

Caracterizando a Atividade de Code Review no GitHub

Pedro Franco e Gabriel Pongelupe

Outubro de 2025

1 Informações do Grupo

- **Integrantes:** Pedro Franco e Gabriel Pongelupe
- **Professor:** Danilo Quadros
- **Curso:** Engenharia de Software
- **Disciplina:** Laboratório de Experimentação de Software
- **Período:** 6º

2 Introdução

O processo de *code review* é uma prática essencial nas metodologias ágeis, pois garante que o código desenvolvido seja revisado antes de ser integrado à base principal do projeto. Essa revisão colabora para manter a qualidade do software, corrigir falhas e promover o aprendizado coletivo entre os desenvolvedores.

No contexto do GitHub, o processo é conduzido por meio de *Pull Requests* (PRs), onde colaboradores submetem alterações que passam por análise de revisores. Cada PR pode conter informações como o tamanho da modificação, o tempo até sua aprovação, o nível de detalhamento da descrição e a quantidade de comentários trocados.

A solução proposta neste trabalho consiste em uma análise empírica dessas variáveis, buscando compreender quais fatores influenciam o resultado final de uma revisão (merge ou close) e o número de revisões realizadas em cada PR.

3 Objetivos e Questões de Pesquisa

O objetivo principal deste estudo é **caracterizar o comportamento das revisões de código em repositórios populares do GitHub**, avaliando como diferentes métricas de Pull Requests se relacionam com os resultados das revisões.

Para isso, foram definidas as seguintes **Questões de Pesquisa (RQs)** e suas respectivas métricas analisadas:

- **RQ01:** Qual a relação entre o tamanho dos PRs e o feedback final das revisões?
Métricas coletadas: número de arquivos modificados, linhas adicionadas e linhas removidas.

- **RQ02:** Qual a relação entre o tempo de análise dos PRs e o feedback final das revisões? *Métricas coletadas: tempo total de revisão (diferença entre criação e fechamento/merge, em horas).*
- **RQ03:** Qual a relação entre a descrição dos PRs e o feedback final das revisões? *Métricas coletadas: número total de caracteres presentes no corpo descritivo do PR.*
- **RQ04:** Qual a relação entre as interações nos PRs e o feedback final das revisões? *Métricas coletadas: número de comentários realizados durante a revisão.*
- **RQ05:** Qual a relação entre o tamanho dos PRs e o número de revisões realizadas? *Métricas coletadas: número de arquivos modificados, linhas adicionadas e linhas removidas.*
- **RQ06:** Qual a relação entre o tempo de análise dos PRs e o número de revisões realizadas? *Métricas coletadas: tempo total de revisão (em horas).*
- **RQ07:** Qual a relação entre a descrição dos PRs e o número de revisões realizadas? *Métricas coletadas: número de caracteres no corpo descritivo do PR.*
- **RQ08:** Qual a relação entre as interações nos PRs e o número de revisões realizadas? *Métricas coletadas: número de comentários realizados por participantes.*

Essas RQs orientam o estudo e estruturam a análise dos resultados obtidos, permitindo identificar padrões nas revisões de código no GitHub.

4 Tecnologias e Ferramentas Utilizadas

Para responder às RQs e realizar as análises, foram utilizadas as seguintes tecnologias:

- **Python:** Linguagem base para coleta e processamento dos dados.
- **Pandas:** Biblioteca responsável pela manipulação, limpeza e consolidação dos dados em arquivos CSV.
- **GitHub REST API:** Utilizada para coletar informações de PRs (status, revisões, descrições, comentários, etc.).
- **JavaScript (Chart.js):** Empregada para geração dos gráficos a partir dos dados CSV, visualizando os resultados de cada RQ.

A combinação dessas ferramentas possibilitou uma análise quantitativa das revisões, com visualização clara das relações entre as métricas e os resultados.

5 Hipóteses

Com base nas RQs, foram formuladas as seguintes hipóteses:

- **H1:** PRs menores tendem a ser aceitos com maior frequência.
- **H2:** PRs com descrições mais completas possuem maior taxa de merge.

- **H3:** Um número elevado de interações indica maior complexidade na revisão e menor chance de aceitação.
- **H4:** PRs revisados rapidamente tendem a ser aceitos, enquanto PRs que demoram mais passam por revisões adicionais.

6 Metodologia

A metodologia adotada neste estudo foi dividida em quatro etapas principais: coleta de dados, processamento, análise e visualização. O objetivo foi construir um pipeline completo capaz de extrair informações relevantes sobre Pull Requests (PRs) de repositórios populares do GitHub e gerar gráficos automatizados a partir dessas informações.

6.1 Coleta de Dados

A coleta de dados foi realizada utilizando a **GitHub REST API**, por meio de scripts Python desenvolvidos especificamente para esse propósito:

- **fetch_repos.py:** Responsável por coletar os 200 repositórios mais populares do GitHub e armazená-los em `data/processed/top_repos.csv`.
- **fetch_prs.py:** Coleta os Pull Requests (PRs) de cada repositório, registrando informações como título, autor, status final (*merged* ou *closed*), tempo de análise, número de arquivos modificados, adições, remoções e interações.

As informações foram salvas em formato CSV dentro do diretório `data/raw/`, garantindo reprodutibilidade e facilidade de acesso.

6.2 Processamento dos Dados

Após a coleta, os dados foram tratados utilizando o script `process_data.py`, que:

- Removeu duplicatas e PRs sem revisões manuais.
- Calculou métricas derivadas, como:
 - **Tempo de análise:** diferença entre criação e fechamento/merge (em horas);
 - **Tamanho do PR:** número total de arquivos modificados e linhas alteradas;
 - **Interações:** total de comentários associados ao PR;
 - **Descrição:** contagem de caracteres no corpo textual do PR.
- Gerou a base final de análise em `data/processed/prs_clean.csv`.

Além disso, o script `correlacao.py` foi utilizado para realizar análises estatísticas de correlação (Spearman), permitindo explorar possíveis relações entre as métricas coletadas.

6.3 Visualização e Análise dos Resultados

Os resultados foram visualizados por meio de gráficos interativos gerados com **Chart.js**, uma biblioteca JavaScript integrada no arquivo `index.html`. Essa abordagem permitiu representar visualmente os dados processados a partir do CSV (`prs_clean.csv`), utilizando gráficos de barras e dispersão para cada Questão de Pesquisa (RQ).

6.4 Fluxo de Execução

O fluxo completo do projeto pode ser resumido da seguinte forma:

1. Coleta dos repositórios e PRs via scripts Python.
2. Processamento e limpeza dos dados para gerar o dataset final.
3. Cálculo de métricas e correlações.
4. Geração automática de gráficos via **Chart.js**, embutidos em HTML.

O processo foi desenvolvido e executado em ambiente **Python 3.12**, com dependências gerenciadas pelo arquivo `requirements.txt`, e visualização final implementada em **JavaScript (Chart.js)**.

7 Resultados e Análise das Questões de Pesquisa

A seguir, apresentamos os resultados obtidos para cada questão de pesquisa (RQ), com análise dos valores medianos e suas implicações no processo de code review.

RQ01 – Tamanho dos PRs e Feedback Final

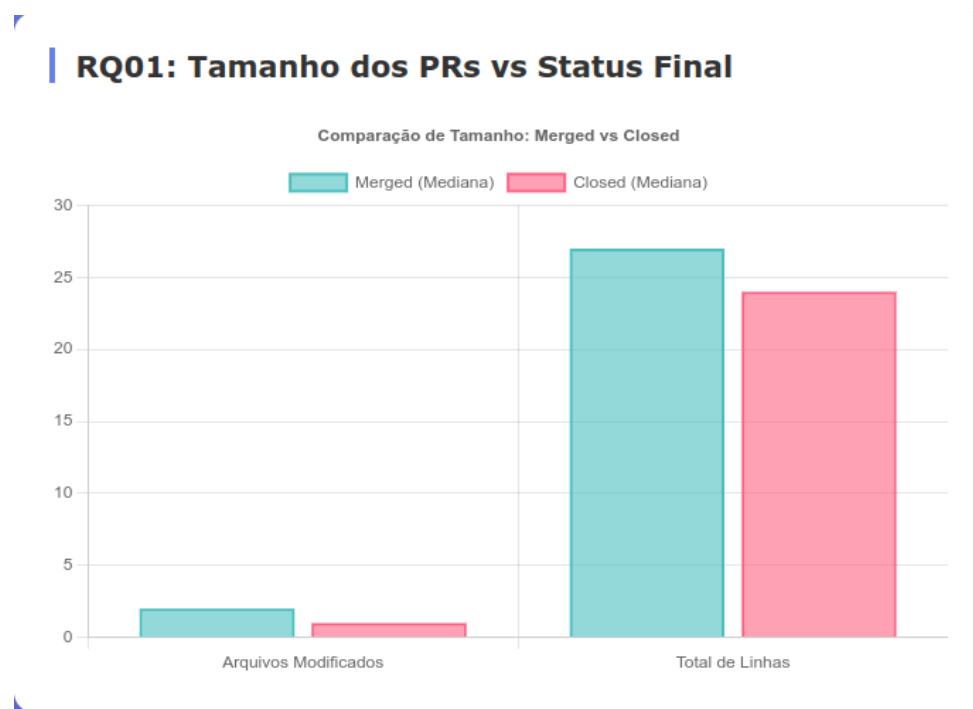


Figura 1: RQ01 — Relação entre tamanho dos PRs e feedback final.

Resultados:

- Arquivos Modificados: Merged (mediana) = 2; Closed (mediana) = 1
- Total de Linhas: Merged (mediana) = 27; Closed (mediana) = 24

Interpretação: Os dados revelam que PRs aceitos (merged) tendem a ser ligeiramente maiores que os rejeitados (closed). PRs aceitos modificam, em média, o dobro de arquivos (2 vs 1) e possuem total de linhas levemente superior (27 vs 24). Embora as diferenças sejam pequenas, este resultado contraria a hipótese inicial de que PRs menores seriam mais aceitos. PRs muito pequenos podem indicar mudanças triviais ou incompletas, enquanto PRs de tamanho moderado demonstram contribuições substanciais e bem estruturadas.

RQ02 – Tempo de Análise e Feedback Final

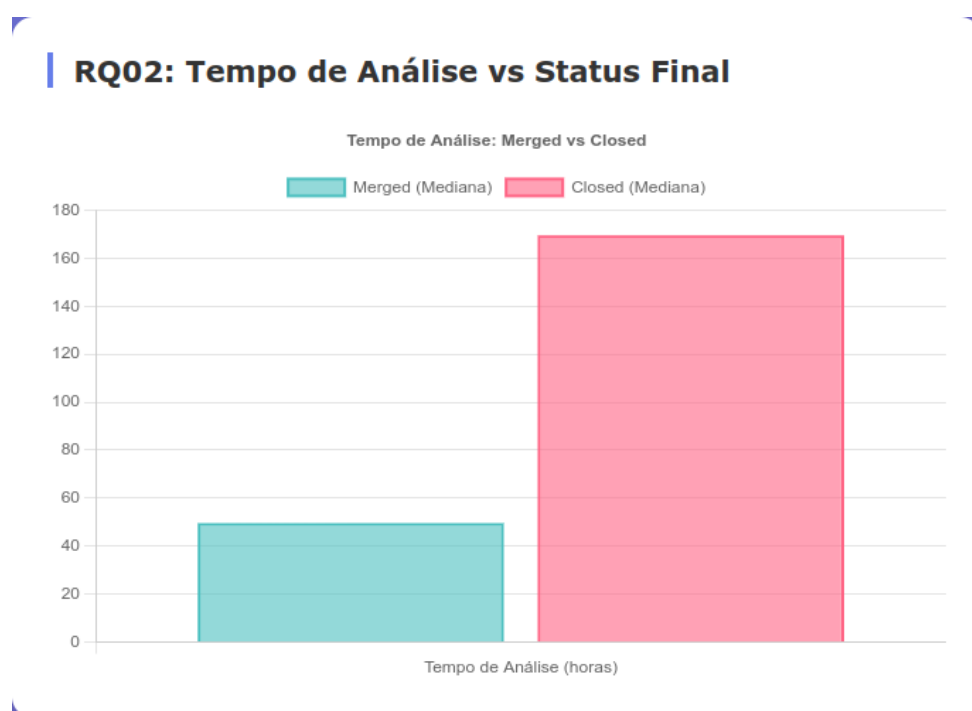


Figura 2: RQ02 — Relação entre tempo de análise e feedback final.

Resultados:

- Tempo de Análise: Merged (mediana) = 49,58 horas; Closed (mediana) = 169,52 horas

Interpretação: Os resultados mostram uma diferença significativa: PRs rejeitados levam aproximadamente 3,4 vezes mais tempo para serem fechados em comparação aos aceitos. PRs merged são processados em cerca de 2 dias, enquanto PRs closed permanecem abertos por aproximadamente 7 dias. PRs que apresentam problemas de qualidade, divergências conceituais ou falta de adesão aos padrões do projeto tendem a gerar discussões mais longas antes da decisão final de rejeição. Por outro lado, PRs bem elaborados são rapidamente identificados e integrados.

RQ03 – Descrição e Feedback Final

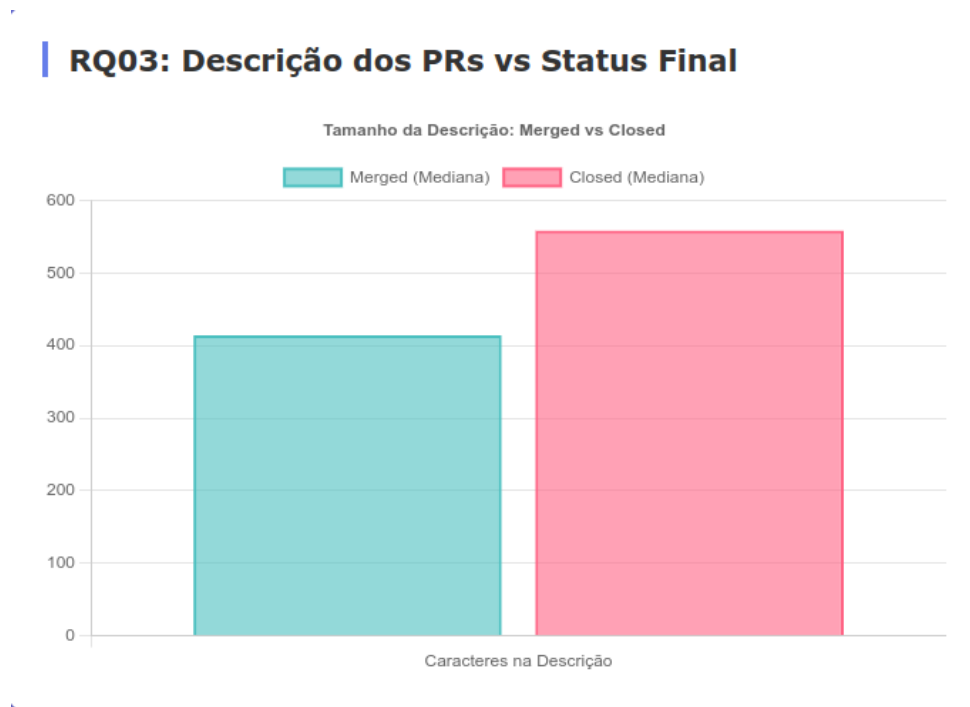


Figura 3: RQ03 — Relação entre descrição e feedback final.

Resultados:

- Caracteres na Descrição: Merged (mediana) = 414; Closed (mediana) = 558

Interpretação: Contrariando a hipótese H2, PRs rejeitados possuem descrições mais extensas (558 caracteres) do que os aceitos (414 caracteres). A diferença de aproximadamente 35% sugere que descrições longas não garantem aprovação. É possível que desenvolvedores tentem justificar mudanças complexas ou controversas com descrições mais detalhadas. Alternativamente, descrições muito longas podem indicar PRs confusos ou que tentam abordar múltiplos problemas simultaneamente. A concisão e clareza parecem ser mais valorizadas que extensão.

RQ04 – Interações e Feedback Final

RQ04: Interações nos PRs vs Status Final

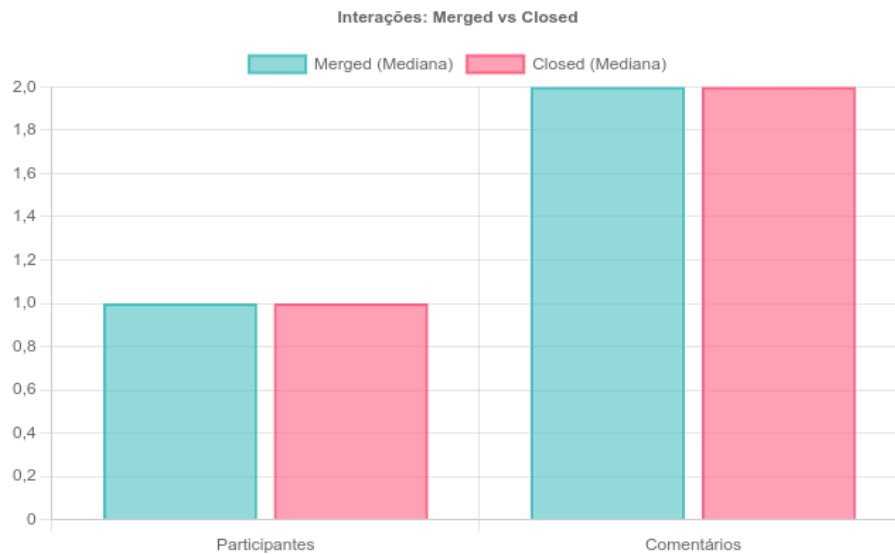


Figura 4: RQ04 — Relação entre número de comentários e feedback final.

Resultados:

- Participantes: Merged (mediana) = 1; Closed (mediana) = 1
- Comentários: Merged (mediana) = 2; Closed (mediana) = 2

Interpretação: Tanto PRs aceitos quanto rejeitados apresentam o mesmo padrão de interação: mediana de 1 participante (além do autor) e 2 comentários. Essa uniformidade indica que o volume de interações não é um fator discriminante no resultado final. A maioria dos PRs passa por revisões objetivas e diretas, sem extensas discussões. O número de comentários não reflete necessariamente complexidade ou problemas — pode simplesmente indicar esclarecimentos pontuais ou aprovação formal.

RQ05 – Tamanho dos PRs e Número de Revisões

RQ05: Tamanho dos PRs vs N° de Revisões

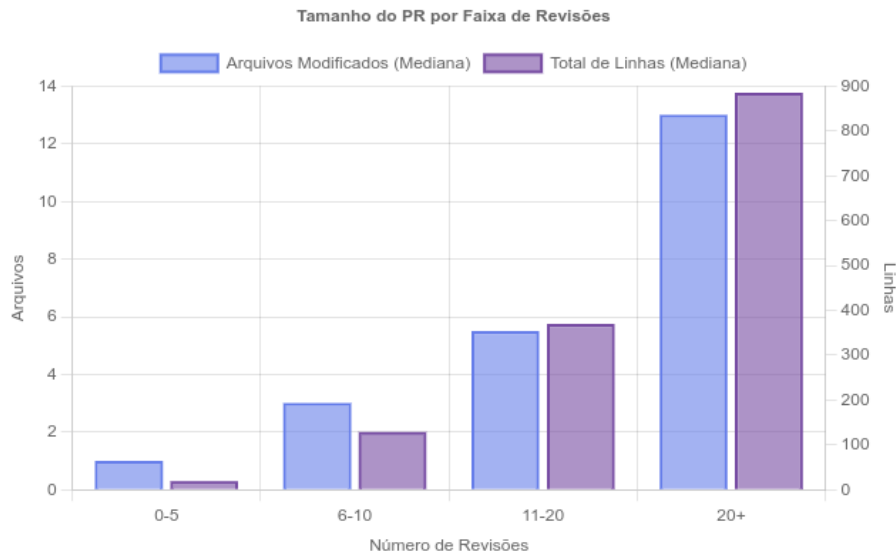


Figura 5: RQ05 — Relação entre tamanho e número de revisões.

Resultados:

- 0-5 revisões: Arquivos = 1; Linhas = 19
- 6-10 revisões: Arquivos = 3; Linhas = 129
- 11-20 revisões: Arquivos = 5,5; Linhas = 369,5
- 20+ revisões: Arquivos = 13; Linhas = 885,5

Interpretação: Existe uma clara correlação positiva entre o tamanho do PR e o número de revisões. PRs com mais de 20 revisões são 13 vezes maiores em arquivos e 46 vezes maiores em linhas modificadas em comparação aos com até 5 revisões. PRs maiores exigem mais ciclos de revisão para garantir qualidade, identificar possíveis problemas e validar todas as mudanças. A complexidade aumenta exponencialmente com o tamanho, justificando múltiplas rodadas de feedback.

RQ06 – Tempo de Análise e Número de Revisões

RQ06: Tempo de Análise vs N° de Revisões

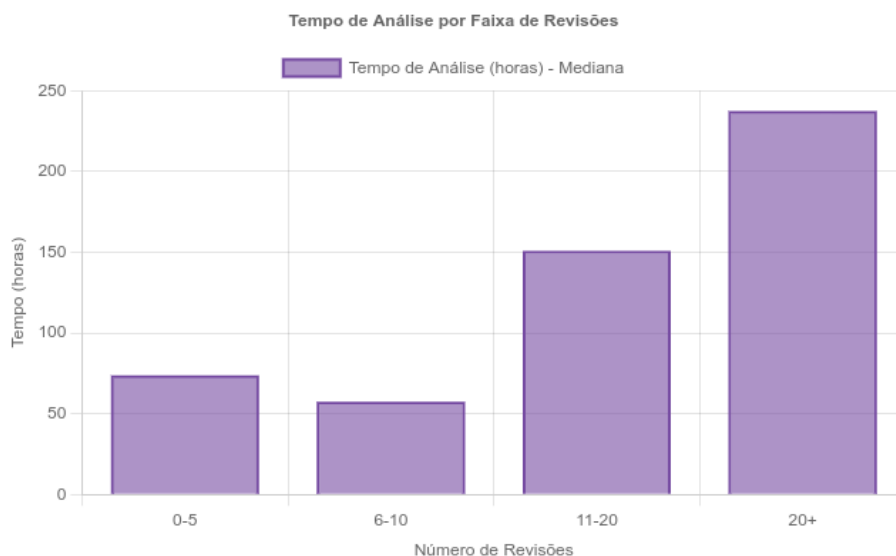


Figura 6: RQ06 — Relação entre tempo de análise e número de revisões.

Resultados:

- 0-5 revisões: 76,87 horas
- 6-10 revisões: 57,44 horas
- 11-20 revisões: 150,83 horas
- 20+ revisões: 237,61 horas

Interpretação: Observa-se uma relação não-linear. PRs com 6-10 revisões apresentam o menor tempo mediano, enquanto aqueles com mais de 20 revisões levam mais de 9 dias. Curiosamente, PRs com poucas revisões (0-5) levam mais tempo que os de 6-10 revisões. PRs simples podem ficar aguardando revisão por falta de prioridade, enquanto PRs moderados recebem atenção mais rápida. PRs com muitas revisões inevitavelmente demandam mais tempo devido aos múltiplos ciclos de feedback e correções.

RQ07 – Descrição dos PRs e Número de Revisões

RQ07: Descrição dos PRs vs N° de Revisões

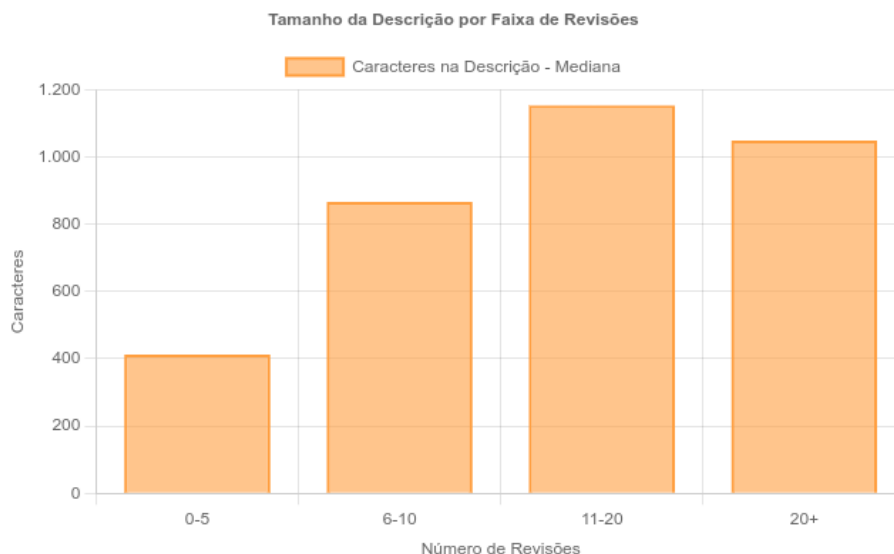


Figura 7: RQ07 — Relação entre descrição e número de revisões.

Resultados:

- 0-5 revisões: 412 caracteres
- 6-10 revisões: 867 caracteres
- 11-20 revisões: 1.152,5 caracteres
- 20+ revisões: 1.049 caracteres

Interpretação: Há uma tendência de aumento no tamanho das descrições conforme o número de revisões cresce, com pico em PRs de 11-20 revisões. PRs com mais de 20 revisões apresentam descrições ligeiramente menores, possivelmente indicando casos especiais. Desenvolvedores tendem a fornecer descrições mais detalhadas para PRs complexos que antecipam múltiplas revisões. Descrições elaboradas podem facilitar o entendimento, mas não eliminam a necessidade de várias rodadas de feedback quando a mudança é substancial.

RQ08 – Interações nos PRs e Número de Revisões

RQ08: Interações nos PRs vs N° de Revisões



Figura 8: RQ08 — Relação entre interações e número de revisões.

Resultados:

- 0-5 revisões: Participantes = 1; Comentários = 2
- 6-10 revisões: Participantes = 1; Comentários = 3
- 11-20 revisões: Participantes = 1; Comentários = 5
- 20+ revisões: Participantes = 1; Comentários = 5,5

Interpretação: O número mediano de participantes permanece constante (1 revisor), mas o número de comentários aumenta progressivamente com mais revisões, passando de 2 para 5,5. Mesmo com múltiplas revisões, a maioria dos PRs é gerenciada por um único revisor principal, indicando processos bem definidos de responsabilidade. O aumento de comentários reflete as discussões necessárias em PRs mais complexos.

Matriz de Correlação Observada

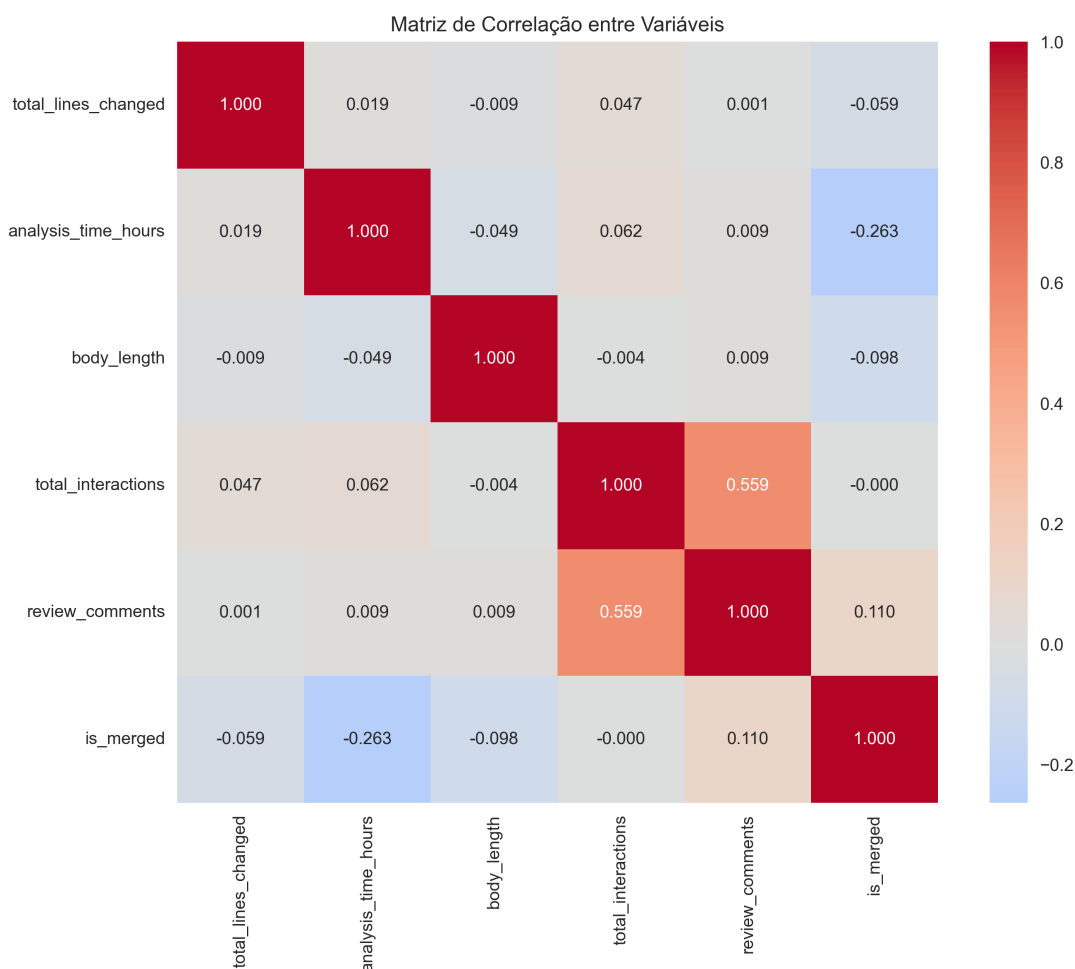


Figura 9: Matriz de correlação (Spearman) entre variáveis do dataset.

Para quantificar as associações entre as métricas, utilizamos o coeficiente de Spearman (ρ), adequado para relações monotônicas e para dados que podem não ser estritamente paramétricos. A matriz acima foi calculada entre as seguintes variáveis (nomes aqui traduzidos para o relatório):

- **total_lines_changed:** soma de adições e remoções (linhas alteradas).
- **analysis_time_hours:** duração da revisão (horas).
- **body_length:** tamanho da descrição do PR (nº de caracteres).
- **total_interactions:** comentários totais (comments + review_comments).
- **review_comments:** número de comentários de revisão.
- **is_merged:** indicador binário de merge (1) ou closed (0).

Em resumo: os sinais mais claros na matriz são (i) **tempo de análise negativo** com merge (maior tempo \rightarrow menor chance de merge) e (ii) **forte associação entre interações e comentários de revisão**. As demais associações com o status do PR são fracas ou desprezíveis.

8 Discussão

Com base nas correlações observadas, reavaliemos as hipóteses propostas:

H1 (PRs menores tendem a ser aceitos): *Refutada ou, na melhor das hipóteses, não suportada.* A correlação entre tamanho (linhas alteradas) e merge é muito fraca ($\rho \approx -0.059$) — não há evidência de que PRs menores tenham probabilidade maior de merge no dataset analisado.

H2 (PRs com descrições mais completas possuem maior taxa de merge): *Refutada.* A correlação entre `body_length` e `is_merged` é fraca negativa ($\rho \approx -0.098$), indicando que descrições mais longas não aumentam a chance de merge — se há efeito, é mínimo e ligeiramente negativo.

H3 (Mais interações indicam menor chance de aceitação): *Não confirmada.* A correlação entre `total_interactions` e `is_merged` é praticamente nula ($\rho \approx -0.000$). Já o número de comentários de revisão tem ligeira correlação positiva com merge ($\rho \approx 0.110$). Isso sugere que nem todo aumento de interação indica problema — conversas de revisão podem também levar à melhoria do PR e ao merge.

H4 (PRs revisados rapidamente tendem a ser aceitos, enquanto PRs que demoram mais passam por revisões adicionais): *Parcialmente confirmada.* Observa-se uma correlação **negativa moderada** entre tempo de análise e merge ($\rho \approx -0.263$): PRs que demoram mais têm menor probabilidade de serem mergeados. Isso corrobora o aspecto temporal da hipótese. Quanto a “passarem por revisões adicionais”, a matriz mostra que tempo tem correlação pequena com `review_comments` (0.009) — ou seja, o tempo por si só não está fortemente associado ao número de comentários de revisão na mesma matriz, mas está associado ao desfecho negativo (closed). Possíveis interpretações: PRs que geram impasse (e.g., discussões conceituais, falta de acordo) permanecem abertas por mais tempo e acabam fechadas; PRs com ciclo de revisão normal e produtivo tendem a ser resolvidos mais rápido e mergeados.

9 Conclusão

Reavaliando o estudo com base na matriz de correlação real:

- O **tempo de análise** é o fator mais claramente associado ao **desfecho negativo** (PR fechado), com correlação negativa moderada ($\rho \approx -0.263$).
- O **número de interações/comentários** está fortemente associado entre si (comentários \times interações; $\rho \approx 0.559$) e os comentários de revisão mostram uma fraca correlação positiva com merge ($\rho \approx 0.110$), sugerindo que discussões construtivas ajudam na integração.
- **Tamanho do PR** (linhas alteradas) e **tamanho da descrição** não se mostram bons preditores do status final (correlações próximas de zero ou fracas).

Portanto, as recomendações práticas derivadas destes achados são:

- Mantenedores devem priorizar revisões ágeis para reduzir a chance de fechamento por abandono ou impasse.
- Desenvolvedores devem engajar-se em diálogos produtivos com revisores (comentários focados) — isso pode aumentar a probabilidade de merge.

- Evitar assumir que apenas reduzir o tamanho do PR ou aumentar a extensão da descrição por si só aumentará a chance de aceitação.

Respostas às Questões de Pesquisa (RQs)

Com base nos resultados e análises apresentadas, as RQs foram respondidas da seguinte forma:

- **RQ01:** PRs aceitos tendem a ser levemente maiores (em arquivos e linhas modificadas) que os rejeitados, contrariando a hipótese de que PRs menores seriam mais aceitos.
- **RQ02:** Existe uma relação clara entre tempo de análise e feedback final — PRs mergeados são revisados cerca de 3 vezes mais rápido que PRs fechados, indicando que revisões ágeis favorecem o merge.
- **RQ03:** PRs rejeitados possuem descrições mais longas que os aceitos. Descrições muito extensas não aumentam a taxa de merge e podem indicar PRs complexos ou confusos.
- **RQ04:** O número de interações (comentários) é semelhante em PRs aceitos e rejeitados, sugerindo que o volume de comentários não influencia diretamente o resultado final.
- **RQ05:** Há forte relação positiva entre o tamanho do PR e o número de revisões — PRs maiores passam por mais ciclos de feedback.
- **RQ06:** O tempo de análise cresce à medida que o número de revisões aumenta, especialmente em PRs com mais de 20 revisões, que podem levar mais de 9 dias para conclusão.
- **RQ07:** PRs com mais revisões tendem a ter descrições mais detalhadas, sugerindo que desenvolvedores fornecem mais contexto em mudanças complexas.
- **RQ08:** O número de comentários cresce com o número de revisões, mas o número de participantes permanece constante (geralmente um revisor principal), indicando revisões concentradas e consistentes.

Em síntese, o estudo mostrou que o fator mais relevante para o sucesso de um *Pull Request* é o **tempo de análise**, seguido pela **complexidade medida em número de revisões**, enquanto o tamanho e a extensão das descrições possuem efeito limitado no resultado final.