

Impacto da relevância temática e outros atributos relacionados à comunicação em issue tracking no tempo de resolução das issues

Fábio Trindade Ramos
Federal University of Viçosa
Florestal, Brasil
fabio.ramos@ufv.br

Gláucia Braga e Silva
Federal University of Viçosa
Florestal, Brasil
glauucia@ufv.br

RESUMO

Contexto: Em um projeto de desenvolvimento de *software*, a resolução de uma *issue* pode ser influenciada por diversos fatores, como, o número de desenvolvedores envolvidos, a complexidade da *issue*, a experiências dos desenvolvedores, a multitarefa e os comentários educados. Assim, entender quais fatores influenciam o seu tempo de resolução é crucial para diminuí-lo e, conseqüentemente, para os resultados de um projeto. **Objetivo:** Este trabalho objetiva avaliar o impacto de atributos extraídos de repositórios de *issue tracking* do GitHub no tempo de resolução das *issues*. Esses atributos abrangem a relevância temática dos comentários postados, a quantidade de comentários e de desenvolvedores em uma *issue*, e a participação de top-devs, seja na criação, na discussão ou na resolução das *issues*. **Materiais e Métodos:** A organização Ruby on Rails foi alvo de estudo deste trabalho, devido à sua alta atividade em *issue tracking*. A partir dos dados obtidos, foi aplicado o método de correlação de Spearman para quantificar a relação entre dois atributos, e testes de hipótese para analisar as amostras. **Resultados:** Os resultados mostram que *issues* e *pull requests* criadas por top-devs possuem um tempo médio menor de resolução. Além disso, foi encontrada uma tendência de aumento no tempo de resolução das *issues* e *pull requests*, e apenas *issues*, respectivamente, quando a quantidade de comentários e de desenvolvedores aumentam. Por fim, observou-se que a relevância temática dos comentários possuem valores baixos, em sua maioria, entre 0 e 0,1.

1 INTRODUÇÃO

Em ambientes de *software*, há a colaboração constante de pessoas na produção de artefatos, como a construção de código novo, correção e discussão de erros, integração de código, entre outros. Nesses ambientes, a comunicação entre os participantes mostra-se essencial e também representa um dos principais gargalos durante o desenvolvimento. Para lidar com esses gargalos, repositórios de *software*, como o GitHub, desempenham um papel fundamental ao armazenar informações acerca dessas colaborações. De acordo com Vidoni [17], minerar esses repositórios permite uma exploração empírica dos dados, sendo os resultados diretamente aplicáveis em

evolução de software, redes e caracterização de desenvolvedores, previsão de *bugs*, estimativa de esforço, entre outros.

Repositórios de *software* contam com um sistema de rastreamento de questões (*issue tracking*), cuja função é manter registro de *issues*, como por exemplo, *bugs*, novos requisitos e sugestões de melhorias. Esses problemas geralmente indicam que o projeto não está sendo desenvolvido conforme o planejado. Portanto, é natural esperar que a resolução ágil de uma *issue* evite atrasos em um projeto. No entanto, a sua resolução pode ser influenciada por diversos fatores, como por exemplo, o número de desenvolvedores envolvidos, complexidade da *issue*, experiências dos desenvolvedores, multitarefa [10], comentários educados [15], entre outros. Assim, entender quais fatores influenciam o seu tempo de resolução é crucial para diminuí-lo e, conseqüentemente, para os resultados do projeto.

Este trabalho objetiva avaliar o impacto de alguns atributos extraídos de repositórios de *issue tracking* do GitHub no tempo de resolução das *issues*. Esses atributos abrangem a relevância temática dos comentários postados, a quantidade de comentários e a quantidade de desenvolvedores em uma *issue*, e a participação de top-devs (desenvolvedores que se destacaram em função da relevância dos comentários postados), seja na criação, na discussão ou na resolução das *issues*. No contexto de *issue tracking*, a métrica de relevância temática representa o quanto um texto é relevante no contexto que está inserido [2, 14]. Dessa forma, este estudo foi guiado pelas seguintes questões de pesquisa:

- **QP01** Existe relação entre a quantidade de comentários em uma *issue* e suas respectivas relevâncias temáticas e o tempo de resolução?
- **QP02**. Existe relação entre a quantidade de desenvolvedores participando de uma *issue* e o tempo de resolução?
- **QP03**. Existe diferença no tempo de resolução entre as *issues* que foram criadas, comentadas ou atribuídas por/para top-devs em *issue tracking*?

O restante do artigo está organizado da seguinte forma: a seção 2 apresenta informações sobre comunicações em *issue tracking* e o conceito de relevância temática. Na seção 3, são apresentados os trabalhos relacionados. A seção 4 contém os materiais e métodos utilizados no trabalho. Os resultados e discussões estão presentes na seção 5. Por fim, as ameaças à validade e as considerações finais estão, respectivamente, nas seções 6 e 7.

2 COMUNICAÇÕES EM ISSUE TRACKING

Frequentemente, problemas ou melhorias reportadas por desenvolvedores são discutidos durante o desenvolvimento de um projeto. Essas discussões geralmente ocorrem por meio de um sistema de rastreamento de questões comumente disponível em repositórios

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/XXXXXXXX.XXXXXXX>

de *software*. De acordo com Bertram et al. [4], esses sistemas são responsáveis por manter informações valiosas, como a comunicação realizada pelos desenvolvedores e ideias de recursos parcialmente envolvidas, sendo útil de diversas formas para um projeto.

2.1 Issue Tracking no GitHub

No GitHub, as *issues* são reportadas após a inserção de informações obrigatórias, como o título, e opcionais, como rótulos, atribuição de *issue* a um desenvolvedor e descrição. Há ainda informações sobre a *issue* que são geradas automaticamente, como data da criação, estado (fechada ou aberta), criador da *issue*, entre outras. Após essa etapa, qualquer colaborador do projeto pode participar da discussão, em caso de projetos privados. Em projetos *open-source*, qualquer desenvolvedor que possua uma conta na plataforma pode colaborar com as questões reportadas. Além disso, vale ressaltar que o GitHub considera *pull requests* (PRs) como um tipo especial de *issue*, utilizadas para solicitar a alteração em uma ramificação (*branch*).

2.2 Relevância Temática em Issue Tracking

A relevância temática, métrica que visa avaliar a relevância de um texto em relação ao assunto em que está inserido, foi inicialmente aplicada em fóruns educacionais [1, 9] com o propósito de identificar quais alunos estavam contribuindo, ou não, com textos relacionados ao tema em discussão. Posteriormente, Neto e Braga e Silva [14] adaptaram essa métrica para analisar a relevância dos comentários postados em ambientes de *issue tracking*, sendo o título da *issue* e a sua descrição utilizados para identificar o contexto do problema discutido.

Neto e Braga e Silva [14] aplicaram em sua abordagem a técnica de similaridade de grafos. Em seguida, Batista et al. [2] adaptaram essa abordagem, substituindo a técnica por similaridade de cossenos. De acordo com Machado et al. [9], a similaridade de grafos utiliza uma estrutura baseada em vértices e arestas (grafo), no qual os vértices representam os termos mais relevantes e as arestas conectam os termos próximos entre si do texto. Com base nessa estrutura são realizados cálculos matemáticos e outras técnicas para obter a relevância do comentário. Por outro lado, a similaridade de cossenos se baseia em vetorizar os textos para fazer uma análise angular entre os conteúdos textuais. Assim, através da distância angular entre os vetores é obtida a relevância do comentário.

Associar um texto a outros atributos quali-quantitativos não é viável com sua representação original. Nesse sentido, a relevância temática desempenha um papel fundamental ao expressar numericamente a relevância de um comentário, pois permite estabelecer relações com outras métricas ou atributos, como tempo de resolução, gênero e complexidade da *issue*.

2.2.1 RIT. Batista et al. [2], desenvolveram uma biblioteca, chamada RIT, para automatizar o cálculo da relevância temática em ambientes de *issue tracking* do GitHub. A RIT funciona em três etapas consecutivas: inicialmente, ocorre a extração dos dados das *issues* de um repositório; logo após, é calculada a relevância temática (um valor entre 0 e 1, com os extremos inclusos), para cada comentário, a partir dos dados textuais brutos (sem manipulações prévias) do título, descrição e dos próprios comentários das *issues*;

por último, são exportados dados acerca das *issues* e comentários, junto com a relevância temática, para um arquivo CSV.

Neste trabalho, o código da biblioteca RIT foi adaptado/estendido com o intuito de melhorar o cálculo da métrica para o contexto de *issue tracking* e também para facilitar o uso dos dados gerados e eliminar informações redundantes. Inicialmente, observou-se que os atributos textuais presentes nesses ambientes podem conter elementos irrelevantes dependendo do escopo da pesquisa. Por exemplo, nos trabalhos de Lima et al. [8] e Zhang et al. [18], foram removidos elementos como imagens, trechos de código e URLs de comentários extraídos também desses sistemas por não agregarem valor ou prejudicarem o resultado da pesquisa. O cálculo de relevância temática empregada pela RIT pode ser prejudicado por esses elementos, pois a forma que são dispostos estruturalmente no texto são diferentes da gramática utilizada pelos humanos. Dessa forma, neste trabalho, foi inserida uma etapa de pré-processamento na RIT com o objetivo de melhorar o cálculo da relevância temática. Esta etapa retira os seguintes elementos da descrição e comentários das *issues*:

- | | |
|----------------------|---------------------------------|
| (1) URL; | (5) Caminho de arquivo e pasta; |
| (2) Email; | (6) Referência de imagem; |
| (3) Emoji; | (7) Trecho de código. |
| (4) Nome de usuário; | |

Houve também a modificação na etapa de exportação da biblioteca com o objetivo de normalizar os dados exportados, visto que o arquivo CSV exportado pela biblioteca continha diversas informações redundantes. Assim, os dados gerados pela biblioteca foram separados em 2 arquivos CSV, conforme pode ser visto na Tabela 1. Além disso, as informações exportadas foram alteradas, as quais podem ser vistas na Tabela 1. O dado Tags, presente nos dados dos comentários, indicam se os elementos 1, 5, 6 e 7 foram removidos do comentário. Por exemplo, se foi retirado uma referência de uma imagem e um trecho de código do comentário, o arquivo CSV na coluna Tags desse comentário conterá o valor "IMAGEM\nTRECHO_CODIGO\n".

Tabela 1: Dados Exportados Após a Adaptação da RIT

Dados da <i>issue</i>	Dados do comentário
Número da <i>issue</i>	ID da <i>issue</i>
ID da <i>issue</i>	Número do comentário
Título da <i>issue</i>	ID do comentário
Data de criação da <i>issue</i>	Data do comentário
Nome do repositório da <i>issue</i>	Relevância temática
Link da <i>issue</i>	Autor do comentário
-	Tags

3 TRABALHOS RELACIONADOS

Nessa seção serão discutidos os trabalhos que investigam a influência de elementos no tempo de resolução de *issue*, assim como trabalhos que aplicam a métrica de relevância temática.

Ortu et al. [15] utilizam um modelo de regressão para entender como as emoções humanas afetam o tempo de resolução de *issues*. Como resultado, os modelos mostram que comentários expressando

emoções felizes e amáveis estão relacionados a *issues* com menor tempo de resolução, enquanto os que expressam emoções tristes e agressivas estão relacionadas a *issues* com maior tempo de resolução. Entender a relação de atributos com o tempo de resolução de uma *issue* é essencial para minimizá-lo. Assim, o atual trabalho avalia se a quantidade de comentários e desenvolvedores, bem como os desenvolvedores que possuem comentários mais relevantes, causam impacto no tempo de resolução de uma *issue*.

Neto et al. [12] utilizam um modelo baseado em redes neurais para estimar o tempo de resolução de *issue*, a partir do uso de atributos textuais, temporais e dinâmicos. Este trabalho seleciona alguns desses atributos, incluindo a quantidade de comentários, a participação de desenvolvedores, a criação de *issues* por um desenvolvedor e a atribuição de *issue* a um desenvolvedor.

No contexto de *issue tracking*, foram desenvolvidas algumas abordagens para aplicar a métrica de relevância temática. Neto e Braga e Silva [14] criaram uma ferramenta que identifica desenvolvedores-chave em um projeto, com foco na comunicação. Neto [13] aprimorou essa ferramenta, adicionando recursos como análise de sentimentos e análise de rede social, junto com a relevância temática, para criar uma rede social entre os envolvidos no projeto (*stakeholders*), facilitando a investigação do perfil destes *stakeholders* e a identificação dos *stakeholders* mais importantes na comunicação. Batista et al. [3], por sua vez, utilizaram essa métrica para analisar possíveis divergências na relevância dos comentários postados por homens e mulheres em *issues* do GitHub. A relevância temática possibilita a associação da relevância de um texto com outras métricas quali-quantitativas, tornado-a de grande valor para análise de dados envolvendo atributos textuais. Dessa forma, esse trabalho utiliza esta métrica com o objetivo de analisar a relação da relevância dos comentários com o tempo de resolução de *issue*.

O trabalho de Huang et al. [7] investiga se *issues* que possuem perguntas na discussão do problema diferem das que não possuem em termos de tempo de resolução de *issue*, número de comentários, número de desenvolvedores participando da *issue* e número de reatribuições. Enquanto o trabalho anterior avalia o impacto de características específicas da *issue* no tempo de resolução de *issue*, esse estudo é guiado pela relevância dos comentários para avaliar o impacto dos top-devs no tempo de resolução de *issue*, seja comentando, criando ou sendo atribuído a uma *issue*.

Por fim, o presente trabalho se destaca em relação aos demais pelo uso da relevância temática para analisar a sua relação com o tempo de resolução das *issues* e identificar desenvolvedores-chave (top-devs). A Tabela 2 apresenta um comparativo dos trabalhos relacionados com este trabalho, destacando as similaridades.

4 MATERIAIS E MÉTODOS

Esta seção aborda as etapas do processo de mineração de dados deste estudo, bem como as métricas utilizadas para análise dos dados.

4.1 Seleção e Extração dos Dados

Para a seleção do repositório de *issue tracking* utilizado na extração, optou-se por usar como critério uma organização com alta atividade em *issue tracking*. Nesse sentido, a organização Ruby on

Rails¹, criada em 2008, próxima à fundação do GitHub, apresentou-se como uma fonte rica de informações sobre a comunicação, sendo selecionada nesta pesquisa.

Os atributos como a relevância temática, quantidade de comentários e quantidade de desenvolvedores foram obtidos a partir da RIT, enquanto para obter dados acerca da atribuição de *issues*, tempo de resolução de *issue* e autor da *issue*, foi necessário recorrer a outra abordagem. São dois os principais métodos encontrados para obter dados do GitHub: GitHub API² (REST e GraphQL) e projetos que armazenam e disponibilizam dados públicos do GitHub, conhecidos como *datasets*. As APIs do GitHub permitem acesso aos projetos hospedados na plataforma [11], enquanto os *datasets* facilitam o acesso a esses dados para desenvolvedores e pesquisadores [5], contornando as limitações de requisições diárias impostas pelas APIs, o que dificultaria a análise de grandes volumes de dados. Entretanto, os trabalhos de Mombach [11] e Cosentino et al. [6] apontam que *datasets* como o GHTorrent³, GitHub Archive⁴ e BOA⁵ não estão atualizados em relação aos dados obtidos através das APIs.

Para garantir uma melhor consistência dos dados, este trabalho adotou a mesma abordagem da RIT, utilizando a API REST do GitHub para obter as informações não fornecidas pela biblioteca. Além disso, foi criado um script para automatizar a execução do cálculo de relevância temática para todos os repositórios da organização, dado que a RIT funciona para um repositório por vez, sendo os nomes dos repositórios obtidos através também da API. A extração dos dados foi realizada em maio de 2023.

4.2 Definição das Métricas

Para a condução das análises, foram utilizadas as seguintes métricas:

- **M01:** Relevância temática dos comentários - calculada via biblioteca RIT;
- **M02:** Quantidade de comentários da *issue* - número de comentários postados em uma *issue*;
- **M03:** Quantidade de desenvolvedores envolvidos na *issue* - número de desenvolvedores que postaram comentários na *issue*;
- **M04:** Tempo de resolução da *issue* - tempo total de resolução da *issue*, desde sua criação até seu fechamento;
- **M05:** Top-devs em *issue tracking* - desenvolvedores com alta atividade em *issue tracking*.

A métrica M01 foi utilizada com duas finalidades: calcular a sua relação com o tempo de resolução de *issue* e identificar os desenvolvedores da métrica M05. As métricas M02 e M03 foram obtidas a partir da biblioteca RIT, no entanto, foi necessário utilizar a biblioteca Pandas para obtê-las, sendo que essas métricas vêm de forma implícita no CSV disponibilizado pela RIT. Por fim, a métrica M05 foi obtida selecionando os desenvolvedores que postaram comentários com relevância temática maior do que 0,5.

¹<https://github.com/rails>

²<https://docs.github.com/en/developers/overview/about-githubs-api>

³<https://ghtorrent.org/>

⁴<https://www.gharchive.org/>

⁵<https://boa.cs.iastate.edu/>

Tabela 2: Similaridades Entre os Trabalhos Relacionados e Este Trabalho

	Uso de atributos textuais	Uso de atributos temporais	Uso de atributos relacionados aos desenvolvedores	Usa a relevância Temática
Ortu et al. [15]	Análise de sentimentos	Classificar o tempo de resolução das <i>issues</i>		Não
Neto e Braga e Silva [14]	Calcular a relevância dos comentários	Calcular o tempo de resolução das <i>issues</i>	Identificar <i>stakeholders</i>	Sim
Neto [13]	Análise de sentimentos e calcular a relevância dos comentários	Calcular o tempo de resolução das <i>issues</i>	Identificar <i>stakeholders</i>	Sim
Neto et al. [12]	Criar um modelo de predição	Criar um modelo de predição	Criar um modelo de predição	Não
Batista et al. [3]	Calcular a relevância dos comentários		Identificar o gênero dos desenvolvedores	Sim
Huang et al. [7]	Identificar comentários com perguntas na discussão das <i>issues</i>	Calcular o tempo de resolução das <i>issues</i>		Não
Este trabalho	Calcular a relevância dos comentários	Calcular o tempo de resolução das <i>issues</i>	Identificar papéis dos desenvolvedores nas <i>issues</i>	Sim

4.3 Análise dos Dados

As próximas subseções abordam o método utilizado para quantificar a relação entre as métricas M01 com M04, utilizada para responder as QP01; e as abordagens aplicadas para a QP03; e a limpeza de dados realizada. Vale ressaltar que gráficos de boxplots e dispersão foram utilizados para analisar a relação entre duas variáveis na QP01 e QP02.

4.3.1 Correlação de Spearman. Para analisar a relação entre duas variáveis, foi aplicado o método de correlação de Spearman. Esse método calcula o grau de relacionamento, linear ou não-linear, entre duas variáveis, gerando um valor entre -1 e 1. De acordo com Schober et al. [16], o coeficiente de correlação pode ser interpretado com base na Tabela 3.

Tabela 3: Interpretação dos Coeficientes de Correlação

Valor absoluto do coeficiente de correlação	Interpretação
[0,00, 0,10)	Correlação insignificante
[0,10, 0,40)	Correlação fraca
[0,40, 0,70)	Correlação moderada
[0,70, 0,90)	Correlação forte
[0,90, 1,00]	Correlação muito forte

Valores positivos e negativos indicam, respectivamente, uma correlação positiva e negativa, enquanto zero representa a ausência de correlação. Uma correlação positiva significa que o aumento de uma variável está relacionado ao aumento da outra variável, enquanto uma correlação negativa indica que o aumento de uma variável está relacionado à diminuição da outra variável.

O cálculo do método de correlação foi realizado utilizando a função *corr* disponível na classe *DataFrame* da biblioteca *Pandas*.

4.3.2 Teste Shapiro-Wilk. Em estatística, uma população se refere a um conjunto de elementos a ser estudado, ao passo que uma amostra se refere a um subconjunto de uma população. Quando se deseja comparar a média de duas amostras e não há todos os dados da população, é necessário realizar inferências sobre as mesmas. Um dos métodos utilizados para fazer essa comparação, é o teste t. Contudo esse teste parte de um pressuposto de que as amostras a serem comparadas seguem uma distribuição normal.

Para responder à QP03, surgiu a necessidade de aplicar um teste para comparar duas amostras, como o teste t. Para isso, foi necessário verificar a normalidade das amostras, a qual foi feita através do teste de Shapiro-Wilk. Esse teste verifica, estatisticamente, se uma amostra possui um distribuição normal, partindo das seguintes hipóteses:

- Hipótese nula (H_0): A amostra segue uma distribuição normal.
- Hipótese alternativa (H_1): A amostra não segue uma distribuição normal.

O resultado desse teste retorna um valor de p, que é utilizado para avaliar a rejeição ou não da hipótese nula, com base em um limiar α , estabelecido em 5% nesse estudo. Em outras palavras, se o valor de p for menor ou igual a 5%, a hipótese nula é rejeitada, caso contrário, a hipótese nula é aceita. Esse método foi aplicado diretamente através do módulo *stats* da biblioteca *SciPy*.

4.3.3 Mann-Whitney U. Quando duas amostras não seguem uma distribuição normal e há o intuito de compará-las, é possível aplicar um teste não paramétrico. Para responder à QP03, foi necessário comparar duas amostras que não apresentavam uma distribuição normal, sendo feita através do teste de Mann-Whitney U. Esse método segue as seguintes hipóteses:

- Hipótese nula (H_0): Não há diferença significativa entre as medianas das duas amostras.

- Hipótese alternativa (H_1): Há diferença significativa entre as medianas das duas amostras.

O resultado desse teste retorna um valor de p , com o mesmo objetivo do teste de Shapiro-Wilk. Para esse método, o valor de α também foi estabelecido em 5%, além de também ser aplicado através do módulo *stats* da biblioteca SciPy.

5 RESULTADOS E DISCUSSÕES

Esta seção apresenta os resultados das métricas propostas na seção 4.2 sobre dados de *issue tracking* da organização Ruby on Rails no GitHub. Os dados foram extraídos com a biblioteca RIT e a API REST do GitHub. Ao todo, foram obtidos 228.439 comentários provenientes de 49.108 *issues* de 88 repositórios da referida organização através da RIT. Ao passo que, a API REST do GitHub forneceu dados das datas de fechamento e abertura de 60.564 *issues* de 116 repositórios.

A diferença na quantidade de *issues* e repositórios obtidas para cada extração se dá por dois motivos: a biblioteca RIT não considera *issues* que não possuem comentários; e a extração dos dados necessários para calcular o tempo de resolução foi feita em um momento posterior à da extração feita pela RIT, por meio da API REST.

Das 49.108 *issues* exportadas pela RIT, duas foram removidas devido à presença de valores nulos para a relevância temática e duas não puderam ser analisadas devido à reabertura dessas *issues* entre a extração inicial realizada pela RIT e a obtenção dos dados via API REST. Consequentemente, não foi possível calcular o tempo de resolução para as *issues* reabertas. Como resultado dessa remoção, houve perda de informações de 10 comentários.

A Tabela 4 apresenta um resumo da quantidade de dados obtidos e analisados. Essa tabela mostra que a organização Ruby on Rails concentra um elevado número de *issues*, comentários e colaboradores, com uma amostra significativa de dados de comunicação em *issue tracking*. Além disso, no que compete a quantidade de desenvolvedores, 2.354 participam tanto de *issues* quanto de PRs.

	Issues	PRs	Total
Repositórios	-	-	116
Issues (incluindo PRs)	21.859	38.705	60.564
Comentários	107.130	121.299	228.439
Desenvolvedores	16.300	8.488	20.899

Tabela 4: Visão Geral dos Dados Obtidos

Para responder às Questões de Pesquisa propostas neste trabalho, as *issues* foram separadas em dois grupos: *issues* e *pull requests*. As *issues* referem-se a problemas reportados, enquanto as *pull requests* são solicitações de alteração de branches. Essa separação foi realizada devido aos diferentes objetivos desses elementos. As *issues* são voltadas para discussão e resolução de problemas, enquanto as *pull requests* são mais relacionadas à revisão de código, não havendo necessariamente discussão sobre elas. Dessa forma, é esperado que esses dois grupos tenham características diferentes, levando em consideração seus propósitos distintos.

As próximas subseções apresentam os resultados obtidos através da aplicação das métricas de M01 a M05, seguindo cada questão de pesquisa.

5.1 QP01. Existe Relação entre a Quantidade de Comentários em uma Issue e suas Respective Relevâncias Temáticas e o Tempo de Resolução?

Para responder a esta Questão de Pesquisa, realizou-se duas análises, uma para a relevância temática e o tempo de resolução, e outra para a quantidade de comentários e o tempo de resolução.

5.1.1 *Existe Relação Entre a Relevância Temática dos Comentários e o Tempo de Resolução?* Para analisar a relação entre a relevância temática dos comentários e o tempo de resolução, primeiramente foi calculado a média aritmética da relevância temática para cada *issue* e *pull request* (PR). Logo após, foi criado um gráfico de dispersão para observar a relação entre essas variáveis, visto na Figura 1. Por fim, o método de correlação de Spearman foi aplicado para quantificar essa relação.

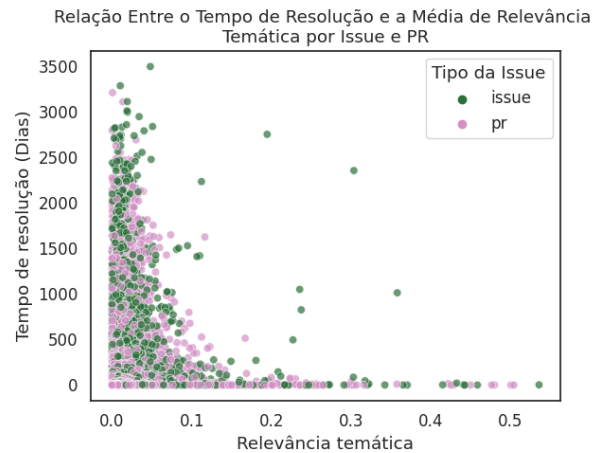


Figura 1: Gráfico de Dispersão da Média da Relevância Temática por *issues* e PRs e o Tempo de Resolução

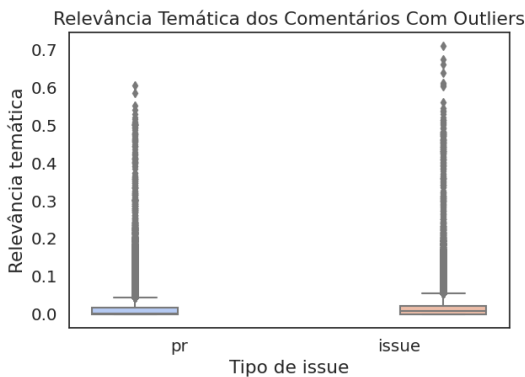
É possível observar que os dados estão mais concentrados entre 0 e 0.1 no eixo da relevância temática. Em outras palavras, a média de relevância temática dos comentários por *issue* e por PR é muito baixa, como já observado no trabalho de Batista et al. [3]. Além disso, não parece haver uma relação monótona entre as variáveis, ou seja, o aumento (ou diminuição) de uma variável não implica o aumento (ou diminuição) da outra.

Os valores de correlação calculados comprovam essa observação, sendo obtido para *issues* e PRs, respectivamente, 0,08 e 0,18. Conforme a Tabela 3, pode-se dizer que há uma correlação insignificante entre a média da relevância temática por *issue* e o tempo de resolução, e uma correlação fraca entre a média de relevância temática por PR e o tempo de resolução.

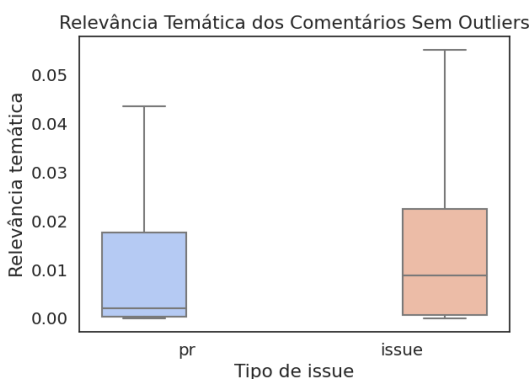
Embora a relevância temática e o tempo de resolução não estejam diretamente relacionados, não é possível afirmar que a relevância temática não influencie o tempo de resolução de uma *issue*. No entanto, observou-se que esse atributo sozinho não influencia o tempo de resolução. O processo de comunicação abrange um conjunto grande de variáveis, sendo em algumas ocasiões necessário a análise dessas variáveis em conjunto para identificar uma relação. Como

exemplo, Neto et al. [12] fizeram o uso de dezenas de atributos para criar um modelo capaz de prever o tempo de resolução de uma *issue*.

Para olhar um pouco mais de perto para os valores de relevância temática para *issues* e PRs, os dados foram analisados por meio de boxplots, apresentados na Figura 2. De forma similar ao gráfico de dispersão, é possível observar que os valores de relevância temática tanto para *issues* quanto para PRs são muito baixos. Para entender esse comportamento, alguns comentários foram analisados manualmente e observou-se que diversos deles possuem agradecimentos e emojis, os quais dificilmente vão possuir relevância no contexto desse trabalho. Além disso, identificou-se que aproximadamente 54% das *issues* e 38% dos PRs contêm elementos como URL, trecho de código, caminhos e referências de imagem. Dessa forma, há a possibilidade dos desenvolvedores terem se comunicado, em sua maior parte, através desses elementos, havendo assim pouco elemento textual para ser analisado, devido a execução da etapa de pré-processamento.



(a) Boxplot da Relevância Temática das *Issues* e PRs com *Outliers*



(b) Boxplot da Relevância Temática das *Issues* e PRs sem *Outliers*

Figura 2: Relevâncias Temáticas de *Issues* e PRs, ambas com valores baixos

5.1.2 *Existe relação entre a Quantidade de Comentários em uma Issue e o Tempo de Resolução?* Para analisar a relação entre a quantidade de comentários e o tempo de resolução, foram criados boxplots do tempo de resolução das *issues* e PRs de acordo com diferentes faixas de quantidades de comentários, como visto na Figura 3. Dessa forma, é possível analisar o comportamento do tempo de resolução conforme a variação da quantidade de comentários.

Visualmente, a Figura 3 mostra que o tempo de resolução, tanto para *issues* quanto para PRs, tende a aumentar conforme a quantidade de comentários aumenta. Aparentemente, colocar cada vez mais esforços na discussão não contribui unicamente para que uma *issue* ou PR seja resolvida mais rapidamente. Conforme apresentado por Huang et al. [7], perguntas que surgem no meio das discussões aumentam esse tempo de resolução. Sendo assim, quando *issues* ou PRs apresentarem cada vez mais um aumento na quantidade de comentários, os desenvolvedores devem investigar se esse aumento está sendo benéfico para a discussão ou não. Em caso negativo, é possível alocar mais esforços na modelagem do problema para que o mesmo seja entendido pelos colaboradores mais facilmente.

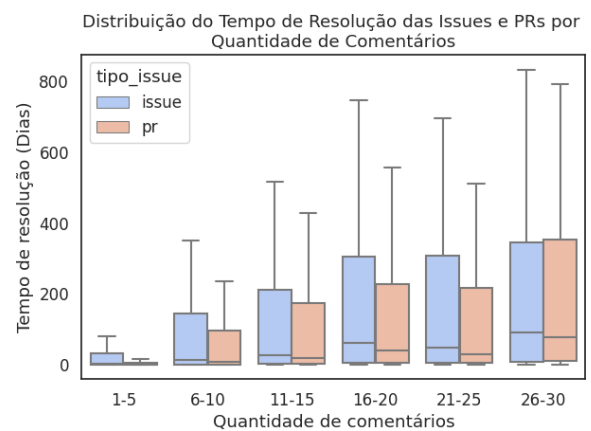
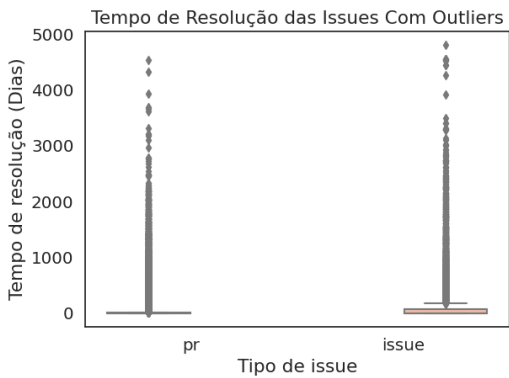
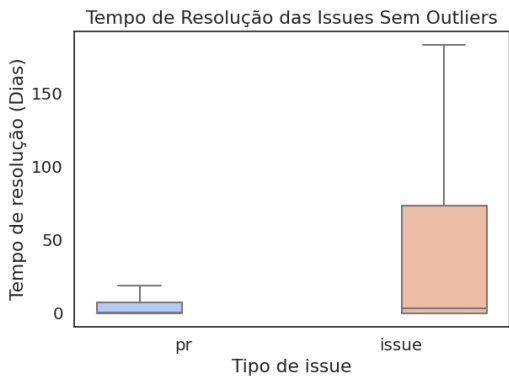


Figura 3: Boxplots sem *Outliers* do Tempo de Resolução das *Issues* e PRs por Quantidades de Comentários

5.1.3 *Issues e PRs.* Os resultados obtidos mostram que, apesar das *issues* e PRs possuírem objetivos diferentes, existem algumas características em comum entre eles. Por exemplo, ambos apresentaram baixa relevância temática, embora as *issues* demonstraram valores de relevância temática mais elevados em comparação aos PRs, como foi visto na Figura 2. Além disso, em ambos tipos de *issues*, há uma tendência do aumento no tempo de resolução conforme a quantidade de comentários aumenta. Contudo, a Figura 4 mostra que os PRs possuem um tempo de resolução proporcionalmente menor em relação às *issues*. Enquanto, aproximadamente, 75% das *issues* são solucionadas em até 3 meses, a mesma proporção dos PRs é resolvida em menos de 1 mês. Essa diferença no tempo de resolução pode-se dar pelo fato de PRs serem mais valorizadas pelos desenvolvedores do que as *issues*, uma vez que trazem contribuições concretas para o projeto.



(a) Boxplot do Tempo de Resolução das *Issues* e PRs com *Outliers*



(b) Boxplot do Tempo de Resolução das *Issues* e PRs sem *Outliers*

Figura 4: Boxplots do Tempo de Resolução das *Issues* e PRs

Além disso, diante da Figura 4, pode-se pensar em dois motivos para uma *issue* ou PR levarem aproximadamente décadas para serem resolvidas: ou foram reabertas ou foram esquecidas. No caso de terem sido reabertas, o tempo de resolução não diz exatamente quantos dias corridos esses elementos demoraram para serem resolvidos. Dessa forma, identificar quais *issues* e PRs foram reabertas e desconsiderá-las da análise pode trazer melhores resultados a respeito do impacto dos atributos aplicados neste trabalho no tempo de resolução.

5.2 QP02. Existe Relação entre a Quantidade de Desenvolvedores Participando de uma *Issue* e o Tempo de Resolução?

O método utilizado para responder a pergunta anterior foi o mesmo aplicado nesta Questão de Pesquisa. Dessa forma, foram criados boxplots do tempo de resolução das *issues* e PRs de acordo com diferentes faixas de quantidades de desenvolvedores, como feito para os comentários. Vale ressaltar que esses desenvolvedores se referem aos que participaram da discussão das *issues* ou PRs, postando comentários.

O gráfico da Figura 5 mostra que para as *issues*, há uma tendência de aumento no tempo de resolução de acordo com o aumento da quantidade de desenvolvedores em quase todas as faixas de comentários. Já os PRs apresentam essa mesma relação até as 3 primeiras faixas, sendo que nas faixas seguintes o tempo de resolução das PRs diminui conforme o aumento da quantidade de desenvolvedores, não havendo uma relação bem definida.

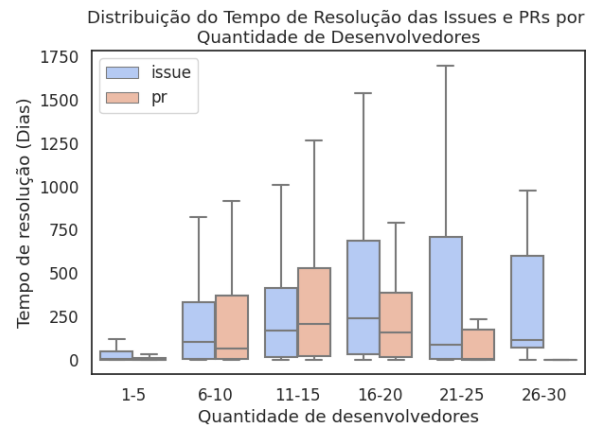


Figura 5: Boxplots Sem *Outliers* do Tempo de Resolução das *Issues* e PRs por Quantidades de Desenvolvedores

A mesma discussão feita sobre a relação entre a quantidade de comentários e o tempo de resolução das *issues* e PRs na QP01 pode ser aplicada nos resultados obtidos para as *issues* nesta Questão de Pesquisa, ou seja, quando as *issues* apresentarem cada vez mais um aumento na quantidade de desenvolvedores, deve-se investigar se esse aumento está sendo benéfico para a discussão ou não. Em caso negativo, investigar se os esforços devem ser investidos na elaboração da *issue*, pois aumentar o número de envolvidos não garante um tempo de resolução menor nem um melhor entendimento. Pelo contrário, pode inclusive aumentar o ruído na comunicação e estender o tempo de resolução.

5.3 QP03. Existe Diferença no Tempo de Resolução entre as *Issues* que foram Criadas, Comentadas ou Atribuídas por/para Top-Devs?

Primeiramente, para responder a essa pergunta, foi necessário identificar os top-devs. Como resultado dessa identificação, apenas 24 de 20.899 desenvolvedores foram selecionados, sendo responsáveis por 43 comentários com relevância temática maior do que 0,5. Vale lembrar que a relevância temática varia entre 0 e 1. Logo após, foi necessário definir as amostras de duas populações: *issues* e PRs. As amostras podem ser vistas a seguir:

- A = *Issues*/PRs que os top-devs em *issue tracking* participam;
- \bar{A} = *Issues*/PRs que os top-devs em *issue tracking* não participam;
- B = *Issues*/PRs criadas pelos top-devs em *issue tracking*;

- \tilde{B} = *Issues*/PRs que não foram criadas por top-devs em *issue tracking*;
- C = *Issues*/PRs atribuídas aos top-devs em *issue tracking*;
- \tilde{C} = *Issues*/PRs que não foram atribuídas aos top-devs em *issue tracking*.

As amostras B e \tilde{B} , bem como C e \tilde{C} , juntas formam a sua respectiva população. Neste caso, como há acesso a todas as informações da população, não é necessário aplicar métodos de inferência estatística para comparar essas amostras. É possível simplesmente analisar os dados brutos e fazer as comparações desejadas.

Por outro lado, as amostras A e \tilde{A} juntas não formam as populações completas. Isso ocorre porque para identificar as *issues* e PRs pertencentes a cada amostra foi necessário utilizar duas bases de dados, uma obtida pela RIT e outra obtida pela API REST do GitHub. Os dados obtidos da API REST possuem o tempo de resolução de todas as *issues* e PRs da organização Ruby on Rails. Por outro lado, a RIT não fornece informações sobre *issues* e PRs com quantidade de comentários igual a 0. Dessa forma, como a extração feita pela API REST foi realizada um momento posterior ao da RIT, não foi possível considerar que as *issues* e PRs que não estavam na base de dados obtidos pela RIT não possuíam participação com comentários dos top-devs (já que possuem quantidade de comentários igual a 0), pois *issues* e PRs podem ter sido fechadas entre o período de obtenção das duas base de dados, não havendo possibilidade de identificar se esses desenvolvedores participaram ou não da sua discussão. Portanto, para essas amostras, é necessário realizar inferências estatísticas para realizar comparações.

As Tabelas 5 e 6 apresentam, respectivamente, a quantidade de *issues* e PRs encontradas para essas amostras e a média aritmética e desvio padrão do tempo de resolução de cada uma. Os valores de desvios padrões presentes nessa tabela apontam que os dados das amostras apresentam uma alta variabilidade, ou seja, os dados estão bem dispersos da sua respectiva média.

Tabela 5: Quantidade de *Issues* e PRs para cada Amostra

	A	\tilde{A}	B	\tilde{B}	C	\tilde{C}
<i>Issues</i>	7.131	13.466	552	21.307	270	21.589
PRs	15.128	13.379	1.425	37.280	2.002	36.703

De acordo com a Tabela 5, os desenvolvedores selecionados, apesar de serem poucos, participam de mais de 22.000 *issues* e PRs, mostrando serem bem participativos nas resoluções dos problemas e revisões de códigos. No entanto, esses desenvolvedores não foram atribuídos e nem criaram muitas *issues* e PRs se comparados com a quantidade total.

Em particular, para responder se existe diferença no tempo de resolução de *issues* e PRs para as amostras A e \tilde{A} , foi realizado um teste de Shapiro-Wilk para verificar a normalidade das amostras. Foi obtido um valor de p próximo de 0 para ambas as amostras, rejeitando a hipótese nula, ou seja, os dados não seguem uma distribuição normal. Dessa forma, foi aplicado o teste de Mann-Whitney. Como resultado desse teste, o valor de p retornado foi próximo de 0 para as *issues* e 0,12 para os PRs, ou seja, as amostras A e \tilde{A} das *issues* são estatisticamente diferentes, enquanto os PRs não. De outra forma, *issues* com comentários de top-devs levam, em média,

mais tempo para serem resolvidas do que as que não contam com a sua participação, enquanto os PRs que esses top-devs participam levam, em média, o mesmo tempo para serem resolvidos do que as que eles não participam.

Apesar das *issues* contarem com comentários de desenvolvedores que sabem se comunicar, as mesmas podem ser criadas por desenvolvedores que não possuem essa habilidade. Por exemplo, Huang et al. [7] identificou que *issues* que contém perguntas na discussão estão relacionadas a um maior tempo de resolução. Isso pode ocorrer principalmente pela má escrita do relatório do problema, gerando dúvidas e exigindo mais tempo para a sua resolução. Os resultados posteriores mostram que *issues* e PRs criadas por desenvolvedores que postam comentários de qualidade estão associados a um menor tempo médio de resolução.

Como não é necessário realizar inferência estatística para comparar as amostras B e \tilde{B} , e C e \tilde{C} para cada tipo de *issue*, foi observado unicamente as médias para responder se há diferença nas *issues* ou PRs que os top-devs criaram ou foram atribuídos. Dessa forma, é possível interpretar da seguinte forma os valores da tabela para essas amostras:

- (1) *Issues* criadas por top-devs em *issue tracking* são resolvidas, em média, mais rápido do que as que eles não criaram, apesar de pouca diferença. Por outro lado, os PRs que esses desenvolvedores criaram possuem uma média de tempo de resolução bem menor do que as que eles não criaram.
- (2) Tanto as *issues* quanto os PRs atribuídos aos top-devs em *issue tracking* demoram, em média, muito mais tempo para serem resolvidas do que as que eles não foram atribuídos.

Diante da interpretação do item 1, pode-se pensar que como esses desenvolvedores postam comentários relevantes, ou seja, são bons na comunicação, eles também são capazes de criar títulos, descrições e até mesmo de códigos melhores para as *issues* e PRs, facilitando o processo de entendimento e resolução das mesmas. Além disso, essa interpretação reforça a ideia dos desenvolvedores alocarem mais esforço na modelagem do problema do que em discussões, quando a mesma não está colaborando para a resolução da *issue* ou PR.

A Tabela 5 mostra que os top-devs em *issue tracking* participam ativamente nas discussões das *issues* e PRs. Sendo assim, o fato do item 2 mostrar um elevado tempo médio de resolução nas *issues* que a eles são atribuídas pode indicar uma sobrecarga em cima desses desenvolvedores. O trabalho de Martins [10] mostra que *issues* atribuídas a desenvolvedores com multitarefa possui mediana igual a 232 dias, enquanto os que não estão, apresentam mediana igual a 32 dias. A multitarefa nesse estudo é usada para identificar se um desenvolvedor está atribuído a mais de uma *issue*. No contexto deste trabalho, pode-se pensar em multitarefa também como a participação de um desenvolvedor na discussão em mais de uma *issue* ao mesmo tempo.

Além disso, para melhorar a interpretação desses dados, as Figuras 6, 7, 8 apresentam, respectivamente os boxplots para as amostras A e \tilde{A} , B e \tilde{B} , e C e \tilde{C} , as quais reforçam as afirmações feitas nesta Questão de Pesquisa em relação a essa amostras.

Tabela 6: Médias e Desvios Padrões das Amostras em Dias

	A		Ã		B		B̃		C		C̃	
	MA	DP	MA	DP	MA	DP	MA	DP	MA	DP	MA	DP
Issues	135,82	324,95	79,25	220,41	99,33	258,67	100,50	287,80	197,40	393,17	99,25	285,33
PRs	86,69	266,12	59,25	205,79	36,13	187,93	59,12	221,40	227,36	479,46	49,05	192,35

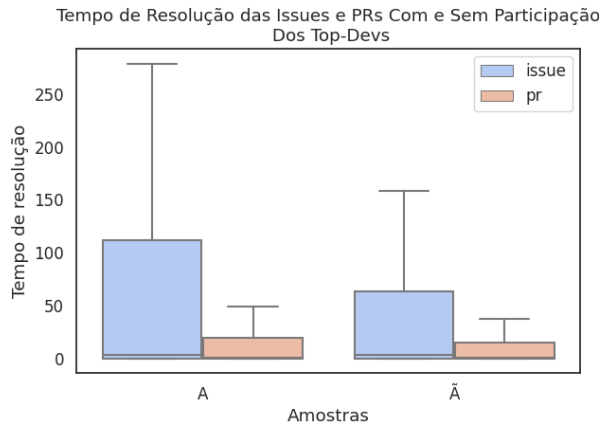


Figura 6: Boxplots Sem Outliers do Tempo de Resolução das *issues* e PRs Para as Amostras A e Ã

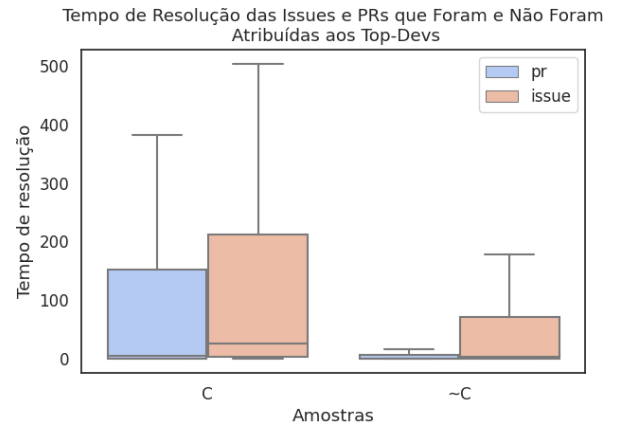


Figura 8: Boxplots Sem Outliers do Tempo de Resolução das *issues* e PRs Para as Amostras C e C̃

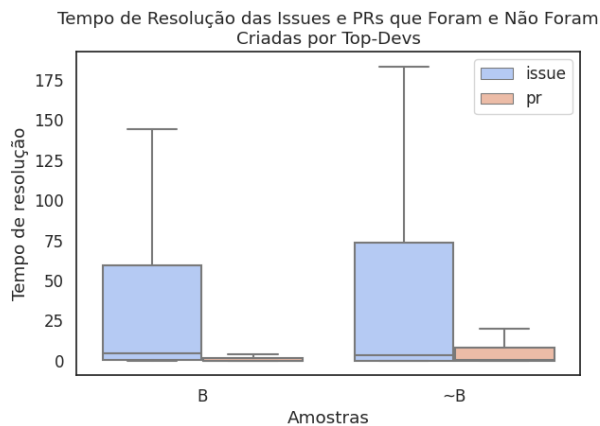


Figura 7: Boxplots Sem Outliers do Tempo de Resolução das *issues* e PRs Para as Amostras B e B̃

6 AMEAÇAS À VALIDADE

Como a RIT não exporta informações sobre *issues* e PRs que não possuem comentários, para a análise da QP01 e QP02, foram considerados apenas as *issues* e PRs fornecidos pela RIT. Conseqüentemente, não foi possível incluir *issues* e PRs com quantidades de comentários e desenvolvedores iguais a 0 nos boxplots apresentados. Dessa forma, a inserção dessas *issues* e PRs na análise pode alterar os resultados obtidos na faixa inicial de cada boxplot.

Ademais, os boxplots apresentados na QP03 mostram que *issues* e PRs levam milhares de dias para serem resolvidas. Isso pode ocorrer devido à reabertura das mesmas, ou seja, não há certeza que as *issues* e PRs demoram milhares de dias corridos para serem resolvidas. Desconsiderar a reabertura pode trazer resultados mais consistentes em relação ao tempo de resolução das *issues* e PRs, pois a quantidade de *outliers* provavelmente será reduzida.

7 CONSIDERAÇÕES FINAIS

O presente trabalho apresentou uma análise de dados em sistema de *issue tracking* com o intuito de avaliar o impacto de atributos como a relevância temática, quantidade de comentários, quantidade de desenvolvedores e de top-devs no tempo de resolução das *issues* e *pull requests*.

Através dessa pesquisa, observou-se que a média do tempo de resolução por quantidade de comentários desses elementos tende a aumentar conforme o número de comentários presente aumenta. Além disso, *issues* e PRs que foram criadas por top-devs são resolvidas, em média, mais rapidamente do que as que eles não criaram. Por outro lado, quando esses desenvolvedores são responsáveis por uma *issue* ou PR, o tempo médio de resolução é maior. Ademais, observando quando esses desenvolvedores estão participando da discussão, as *issues* apresentaram um tempo de resolução médio maior do que quando eles não participam, enquanto não foi encontrada diferença para os PRs.

Embora não foi encontrada correlação entre o tempo de resolução de *issues* e PRs e a relevância temática, a mesma pode ser analisada em conjunto com outras variáveis do processo de comunicação para que seja identificada alguma relação, visto que a comunicação

envolve uma gama de atributos e variáveis que influenciam o tempo de resolução das *issues* e PRs, tais como, complexidade, gravidade e prioridade. Além disso, observou-se que a relevância temática apresentou valores muito baixos, em sua maioria, entre 0 e 0,1. Já o tempo de resolução chegava na casa de milhares de dias, sendo uma possível causa a reabertura desses elementos. Contudo, cerca de 75% das *issues* e PRs são resolvidas, respectivamente, em menos de 3 meses e em menos de 1 mês.

Por fim, como trabalhos futuros, destacam-se: a) realizar novas análises, retirando da amostra *issues* que foram reabertas; b) calcular uma métrica de reputação para desenvolvedores com base na sua atividade em *issue tracking*; c) implementar ajustes na biblioteca RIT, tal como a exportação de informações sobre *issues* e *pull requests* que contenham quantidade de comentários igual a zero.

REFERÊNCIAS

- [1] Breno Fabrício Terra Azevedo. 2011. MineraFórum: um recurso de apoio para análise qualitativa em fóruns de discussão. (2011).
- [2] Estela Miranda Batista, Melissa Araújo, Fábio Trindade Ramos, Gláucia Braga, Luiz Eugênio Coelho Neto, Santiago de Oliveira Souza, et al. 2023. RIT: uma Biblioteca para Análise da Relevância Temática dos Comentários Postados no Issue Tracking do GitHub. *Anais do Computer on the Beach* 14 (2023), 094–101.
- [3] Estela Miranda Batista, Gláucia Braga e Silva, and Thais Regina de Moura Braga e Silva. 2023. Gender Diversity on GitHub Issue Tracking: What's the Difference? *Journal on Interactive Systems* 14, 1 (Jun. 2023), 128–137. <https://doi.org/10.5753/jis.2023.3126>
- [4] Dane Bertram, Amy Volda, Saul Greenberg, and Robert Walker. 2010. Communication, collaboration, and bugs: the social nature of issue tracking in small, collocated teams. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. 291–300.
- [5] Hudson Borges, Jailton Coelho, Paulo Carvalho, Mariane Fernandes, and Marco Tulio Valente. 2016. Como Pesquisadores Usam o Dataset GHTorrent?. In *5th Brazilian Workshop on Software Visualization, Evolution and Maintenance (VEM)*. 1–8.
- [6] Valerio Cosentino, Javier Luis, and Jordi Cabot. 2016. Findings from GitHub: methods, datasets and limitations. In *Proceedings of the 13th International Conference on Mining Software Repositories*. 137–141.
- [7] Yonghui Huang, Daniel Alencar da Costa, Feng Zhang, and Ying Zou. 2019. An empirical study on the issue reports with questions raised during the issue resolving process. *Empirical Software Engineering* 24 (2019), 718–750.
- [8] Marcia Lima, Igor Steinmacher, Denae Ford, Evangeline Liu, Grace Vorreuter, Tayana Conte, and Bruno Gadelha. 2022. Looking for related discussions on GitHub Discussions. *arXiv preprint arXiv:2206.11971* (2022).
- [9] Crystiano José Richard Machado, Alexandre Magno Andrade Maciel, Rodrigo Lins Rodrigues, and Ronaldo Menezes. 2019. An approach for thematic relevance analysis applied to textual contributions in discussion forums. *International Journal of Distance Education Technologies (IJDET)* 17, 3 (2019), 37–51.
- [10] Matheus Felipe F Martins. 2022. Análise da Influência da Multitarefa de Desenvolvedores no Ciclo de Vida de Issues do GitHub. Pontifícia Universidade Católica de Minas Gerais.
- [11] Marco Tulio Mombach, Thais e Valente. 2018. GitHub REST API vs GHTorrent vs GitHub Archive: A comparative study. Universidade Federal de Minas Gerais.
- [12] Luiz Neto, Gláucia Silva, and Giovanni Comarela. 2021. Estimativa do tempo de resolução de issues no github usando atributos textuais e temporais. In *Proceedings of the XXXV Brazilian Symposium on Software Engineering*. 253–262.
- [13] Luiz Eugênio Neto. 2018. Quem faz a diferença? Uma abordagem para análise da comunicação entre os stakeholders de um projeto de software. Universidade Federal de Viçosa - campus Florestal.
- [14] Neto e Braga e Silva Neto e Braga e Silva. 2018. ColMiner: uma ferramenta de apoio ao gerenciamento das comunicações em um ambiente de issue tracking. In *Anais do XIV Simpósio Brasileiro de Sistemas de Informação*. SBC, 392–399.
- [15] Marco Ortu, Bram Adams, Giuseppe Destefanis, Parastou Tourani, Michele Marchesi, and Roberto Tonelli. 2015. Are bullies more productive? Empirical study of affectiveness vs. issue fixing time. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*. IEEE, 303–313.
- [16] Patrick Schober, Christa Boer, and Lothar A Schwarte. 2018. Correlation coefficients: appropriate use and interpretation. *Anesthesia & analgesia* 126, 5 (2018), 1763–1768.
- [17] M Vidoni. 2022. A systematic process for Mining Software Repositories: Results from a systematic literature review. *Information and Software Technology* 144 (2022), 106791.
- [18] Feng Zhang, Foutse Khomh, Ying Zou, and Ahmed E Hassan. 2012. An empirical study on factors impacting bug fixing time. In *2012 19th Working conference on reverse engineering*. IEEE, 225–234.