Caracterizando a Atividade de Code Review no GitHub

Matheus Machado de Oliveira Andrade[1], Rafael Parreira Chequer[2], Samara Martins Ferreira[3], Victor Alexandre Peters Fonseca[4]

Instituto de Ciências Exatas e Informática
Pontifícia Universidade de Minas Gerais (PUC Minas)
Belo Horizonte – MG – Brasil

{500391, 1275759, 1328767, 1314281}@sga.pucminas.br

Resumo. *Este laboratório, fará uma análise da atividade de* ***code review*** *em repositórios populares do* ***GitHub****. O foco será identificar as variáveis que influenciam na decisão de* ***merge*** *de um* ***Pull Request (PR)****. A análise será feita sob a perspectiva dos desenvolvedores que submetem código aos repositórios selecionados. Isso envolve a inspeção do código produzido antes de sua integração à base principal, garantindo a qualidade do código e evitando a inclusão de defeitos. Além disso, será considerado o papel das ferramentas de verificação estática que avaliam requisitos de estilo de programação ou padrões definidos pela organização.*

1. Introdução

A prática de code review tornou-se uma constante nos processos de desenvolvimento ágeis. Em linhas gerais, ela consiste na interação entre desenvolvedores e revisores visando inspecionar o código produzido antes de integrá-lo à base principal. Assim, garante-se a qualidade do código integrado, evitando-se também a inclusão de defeitos. No contexto de sistemas open source, mais especificamente dos desenvolvidos através do GitHub, as atividades de code review acontecem a partir da avaliação de contribuições submetidas por meio de Pull Requests (PR). Ou seja, para que se integre um código na branch principal, é necessário que seja realizada uma solicitação de pull, que será avaliada e discutida por um colaborador do projeto. Ao final, a solicitação de merge pode ser aprovada ou rejeitada pelo revisor. Em muitos casos, ferramentas de verificação estática realizam uma primeira análise, avaliando requisitos de estilo de programação ou padrões definidos pela organização.

Neste contexto, o objetivo deste laboratório é analisar a atividade de code review desenvolvida em repositórios populares do GitHub, identificando variáveis que influenciam no merge de um PR, sob a perspectiva de desenvolvedores que submetem código aos repositórios selecionados.

1.1. Questões de Pesquisa e Hipóteses Informais

A. Feedback Final das Revisões (Status do PR):

RQ 01 - Qual a relação entre o tamanho dos PRs e o feedback final das revisões?

O tamanho dos PRs pode influenciar o feedback final das revisões. PRs muito grandes podem ser mais difíceis de revisar e, portanto, podem ter mais chances de serem rejeitados.

RQ 02 - Qual a relação entre o tempo de análise dos PRs e o feedback final das revisões?

O tempo de análise dos PRs também pode afetar o feedback final. PRs que levam muito tempo para serem analisados podem indicar problemas complexos, o que pode levar a um feedback negativo.

RQ 03 - Qual a relação entre a descrição dos PRs e o feedback final das revisões?

A descrição dos PRs é crucial para o feedback final. Descrições claras e detalhadas podem facilitar o processo de revisão e aumentar as chances de aprovação do PR.

RQ 04 - Qual a relação entre as interações nos PRs e o feedback final das revisões?

As interações nos PRs, como comentários e discussões, podem influenciar o feedback final. PRs com muitas interações podem indicar um alto nível de colaboração, o que pode levar a um feedback mais positivo.

B. Número de Revisões:

RQ 05 - Qual a relação entre o tamanho dos PRs e o número de revisões realizadas?

O tamanho dos PRs pode influenciar o número de revisões realizadas. PRs maiores podem exigir mais revisões para garantir a qualidade do código.

RQ 06 - Qual a relação entre o tempo de análise dos PRs e o número de revisões realizadas?

O tempo de análise dos PRs pode afetar o número de revisões. PRs que levam mais tempo para serem analisados podem exigir mais revisões.

RQ 07 - Qual a relação entre a descrição dos PRs e o número de revisões realizadas?

A descrição dos PRs pode influenciar o número de revisões. PRs com descrições claras e detalhadas podem facilitar o processo de revisão e, portanto, exigir menos revisões.

RQ 08 - Qual a relação entre as interações nos PRs e o número de revisões realizadas?

As interações nos PRs podem afetar o número de revisões. PRs com muitas interações podem indicar um alto nível de colaboração e, portanto, podem exigir mais revisões para considerar todas as opiniões e sugestões.

2. Metodologia

2.1. Definição do Problema e Formulação das hipóteses

O problema em questão envolve a análise da atividade de code review em repositórios populares do GitHub, com o objetivo de identificar as variáveis que influenciam na decisão de mesclar um Pull Request (PR). Isso implica entender como os desenvolvedores submetem e revisam código, e quais fatores levam à aprovação ou rejeição de uma contribuição. A partir da definição do problema foram estabelecidas hipóteses para cada uma das questões de pesquisa que são a base para a fase de coleta de dados.

2.2. Coleta de Dados

Para a etapa de coleta de dados foi desenvolvido um código Python para coletar os dados e armazenar informações sobre repositórios do GitHub e seus respectivos pull requests. O código utiliza uma lista de tokens para autenticação nas requisições à API do GitHub que são rotacionados para evitar overflow e timeout no fluxo. As requisições são realizadas manipulando e gerenciando possíveis erros e estabelecendo uma pausa caso o acesso seja interrompido pelo limite da API.

Executa-se uma consulta GraphQl para buscar repositórios mais populares do GitHub baseado no número de estrelas, obtendo os 200 repositórios que atendam ao critério utilizando paginação. A lista com os repositórios é transformada em um DataFrame do pandas e depois é salva em um arquivo CSV (processed_data). Para cada repositório, coleta dados detalhados de seus pull requests usando outra consulta GraphQL. Para cada um desses repositórios listados é executado uma nova query GraphQl para obter os dados dos pull requests relacionados a eles, foram selecionados mais de 700 mil linhas de dados e esses foram salvos em CSV (processed_data_pullRequests).

2.3. Processamento

Um segundo script Python foi confeccionado para analisar os dados de pull requests armazenados no arquivo CSV, os dados são convertidos para o tipo correto e as datas são manipuladas. Estas funções de conversão permitem que as operações matemáticas não falhem devido a tipos de dados incorretos. A manipulação das datas de criação, merge e fechamento dos pull requests gera o dado de “tempo de análise” como a diferença em dias entre a data de criação e a última atividade (mesclada ou fechada).

Após processar todas as linhas do CSV, o script cria um DataFrame do pandas com esses dados processados, removendo as linhas com dados não utilizados ou nulos. Os dados são agrupados por nome e dono do repositório e para cada grupo, o script calcula: contagem total de pull requests; soma dos reviews; total de arquivos; total de interações, pelos comentários; comprimento dos textos de descrição, pelo número de caracteres; e a média do tempo de análise.

Os resultados são direcionados a um DataFrame que é salvo em um novo arquivo CSV (processed_data_summary.csv).

2.4. Análise de Dados e Interpretação dos Resultados

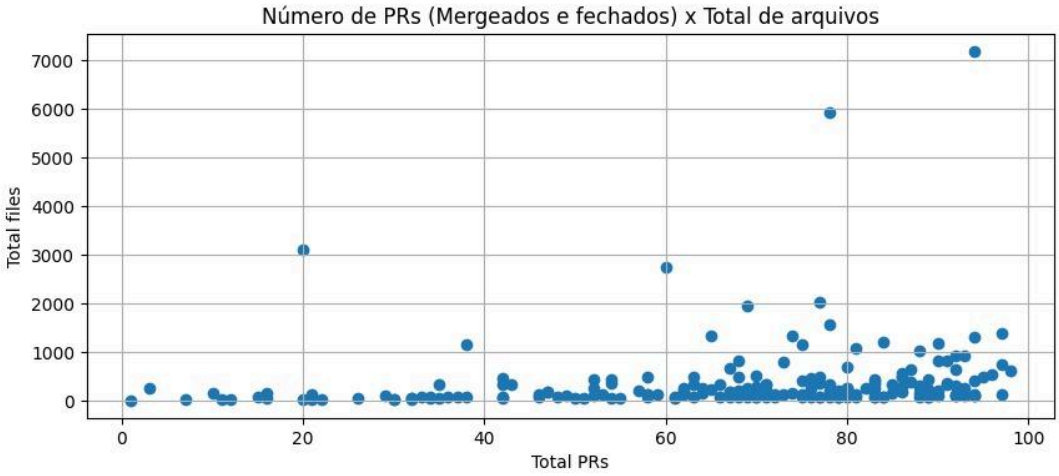
Um terceiro código Python faz uma análise exploratória de dados extraídos dos pull requests. Ele foca nas relações entre diferentes métricas associadas aos dados que foram solicitadas pelas Questões de Pesquisa.

Os dados do arquivo CSV (processed_data_summary.csv) é estruturado em um DataFrame do pandas. Os dados são carregados e para cada Questão de Pesquisa é gerado um gráfico de dispersão e a partir desses gráficos é feita uma análise dos plots para gerar uma interpretação fundamentada das informações geradas. Além disso, o código gerou uma matriz de correlação de Spearman para todas as métricas que foram utilizadas.

3. Resultados

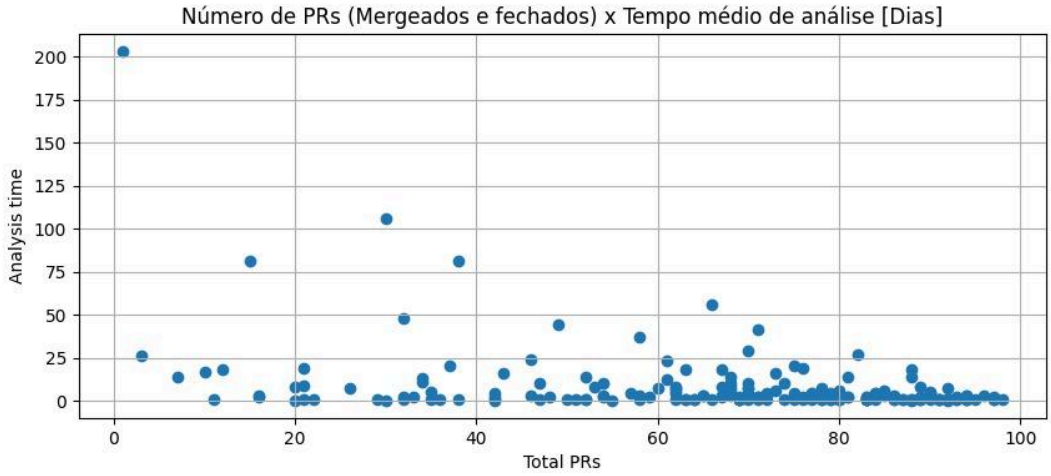
Para a etapa de coleta de resultados serão apresentados os gráficos gerados para responder a cada uma das Research Questions solicitadas, assim como as suas análises.

RQ 01 - Qual a relação entre o tamanho dos PRs e o feedback final das revisões?



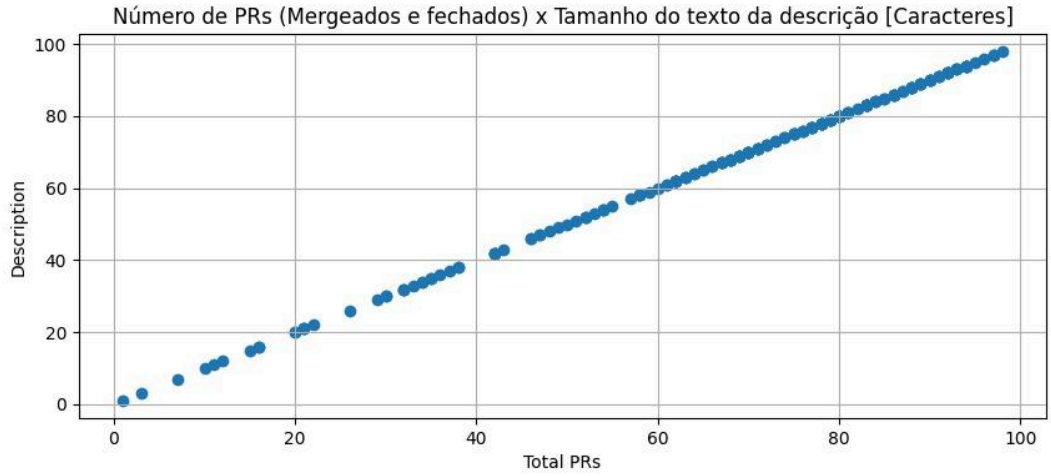
O gráfico referente à RQ 01 mostra principalmente uma aglomeração de pontos próximo ao eixo horizontal, indicando que repositórios com um menor número de PRs tendem a ter menos arquivos alterados. Observam-se alguns outliers com uma quantidade significativamente maior de arquivos para um número baixo de PRs. Além disso, a frequência de PRs com muitos arquivos cai conforme aumenta o número de PRs, sugerindo que repositórios mais ativos possuem, em média, PRs menores ou com uma quantidade moderada de arquivos.

RQ 02 - Qual a relação entre o tempo de análise dos PRs e o feedback final das revisões?



Analisando o gráfico de RQ 02, vemos uma clara concentração de PRs que possuem um tempo de análise baixo, o que sugere que a maioria dos PRs é processada rapidamente. Existem alguns casos atípicos onde o tempo de análise é muito maior, indicados pelos pontos afastados da concentração principal. Esses outliers podem representar PRs que são mais complexos ou que encontraram algum tipo de atraso no processo de revisão. Quanto à distribuição dos PRs, a densidade de pontos diminui à medida que o tempo de análise aumenta, o que é esperado já que menos PRs tendem a ter problemas que levam a análises mais longas.

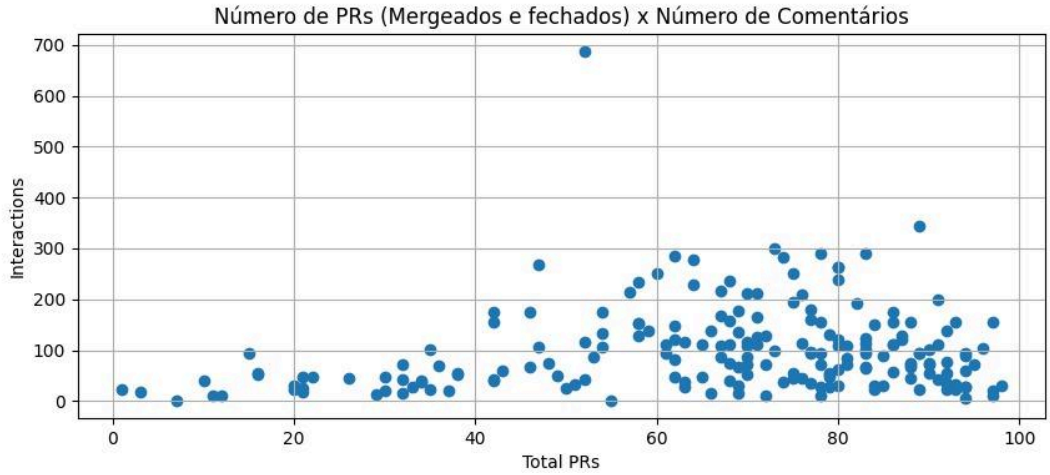
RQ 03 - Qual a relação entre a descrição dos PRs e o feedback final das revisões?



No gráfico referente à RQ 03, que relaciona o número de PRs com o tamanho do texto de descrição, percebe-se uma tendência linear muito clara: à medida que o número de PRs aumenta, o tamanho da descrição também aumenta proporcionalmente. Não há outliers evidentes, e a distribuição dos pontos sugere uma correlação direta e forte entre as duas

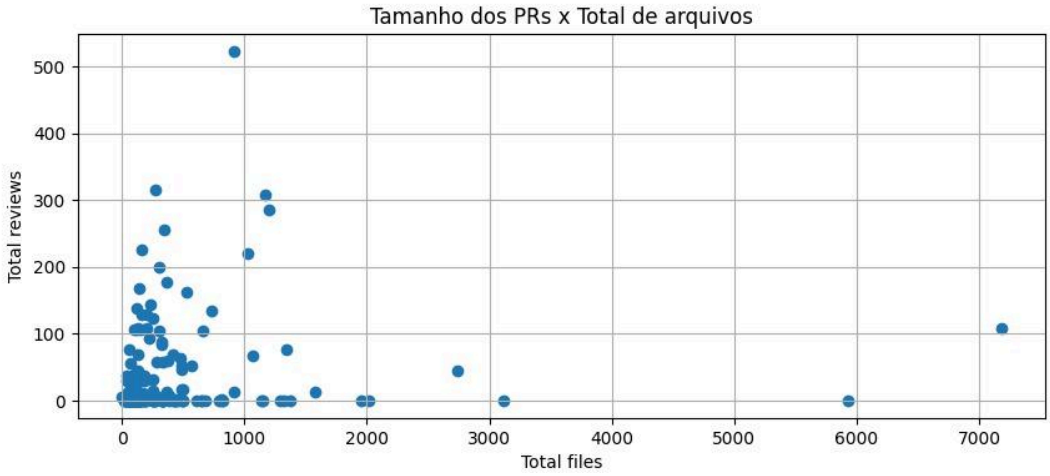
variáveis, o que pode indicar uma tendência dos desenvolvedores de fornecer descrições mais detalhadas em repositórios com maior atividade de PRs.

RQ 04 - Qual a relação entre as interações nos PRs e o feedback final das revisões?



No gráfico para RQ 04, que relaciona o número de PRs com o número de comentários, vemos que a maioria dos PRs tem uma quantidade relativamente baixa de comentários, indicado pela alta concentração de pontos perto do eixo horizontal. Alguns PRs apresentam um número maior de comentários, destacando-se como outliers. A distribuição dos comentários pelos PRs não sugere uma relação clara e direta com o número de PRs, já que não observamos um padrão consistente de aumento ou diminuição. Isso pode indicar que o volume de interações em PRs depende mais do conteúdo ou complexidade específica de cada PR do que do número total de PRs.

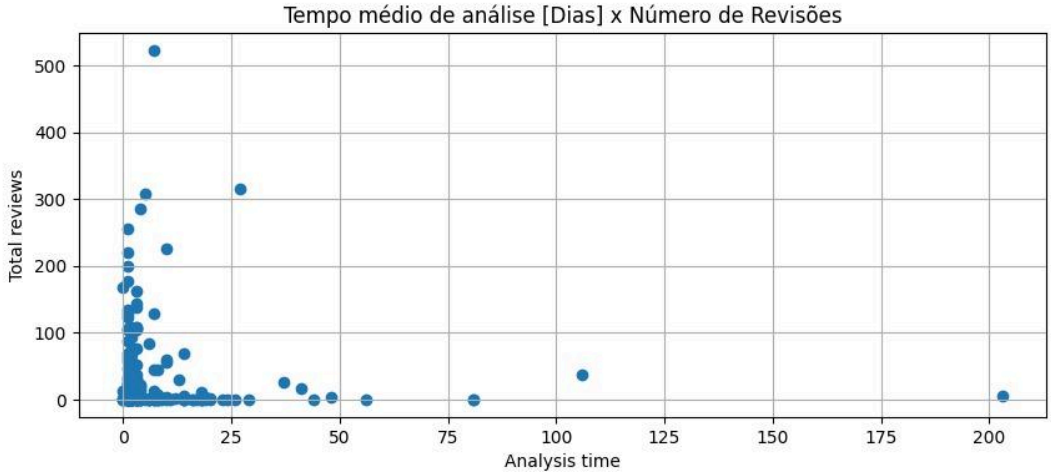
RQ 05 - Qual a relação entre o tamanho dos PRs e o número de revisões realizadas?



O gráfico correspondente à RQ 05 mostra a relação entre o tamanho dos PRs, medido pelo total de arquivos, e o número de revisões. A maioria dos PRs, que têm um número

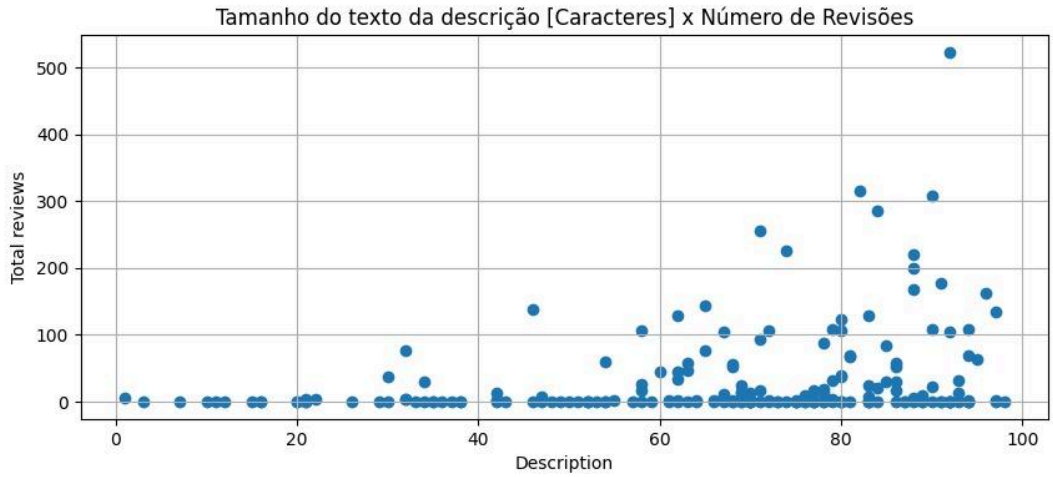
menor de arquivos, concentra-se na região inferior do gráfico, indicando um número menor de revisões. Existem alguns PRs com um grande número de arquivos que apresentam mais revisões, visíveis como outliers. No entanto, não há uma clara tendência de aumento no número de revisões à medida que o número de arquivos aumenta, sugerindo que o tamanho dos PRs não é o único fator que afeta o número de revisões.

RQ 06 - Qual a relação entre o tempo de análise dos PRs e o número de revisões realizadas?



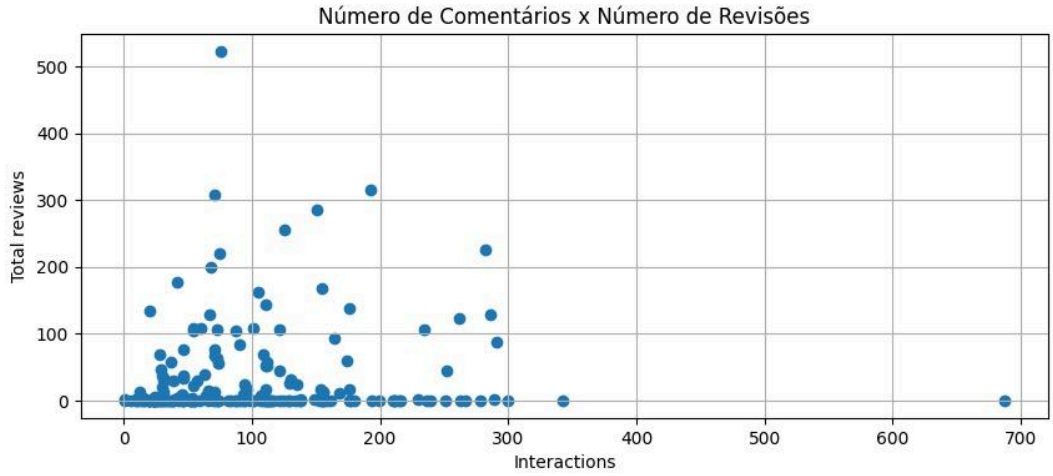
Observando o gráfico relativo à RQ 06, nota-se uma concentração de dados com um número menor de revisões, principalmente para PRs com tempos de análise mais curtos, indicando que a maior parte das revisões é realizada rapidamente. Existem pontos dispersos que sugerem uma quantidade maior de revisões espalhadas por diferentes durações de análise, refletidos como outliers. Isso mostra que, enquanto muitos PRs são concluídos com poucas revisões, alguns exigem uma quantidade substancialmente maior, independentemente do tempo de análise. A distribuição não mostra um padrão claro que ligue diretamente o tempo de análise ao número de revisões, o que pode indicar que a complexidade ou a natureza dos PRs influencia mais no número de revisões do que o tempo que eles ficam abertos.

RQ 07 - Qual a relação entre a descrição dos PRs e o número de revisões realizadas?



Analisando o gráfico referente à RQ 07, que compara o tamanho do texto da descrição dos PRs com o número de revisões, podemos perceber que há uma densa concentração de pontos na região de descrições mais curtas e um número menor de revisões. Isso indica que, frequentemente, PRs com descrições sucintas passam por menos revisões. Enquanto isso, percebemos alguns pontos espalhados ao longo do eixo vertical, que representam PRs com um número variado de revisões, independentemente do comprimento da descrição, agindo como outliers. Essa distribuição sugere que a extensão da descrição do PR não é consistentemente um indicador da quantidade de revisões que o PR receberá.

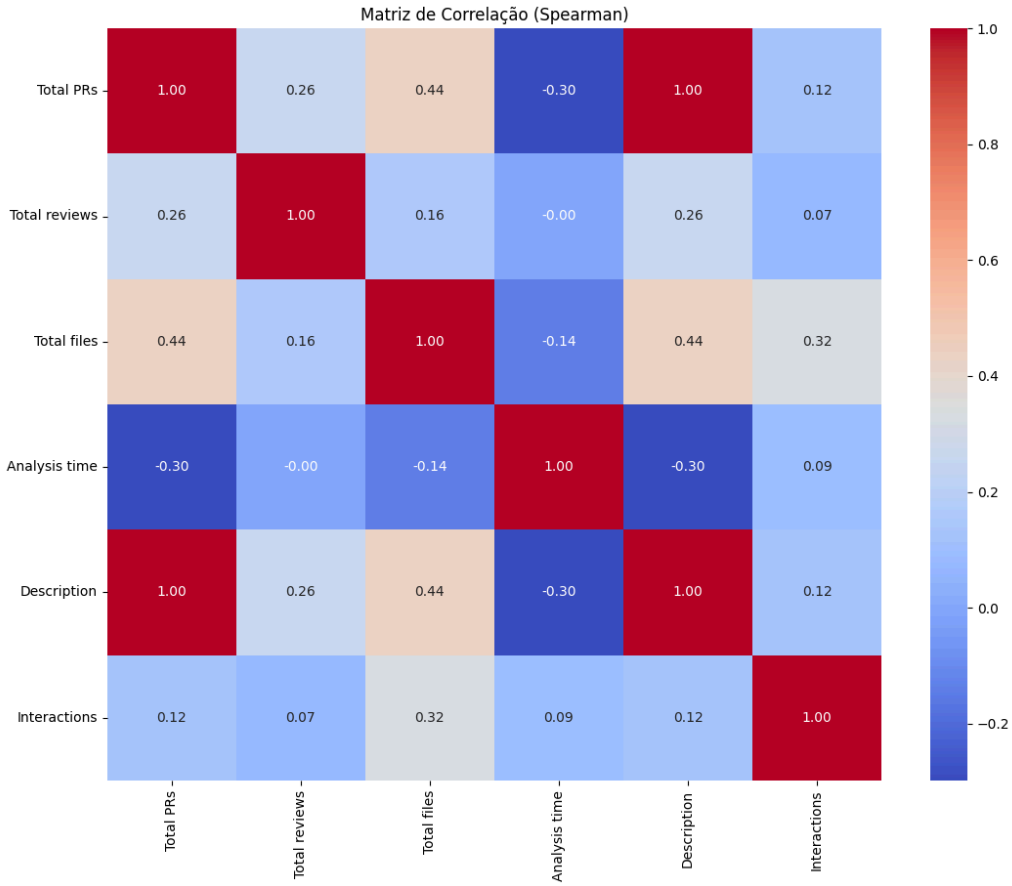
RQ 08 - Qual a relação entre as interações nos PRs e o número de revisões realizadas?



O gráfico para RQ 08, que examina a relação entre interações nos PRs e o número de revisões, mostra a maioria dos PRs agrupada com um número relativamente baixo tanto

de interações quanto de revisões. Alguns PRs destacam-se com um número elevado de revisões, mesmo com um número moderado de interações, agindo como outliers. A distribuição sugere que não há uma relação direta e consistente entre o número de comentários e revisões, já que muitos PRs com poucas interações têm um espectro variado de revisões. Isso indica que a quantidade de revisões pode ser influenciada por outros fatores além das interações.

Correlação de Spearman - Métricas



Em primeira instância, o teste de correlação de Spearman foi escolhido por ser mais apropriado para dados não paramétricos e pode revelar relações monotônicas entre as variáveis. Ademais, a matriz de correlação de Spearman nos dá uma visão sobre as relações entre diferentes aspectos dos Pull Requests em um repositório. Um valor de 0.44 entre o total de PRs e o total de arquivos sugere uma correlação moderada; projetos com mais PRs tendem a modificar mais arquivos. O total de PRs e o tempo de análise têm uma correlação negativa de -0.30, indicando que repositórios com mais PRs podem ter um fluxo de trabalho mais eficiente, levando a análises mais rápidas.

Os valores de correlação de 0.26 entre total de PRs e total de revisões e de 0.16 entre total de revisões e total de arquivos são relativamente baixos, mostrando que mais PRs e um maior número de arquivos não necessariamente resultam em mais revisões. Interessantemente, a descrição está moderadamente correlacionada com o total de arquivos (0.44), mas inversamente correlacionada com o tempo de análise (-0.30), talvez sugerindo que descrições mais completas levam a uma revisão mais eficiente. Por fim, a correlação entre interações e o total de arquivos (0.32) é fraca a moderada, implicando

que PRs com mais arquivos podem gerar mais comentários, embora isso não seja uma regra fixa, dada a natureza moderada da correlação.

4. Discussões

Os PRs com maior volume não só desafiam a revisão, mas também trazem uma chance maior de serem rejeitados. Por outro lado, PRs que passam mais tempo sob análise podem indicar complexidades que requerem mais atenção.

Nota-se que descrições mais elaboradas e maiores nos PRs parecem ser um fator positivo, facilitando a revisão e podendo aumentar as chances de aceitação. As interações dentro dos PRs, refletidas em comentários e discussões, também contribuem positivamente, talvez apontando para uma colaboração efetiva que influencia o feedback final de forma benéfica.

A quantidade de revisões é afetada pelo tamanho dos PRs, mas a relação não é diretamente proporcional como se poderia esperar. PRs mais amplos não necessariamente implicam em um maior número de revisões. Isto sugere que há outros aspectos em jogo que determinam quantas revisões um PR requer.

As análises de correlação de Spearman, permitiram uma visão detalhada das dinâmicas das métricas relativas aos PRs. O estudo destaca que o processo de revisão e aceitação de PRs é multifacetado e não pode ser reduzido apenas a números e volume de dados. As práticas de code review são complexas e envolvem diversos fatores que transcendem o escopo técnico, ressaltando a interação humana e comunicação no desenvolvimento de software colaborativo.

5. Conclusão

Esse trabalho detalha uma investigação das atividades de pull request nos repositórios mais populares do GitHub. Buscou-se desvendar os fatores que impactam na decisão de aceitar ou rejeitar um Pull Request (PR). A pesquisa sugere que PRs extensos podem ser mais difíceis de revisar, ou seja, podem ter maior probabilidade de rejeição; enquanto PRs que demoram mais para serem analisados podem possuir problemas mais complexos.

A pesquisa apontou que descrições maiores, claras e detalhadas, de PRs tendem a facilitar o processo de revisão e podem aumentar a probabilidade de aceitação de um PR. Além disso, a interação nos PRs, como comentários, aparenta ter um impacto positivo no processo de revisão, indicando um nível mais alto de colaboração e conduzindo a um feedback mais favorável.

A quantidade de revisões necessárias em PR é impactada pelo seu tamanho, PRs com maior volume geralmente demandando mais análises. Contudo, estudos indicam que a correlação entre o número de revisões e o tamanho ou quantidade de arquivos alterados não é particularmente forte. Observa-se que PRs com descrições detalhadas e um maior número de interações não resultam necessariamente em uma maior frequência de revisões. Isso sugere que existem outros elementos que podem determinar o número necessário de revisões.

Este trabalho fornece uma base importante para aprimorar as práticas de revisão de código, potencialmente levando a uma prática de desenvolvimento de software mais eficaz.