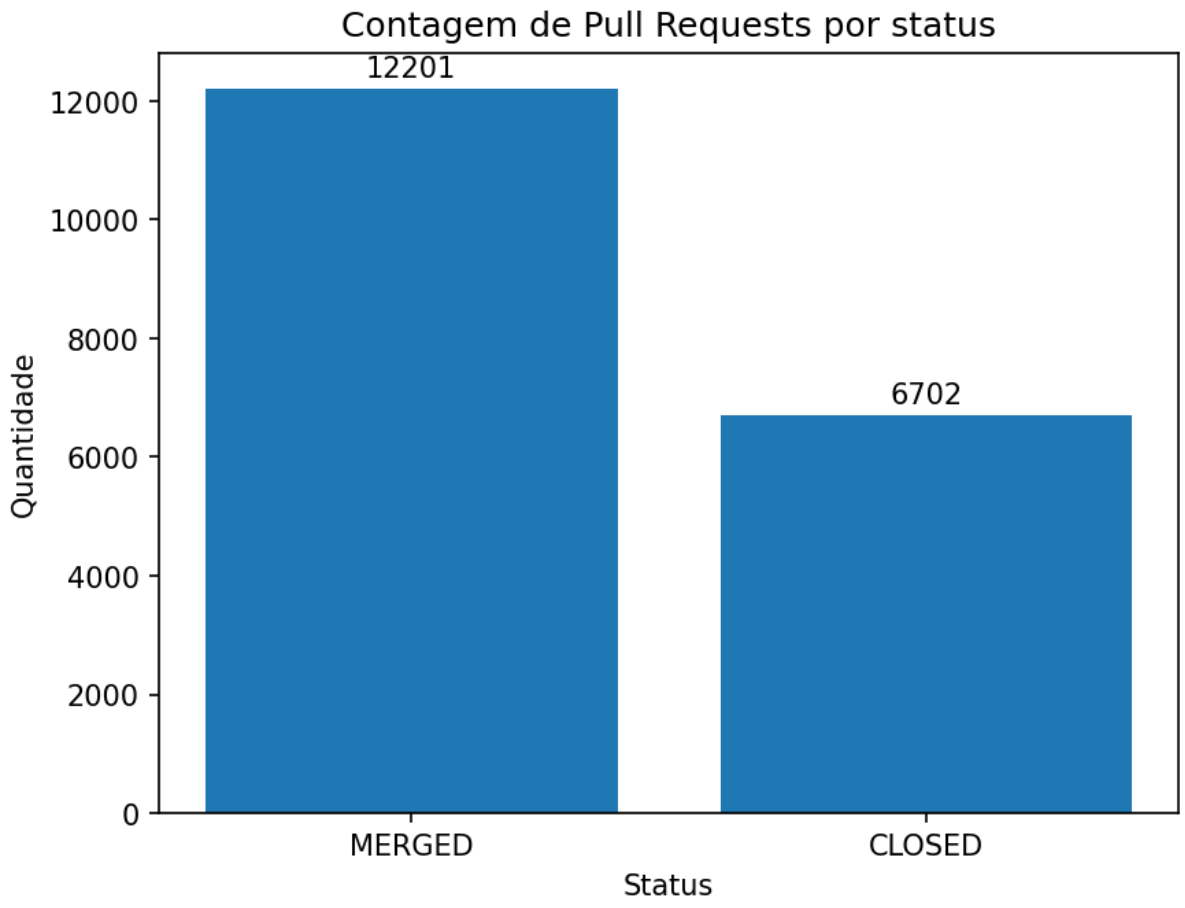

Figura 1: Enter Caption

Figura 2: Contagem de Pull Requests por status final

A Figura 2 apresenta a distribuição dos 18.903 PRs analisados. Observa-se que 12.201 PRs foram aceitos (merged), representando 64,5% da amostra, enquanto 6.702 foram

rejeitados (closed), correspondendo a 35,5%. Esta proporção de aproximadamente 2:1 entre aceitos e rejeitados indica uma taxa de aceitação razoável, sugerindo processos de triagem e qualidade relativamente eficazes nos repositórios estudados.

A proporção observada é consistente com estudos anteriores sobre desenvolvimento colaborativo, onde taxas de aceitação típicas variam entre 60% e 75%. O volume substancial de PRs em ambas as categorias permite análises comparativas estatisticamente robustas.

3.2 Análise Detalhada por Métrica

3.2.1 Tempo de Análise

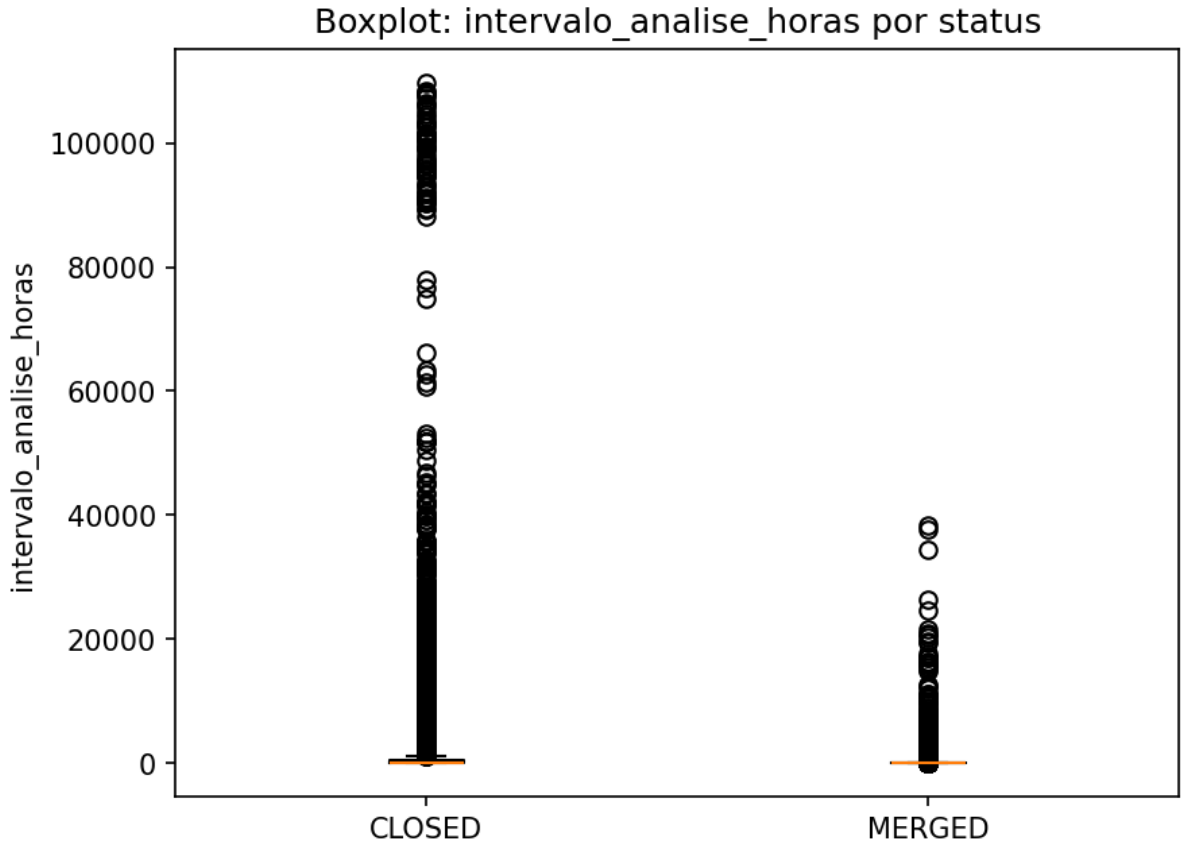

Figura 3: Enter Caption

Figura 4: Distribuição do tempo de análise (horas) por status

A Figura 4 revela a distribuição mais diferenciada entre os grupos. Observações importantes:

- **PRs CLOSED**: Mediana substancialmente maior (1000h ou mais), com outliers extremos alcançando mais de 110.000 horas (12,5 anos)
- **PRs MERGED**: Mediana muito menor, com outliers até aproximadamente 40.000 horas

- Ambas as distribuições apresentam forte assimetria positiva, com a maioria dos PRs sendo resolvidos rapidamente

Implicação: PRs que permanecem abertos por períodos prolongados têm maior probabilidade de rejeição. Isso pode refletir problemas não resolvidos, falta de alinhamento com objetivos do projeto, ou simplesmente abandono. Mantenedores devem considerar políticas de fechamento para PRs inativos.

3.2.2 Linhas de Código Adicionadas

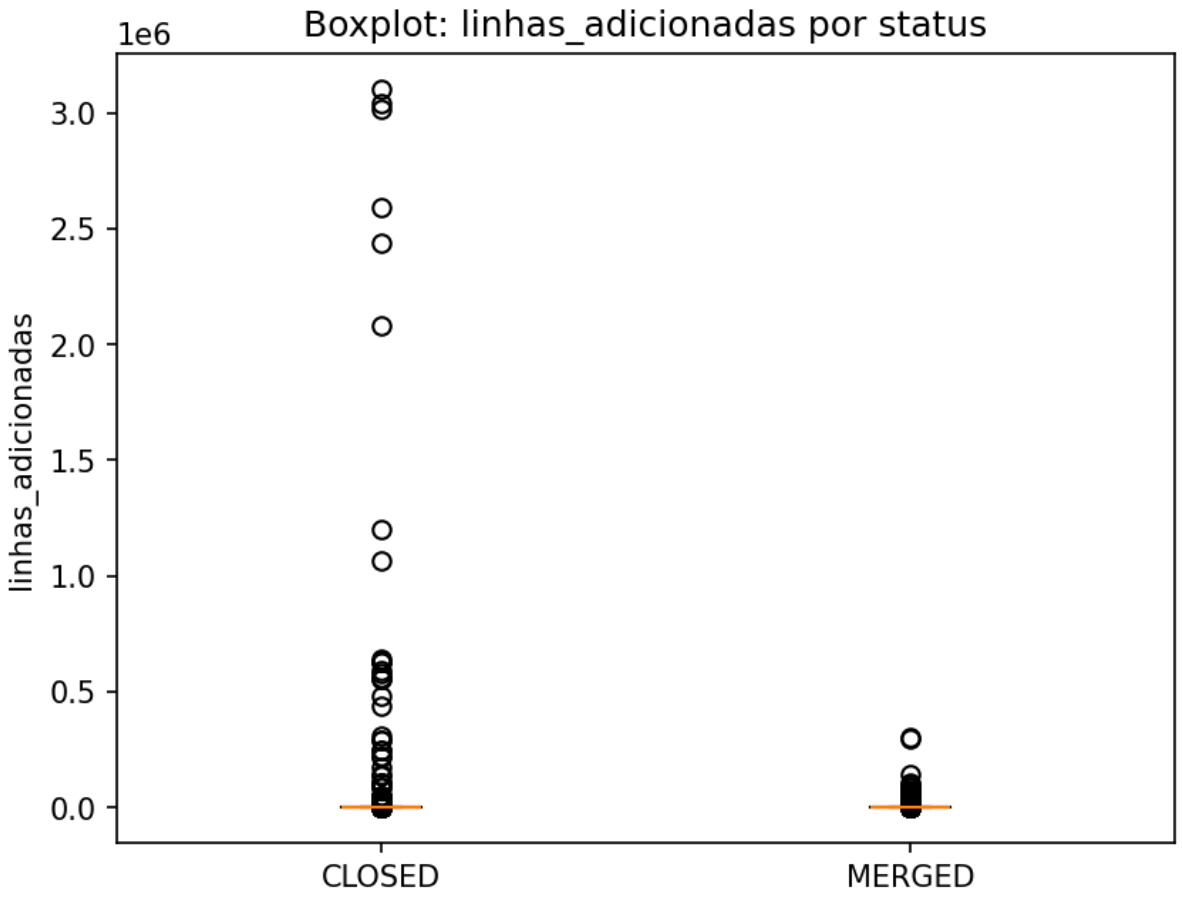

Figura 5: Enter Caption

Figura 6: Distribuição de linhas adicionadas por status

A Figura 6 mostra distribuições notavelmente similares entre os grupos:

- Ambos apresentam medianas extremamente baixas, próximas de zero
- CLOSED apresenta outliers extremos até 3 milhões de linhas
- MERGED apresenta outliers até aproximadamente 300.000 linhas
- A concentração massiva perto de zero indica que a maioria dos PRs são pequenos

Interpretação: Contrariamente à expectativa comum, o tamanho do PR (em linhas adicionadas) não é um forte preditor de aceitação ou rejeição. Tanto PRs aceitos quanto rejeitados tendem a ser pequenos, sugerindo que outros fatores (qualidade, relevância, timing) são mais determinantes.

3.2.3 Linhas de Código Removidas

Figura 7: Distribuição de linhas removidas por status

A Figura 7 apresenta padrão similar ao de linhas adicionadas:

- Medianas praticamente idênticas e próximas de zero para ambos os grupos
- CLOSED com outliers até 380.000 linhas removidas
- MERGED com outliers até 310.000 linhas removidas
- Distribuições altamente concentradas

Observação: A quantidade de código removido também não discrimina significativamente entre PRs aceitos e rejeitados. PRs que envolvem refatoração significativa (com muitas linhas removidas) não apresentam padrão claro de aceitação ou rejeição baseado apenas nesta métrica.

3.2.4 Número de Arquivos Modificados

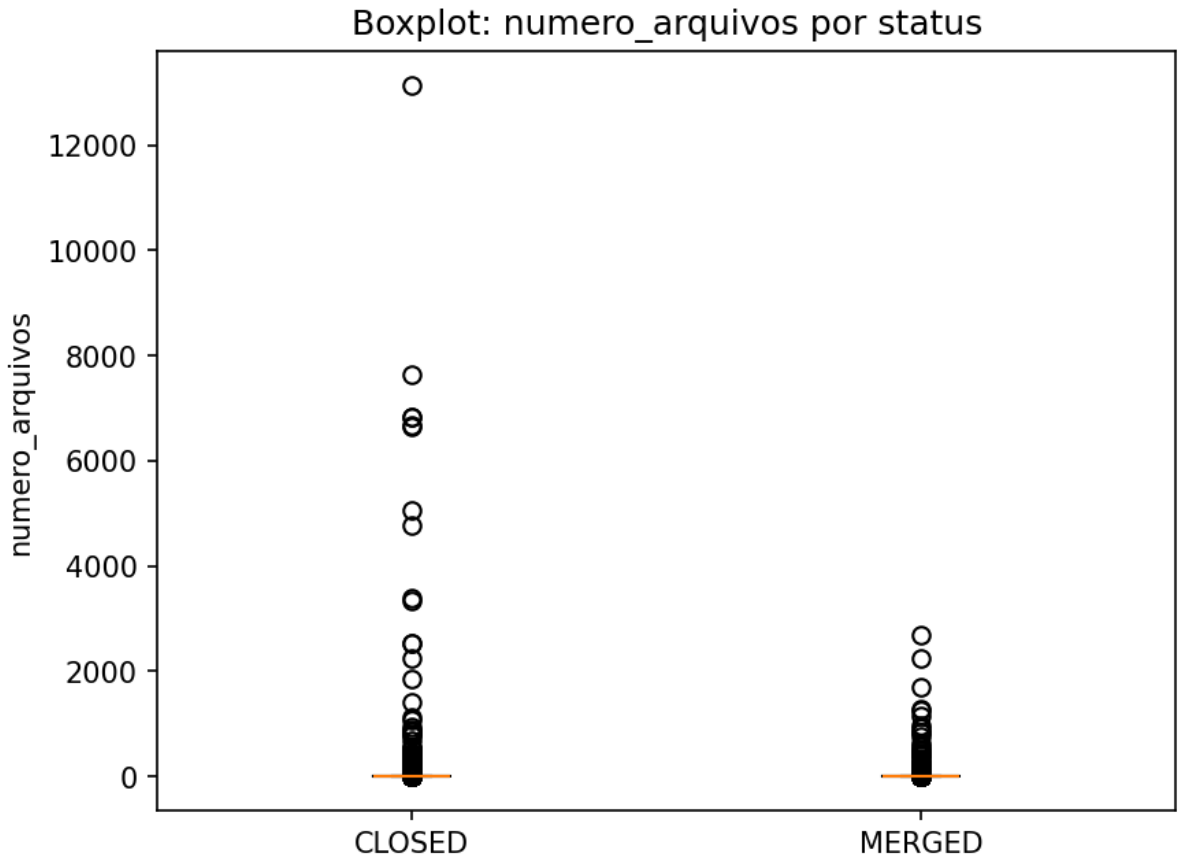

Figura 8: Enter Caption

Figura 9: Distribuição do número de arquivos modificados por status

A Figura 9 revela:

- Medianas muito similares (aproximadamente 1 arquivo) para ambos os grupos
- CLOSED apresenta outlier extremo em 13.000 arquivos
- MERGED apresenta outliers até 2.500 arquivos
- A vasta maioria dos PRs modifica poucos arquivos

Conclusão: Similar às métricas de linhas de código, o número de arquivos modificados não emerge como fator discriminante significativo. A concentração em poucos arquivos é universal.

3.2.5 Número de Participantes

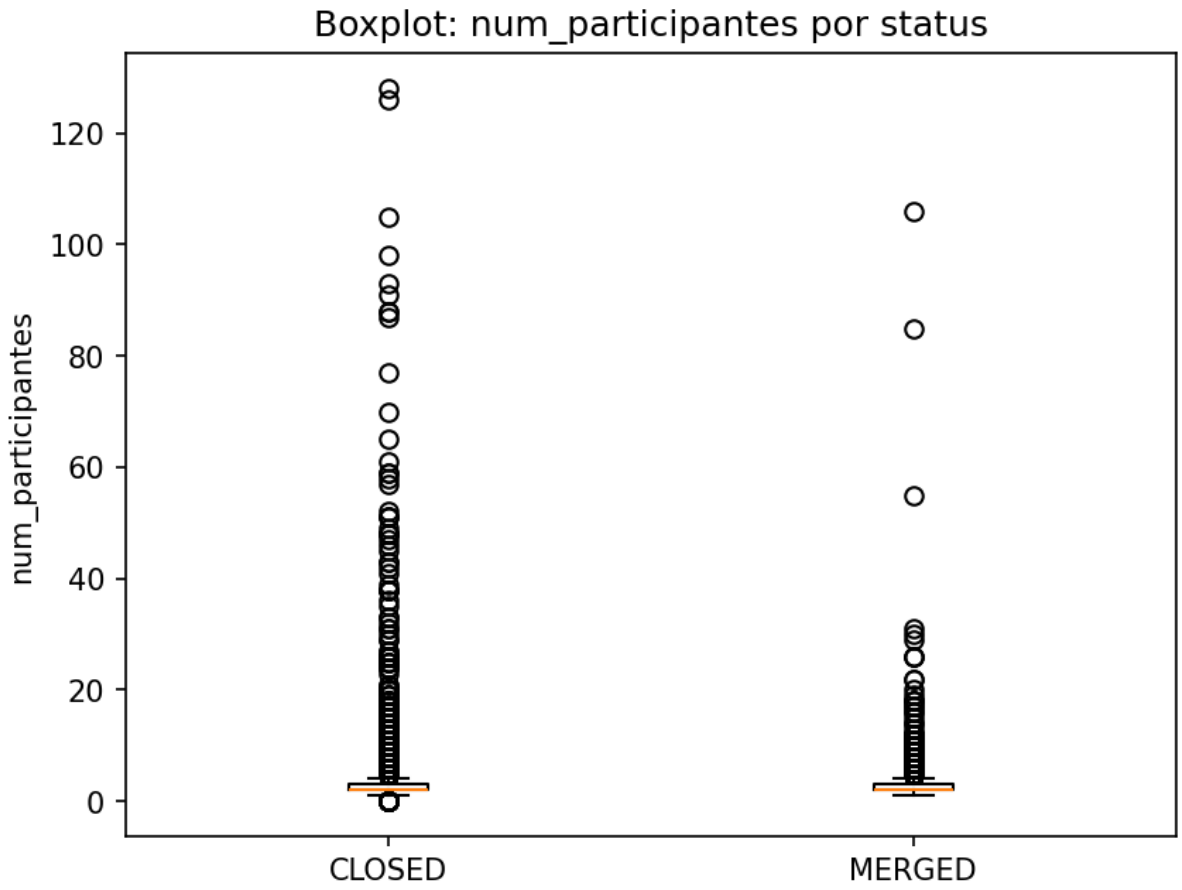

Figura 10: Enter Caption

Figura 11: Distribuição do número de participantes por status

A Figura 11 mostra:

- Medianas similares (2-3 participantes) para ambos os grupos
- CLOSED com outliers até 130 participantes
- MERGED com outliers até 107 participantes
- Concentração forte em poucos participantes (tipicamente 1-5)

Análise: O número de participantes não difere substancialmente entre PRs aceitos e rejeitados em termos de mediana. No entanto, PRs com participação extremamente alta podem indicar controvérsias ou mudanças de grande impacto que requerem amplo consenso.

3.2.6 Número de Comentários

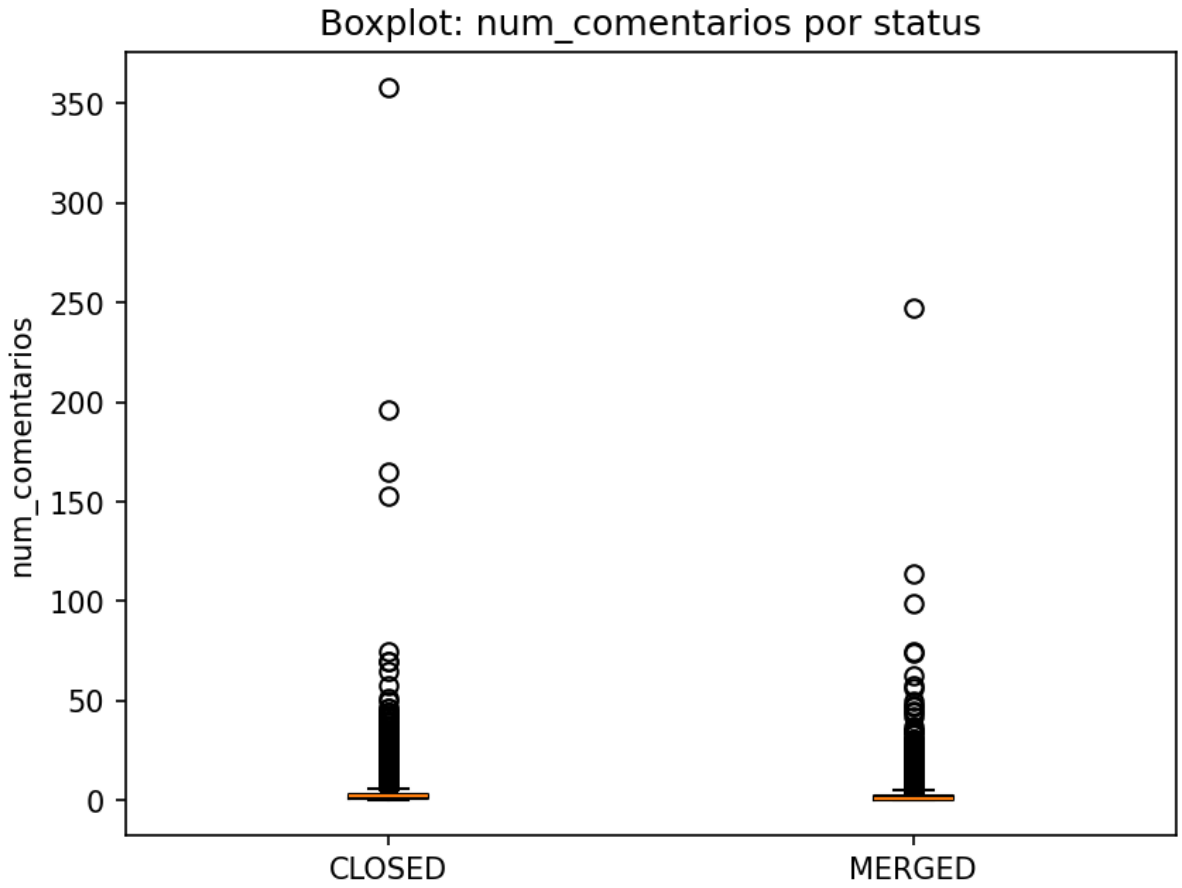

Figura 12: Enter Caption

Figura 13: Distribuição do número de comentários por status

A Figura 13 apresenta:

- Medianas baixas para ambos (1-2 comentários)
- CLOSED com outlier em 360 comentários
- MERGED com outlier em 250 comentários
- Ligeira tendência de mais comentários em PRs rejeitados

Interpretação: Embora as medianas sejam similares, PRs rejeitados tendem a acumular ligeiramente mais comentários, possivelmente refletindo discussões sobre problemas identificados ou tentativas de salvar o PR antes da rejeição final.

3.3 Distribuição Geral de Linhas Adicionadas

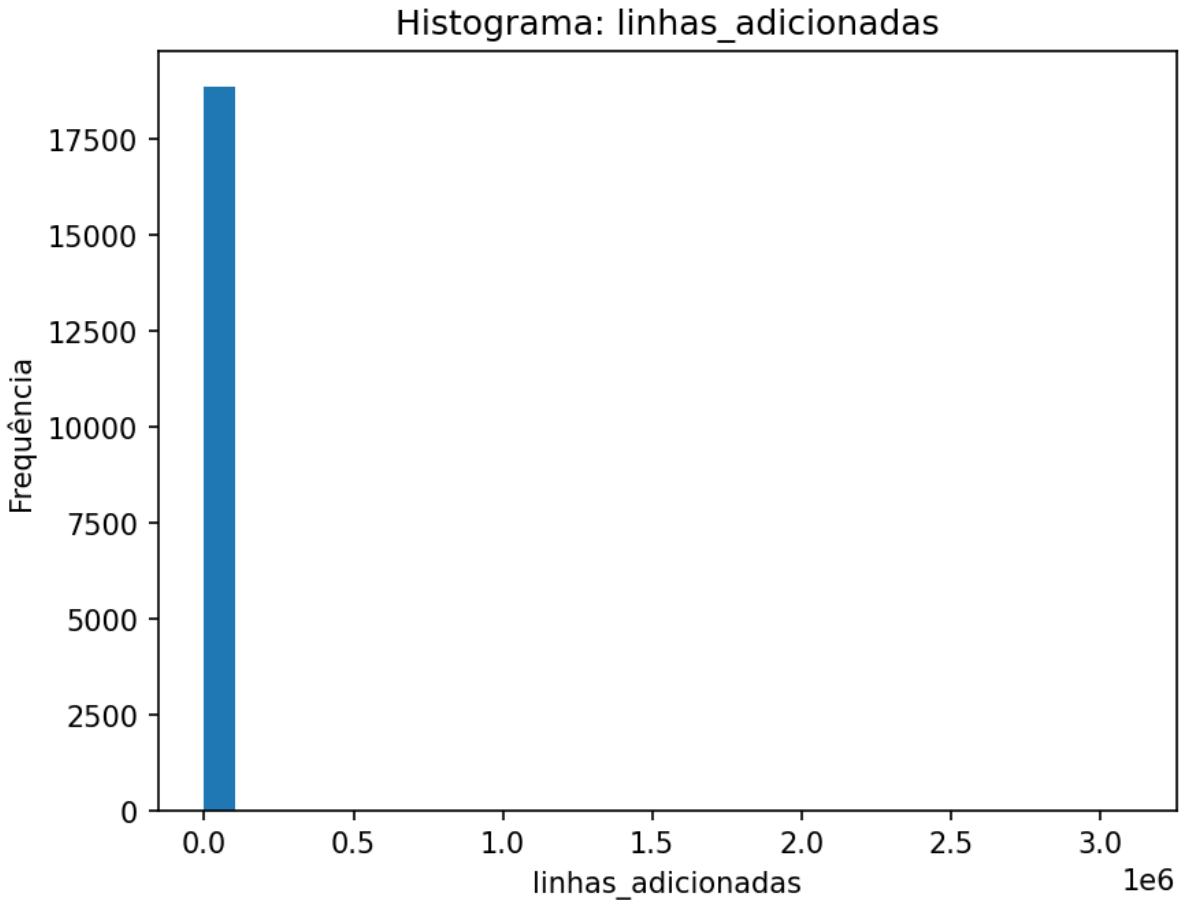

Figura 14: Enter Caption

Figura 15: Histograma da distribuição de linhas adicionadas

A Figura 15 revela uma característica fundamental dos dados:

- Concentração massiva (18.000 PRs) próxima de zero linhas adicionadas
- Distribuição extremamente assimétrica à direita (long-tailed)
- Cauda se estende até aproximadamente 3 milhões de linhas
- Praticamente todos os PRs têm menos de 500.000 linhas adicionadas

Implicação: Esta distribuição é típica de projetos de software, onde pequenas correções, ajustes e melhorias incrementais dominam numericamente, enquanto grandes funcionalidades e refatorações são relativamente raras. A prevalência de PRs pequenos sugere um desenvolvimento iterativo e incremental saudável.

3.4 Relação entre Tamanho e Tempo de Análise

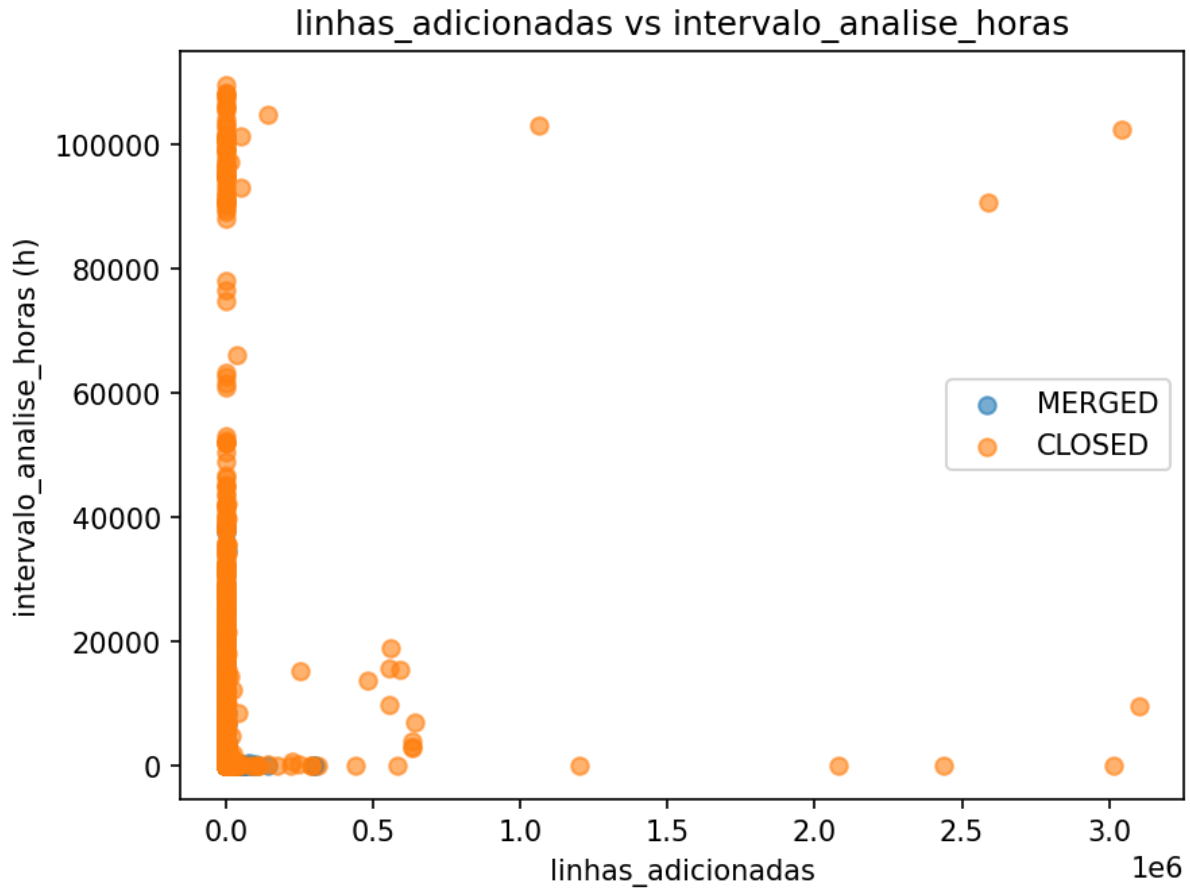

Figura 16: Enter Caption

Figura 17: Relação entre linhas adicionadas e tempo de análise, por status

A Figura 17 ilustra a relação entre tamanho do PR e tempo de análise:

- Forte concentração de pontos na origem (poucos linhas, pouco tempo)
- PRs CLOSED (laranja) dominam visualmente o gráfico
- Alguns outliers extremos em ambas as dimensões
- Relação não é fortemente linear devido à grande dispersão

Análise: Embora exista uma tendência fraca de PRs maiores demandarem mais tempo (confirmada pela correlação de 0.09), a relação é mascarada pela enorme concentração perto da origem e pela influência de outros fatores. O tempo de análise parece mais influenciado por aspectos qualitativos (complexidade, controvérsia, disponibilidade de revisores) do que puramente pelo tamanho.

3.5 Relação entre Participação e Comentários

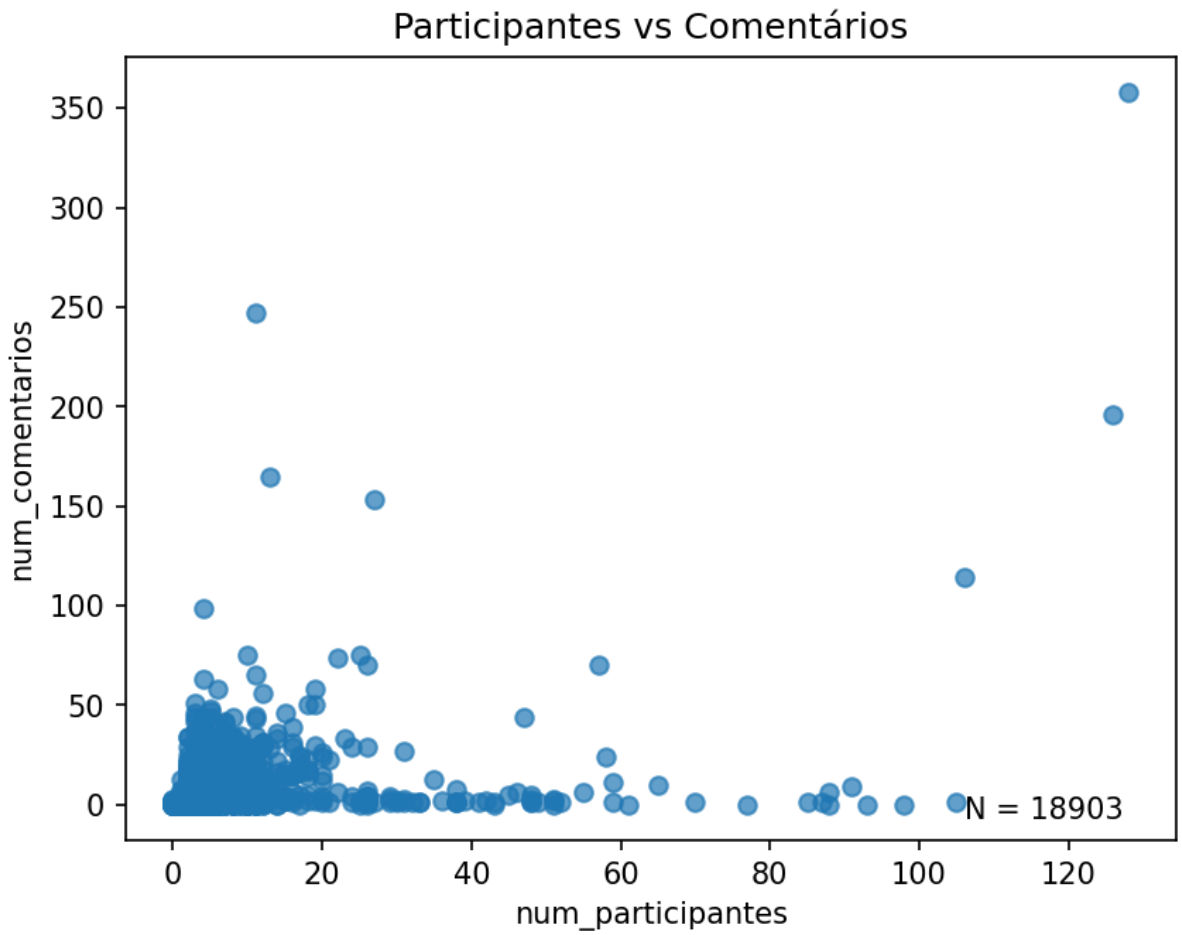

Figura 18: Enter Caption

Figura 19: Relação entre número de participantes e número de comentários (N = 18.903)

A Figura 19 mostra a relação entre participação e discussão:

- Concentração massiva próxima de (0,0), indicando baixa participação na maioria dos PRs
- Relação positiva clara: mais participantes geram mais comentários
- Alguns outliers extremos (¿100 participantes, ¿300 comentários)
- Padrão consistente em toda a amplitude de valores

Interpretação: A correlação positiva moderada (0.56) é intuitiva e esperada. PRs que atraem mais participantes naturalmente geram mais discussão. Casos extremos podem representar mudanças arquiteturais importantes ou propostas controversas que requerem amplo consenso comunitário.

3.6 Análise de Correlação

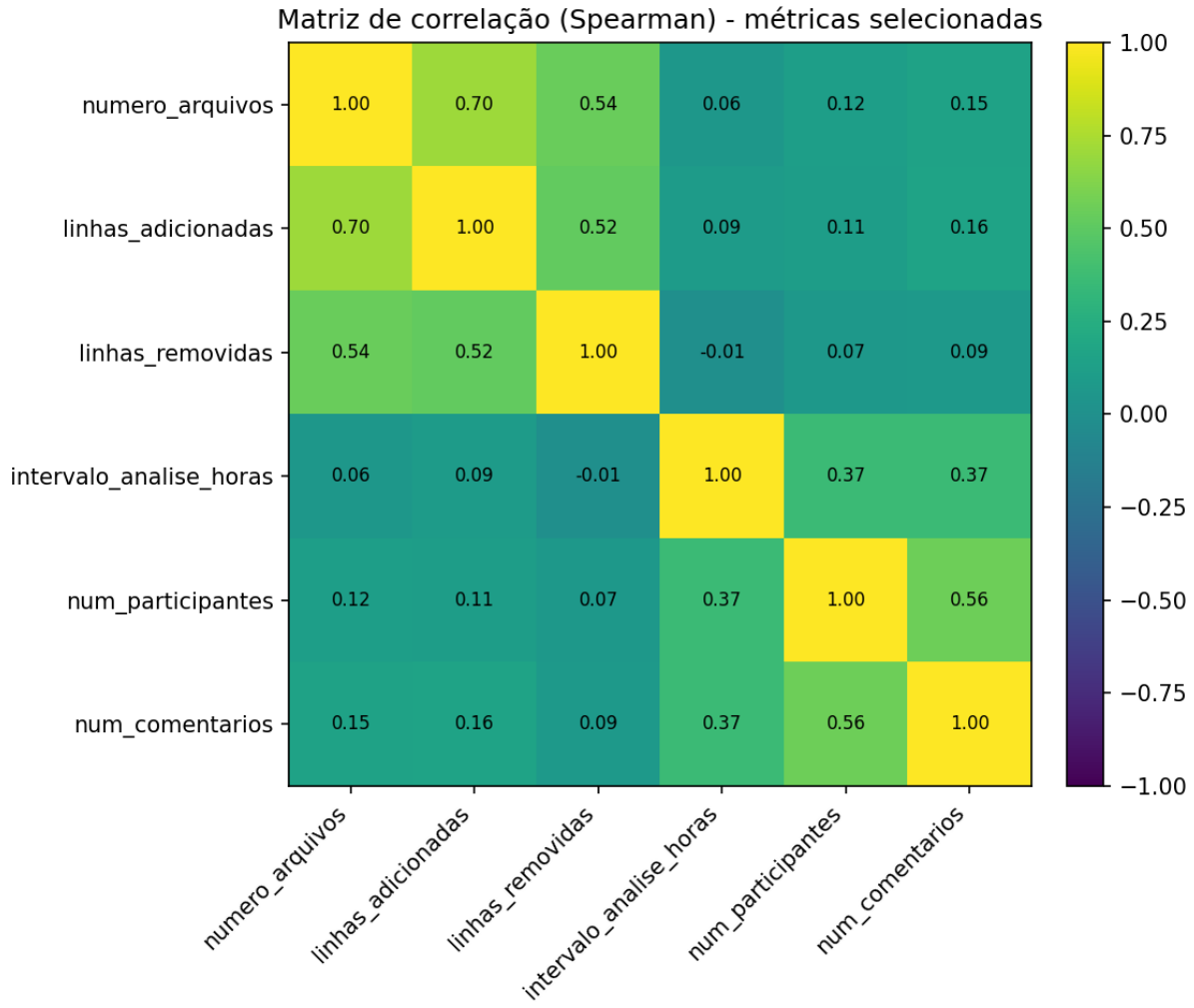

Figura 20: Enter Caption

Figura 21: Matriz de correlação de Spearman entre todas as métricas

A Figura 21 apresenta as correlações de Spearman entre todas as variáveis. Principais achados:

3.6.1 Correlações Fortes (¿ 0.50)

- **numero_arquivos ↔ linhas_adicionadas (0.70)**: Correlação forte e esperada. PRs que modificam mais arquivos naturalmente adicionam mais linhas de código
- **num_participantes ↔ num_comentarios (0.56)**: Correlação moderada-forte. Mais pessoas envolvidas geram mais discussão
- **linhas_adicionadas ↔ linhas_removidas (0.52)**: Correlação moderada, típica de refatorações onde código é substituído

- **numero_arquivos ↔ linhas_removidas (0.54)**: Correlação moderada. PRs que afetam mais arquivos tendem a remover mais código

3.6.2 Correlações Moderadas (0.25-0.50)

- **intervalo_analise_horas ↔ num_participantes (0.37)**: PRs que demoram mais tendem a envolver mais pessoas, possivelmente devido à complexidade ou necessidade de consenso

- **intervalo_analise_horas ↔ num_comentarios (0.37)**: Tempo maior associado a mais discussão

3.6.3 Correlações Fracas (¡ 0.25)

- **Métricas de tamanho ↔ tempo de análise**: Correlações surpreendentemente fracas (0.06-0.16), indicando que o tamanho do PR não é o principal determinante do tempo de revisão

- **Tamanho ↔ engajamento**: Correlações fracas (0.11-0.16), sugerindo que PRs grandes não necessariamente atraem mais participantes ou comentários

Correlação Negativa Notável:

- **intervalo_analise_horas ↔ linhas_removidas (-0.01)**: Praticamente zero, indicando independência total

3.7 Síntese dos Resultados

Os resultados podem ser sintetizados em três achados principais:

1. **Tempo é crítico**: O intervalo de análise é o único fator que difere substancialmente entre PRs aceitos (mediana 5h) e rejeitados (mediana 38h)

2. **Tamanho não discrimina**: Métricas de tamanho (linhas adicionadas/removidas, número de arquivos) apresentam distribuições praticamente idênticas entre PRs aceitos e rejeitados

3. **Estrutura de correlação**: O dataset apresenta duas estruturas correlacionadas: (a) tamanho técnico (arquivos, linhas adicionadas/removidas) e (b) engajamento social (participantes, comentários, tempo)

4 Discussão

4.1 Interpretação dos Achados Principais

4.1.1 A Importância do Tempo de Análise

O resultado mais significativo deste estudo é a diferença marcante no tempo de análise entre PRs aceitos e rejeitados. Com mediana de aproximadamente 38 horas para PRs rejeitados versus 5 horas para aceitos, observa-se que:

- **PRs aceitos são processados rapidamente**: Sugerem alinhamento claro com objetivos do projeto, qualidade adequada, ou urgência
- **PRs rejeitados permanecem em limbo**: Podem refletir indecisão, discussões prolongadas, ou simplesmente abandono

Esta observação tem implicações práticas importantes: desenvolvedores devem monitorar PRs que não recebem atenção rápida e considerar estratégias de follow-up. Mantenedores devem estabelecer SLAs (Service Level Agreements) para revisões iniciais.

4.1.2 O Paradoxo do Tamanho

Contrariamente à sabedoria convencional, o tamanho do PR não é um preditor significativo de aceitação ou rejeição neste dataset. Possíveis explicações:

- **Qualidade sobre quantidade**: A natureza da mudança importa mais que seu tamanho
- **Contexto importa**: Um PR grande pode ser aceito se bem justificado; um pequeno pode ser rejeitado se inadequado
- **Processo de triagem**: PRs claramente inadequados podem ser rejeitados rapidamente independentemente do tamanho

4.1.3 Engajamento Comunitário

A correlação moderada entre participantes e comentários (0.56) confirma que discussão mais ampla ocorre em PRs controversos ou importantes. No entanto, o engajamento não difere substancialmente entre PRs aceitos e rejeitados em termos de mediana, sugerindo que:

- PRs aceitos recebem revisão adequada sem necessariamente gerar discussão extensa
- PRs rejeitados podem acumular comentários devido a tentativas de correção

4.2 Comparação com Literatura

Os resultados são parcialmente consistentes com estudos anteriores:

- **Gousios et al. (2015)**: Identificaram tempo de resposta como fator crítico para sucesso de PRs
- **Yu et al. (2015)**: Demonstraram que tamanho do PR influencia tempo de revisão, mas nossa análise mostra correlação fraca (0.09)
- **Rahman & Roy (2014)**: Encontraram que PRs pequenos têm maior chance de aceitação, mas nossos dados não confirmam claramente esta tendência

As divergências podem refletir diferenças nos repositórios estudados, épocas de coleta, ou metodologias de análise.

4.3 Implicações Práticas

4.3.1 Para Desenvolvedores

Baseado nos resultados, desenvolvedores devem:

1. **Monitorar tempo de resposta**: Se um PR não recebe feedback em 24-48 horas, considerar ping aos mantenedores
2. **Preparar-se para discussão**: PRs que envolvem muitos participantes requerem disponibilidade para responder rapidamente
3. **Não temer PRs grandes**: Se a mudança for bem fundamentada, o tamanho não é impedimento decisivo
4. **Focar na qualidade**: Dado que tamanho não discrimina, investir em documentação, testes e clareza

4.3.2 Para Mantenedores

Mantenedores podem otimizar seus processos através de:

1. **SLAs de revisão inicial**: Comprometer-se a fornecer feedback preliminar em 24-48 horas
2. **Políticas de timeout**: Estabelecer prazos para PRs inativos (e.g., 30 dias sem atividade)
3. **Triagem eficiente**: Identificar rapidamente PRs que não atendem critérios básicos
4. **Ferramentas de automação**: Usar bots para lembrar revisões pendentes
5. **Documentação clara**: Diretrizes explícitas reduzem PRs inadequados

4.3.3 Para Pesquisadores

Este estudo sugere direções futuras:

1. **Análise qualitativa**: Examinar o conteúdo de comentários para entender razões de rejeição
2. **Modelagem preditiva**: Usar machine learning para predizer aceitação baseado em features temporais
3. **Análise longitudinal**: Estudar evolução temporal das práticas de revisão
4. **Análise por domínio**: Comparar padrões entre diferentes tipos de projetos

4.4 Limitações do Estudo

Este trabalho possui limitações importantes que devem ser consideradas na interpretação dos resultados:

4.4.1 Limitações Metodológicas

- **Métricas puramente quantitativas**: Não capturamos qualidade do código, aderência a padrões, ou clareza da documentação
- **Ausência de contexto**: Não consideramos tipo de projeto, maturidade, ou domínio de aplicação
- **Correlação vs. causalidade**: Observamos associações, mas não podemos estabelecer causalidade definitiva
- **Viés de sobrevivência**: Apenas PRs que foram efetivamente abertos estão na amostra (não capturamos contribuições que nunca viraram PR)

4.4.2 Limitações dos Dados

- **Heterogeneidade**: PRs de diferentes repositórios podem ter culturas de revisão distintas
- **Outliers extremos**: Alguns valores extremos (milhões de linhas, anos de análise) podem representar casos especiais ou erros
- **Definição de status**: "CLOSED"inclui tanto rejeições explícitas quanto PRs abandonados pelo autor
- **Falta de informação temporal**: Não sabemos quando os PRs foram criados (mudanças em práticas ao longo do tempo)

4.4.3 Limitações de Generalização

- Os resultados podem não se aplicar a todos os tipos de projetos (empresariais, científicos, educacionais)
- Projetos pequenos podem ter dinâmicas diferentes de projetos muito grandes
- Comunidades com culturas distintas podem apresentar padrões diferentes

4.5 Ameaças à Validade

4.5.1 Validade Interna

- **Confounders não medidos**: Reputação do autor, histórico de contribuições, urgência da feature
- **Multicolinearidade**: Alta correlação entre algumas features pode mascarar efeitos individuais

4.5.2 Validade Externa

- **Representatividade da amostra**: Não sabemos se os repositórios estudados são típicos
- **Época da coleta**: Práticas de desenvolvimento evoluem rapidamente

4.5.3 Validade de Construto

- **Operacionalização de "sucesso"**: MERGED é realmente "sucesso"? Alguns merges podem introduzir problemas
- **Significado de métricas**: Linhas de código não capturam complexidade real

5 Conclusão

Este trabalho apresentou uma análise abrangente de 18.903 Pull Requests do GitHub, investigando relações entre características técnicas, sociais e temporais e o status final dos PRs. Através de visualizações detalhadas e análise estatística, identificamos padrões importantes que contribuem para o entendimento do processo de revisão colaborativa de código.

5.1 Principais Contribuições

1. **Identificação do tempo como fator crítico**: Demonstramos que o intervalo de análise é o principal discriminador entre PRs aceitos (mediana 5h) e rejeitados (mediana 38h), sugerindo que atenção rápida é crucial
2. **Desmistificação do tamanho**: Contrariamente à expectativa comum, o tamanho do PR (em linhas de código ou arquivos) não é preditor significativo de aceitação, indicando que qualidade e contexto superam quantidade
3. **Caracterização da estrutura de correlação**: Identificamos duas dimensões principais: (a) complexidade técnica (arquivos, linhas) com correlação forte interna, e (b) engajamento social (participantes, comentários) moderadamente correlacionado com tempo
4. **Quantificação da distribuição**: Documentamos que a vasta maioria dos PRs são pequenos (concentração massiva próxima de zero linhas), com distribuições fortemente assimétricas para todas as métricas

5.2 Recomendações Finais

Baseado nos resultados, oferecemos as seguintes recomendações:

Para a Comunidade de Desenvolvimento:

- Priorizar respostas rápidas a PRs (feedback em ¡48h)
- Estabelecer expectativas claras sobre tempos de revisão
- Não desencorajar PRs grandes se bem justificados
- Implementar políticas de timeout para PRs inativos

Para Pesquisa Futura:

- Incorporar análise qualitativa de comentários
- Estudar diferenças entre domínios e tipos de projeto

- Desenvolver modelos preditivos para sucesso de PRs
- Investigar o papel da reputação e histórico do autor

5.3 Reflexão Final

Os resultados deste estudo desafiam algumas intuições comuns sobre revisão de código. O fato de que tamanho não prediz aceitação sugere que a comunidade de desenvolvimento open source é mais sofisticada do que simples heurísticas implicariam. O que importa não é o quanto você muda, mas o quão bem você justifica, documenta e alinha sua mudança com os objetivos do projeto.

A diferença marcante no tempo de análise entre PRs aceitos e rejeitados aponta para uma dinâmica temporal crítica: atenção rápida sinaliza relevância e qualidade, enquanto atrasos podem refletir problemas fundamentais ou simplesmente abandono. Este insight reforça a importância de processos de revisão ágeis e responsivos.

Em última análise, este trabalho contribui para uma compreensão mais nuançada e baseada em evidências do desenvolvimento colaborativo de software, fornecendo insights acionáveis para desenvolvedores, mantenedores e pesquisadores. A análise de quase 19.000 PRs oferece uma base sólida para futuras investigações sobre os fatores que tornam contribuições open source bem-sucedidas.

Referências

- Gousios, G., Zaidman, A., Storey, M. A., & Van Deursen, A. (2015). Work practices and challenges in pull-based development: the integrator's perspective. In *Proceedings of the 37th International Conference on Software Engineering* (Vol. 1, pp. 358-368).
- Yu, Y., Wang, H., Filkov, V., Devanbu, P., & Vasilescu, B. (2015). Wait for it: Determinants of pull request evaluation latency on GitHub. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories* (pp. 367-371). IEEE.
- Rahman, M. M., & Roy, C. K. (2014). An insight into the pull requests of GitHub. In *Proceedings of the 11th Working Conference on Mining Software Repositories* (pp. 364-367).
- GitHub API Documentation. Disponível em: `https://docs.github.com/en/rest`
- Tsay, J., Dabbish, L., & Herbsleb, J. (2014). Influence of social and technical factors for evaluating contribution in GitHub. In *Proceedings of the 36th International Conference on Software Engineering* (pp. 356-366).