



**INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA - PRAÇA DA LIBERDADE**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE**

Matheus Santos Rosa Carneiro

Raíssa Carolina Vilela da Silva

Vitor Augusto Alves de Jesus

**TRABALHO PRÁTICO DE LABORATÓRIO DE EXPERIMENTAÇÃO DE SOFTWARE**

**BELO HORIZONTE**

**2021**

## 1. INTRODUÇÃO

O GitHub é uma plataforma de armazenamento e versionamento de código que permite a criação, gestão e compartilhamento de repositórios gratuitamente a partir de qualquer máquina que tenha conexão à internet. Essa ferramenta oferece uma API gratuita que permite a obtenção de dados sobre qualquer repositório público utilizando queries com parâmetros bem definidos. Desta forma, utilizaremos essas ferramentas para buscar nossos dados e analisar os resultados obtidos.

A partir do desenvolvimento de sistemas *open-source*, temos o fator que diversos desenvolvedores contribuem em partes diferentes do código e um dos riscos a serem gerenciados diz respeito à evolução dos seus atributos de qualidade interna. Neste contexto, o objetivo deste projeto é analisar os aspectos relacionados a adoção de práticas de revisão de código.

Este trabalho foi proposto pelo professor orientador da disciplina de Laboratório de Experimentação de *Software*, com o intuito de fazer com que respondêssemos as seguintes perguntas:

- 1.1.1 – Qual a relação entre o tamanho dos PRs e o *feedback* final das revisões?
- 1.1.2 – Qual a relação entre o tempo de análise dos PRs e o *feedback* final das revisões?
- 1.1.3 – Qual a relação entre a descrição dos PRs e o *feedback* final das revisões?
- 1.1.4 – Qual a relação entre as interações nos PRs e o *feedback* final das revisões?
- 1.1.5 – Qual a relação entre o tamanho dos PRs e o número de revisões realizadas?
- 1.1.6 – Qual a relação entre o tempo de análise dos PRs e o número de revisões realizadas?
- 1.1.7 – Qual a relação entre a descrição dos PRs e o número de revisões realizadas?
- 1.1.8 – Qual a relação entre as interações nos PRs e o número de revisões realizadas?

As métricas definidas para a análise dos dados são:

- 1.2.1 – Tamanho: número de arquivos; total de linhas adicionadas e removidas.
- 1.2.2 – Tempo de Análise: intervalo entre a criação do PR e a última atividade.
- 1.2.3 – Descrição: número de caracteres do corpo de descrição do PR.
- 1.2.4 – Interações: número de participantes; número de comentários.

## 2. METODOLOGIA

Este é um estudo da disciplina de Laboratório de Medição e Experimentação de *Software*, que consiste em realizar uma abordagem quantitativa. Esta abordagem, ocorre a partir da migração de PRs submetidos à repositórios populares do GitHub, que possuam pelo menos cem PRs com *status MERGED* ou *CLOSED*. Além disso, a fim de analisar os *pull requests* que tenham passado pelo processo de *code review*, serão analisados apenas os PRs que possuem os *status* citados anteriormente e que possuam pelo menos uma revisão. Além disso, foram filtrados os PRs que tenham sido revisados em pelo menos uma hora, ou seja, o cálculo entre a data de criação do PR e de seu encerramento (merge/closed) deve ser maior que uma hora.

Para a coleta dos dados foi elaborado um *script* na linguagem de programação *Python* para que fosse possível realizar capturar apenas os dados relevantes à pesquisa. Além disso, foram propostas métricas que facilitasse a análise dos resultados obtidos. A coleta dos dados foi feita através da API do GitHub conhecida como *GraphQL* e em seguida os dados foram exportados para um arquivo *.csv* para que fosse possível analisar os dados em conjunto.

### 3. RESULTADOS OBTIDOS

Os resultados obtidos estão diretamente relacionados às métricas citadas e a conclusão da análise este sendo realizada em resposta às *questions* propostas.

➤ **Quantidade de arquivos X Quantidade de PRs**

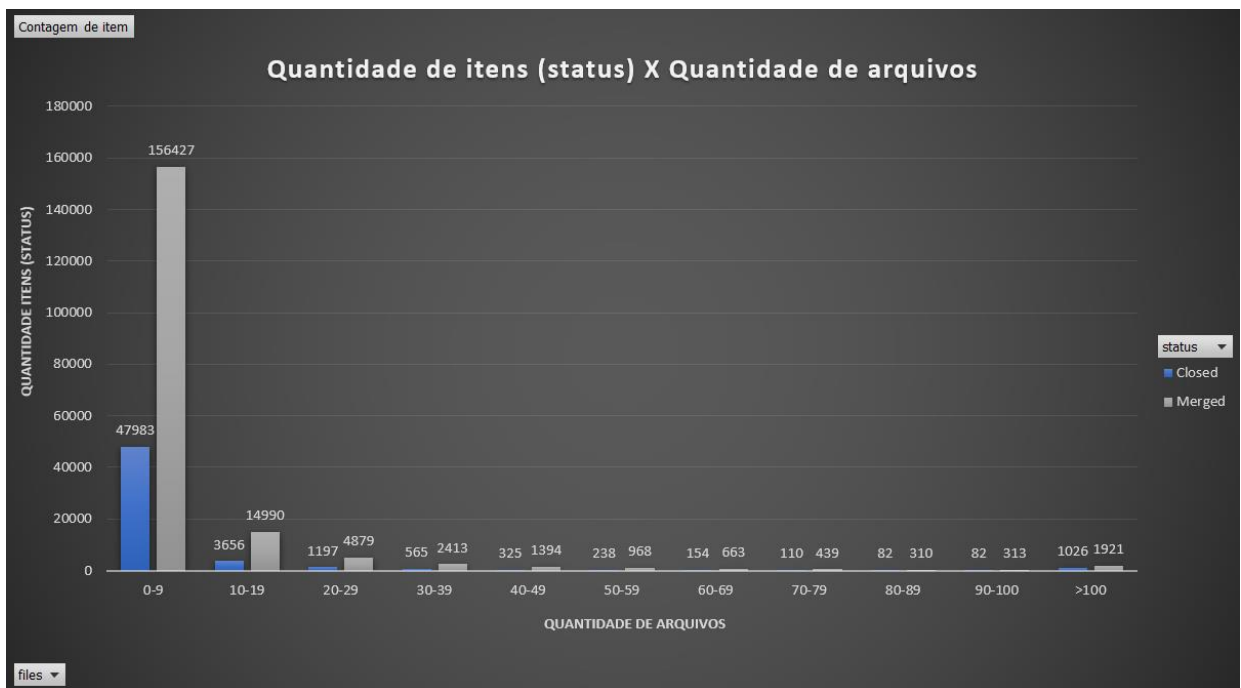
➤ **Hipótese**

Quanto maior a quantidade de arquivos, menor será o número de pull requests aceitos.

➤ **Análise**

Através da análise dos dados, é possível notar que os PRs que possuem entre 0-9 arquivos, são os PRs que são mais analisados. No entanto, pode-se concluir que a quantidade de PRs que são aprovados, ou seja, que são *MERGED*, são aproximadamente três vezes maiores do que as quantidade de PRs que são *CLOSED*, ou seja, a maior parte dos PRs que possuem entre 0-9 arquivos, costumam ser aceitos. Dessa forma, a hipótese se mostra verdadeira.

	MÉDIA	MODA	MEDIANA
FILES-PR-CLOSED	17,1	1	2
FILES-PR-MERGED	9,4	1	2



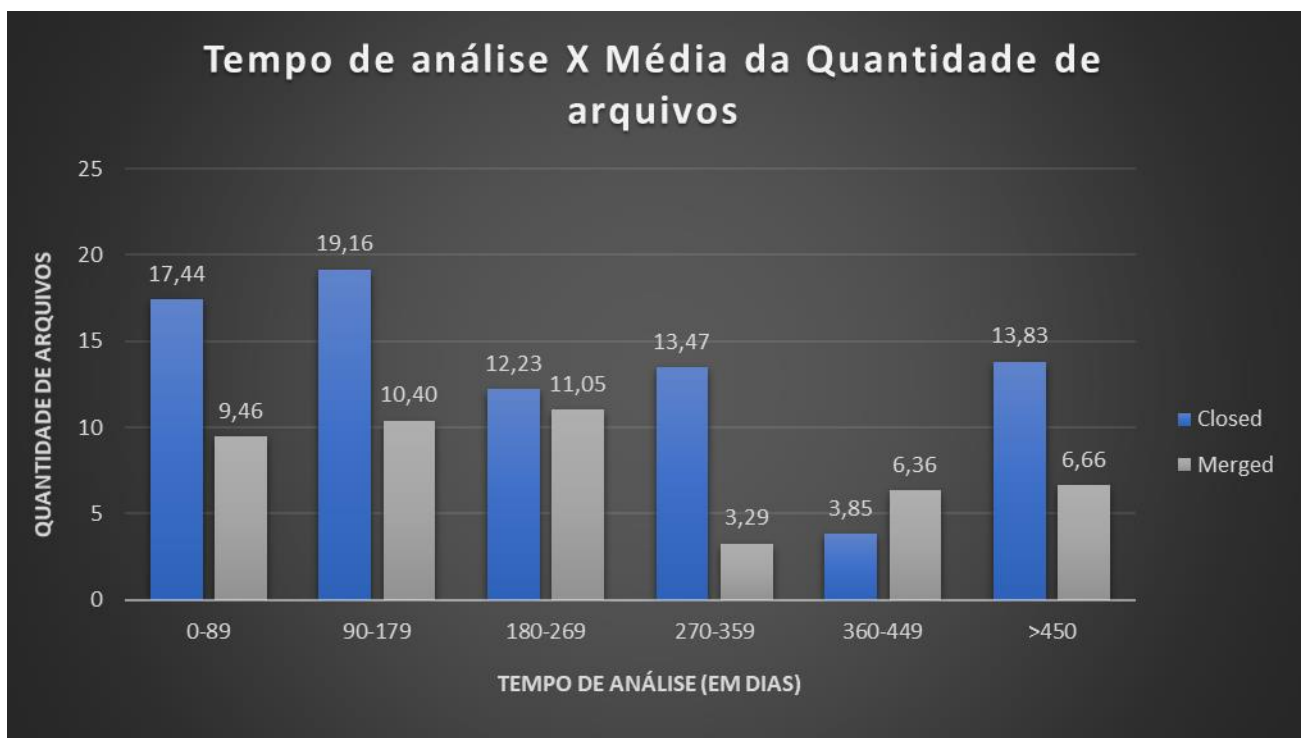
➤ **Tempo de análise X Quantidade de arquivos**

➤ **Hipótese**

Quanto maior o número de arquivos, maior a quantidade de PRs *MERGED*.

➤ **Análise**

Levando em consideração, o tempo de análise contado em dias, podemos perceber que os PRs que possuem mais arquivos, normalmente não possuem status merged, ou seja, a chance de um PR com muitos arquivos serem aprovados é mais baixa que os PRs com menos arquivos, conforme mostra gráfico abaixo. Dessa forma, a hipótese apontada é falsa.



➤ **Quantidade de caracteres X Status dos PRs**

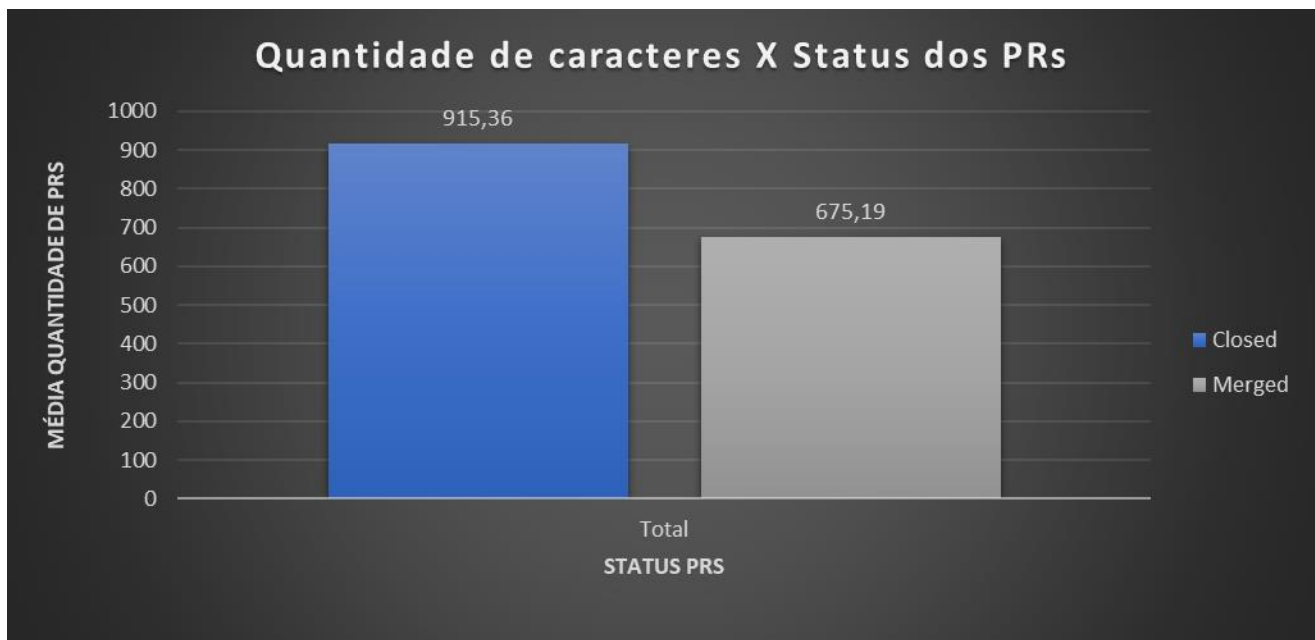
➤ **Hipótese**

Quanto maior a quantidade de linhas, maior a chance do PR ser aprovado.

➤ **Análise**

Analisando a quantidade de caracteres e o *feedback* final das revisões, é possível perceber que os PRs que são aprovados, ou seja, *MERGED* são os que possuem menor quantidade de caracteres. Além disso, também se percebe que os caracteres dos PRs que são *CLOSED*, ou seja, rejeitados, possuem 25% maior quantidade de caracteres em relação a quantidade de caracteres dos PRs aprovados. Dessa forma, a hipótese se mostrou falsa.

	MÉDIA	MODA	MEDIANA
BODY-PR-CLOSED	915,4	0	2
BODY-PR-MERGED	675,2	0	2



➤ **Quantidade de comentários X Status dos PRs**

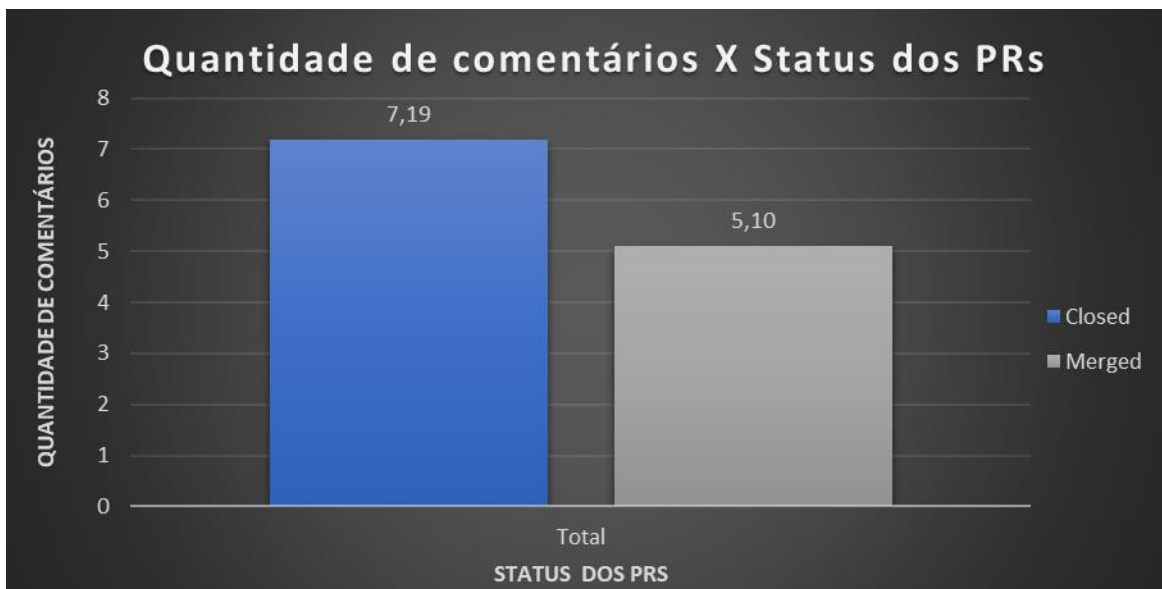
➤ **Hipótese**

Quanto maior o número de comentários, maior a tendência de o PR ser aprovado.

➤ **Análise**

Ao analisar a quantidade comentários em relação ao *status* dos PRs, pode-se perceber que os comentários dos PRs reprovados (*CLOSED*) são maiores do que os comentários dos PRs *MERGED*. Dessa forma, a hipótese prova-se falsa conforme os gráficos e tabelas abaixo.

	MÉDIA	MODA	MEDIANA
COMMENTS-PR-CLOSED	7,2	2	4
COMMENTS-PR-MERGED	5,1	1	2



➤ **Quantidade de arquivos X Quantidade de revisões**

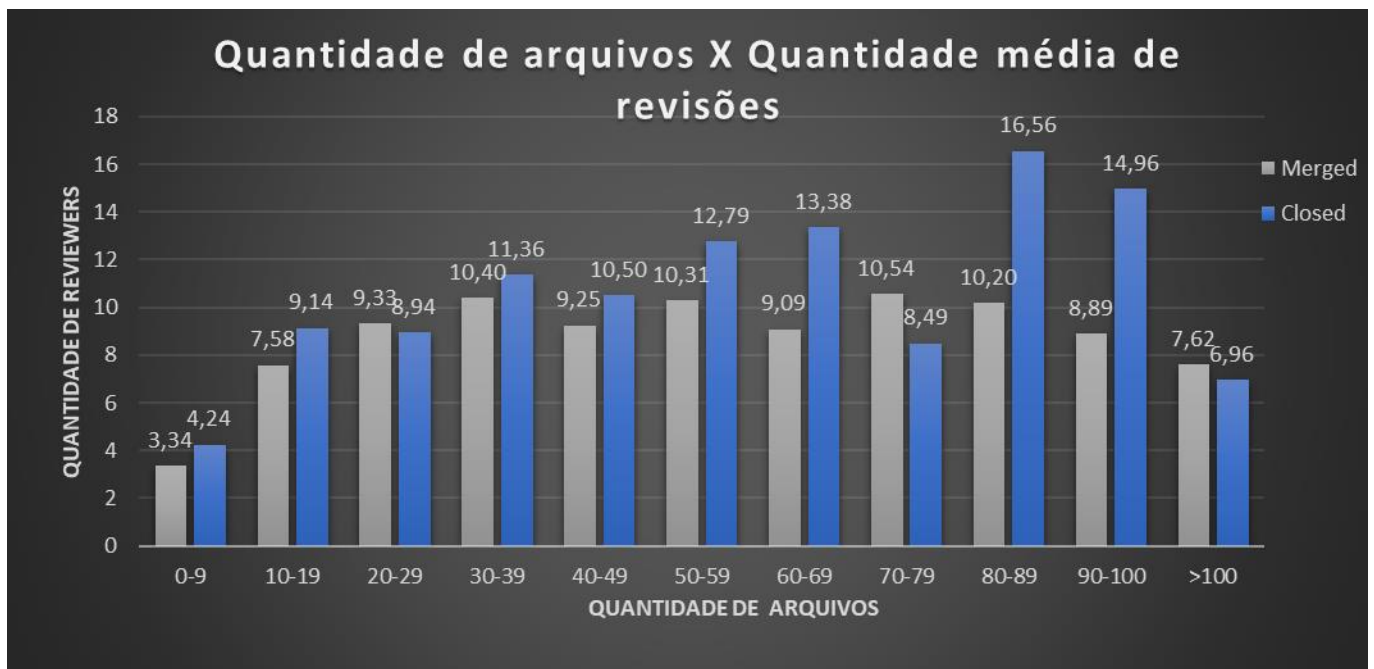
➤ **Hipótese**

Quanto maior a quantidade de revisores, menor a chance do PR ser aprovado (*MERGED*).

➤ **Análise**

Ao analisar os dados coletados, é possível notar que os PRs que possuem maior número de revisores e maior quantidade de arquivos, tendem a ser rejeitados. Em contrapartida, PRs mais enxutos e com menos arquivos, possuem maior chance de serem aprovados (*MERGED*). Dessa forma, a hipótese mostra-se verdadeira.





	MÉDIA	MODA	MEDIANA
REVIEW-PR-CLOSED	4,9	1	3
REVIEW-PR-MERGED	4,1	1	2

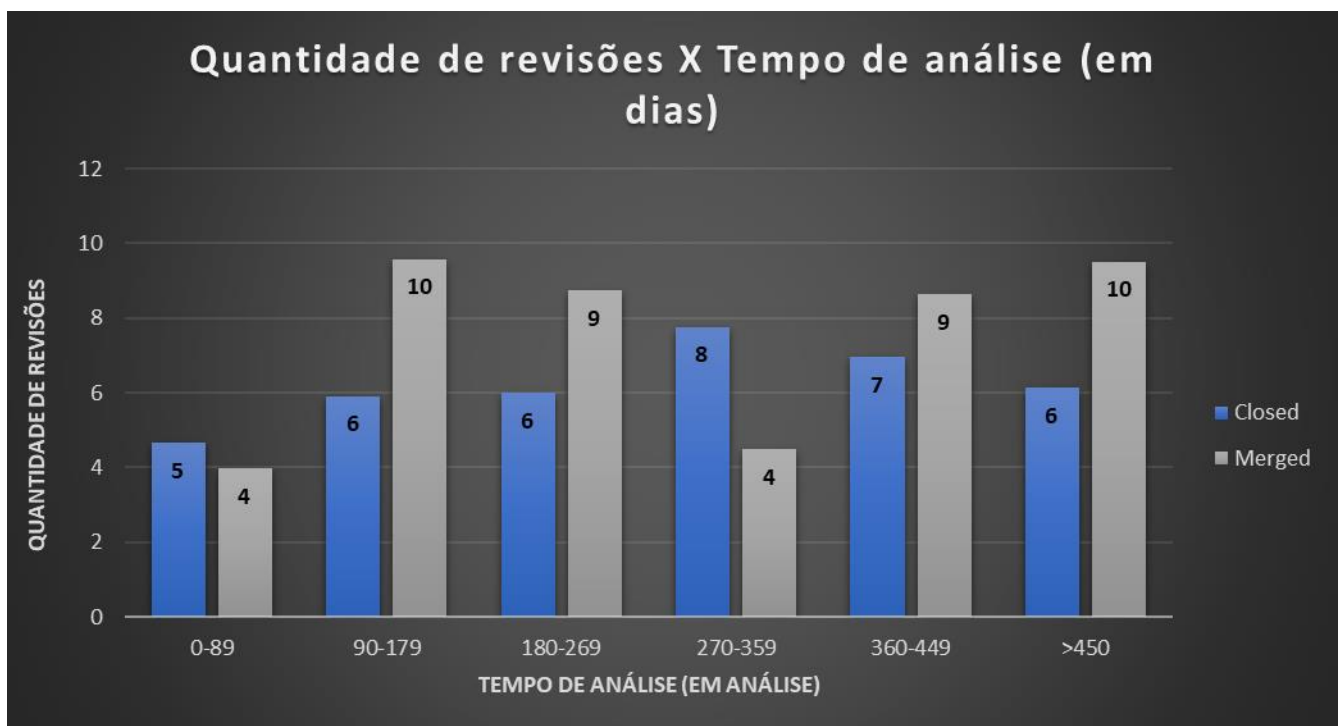
#### ➤ Quantidade de revisões X Tempo análise (em dias)

#### ➤ Hipótese

Quanto maior a quantidade de revisores, maior o tempo de análise dos PRs.

#### ➤ Análise

Analisando a quantidade de revisões em relação ao tempo de análise dos dados, é possível perceber que quanto maior a quantidade de revisores, maior será o tempo de análise, dessa forma, a hipótese prova-se verdadeira.



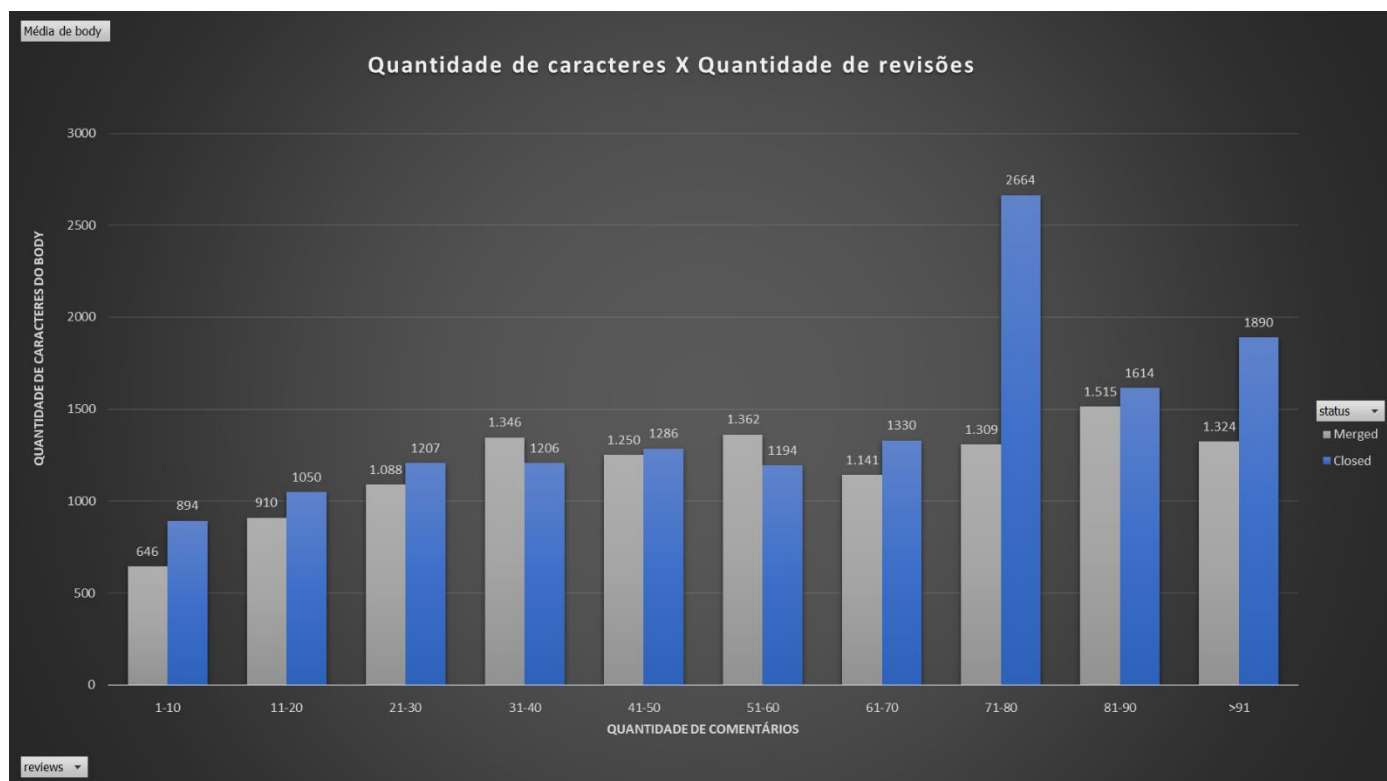
➤ **Quantidade de caracteres X Quantidade de revisões**

➤ **Hipótese**

Quanto mais revisões possuir o pull request, maior será o número de caracteres.

➤ **Análise**

Através da análise dos dados, é possível notar que os arquivos que possuem mais caracteres, conseqüentemente são também os que possuem mais revisões. Dessa forma a hipótese prova-se verdadeira.



➤ **Quantidade de revisões X Quantidade de participantes**

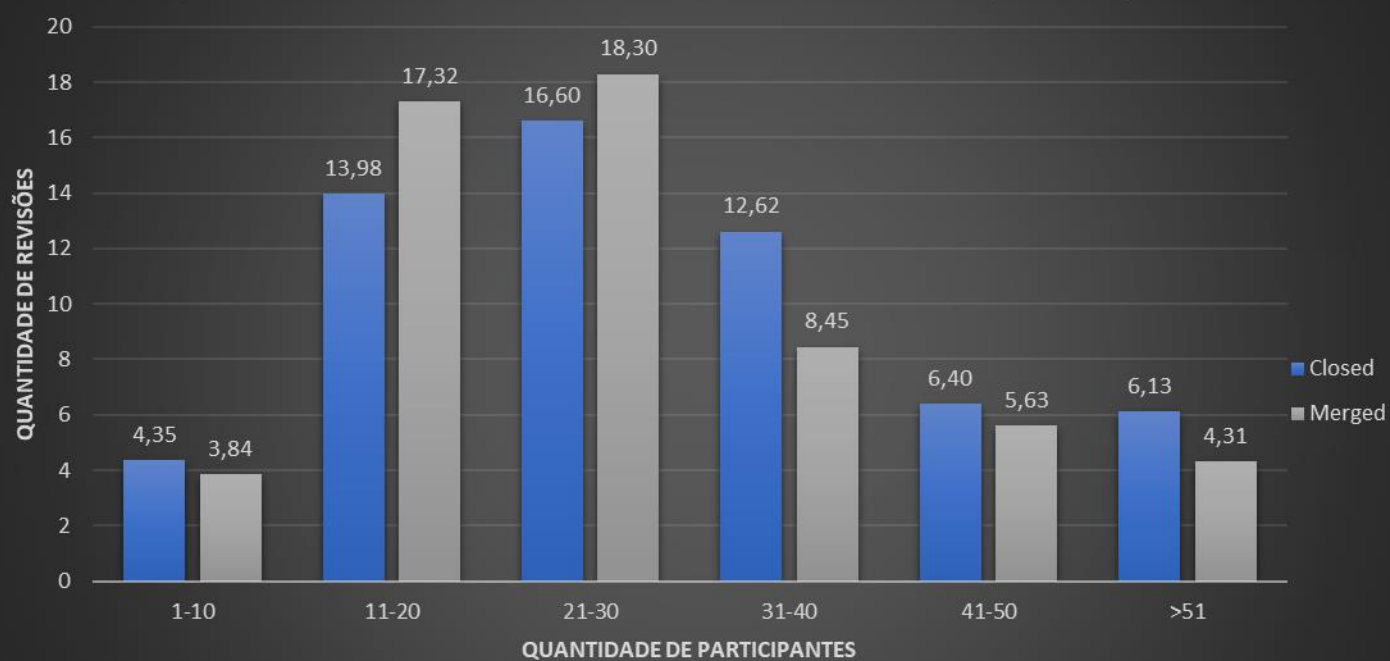
➤ **Hipóteses**

Quanto maior o número de revisores, maior será a quantidade de participantes.

➤ **Análise**

Analisando os dados coletados, é possível perceber que os PRs mais revisados são os que possuem entre 11 e 30 participantes. Dessa forma a hipótese levantada é falsa, conforme apontado pelo gráfico abaixo.

## Quantidade de revisões X Quantidade de participantes



## 4. CONCLUSÃO

A partir da coleta de dados realizada, pudemos concluir que as métricas relacionadas ao *code review* podem nos gerar importantes dados e relações obtidas através dos dados extraídos dos *pull requests*. É notável que a maior parte de *code review* são provenientes de *pull requests* aceitos, ou seja, que tenham o *status MERGED*. Além disso, foi interessante perceber que as hipóteses que propusemos para que fosse possível realizar a análise dos dados obtidos foi, em sua maioria, comprovada através da análise realizada.

Ao analisar os dados, foi possível perceber que quanto mais revisores se têm para a análise de um *pull request* maior é o tempo gasto para a análise e maior é a chance de reprovação desse PR, em contrapartida, é notável que a maioria dos PRs aprovados têm um tempo considerado curto para a análise. Além disso, foi perceptível também que os PRs rejeitados possuíam em seus arquivos, aproximadamente 25% mais caracteres do que os PRs aprovados.

Dessa forma, podemos concluir que, nem sempre o fato de ter muitos revisores quer dizer agilidade, pois, desta forma o tempo gasto para análise e liberação de um PR acaba se tornando maior. No entanto, não deixa de ser importante ter revisores para aprovar um PR, pois, isto garante que possíveis problemas possam ser resolvidos antes mesmo de fazer o *MERGE* na *branch* mais atualizada.