PGCOMP - Programa de Pós-Graduação em Ciência da Computação
Universidade Federal da Bahia (UFBA)
Av. Milton Santos, s/n - Ondina
Salvador, BA, Brasil, 40170-110

https://pgcomp.ufba.br
pgcomp@ufba.br

A reabertura de issues representa um desafio significativo no desenvolvimento e manutenção de software, aumentando os custos e a complexidade dos esforços envolvidos. Essa ocorrência muitas vezes indica problemas não resolvidos ou mal compreendidos na comunicação entre os colaboradores do projeto e os usuários em plataformas como o GitHub.

Esta tese tem como objetivo aprofundar a compreensão do conceito de reaberturas de issues em repositórios de software de código aberto do GitHub, considerando dados históricos, categorização de issues e análise de sentimentos dos desenvolvedores presentes nas discussões associadas a essas issues.

Nossa metodologia envolveu o uso da ferramenta SentiStrength-SE, adaptada para léxicos da área de Engenharia de Software, para calcular a polaridade e o sentimento nos textos das discussões relacionadas às issues. Em seguida, desenvolvemos um modelo de categorização automática de issues, classificando-as em categorias específicas, como banco de dados, configuração, desempenho, funcional, GUI, info, permissão/obsoleto, redes, segurança e testes. Essa abordagem permite uma priorização mais eficaz na resolução das issues reabertas, direcionando recursos de forma mais precisa. Por último, caracterizamos a reabertura de issues de acordo com os sentimentos dos desenvolvedores contidos nos textos das discussões em cada categoria de issue. Os resultados revelaram que a análise de sentimentos, quando aplicada isoladamente, não mostrou uma métrica eficaz para identificar reaberturas de issues. No entanto, identificamos que certos tipos de categorias de issues estão mais propensos a problemas relacionados à reabertura. Isso aponta para a importância da categorização de issues em conjunto com a análise de sentimentos para uma abordagem mais eficiente na prevenção e tratamento das reaberturas de issues em repositórios de software de código aberto.

Palavras-chave: Reabertura de issues, Análise de sentimentos, Categorização de issues

DSC | 48 | 2024

Uma Investigação sobre Análise de Sentimentos e Categorização de Issues Reabertas do GitHub

Gláucya Carreiro Boechat

UFBA

Uma Investigação sobre Análise de Sentimentos e Categorização de Issues Reabertas do GitHub

Gláucya Carreiro Boechat

Tese de Doutorado

Universidade Federal da Bahia

Programa de Pós-Graduação em
Ciência da Computação

Março | 2024

Universidade Federal da Bahia
Instituto de Computação

Programa de Pós-Graduação em Ciência da Computação

UMA INVESTIGAÇÃO SOBRE ANÁLISE DE SENTIMENTOS E CATEGORIZAÇÃO DE ISSUES REABERTAS DO GITHUB

Gláucya Carreiro Boechat

TESE DE DOUTORADO

Salvador
08 de março de 2024

GLÁUCYA CARREIRO BOECHAT

UMA INVESTIGAÇÃO SOBRE ANÁLISE DE SENTIMENTOS E CATEGORIZAÇÃO DE ISSUES REABERTAS DO GITHUB

Esta Tese de Doutorado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Orientador: Manoel Gomes de Mendonça Neto
Co-orientador: Ivan do Carmo Machado

Salvador
08 de março de 2024

Ficha catalográfica elaborada pela Biblioteca Universitária de Ciências e Tecnologias
Prof. Omar Catunda, SIBI - UFBA.

B669 Boechat, Gláucya Carreiro.
Uma Investigação sobre Análise de Sentimentos e Categorização de Issues Reabertas do GitHub / Gláucya Carreiro Boechat – Salvador, 2024.
166f.

Orientador: Prof. Dr. Manoel Gomes de Mendonça Neto.
Co-orientador: Prof. Dr. Ivan do Carmo Machado.
Tese (Doutorado) – Universidade Federal da Bahia, Instituto de Computação, 2024.

1. Reabertura de issues. 2. Análise de sentimentos. 3. Categorização de issues. I. Mendonça, Manoel Gomes de. II. Machado, Ivan do Carmo. III. Universidade Federal da Bahia. Instituto de Computação. IV. Título.

CDU – 004

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE COMPUTAÇÃO
PGCOMP - Programa de Pós-Graduação em Ciência da Computação
http://pgcomp.ufba.br

"*Uma Investigação sobre Análise de Sentimentos e Categorização de Issues Reabertas do GitHub*"

Glaucya Carreiro Boechat

Esta tese foi julgada adequada à obtenção do título de Doutor em Ciência da Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da UFBA.

Banca Examinadora

Documento assinado digitalmente
gov.br MANOEL GOMES DE MENDONCA NETO
Data: 15/03/2024 11:51:10-0300
Verifique em https://validar.iti.gov.br

_____
Prof. Dr. Manoel Gomes de Mendonça Neto (Orientador - UFBA)

Documento assinado digitalmente
gov.br RODRIGO ROCHA GOMES E SOUZA
Data: 11/04/2024 11:40:38-0300
Verifique em https://validar.iti.gov.br

_____
Prof. Dr. Rodrigo Rocha Gomes e Souza (UFBA)

Documento assinado digitalmente
gov.br EMMANUEL SAVIO SILVA FREIRE
Data: 04/04/2024 15:53:55-0300
Verifique em https://validar.iti.gov.br

_____
Prof. Dr. Emmanuel Sávio Silva Freire (IFCE)

Documento assinado digitalmente
gov.br GLAUCO DE FIGUEIREDO CARNEIRO
Data: 25/03/2024 09:45:22-0300
Verifique em https://validar.iti.gov.br

_____
Prof. Dr. Glauco de Figueiredo Carneiro (UFS)

Documento assinado digitalmente
gov.br MARIO ANDRE DE FREITAS FARIAS
Data: 18/03/2024 16:52:31-0300
Verifique em https://validar.iti.gov.br

_____
Prof. Dr. Mário André de Freitas Farias (IFS)

Salvador, 08 de março de 2024

Dedico a minha familia.

AGRADECIMENTOS

A Deus, pela vida e pelo seu amor, que possibilitou a conclusão de mais uma etapa importante na minha vida.

Agradeço aos meus orientadores, Manoel Mendonça e Ivan Machado, por toda orientação, incentivo, paciência e compreensão.

Ao meu esposo, Rubisley, meus filhos Diego e Gael, meus pais, Joana D'arc e Astrogildo, meus irmãos, Cyntia e Rycardo, tia Tereza(in memoria), sobrinhos, Victor e Isabelle, e demais parentes, pelo amor, incentivo e, principalmente, pela confiança.

Gostaria de deixar meus sinceros agradecimentos aos grupos de pesquisa LABES[2], Aries Lab e Cemantika, Lab Ines, e aos meus amigos e colegas da UFBA, em especial Ana Maria, Michelle, Tiago, Alcemir, Magno, Lari, Leandro, Cretchas, Jonatas, Renata, Savio, Alberto, Jaziel, Thiago, Mirella, Rai, Dhennya, Dusse, Sara, Thialia, Rafael, Danniel, Ailton, e Stéfani.

Agradeço imensamente aos meus queridos amigos Joselito e Hiolanda pela valiosa ajuda no trabalho. Às minhas amigas de longa data, Jenny e Lu, e às amigas de república, Laura, Mile e Nick, bem como a Bela, Mario e minha tia Conchita, pelo constante suporte e amizade ao longo dessa jornada.

Expresso minha gratidão a todos os professores e profissionais do Instituto de Computação por todo o apoio ao longo da realização do meu trabalho. Agradeço também aos professores Reginaldo Soeiro, Akebo Yamakami, Takaaki Ohishi, Antonio Scarpelli e André Gustavo dos Santos, pelas cartas de recomendação.

Agradeço também aos membros da banca examinadora pelo interesse e disponibilidade.

Ao apoio financeiro da Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB).

Por fim, agradeço a todos que contribuíram direta ou indiretamente para o desenvolvimento desta tese.

Emotions change how we see the world and how we interpret the actions of others.

—PAUL EKMAN (Emotions Revealed)

RESUMO

A reabertura de issues representa um desafio significativo no desenvolvimento e manutenção de software, aumentando os custos e a complexidade dos esforços envolvidos. Essa ocorrência muitas vezes indica problemas não resolvidos ou mal compreendidos na comunicação entre os colaboradores do projeto e os usuários em plataformas como o GitHub.

Esta tese tem como objetivo aprofundar a compreensão do conceito de reaberturas de issues em repositórios de software de código aberto do GitHub, considerando dados históricos, categorização de issues e análise de sentimentos dos desenvolvedores presentes nas discussões associadas a essas issues.

Nossa metodologia envolveu o uso da ferramenta SentiStrength-SE, que conta com um léxico especializado para a Engenharia de Software, para calcular a polaridade e o sentimento nos textos das discussões relacionadas às issues. Desenvolvemos também um modelo de categorização automática de issues, que as classifica em categorias específicas, como banco de dados, configuração, desempenho, funcional, GUI, info, permissão/obsoleto, redes, segurança e testes. Essa abordagem permite uma priorização mais eficaz na resolução das issues reabertas, direcionando recursos de forma mais precisa. Além disso, caracterizamos a reabertura de issues de acordo com os sentimentos dos desenvolvedores contidos nos textos das discussões em cada categoria.

Os resultados revelaram que a análise de sentimentos, quando aplicada isoladamente, não mostrou uma métrica eficaz para identificar reaberturas de issues. No entanto, identificamos que certos tipos de categorias de issues estão mais propensos a problemas relacionados à reabertura. Isso aponta para a importância da categorização de issues em conjunto com a análise de sentimentos para uma abordagem mais eficiente na prevenção e tratamento das reaberturas de issues em repositórios de software de código aberto.

Palavras-chave: Reabertura de issues, análise de sentimentos, categorização de issues, mineração de repositório de software.

ABSTRACT

The reopening of issues represents a significant challenge in software development and maintenance, increasing the costs and complexity of the efforts involved. This occurrence often indicates unresolved or misunderstood issues in communication between project collaborators and users on platforms like GitHub.

This thesis aims to deepen the understanding of issue reopenings in open-source GitHub software repositories, considering historical data, issue categorization, and sentiment analysis of developers involved in the associated discussions.

Our methodology involved using the SentiStrength-SE tool, adapted for lexicons in the field of Software Engineering, to calculate polarity and sentiment in the texts of discussions related to issues. Subsequently, we developed an automated issue categorization model, classifying them into specific categories such as configuration, database-related, program anomaly, performance, functional, GUI-related, info, permission/deprecation, network, security, and testing. This approach enables more effective prioritization in resolving reopened issues, directing resources more accurately. Finally, we characterized issue reopenings based on the sentiments of developers expressed in discussions within each issue category.

The results revealed that sentiment analysis, when applied in isolation, did not prove to be an effective metric for identifying issue reopenings. However, we identified that certain types of issue categories are more prone to problems related to reopening. This underscores the importance of combining issue categorization with sentiment analysis for a more efficient approach to preventing and addressing issue reopenings in open-source software repositories.

Keywords: Reopened issues, sentiment analysis, issue classification, mining software repositories

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

Capítulo 1

INTRODUÇÃO

1.1 MOTIVAÇÃO

A manutenção de software é uma etapa vital do ciclo de vida de qualquer projeto, pois abrange desde pequenas correções de erros de codificação até melhorias substanciais que buscam a adaptação a novos requisitos e a correção de erros de design e especificação (SOMMERVILLE, 2010). As atividades de manutenção frequentemente superam os esforços de desenvolvimento do software em termos de custos e complexidade. Em um cenário em que recursos são preciosos, evitar ou minimizar retrabalho é muito importante. Problemas reportados por engenheiros de software devem ser corretamente resolvidos de maneira eficaz, garantindo que não gerem retrabalho na fase de manutenção e evolução do software, como salientado por Xia et al. (2015).

Em geral, problemas de manutenção são relatados através um sistema de rastreamento de issues ou bugs (*Issue Tracking Systems* (ITS)). Em alinhamento com a terminologia atualmente adotada na literatura, este documento utilizará o anglicismo *issues* para se referir aos problemas ou questões de manutenção reportados aos engenheiros de software, e utilizará a expressão ITS para se referir aos sistemas de rastreamento dessas *issues*.

A etapa de manutenção de software vai além da resolução de problemas específicos; ela também abrange a classificação ou categorização das issues de acordo com sua natureza e gravidade. Neste contexto, é crucial a correta categorização ou classificação dessas issues, uma vez que essa organização desempenha um papel fundamental na priorização e alocação eficaz de recursos, garantindo que os problemas mais críticos sejam abordados primeiros, conforme evidenciado em estudos anteriores (TAN et al., 2014; IZQUIERDO et al., 2015; CATOLINO et al., 2019).

No entanto, um desafio recorrente reside na precisão da categorização realizada pelos colaboradores, uma vez que muitas vezes não conseguem fazer uma distinção clara entre as diferentes categorias de issues, conforme apontado por Pandey et al. (2018). Além disso, uma análise recente do GitHub revelou que em menos de 80% dos repositórios, as issues são devidamente rotuladas ou categorizadas, destacando a importância contínua de melhorar esse aspecto na gestão de projetos de software (JÚNIOR; BOECHAT; MACHADO, 2021).

Em sistemas de software complexos, a reabertura de issues é uma ocorrência comum (MI; KEUNG, 2016). Issues podem ser inadequadamente reportadas, e as soluções propostas podem ser : (1) *incorretas*, quando não resolvem o problema reportado; (2) *incompletas*, quando não abarcam todo o espaço de solução demandado pelo problema; ou (3) *inconsistentes*, quando conflitam de forma negativa com o funcionamento de outras partes do sistema de software. Resultando em retrabalho e aumentando os custos de manutenção.

Esse trabalho adicional de resolver issues reabertas pode prejudicar a qualidade geral do software, minar a confiança dos colaboradores, sobrecarregar os desenvolvedores e atrasar a entrega de novas versões (PAN; MAO, 2014; XIA et al., 2015). Portanto, entender os motivos que levam à reabertura de issues é fundamental (CAGLAYAN et al., 2012), bem como identificar se uma issue é mais propensa a ser reaberta (SHIHAB et al., 2013).

Os comentários presentes nas discussões de issues, postados por colaboradores no ITS, são uma fonte de informação valiosa para atacar o problema de rearbetura de issues. Estas discussões contêm não apenas informações técnicas, mas também nuances sobre a percepção e os sentimentos das pessoas em relação a um problema específico (ORTU et al., 2015b). No trabalho de Cheruvelil e Silva (2019) foi encontrado evidência sobre a correlação entre sentimentos negativos em comentários de issues e a reabertura dessas issues. Portanto, esta pesquisa concentra-se na análise de sentimentos e categorização de issues em ITS e sua relação com a reabertura de issues em projetos de engenharia de software.

Em ITS, quando uma issue é reportada, ela deve ser validada por uma equipe de triagem e designada para um ou mais engenheiros de software para resolvê-la (ANVIK; HIEW; MURPHY, 2006). A solução produzida pelos engenheiros deve então ser validada por um líder de equipe, que pode retornar a solução para nova revisão, ou pode então finalmente fechar a issue no ITS, dando-a como solucionada. Este processo de solução e fechamento de issues é geralmente acompanhado de discussões e troca de mensagens suportada pelo próprio ITS. A figura 1.1 representa um exemplo simplificado do ciclo de vida de uma issue do Github, o estado “Open” representa que a issue está aberta e o estado “Closed” representa que a issue está fechada. A transição “Opened” mostra o momento que a issue é reportada em um ITS, a transição “Closed” representa que a issue foi resolvida ou finalizada e a transição “Reopened” representa que a issue foi reaberta.

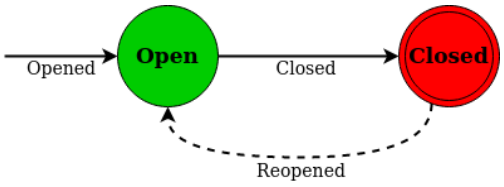


Figura 1.1: Ciclo de vida simplificado de uma issue do Github

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Caracterizar a reabertura de issues em repositórios de software de código aberto com respeito à categoria da issue e ao sentimento dos desenvolvedores nas discussões sobre estas issues.

1.2.2 Objetivos Específicos

Obj 1. Conduzir um levantamento bibliográfico sobre análise de sentimentos, categorização de issues e o problema de reabertura de issues em engenharia de software (Capítulo 2).

Obj 2. Conduzir um estudo piloto para investigar a eficácia ferramentas de análise de sentimentos voltadas para o desenvolvimento de software (Seção 3.1).

Obj 3. Conduzir um estudo empírico para validar a ferramenta de análise de sentimentos e explorar contextos para o seu uso (Seção 3.2).

Obj 4. Conduzir estudos empíricos para avaliar issues reabertas em projetos do GitHub (Seções 3.3 e 3.4).

Obj 5. Criar e desenvolver um categorizador automático de issues do GitHub (Seção 4.1).

Obj 6. Conduzir estudos empíricos para avaliar categorias de issues reabertas em projetos do GitHub (Seção 4.2).

Obj 7. Conduzir um estudo empírico para avaliar os sentimentos de acordo com as categorias de issues reabertas em projetos do GitHub (Capítulo 5).

1.3 QUESTÕES DE PESQUISA

Definimos como pergunta de pesquisa principal: **É possível utilizar a análise de sentimentos para criar um modelo de predição de reabertura de issues em repositórios de software de código aberto?**

A partir dessa pergunta principal, derivamos as seguintes perguntas de pesquisa:

QP1. Qual é a melhor ferramenta de análise de sentimentos voltada para a área de desenvolvimento de software para classificar os sentimentos de issues em larga escala?

- **QP1.1.** Quais são as principais ferramentas de análise de sentimentos voltadas para a área de desenvolvimento de software?
- **QP1.2.** Como validar a classificação de polaridades do dicionário léxico da ferramenta SentiStrength-SE?

QP2. Como caracterizar a influência dos sentimentos nas reabertura de issues?

QP2.1. Existe algum indicativo de que uma issue não será reaberta se ela for fechada com um sentimento positivo?

QP2.2. É possível predizer se uma issue será reaberta quando o número de sentimentos negativos adicionados ao número de sentimentos neutros for maior que o número de sentimentos positivos presentes nas suas discussões?

QP2.3. Os sentimentos dos comentários após o fechamento da issue pode indicar que ela será reaberta?

QP2.4. Existe algum indicativo de que uma issue fechada será reaberta com base no sentimento, pontuação negativa e pontuação positiva do primeiro comentário após a abertura da issue?

QP2.5. Existe algum indicativo de que uma issue fechada será reaberta com base no sentimento, pontuação negativa e pontuação positiva do último comentário antes do primeiro fechamento da issue?

QP2.6. É possível prever se uma issue será reaberta através dos sentimentos presentes nas suas discussões entre os eventos de abertura (open) e o primeiro fechamento (closed)?

QP2.7. Existe uma correlação forte entre os sentimentos das discussões das issues com a probabilidade de reabertura?

QP3. Quais categorias estão mais associadas à reabertura de issues?

QP3.1. Qual é a melhor técnica de subamostragem de dados combinada com a técnica de aprendizado de máquina para a categorização de issues?

QP3.2. O tempo de duração entre a abertura e o primeiro fechamento de uma issue é um indicativo de reabertura nas diferentes categorias?

QP3.3. A quantidade de comentários entre a abertura e o primeiro fechamento de uma issue pode influenciar sua probabilidade de reabertura nas diferentes categorias?

QP4. Como caracterizar os sentimentos nas diferentes categorias no contexto de reaberturas de issues?

QP4.1. Como o sentimento, a pontuação negativa e a pontuação positiva do primeiro comentário após a abertura da issue influenciam a reabertura nas diferentes categorias?

QP4.2. Como o sentimento, a pontuação negativa e a pontuação positiva do último comentário antes do primeiro fechamento da issue influenciam a reabertura nas diferentes categorias?

QP4.3. Como os sentimentos, pontuações negativas e positivas entre a abertura e o primeiro fechamento da issue influenciam a reabertura nas diferentes categorias?

1.4 METODOLOGIA DE PESQUISA

A metodologia de pesquisa da tese está organizada em cinco partes: Parte I - Background, Parte II - Escolha da ferramenta de Análise de Sentimentos, Parte III - Análise de Sentimentos, Parte IV - Categorização de Issues e Parte V - Análise de Sentimentos x Categorização de Issues. A sequência das etapas descritas está ilustrada na figura 1.2.

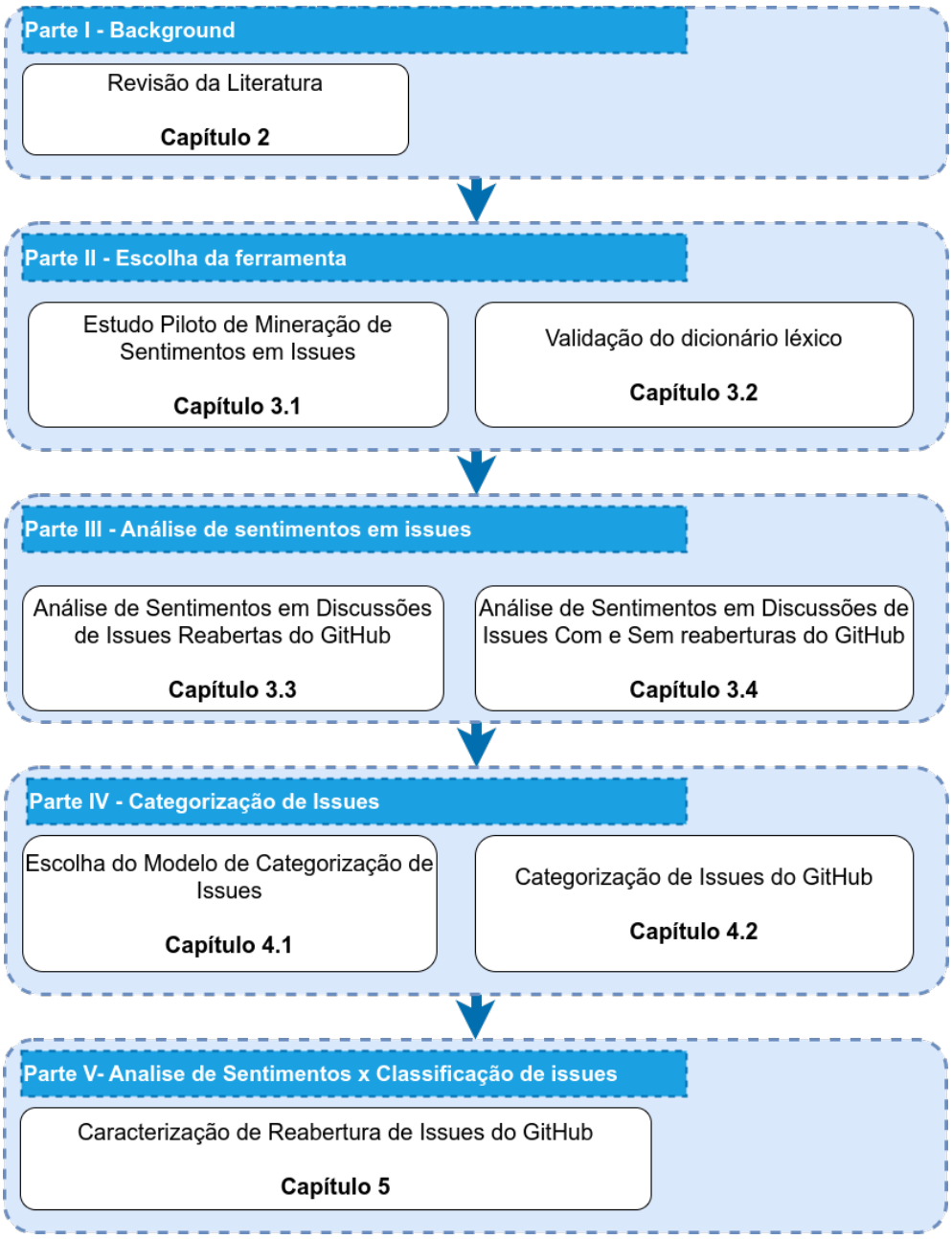


Figura 1.2: Etapas da metodologia.

Cada etapa foi cuidadosamente estruturada para garantir a consistência e a precisão dos resultados obtidos. A seguir, são detalhadas as etapas que compõem a metodologia.

Parte I - Background Esta parte apresenta uma revisão da literatura e mapeamento sistemático sobre análise de sentimentos

- **Revisão da literatura (Capítulo 2)** Apresenta os conceitos básicos sobre mineração de repositórios, ITS, análise de sentimentos, categorização de issues e também apresenta os trabalhos relacionados ao problema de reabertura de issues, aplicação da análise de sentimentos no contexto de mineração de repositórios e ferramentas de análise de sentimentos na área de engenharia de software

Parte II - Escolha da ferramenta de Análise de Sentimentos Nessa parte realizamos uma classificação manual, analisamos as três ferramentas de análise de sentimentos e validamos o léxico da ferramenta *SentiStrength-SE*.

- **Estudo Piloto (Seção 3.1)** Realizamos um estudo piloto de mineração de sentimento em issues, onde foi realizada uma classificação manual de sentimentos das discussões de issues do projeto GeoIP2 Java API do repositório maxmind/GeoIP2-Java[1] no GitHub com aproximadamente 8.1 mil Linhas de código (LOC), 202 forks, 61 watch, 718 star, 35 contribuidores, e usado por 3.6 mil usuários do GitHub, com 113 issues abertas entre 14/08/2013 e 27/08/2018, com 89 usuários do GitHub participando das discussões das issues. Em seguida, analisamos três ferramentas sobre análise de sentimento no contexto de engenharia de software, descritas na seção 2.4.3. A análise foi realizada através da comparação entre a classificação manual e as ferramentas *Senti4SD, SentiCR e SentiStrength-SE*
- **Validação do dicionário léxico (Seção 3.2)**

  A análise de sentimentos faz inferência sobre polaridades em palavras que podem representar possíveis sentimentos. A assertividade dessa classificação é importante para a confiabilidade do resultado esperado. Por esta razão, este trabalho busca investigar, validar e construir um dicionário léxico, no contexto de Engenharia de Software, utilizando como base as palavras, *emoticons* e expressões idiomáticas da ferramenta *SentiStrength-SE*.

  Um experimento com 559 questões respondidas por 48 participantes da área de Computação foi realizado para validação da concordância dos termos léxicos do dicionário. Ao final da coleta dos dados os termos foram reunidos para validação utilizando base de dados *gold standard* [2] composta por posts com perguntas, respostas e comentários do Stack Overflow[3] (CALEFATO; LANUBILE; NOVIELLI, 2018) para encontrar os resultados sobre *accuracy*, *precision*, *recall* e F_1-*score* do novo dicionário.

[1] <https://github.com/maxmind/GeoIP2-java>, atualizado em 09/08/2023
[2] <https://github.com/collab-uniba/Senti4SD/tree/master/Senti4SD_GoldStandard_and_DSM>
[3] <https://stackoverflow.com/>, acessado em 20/02/2024

Parte III - Análise de Sentimentos Nesta parte apresentamos dois estudos sobre reaberturas de issues do GitHub e análise de sentimentos.

- **Análise de Sentimentos em Issues Reabertas do GitHub (Seção 3.3)**

  Analisamos os sentimentos das discussões de issues reabertas dos repositórios listados no *MSR 2014 Mining Challenge* (MSR, 2014) , a lista possui os principais repositórios de acordo com o número de star das principais linguagens de programação do Github. Utilizamos a ferramenta SentiStrength-SE com dicionário léxico voltado para o domínio de Engenharia de Software(ISLAM; ZIBRAN, 2017).

- **Análise de Sentimentos em Issues Com e Sem Reabertas do GitHub (Seção 3.4)**

  Atualização do estudo sobre análise de sentimentos das discussões de issues reabertas dos repositórios listados no MSR 2014 Mining Challenge (MSR, 2014).

  Realizamos uma nova coleta de dados das issues no GitHub, onde acrescentamos os repositórios que migraram para novos repositórios e excluimos os repositórios que possuem menos de cinco issues reabertas. Novas análises são realizadas durante as trocas de comentários entre as mudanças de eventos de abertura e o primeiro fechamento (*Open-Closed*).

Parte IV - Categorização de Issues Nesta parte descrevemos os estudos sobre categorização de issues GitHub.

- **Categorização de Issues (Seção 4.1)** Nesta seção modelamos um classificador automático utilizando técnicas de aprendizado de máquina para categorizar issues do GitHub. As issues pertencem as seguintes categorias: Banco de dados, Configuração, Desempenho, Funcional, GUI, Info, Permissão/Obsoleto, Redes, Segurança e Testes (CATOLINO et al., 2019). Utilizamos 1280 issues de projetos dos ecossistemas como Mozilla, Apache e Eclipse para a construção do categorizador. Para melhorar os resultados do categorizador de issues foi utilizado o balanceamento dos dados com a combinação de subamostragem e sobreamostragem de dados.

- **Categorização de issues com e sem reaberturas do GitHub (Seção 4.2)**

  Nesse estudo analisamos os grupos de issues, agrupadas de acordo com a categoria de issues.

  Primeiro foi realizado categorização das issues da base de dados MSR14. Em seguida foi realizada uma análise estatística média central das issues com e sem reabertura de acordo com cada categoria.

  Para melhorar os resultados das análises foi feita a remoção dos outliers utilizando a medida estátistica *Interquartile Range* (IQR).

Parte V - Análise de Sentimentos x Categorização de Issues Nesta parte caracterizamos as reaberturas de issues de acordo com os sentimentos pelo categoria de issues.

- **Caracterização de Reabertura de Issues do GitHub (Seção 5)** Nessa seção investigamos a análise de sentimentos nos textos contidos em discussões de uma issue e sua categoria como um meio de prever a reabertura dessas issues em repositórios de software do GitHub. Utilizamos a base de dados MSR2014 atualizada com a categorização das issue, pontuação negativa, pontuação positiva e o sentimento do título, da descrição e de cada comentário entre a abertura e fechamento da issue.

1.5 OBSERVAÇÕES FINAIS

Neste capítulo apresentamos nossa motivação, objetivos, questões de pesquisa e metodologia de pesquisa. No próximo capítulo descrevemos o referencial teórico sobre reabertura de issues, categorização de issues, análise de sentimentos e trabalhos relacionados.

Capítulo

2

REVISÃO DA LITERATURA

Neste capítulo, proporcionaremos uma breve introdução à Mineração de Repositórios de Software, aos Sistemas de Rastreamento de Issues (ITS) e à Análise de Sentimentos. Este capítulo está dividido em duas seções principais:

A Seção 2.1 aborda o conceito de mineração de repositórios de software e os Sistemas de Rastreamento de Issues (ITS). Em seguida, na Seção 2.2, abordamos o problema associado à reabertura de issues. A categorização de issues é discutida na Seção 2.3, e posteriormente, a Seção 2.4 apresenta os conceitos básicos sobre Análise de Sentimentos, sua aplicação na área de mineração de repositório de software e as ferramentas desenvolvidas especificamente para o contexto da engenharia de software.

2.1 MINERAÇÃO DE REPOSITÓRIOS DE SOFTWARE

A Mineração de Dados refere-se ao processo de descoberta de conhecimento em bases de dados, conforme destacado por Fayyad, Piatetsky-Shapiro e Smyth (1996). Esse processo consiste na extração automatizada de informações úteis e previamente desconhecidas de repositórios de dados construídos para propósitos distintos da mineração de dados (MENDONCA, 2001). É importante ressaltar que uma informação é considerada útil quando possui valor para o usuário. No entanto, para agregar conhecimento, ela deve ser simultaneamente útil e nova, ou seja, fornecer utilidade e ser previamente desconhecida.

A prática da mineração de dados não se resume apenas à aplicação de algoritmos de aprendizado. Ela também engloba a fase de extração, pré-processamento e transformação dos dados. Essas etapas desempenham um papel fundamental no asseguramento do desempenho eficaz dos algoritmos de aprendizado, como enfatizado por Han, Kamber e Pei (2011).

Por sua vez, o termo Mineração de Repositórios de Software (do inglês Mining Software Repositories (MSR)) representa uma disciplina na engenharia de software com o objetivo específico de descobrir conhecimento em repositórios de software. Nesse contexto, os dados estão armazenados em diversos tipos de repositórios utilizados na engenharia de

software, como sistemas de controle de versão (*Version Control Systems* (VCS)), sistemas de rastreamento de issues (*Issue Tracking Systems* (ITS)), listas de discussão, listas de e-mails, código fonte, entre outros. A premissa fundamental da mineração de dados é que a informação extraída desses repositórios deve ser simultaneamente nova e útil para os processos de engenharia de software MSR (2019).

O propósito das técnicas de MSR é capacitar os gerentes a tomar decisões com base no conhecimento adquirido a partir de dados concretos, reduzindo assim a dependência da intuição e experiência pessoal (Hassan, 2008; FARIAS et al., 2016; MSR, 2019).

Os repositórios de software contêm uma grande quantidade de informações sobre os projetos de software, sua gestão, seus processos e seus produtos. De acordo com Hassan (2008), esses repositórios de software pode ser categorizados em:

- ***Repositórios de Código-Fonte***: Esses repositórios incluem código-fonte, documentação e arquivos de configuração. A mineração desses repositórios abrange diversas tarefas, como análise de código-fonte, detecção de padrões de codificação, identificação de bugs e métricas de qualidade. Exemplos: SourceForge [1], Bitbucket [2].

- ***Repositórios Históricos***: Esses repositórios registram informações sobre a evolução e o progresso de um projeto. Eles englobam repositórios de controle de versão (VCS), repositórios de rastreamento de issues (ITS), repositórios de rastreamento de bugs (*Bug Tracking Systems* (BTS)), repositórios de *releases* e comunicações arquivadas entre a equipe do projeto, como listas de e-mails, fóruns de discussão e sites de perguntas e respostas.

  A mineração desses repositórios históricos permite identificar padrões de desenvolvimento, detecção de ramificações e fusões, e identificação dos principais autores de código, entre outros. Exemplos notáveis incluem Subversion [3] e Git [4].

- ***Repositórios de tempo de execução ou runtime*** : Esses repositórios contêm informações sobre a execução e o uso de um software em produção, como logs de operação, registros de eventos, informações sobre erros ou bugs, métricas de desempenho e outros dados relacionados à operação do software em tempo real. A mineração desses repositórios de tempo de execução permite examinar registros de eventos para identificar problemas em tempo real, monitorar o desempenho do software e tomar decisões automáticas com base em eventos específicos. Exemplos notáveis incluem Elasticsearch [5] e Splunk [6].

A Mineração de Repositórios de Software desempenha um papel essencial no apoio aos gestores de projetos de software para compreender os intricados processos de desen-

[1] <https://sourceforge.net/>
[2] <https://bitbucket.org/>
[3] <https://subversion.apache.org/>
[4] <https://git-scm.com/>
[5] <https://www.elastic.co/elasticsearch/>
[6] <https://www.splunk.com/>

volvimento, manutenção e evolução do software, assim como as dinâmicas entre desenvolvedores e usuários. Além disso, ela proporciona insights valiosos sobre o comportamento do software em tempo de execução.

Seu impacto se estende ainda mais ao ser uma ferramenta valiosa no aprimoramento do design e da arquitetura do software. A Mineração de Repositórios é crucial para promover a reutilização eficaz de código, validar empiricamente novas ideias e técnicas, e oferecer suporte sólido para previsões precisas sobre o desenvolvimento futuro do software (ROBBES; HILL; BIRD, 2018; MSR, 2021).

2.1.1 Sistemas de Rastreamento de Issues

Um Sistema de Rastreamento de Issues (ITS) possibilita que equipes de desenvolvimento de software e seus gerentes relatem e acompanhem issues nos repositórios de software, independentemente da localização (KSHIRSAGAR; CHANDRE, 2015). Essas issues podem abranger desde a correção de bugs, tarefas de refatoração, dívida técnica até requisições de novas funcionalidades ou melhorias do sistema (GIGER; PINZGER; GALL, 2010; RAATIKAINEN et al., 2023).

As issues são compostas por diversos campos de dados, incluindo código de identificação (ID), título, descrição, status (por exemplo, aberta, fechada ou corrigida), prioridade, comentários e datas de envio e alteração. Além disso, a issue mantêm um histórico detalhado de cada alteração de status, atribuições e comentários postados (T.MERTEN et al., 2015; ORTU et al., 2015b; MERTEN et al., 2016). Vale ressaltar que os textos nas issues e seus comentários não se limitam a informações técnicas sobre o software; eles também contêm os sentimentos e emoções dos desenvolvedores de software (ORTU et al., 2015b).

Todas as informações contidas nas issues, incluindo textos e comentários, podem ser recuperadas por meio de APIs, como a PyGithub (WAGNER; FERNáNDEZ, 2015; JACQUES, 2020). Exemplos de ITS incluem Atlassian Jira[7], Bugzilla[8], Redmine[9], Mantis[10], GitLab[11], Bitbucket [12], e GitHub[13]. A seguir, discutiremos mais detalhadamente o GitHub, uma das plataformas colaborativas ITS mais populares da atualidade.

2.1.2 GitHub

O GitHub é uma plataforma de hospedagem de código aberto que possibilita o controle de versão colaborativo em projetos de software, permitindo que usuários trabalhem juntos de qualquer lugar. Com uma comunidade robusta, o GitHub conta com aproximadamente 65 milhões de desenvolvedores, 200 milhões de repositórios, 20 milhões de issues fechadas e 87 milhões de pull requests (OCTOVERSE-GITHUB, 2023). Para o rastreamento e

[7] <https://www.atlassian.com/software/jira>
[8] <https://www.bugzilla.org/>
[9] <https://www.redmine.org/>
[10] <https://www.mantisbt.org/>
[11] <https://about.gitlab.com/>
[12] <https://bitbucket.org/>
[13] <https://github.com/>

gerenciamento eficaz de código-fonte ao longo do tempo, o GitHub utiliza o Sistema de Controle de Versão Distribuída *Distributed Version Control System* (DVCS) - Git [14], conhecido por sua flexibilidade, segurança e eficiência (CHACON, 2014).

Os repositórios hospedados no GitHub cobrem uma ampla gama de tecnologias, com exemplos em diversas áreas, como linguagens de programação (C, C++, C#, Java, JavaScript, Kotlin, PHP, Python, Ruby, Shell, TypeScript), sistemas operacionais (Android, iOS, Linux, macOS, Ubuntu, Windows), sistemas de gerenciamento de banco de dados (MySQL, MongoDB, PostgreSQL), navegadores web (Chrome, Firefox), gerenciadores de pacotes (Homebrew, Npm), **APIs!** (**APIs!**)(GitHub API, OpenGL), criptomoedas (Bitcoin, Monero), sistemas de controle de versão (Git, Mercurial) e outros tópicos (GITHUB-TOPICS, 2020). A figura 2.1 ilustra a página do repositório do projeto *nodejs/node* no GitHub.

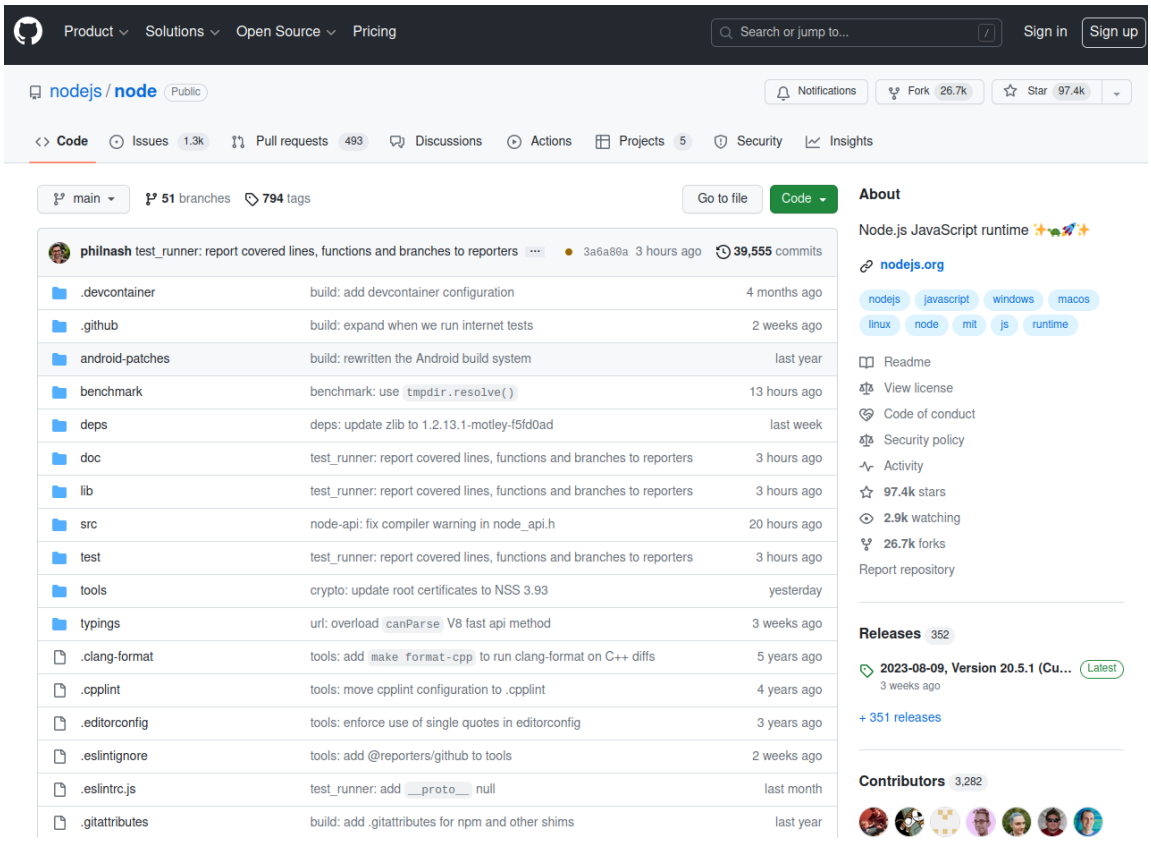


Figura 2.1: Repositório do projeto nodejs/node do GitHub (Acessado em 30/08/2023).

Os usuários do GitHub podem interagir com os repositórios de várias maneiras. Eles podem marcar um repositório com uma estrela, equivalente a salvá-lo como favorito, assistir a um repositório para receber notificações de novas atividades, criar e comentar em issues ou pull requests, criar um fork de um repositório para fazer uma cópia pessoal e realizar alterações sem afetar o repositório original. Cada usuário também tem uma

[14] <http://git-scm.com>

página de perfil personalizada que exibe informações como tipo de usuário, e-mail, URL, localização e empresa, além de mostrar seus repositórios e atividades de contribuição.

Para promover a colaboração efetiva, o GitHub oferece funcionalidades como *branches*, *commits*, discussões da equipe, *pull requests*, issues e repositórios (GITHUB_DOCS, 2023), como descrito a seguir:

- **Repositório** um repositório no GitHub é usado para armazenar projetos de software, ideias ou recursos que podem ser compartilhados.Ele contém arquivos de diversos tipos, como código-fonte, documentos, planilhas, imagens, além de arquivos importantes como a licença e o README, que descreve o projeto. , e esses podem estar organizados em diretórios;

- **Branches:** As *branches* permitem trabalhar com diferentes versões de um repositório simultaneamente. O repositório, por padrão, contém uma *branch* principal,chamada *main*. As *branches* são usadas para experimentar e realizar edições antes de enviá-las à *branch main*;

- **Commits:** Os *commits* representam as alterações salvas em um repositório. Cada *commit* inclui uma mensagem descritiva que explica o que foi alterado e por que a mudança foi feita. As descrições dos commits ajudam a capturar o histórico das alterações realizadas no repositório, para que outros desenvolvedores possam entender o que foi feito e porquê;

- **Discussões da Equipe:** As discussões da equipe são usadas para melhorar a comunicação entre a equipe do projeto de software. Uma discussão pode ser criada por um desenvolvedor de software para discutir uma ideia com toda a equipe, ao invés de abrir uma issue ou *pull request*;

- **Issues:** As issues correspondem às sugestões de melhorias, tarefas ou questões relacionadas ao repositório. As issues podem ser criadas, moderadas e atribuídas aos colaboradores do repositório. Toda issue possui um título, uma descrição, a data e hora de criação, labels ou rótulo, reações, comentários, eventos e usuários ou colaboradores do repositório;

- **Pull Request:** Um *Pull Request* é um tipo especial de issue no GitHub, usado para discutir e revisar um conjunto de alterações propostas de uma *branch* para outra. Essas mudanças podem ser futuramente mescladas na *branch main* do projeto, após revisão e aprovação colaborativa. Embora os *Pull Requests* compartilhem características com as issues, como comentários e discussões, eles são focados na integração de alterações de código.

As issues no GitHub oferecem recursos adicionais para uma melhor organização e comunicação. Elas podem ser categorizadas por *labels*, que podem ser padrões do GitHub ou personalizadas pelos colaboradores do repositório. Além disso, os usuários têm a opção de adicionar reações às discussões, permitindo expressar sentimentos de maneira simples

e eficaz, como *+1 like, -1 dislike, laugh, confused, heart, rocket, laugh, eyes, or hooray* (BORGES; BRITO; VALENTE, 2019).

Cada issue também registra uma série de eventos, correspondendo às ações executadas nela. Esses eventos incluem abertura, fechamento, reabertura, atribuição, desatribuição, edição, exclusão, fixação, desafixação, adição de rótulos, remoção de rótulos, bloqueio, desbloqueio, associação a uma *milestone*, desassociação de *milestone* e transferência (GITHUB_INC., 2023c). Esses recursos proporcionam uma rastreabilidade abrangente das atividades relacionadas a cada issue.

2.2 REABERTURA DE ISSUES

Nosso trabalho se concentra na reabertura de issues, um fenômeno que ocorre quando há a detecção de erro, inconsistência ou incompletude no trabalho associado a uma issue previamente fechada. As issues reabertas podem causar desafios no gerenciamento do esforço de manutenção de software, aumentando a carga de trabalho dos desenvolvedores e atrasando a entrega do software (XIA et al., 2015). Para abordar essas issues, a equipe de desenvolvimento deve tomar medidas proativas para reduzir a probabilidade de reabertura, analisando as possíveis causas por trás desse fenômeno (CAGLAYAN et al., 2012).

A reabertura de uma issue no GitHub ocorre por meio do evento de reabertura (*reopened*), que transforma seu status de fechada (*closed*) para aberta (*open*) (ANVIK; HIEW; MURPHY, 2006). A Figura 2.2 apresenta um exemplo de reabertura da issue, observando a issue de ID 1491 do repositório microsoft/vscode[15]. Nesse exemplo, o colaborador *bpasero* inicialmente fecha a issue, posteriormente a reabre, e são incluídos comentários relevantes ao longo desse processo. O comentário feito por *iamthemovie* ocorre entre o fechamento e a reabertura, proporcionando contexto adicional ao histórico da issue.

[15] <https://github.com/microsoft/vscode/issues/1491>

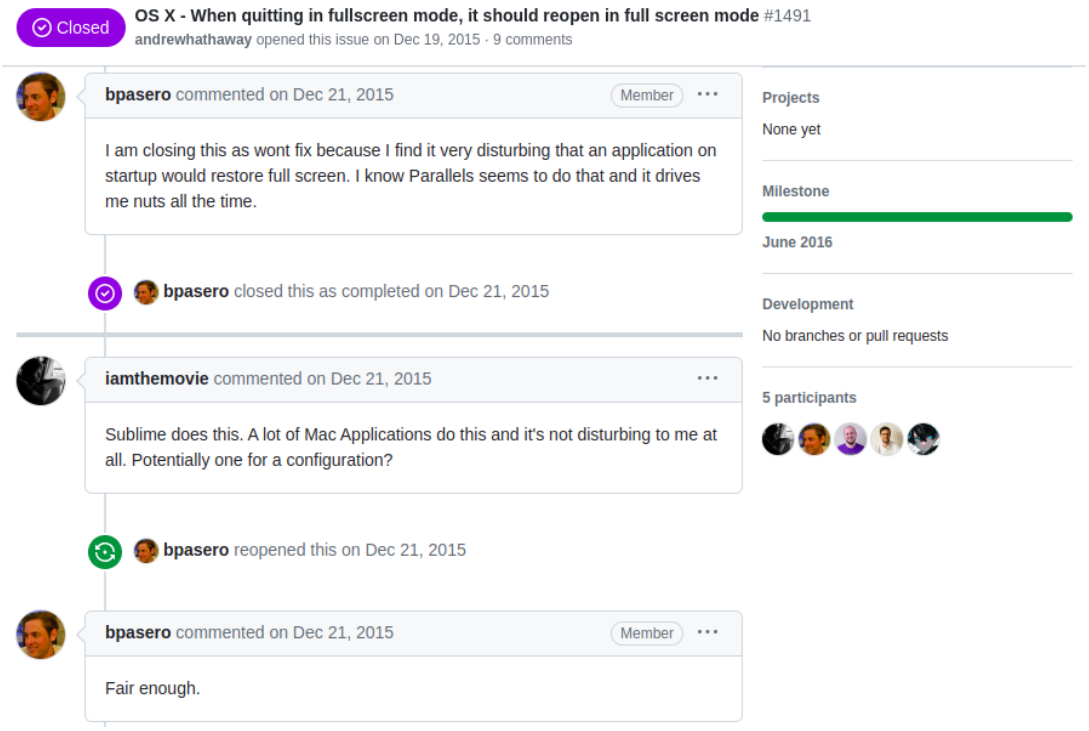


Figura 2.2: Exemplo de relatório de issue com reabertura do repositório microsoft/vscode com ID 1491 (Acessado em 10/09/2023)

Essa capacidade de reabertura reflete a natureza dinâmica do desenvolvimento colaborativo no GitHub, permitindo que as issues sejam revisitadas e reavaliadas conforme necessário ao longo do ciclo de vida do projeto. Essa flexibilidade é valiosa para lidar com problemas persistentes, ajustar prioridades ou incorporar novos insights ao processo de desenvolvimento de software.

As reaberturas de issues relacionadas a bugs podem ser categorizadas em grupos, conforme identificado por Zimmermann et al. (2012). Esses grupos e suas categorias associadas estão resumidos na Tabela 2.1.

Tabela 2.1: Causas de reabertura de bug (ZIMMERMANN et al., 2012).

Grupo	**Categoria**
Not fixed	bugs difíceis de serem reproduzidos
	desenvolvedores não compreendem a causa principal do bug
	relatório bug com informações insuficientes
	a prioridade do bug aumentou
Fixed	bugs de regressão
Process-related	processos

2.2.1 Análise e predição de reabertura de issues

Caglayan et al. (2012) conduziram um estudo de caso com o objetivo de caracterizar os possíveis fatores que podem levar à reabertura de issues. Nesse estudo, uma issue

na ferramenta *IBM Rational ClearQuest*[16] foi considerada reaberta quando mudou de um estado finalizado, como *closed*, *cancelled* ou *postponed*, para um estado ativo, como *assigned* ou *work-in-progress*. A análise incluiu 3.645 relatórios de issues, com o registro mais antigo datando de janeiro de 2005. Os autores identificaram, por meio de um modelo de regressão logística baseado nos dados dos relatórios de issues, que a rede de proximidade das issues e as atividades dos desenvolvedores foram os fatores mais importantes para a reabertura das issues.

Um estudo exploratório foi realizado por Shihab et al. (2013) para investigar os fatores que causam a reabertura de issues relacionadas a bugs. Foram analisados os hábitos de trabalho, *bug report*, *bug fix* e time de colaboradores do Projeto Eclipse. Para realizar a análise, foi construído um modelo de predição de reabertura de issues relacionadas a bugs baseado em árvores de decisão (algoritmo C4.5). Os autores utilizaram a ferramenta *J-REX* para extrair um *snapshot* do repositório *Eclipse CVS*. Foram analisados 1.530 *bug reports* de um total de 18.312 *bug reports* dos projetos *Eclipse Platform 3.0, Apache HTTP Server e OpenOffice*, sendo 246 deles reabertos e 1.284 não. Os principais fatores identificados como causadores da reabertura de issues relacionadas a bugs no *Eclipse* foram a descrição da issue, os textos dos comentários, o tempo para solucionar o bug e o componente onde o bug foi encontrado.

Zimmermann et al. (2012) categorizaram as principais razões que levam as issues relacionadas a defeitos (bugs) a serem reabertas no contexto de duas versões do sistema operacional *Microsoft Windows, Windows Vista e Windows 7*. Eles utilizaram um *survey* realizado com 358 colaboradores da Microsoft e uma análise quantitativa de alto nível de toda a base de dados de bugs do Windows. Esses dados foram usados para construir um modelo de regressão logística que identificou os fatores causadores de reabertura de issues, considerando fatores extraídos dos relatórios de issues e fatores relacionados ao processo e à organização, incluindo a localização e hábitos de trabalho dos desenvolvedores.

Souza, Chavez e Bittencourt (2015) analisaram o projeto *Firefox* para compreender as relações entre as diferentes formas de rejeição de *patches*. Os autores analisaram 82.920 issues do projeto *Firefox*, criados entre 1997 e 2013. Os atributos extraídos das issues foram: número da issue, data de criação, mudanças no status da issue, resolução e sinalizadores, juntamente com as datas de todas as alterações. Para realizar a caracterização da rejeição de *patch*, foi detectado que para cada issue do conjunto de dados existiam três tipos de eventos de rejeição: revisão negativa de um *patch* que foi submetido à revisão por pares; *backout* de um *patch* que foi confirmado (posteriormente dividido em *backout* precoce e tardio); e a reabertura de um relatório de assunto encerrado. Cerca de 5,7% de todas as issues resolvidas do *Firefox* foram reabertas, induzindo uma sobrecarga de discussões entre os colaboradores sobre a reabertura de uma issue. Também foi identificado que 70% das issues fechadas foram reabertas prematuramente devido a equívocos de interpretação dos colaboradores.

No estudo conduzido por Xia et al. (2013), foi realizada uma avaliação da eficácia de diversos algoritmos de aprendizado máquina supervisionado para prever se um relatório de bug seria reaberto. Sete algoritmos de aprendizado de máquina supervisionada

[16] <https://www.ibm.com/products/rational-clearquest>

foram selecionados, incluindo kNN, SVM, regressão logística, Rede Bayesiana, Decision Table, CART e LWL, juntamente com três métodos de classificadores ensemble: AdaBoost, Bagging e Random Forest. Os autores utilizaram bases de dados (Shihab et al., 2010; SHIHAB et al., 2013) com 246 *bug reports*, dos quais foram reabertos e 1.284 não foram reabertos. Os algoritmos que obtiveram os melhores desempenhos foram Bagging e Decision Table (IDTM) com precisão de 92,91% e 92,80%, e F1 de 0,735 e 0,732, respectivamente. A tabela 2.2 detalha as características extraídas dos relatórios de bug, fornecendo uma visão abrangente dos atributos considerados na análise dos algoritmos.

Tabela 2.2: Características extraidas de relatórios de bug reabertos (XIA et al., 2013)

Características	**Descrição**
Time	Momento em que o bug é fechado (hora, dia da semana, mês)
BF	Tempo para resolver o bug, último status do bug, Número de arquivos de código-fonte relacionados ao bug
Human	Nome do relator do bug (reporter), nome do resolvedor do bug (fixer), número de bug reports reportado pelo relator do bug e o número de bug reports resolvidos pelo resolvedor
BR	Componente afetado pelo bug, plataforma afetada pelo bug, Gravidade do bug report, Prioridade do bug report, Número de palavras no texto da descrição, Conteúdo do texto da descrição, Número de comentários, Número de palavras no texto do comentário, Conteúdo do texto do comentário e Se a prioridade foi alterada

Mohamed et al. (2018) propuseram o método DTPre para predição automática da reabertura de *pull requests* baseado em um classificador de árvore de decisão. O objetivo é prever a reabertura de *pull requests* imediatamente após o seu fechamento. A avaliação do preditor foi realizada usando uma base de dados de sete projetos Open Source do GitHub com 100.622 pull requests. Os resultados dos testes apresentaram uma precisão de 95,53%, recall de 99,01% e F1 de 97,23% quando utilizaram a técnica de *oversampling* para balancear os dados. Os autores identificaram que a precisão foi melhor quando usaram apenas *Code features*, a taxa de recall e F1 foram melhores quando utilizaram as três características em conjunto. A tabela 2.3 oferece uma visão abrangente das possíveis causas que podem levar à reabertura do *pull requests*.

Tabela 2.3: Indicadores de possível reabertura de *issue* (MOHAMED et al., 2018)

Classe	**Característica**	**Descrição**
Code features	Número de commits	Número de commits em um pull request no primeiro fechamento.
	Número de arquivos alterados	Número de arquivos alterados, incluindo arquivos adicionados e arquivos excluídos
	Número de linhas adicionadas	Número de linhas para códigos adicionados em um pull request

Continua na próxima página

Tabela 2.3 – Indicadores de possível reabertura de *issue* Cont.

Classe	**Característica**	**Descrição**
	Número de linhas deletadas	Número de linhas para códigos excluídos em um pull request
Review features	Número de comentários	Número de comentários em um pull request
	Evaluation time	Diferença de tempo entre a submissão de um pull request até o primeiro close.
	Closed status	Se o pull request é accepted ou rejected no primeiro close.
Developer feature	Reputação	proporção de pull request anteriores que são enviadas pelo contribuidor e são aceitas.

O estudo conduzido por Cheruvelil e Silva (2019) investigou o impacto do sentimento dos desenvolvedores em relação à reabertura de issues. Utilizando uma amostra de 3.000 issues provenientes de oito projetos open-source hospedados no *Apache* no *Jira*, os pesquisadores extraíram informações como o nome da issue, data de criação, histórico de transição e todos os comentários associados. A análise incluiu o cálculo das pontuações dos sentimentos negativos e positivos com o auxílio da ferramenta SentiStrength-SE. Observando as variações nas pontuações de sentimentos entre as issues sem reabertura, com uma reabertura e com várias reaberturas, os autores encontraram evidências de correlação entre o sentimento e a reabertura das issues, embora o tamanho do efeito pareça ser relativamente pequeno.

Para aprofundar a compreensão da relação entre a reabertura da issue e o sentimento dos desenvolvedores, os pesquisadores aplicaram o teste chi-quadrado de Pearson (x). Os resultados indicaram um valor de chi-quadrado próximo de 0,0, sugerindo uma diferença estatisticamente significativa nas proporções. Esse achado reforça a ideia de que o sentimento dos desenvolvedores desempenha um papel na dinâmica de reabertura de issues, embora a magnitude desse impacto possa ser considerada relativamente modesta.

2.3 CATEGORIZAÇÃO DE ISSUES

No contexto do gerenciamento de projetos de desenvolvimento de software, a categorização apropriada de issues desempenha um papel fundamental. Essa prática envolve o agrupamento de issues em diferentes classes ou categorias, levando em consideração suas características específicas e áreas de foco (TAN et al., 2014). A categorização não apenas facilita a organização das tarefas, mas também simplifica a alocação precisa de recursos de desenvolvimento.

Uma das primeiras e abrangentes classificações de issues relacionadas a tipos de defeitos foi apresentada por Chillarege et al. (1992), denominada Classificação Ortogonal de Defeitos (Orthogonal Defect Classification (ODC)). Essa classificação compreende oito categorias que oferecem uma visão abrangente dos tipos de problemas encontrados

no desenvolvimento de software, permitindo que profissionais identifiquem padrões de defeitos e implementem medidas para aprimorar a qualidade do software. As categorias são as seguintes:

- **Funcionalidade (*function*)** issues com defeitos que afetam recursos significativos, interfaces de usuário final, interfaces de produto, interface com arquitetura de hardware ou estruturas de dados globais. Geralmente, exigem uma alteração formal de design.
- **Atribuição (*assignment*)** refere-se a issue com erros em linhas de código específicas, como inicialização de blocos de controle ou estruturas de dados.
- ***Interface*** correspondem a issue com falhas na interação com outros componentes, módulos ou drivers de dispositivo por meio de macros, instruções de chamada, blocos de controle ou listas de parâmetros.
- **Verificação (*Checking*)** envolvem issue com a lógica do programa que falha ao validar corretamente dados e valores antes de seu uso.
- **Temporização/serialização (*Timing/Serilization*)** issue com erros corrigidos pelo aprimoramento do gerenciamento de recursos compartilhados e em tempo real.
- ***Build/package/merge*** descrevem issue com erros decorrentes de problemas em sistemas de biblioteca, gerenciamento de alterações ou controle de versão.
- **Documentação (*Documentation*)** envolve issue com erros de documentação que pode afetar tanto as publicações quanto as notas de manutenção.
- **Algoritmo (*algorithm*)** incluem issues com problemas de eficiência ou correção que afetam uma tarefa e podem ser corrigidos pela (re)implementação de um algoritmo ou estrutura de dados local sem a necessidade de solicitar uma mudança de design.

O GitHub oferece uma categorização inicial para issues na forma de rótulos ou etiquetas (no inglês *labels*). Esses rótulos são utilizados para categorizar, classificar ou identificar issues e *pull requests* em um repositório. No GitHub, é possível criar novos rótulos, editá-los ou excluí-los posteriormente (GITHUB_INC., 2023b). A seguinte categorização está disponível para novos repositórios:

- ***Bug*** utilizado para indicar problemas ou defeitos inesperados que precisam ser corrigidos.
- **Documentação (*documentation*)** usado para categorizar issue ou pull request relacionada com documentação do projeto.
- **Duplicada (*duplicate*)** significa que a issue, pull requests ou discussões é uma duplicata de outra issue realatada anteriormente no repositório.

- **Melhoria ou Aprimoramento (*enhancement*)** indica issues ou pull requests relacionadas a solicitações de melhorias ou otimizações nas funcionalidades existentes no projeto.

- **Boa primeira issue (*good first issue*)** comumente utilizada para indicar tarefas ou problemas adequados para novos colaboradores do projeto

- **Preciso de Ajuda (*help wanted*)** usado para indicar issues ou pull requst para os quais os colaboradores do projeto estão solicitando ajuda.

- **Inválida (*invalid*)** utilizado quando uma issue, pull request ou uma discussão não é mais relevante ou não está relacionado ao projeto em questão.

- **Pergunta (*question*)**indica que issue, pull request ou uma discussão se refere a uma pergunta ou dúvida.

- **Não resolvido (*wontfix*)** indica que a issue, pull request ou discussão não será tratada.

No estudo conduzido por Catolino et al. (2019), foi proposta uma taxonomia para os tipos de relatórios de bugs, desenvolvida com base em uma análise detalhada de 1280 issues provenientes de 119 projetos dos ecossistemas *Mozilla*, *Apache* e *Eclipse*. A taxonomia compreende diferentes categorias de issues comuns levantadas por meio de relatórios de issues e foi construída manualmente a partir um conjunto grande e diversificado de issues. As principais categorias são as seguintes:

- ***Anomalias de código (Program anomaly issue)*** categoria de issues relacionadas a bugs introduzidos pelos desenvolvedores ao aprimorar o código-fonte existente, envolvendo circunstâncias específicas como exceções, problemas com valores de retorno e falhas inesperadas devido a problemas na lógica do programa.

- ***Configuração (Configuration issue)*** categoria de issues relacionadas à construção de arquivos de configuração, incluindo problemas com bibliotecas externas que precisam ser atualizadas ou corrigidas, nome de diretórios ou caminhos de arquivo errados.

- ***Banco de dados (Database-related issue)*** categoria que relata problemas de consultas, conexão com o banco de dados, conflitos de concorrência, falta de *backup* e recuperação, bem como desempenho do banco de dados.

- ***Desempenho (Performance issue.)*** categoria de issues que reportam o desempenho do software, incluindo o uso excessivo de memória, vazamentos de energia e métodos que causam loops infinitos.

- ***Funcional (Functional Issue)*** categoria de issues referentes à adicição de novos recursos ou aprimoramento dos existentes.

- ***GUI (GUI-related issue)*** categoria de issues referentes à Interface Gráfica do Usuário ou (GUI), como layouts de tela, cores e preenchimento dos elementos, aparência da caixa de texto e botões, bem como falhas inesperadas que aparecem ao usuário final na forma de mensagens de erro incomuns.
- ***Informação (Info)*** categoria de issues referentes à informação.
- ***Permissão/Obsoleto (Permission/deprecation issue.)*** issues relacionadas à presença, modificação ou remoção de chamadas de API ou métodos obsoletos, ou reportam permissões de APIs não utilizadas.
- ***Redes (Network issue)*** categoria de issues relacionadas à problemas de conectividade ou servidor, como configurações incorretas, falhas no servidor, protocolos de comunicação que não são usados corretamente no código-fonte.
- ***Segurança (Security issue)*** issues relacionadas à segurança ou vulnerabilidade de informação, como a remoção de permissões não utilizadas que podem comprometer a confiabilidade geral do sistema.
- ***Testes (Test code-related issue)*** issues relacionadas a testes de software, abrangendo problemas de execução, correção ou atualização de casos de teste.

2.3.1 Classificadores automáticos de issues

O estudo conduzido por Antoniol et al. (2008) demonstrou a viabilidade do uso de modelos de aprendizado de máquina para distinguir entre issues de manutenção corretiva (*bugs*) e atividades não corretivas, como solicitações de melhoria, refatorações, alterações na documentação e outros tipos de issues, conhecidas como não-*bugs* (*non-bugs*). A base de dados foi composta através da classificação manual de aproximadamente 1.800 issues extraídas de três grandes projetos de código aberto: *Mozilla, Eclipse e JBoss*, utilizando as plataformas de rastreamento de problemas *Bugzilla*[17] e *Jira*[18]. Durante o pré-processamento do texto, os símbolos de pontuação foram removidos, aplicou-se uma técnica de stemização, e a frequência do termo-inverso da frequência nos documentos (*Term Frequency - Inverse Document Frequency* (TF-IDF)) foi calculada. Foram empregados modelos de aprendizado de máquina, incluindo Árvore de Decisão (*Decision Tree* (DT)), Naïve Bayes (*Naive Bayes* (NB)) e Regressão Logística (*Logistic Regression* (LR)), para a construção de classificadores voltados para as issues. Os resultados obtidos indicaram que essas issues podem ser classificadas com uma taxa de acerto entre 77% e 82%.

O estudo realizado por Pandey et al. (2017) empregou diversas técnicas de aprendizado de máquina, incluindo Naive Bayes (NB), Análise Discriminante Linear (*Linear discriminant analysis* (LDA)), Máquinas de Vetores de Suporte (*Support Vector Machine* (SVM)), Árvores de Decisão (DT), k-Vizinhos Mais Próximos (k-Nearest Neighbor (kNN)) e Florestas Aleatórias (*Random Forest* (RF)), com o objetivo de automatizar a classificação

[17]https://bugzilla.mozilla.org/home
[18]https://www.atlassian.com/software/jira

de relatórios de issues em duas categorias principais: manutenção corretiva (*BUG*) e issues que não requerem manutenção corretiva (*NUG*). A categoria *NUG* abrange todas as categorias, exceto *BUG*.

O corpus [19] utilizado no estudo foi composto por issues provenientes de três projetos de código aberto: *HttpClient, Lucene e Jackrabbit*, disponíveis no ITS Jira [20]. Durante o pré-processamento, o texto das issues foi convertido para minúsculas, seguido pela remoção de pontuação, números e *stop-words*. No entanto, *stop-words* que contêm conotações negativas, como "não", e auxiliares modais, como "deveria", que influenciam o significado do texto, não foram excluídos. Conectivos temporais como "antes", "depois" e "quando", frequentemente utilizados para descrever cenários de teste, também foram mantidos. Em seguida, aplicou-se a técnica de stemização, seguida pela conversão do texto em uma matriz documento-termo (Document Term Matrix (DTM)). Os classificadores RF e SVM alcançaram taxas médias de acurácia entre 75% e 83%, dependendo do projeto avaliado.

O estudo conduzido por Catolino et al. (2019) apresenta classificadores automáticos de issues do tipo bugs, utilizando o conteúdo textual do relatório de bug para prever sua categoria. As categorias propostas pelos autores foram: Configuração, Redes, Banco de dados, GUI, Desempenho, Permissões/Obsoleto, Segurança , Anomalias de código e Testes. Para a classificação automática das issues, foram utilizados os métodos NB, SVM, LR e RF. A avaliação dos classificadores foi realizada utilizando a validação cruzada denominada k-fold, e as métricas utilizadas foram precisão, revocação, F1, Área Sob a Curva da Característica de Operação do Receptor (AUC-ROC) e coeficiente de correlação de Matthew (MCC). Na etapa de pré-processamento, foram realizadas diversas tarefas, incluindo correção ortográfica, expansão de contrações, conversão do texto para minúsculas, remoção de substantivos, verbos, caracteres especiais, palavras-chave de programação e *stopwords*, além de lematização.

Os resultados dos classificadores apresentaram uma média de precisão para cada categoria entre 36% e 90%, revocação (*recall*) entre 40% e 74%, F1 entre 38% e 79%, AUC-ROC entre 56% e 93%, e MCC entre 59% e 88%. A categoria *Teste* obteve o melhor resultado, com precisão de 90%, revocação de 70%, F1 de 79%, AUC-ROC de 93% e MCC de 88%. Por outro lado, a categoria *Redes* apresentou o pior resultado, com precisão de 36%, revocação de 40%, F1 de 38%, AUC-ROC de 56% e MCC de 59%.

O estudo realizado por Siddiq e Santos (2022) apresenta um classificador automático baseado em Representações de Codificador Bidirecional de Transformadores (textitBidirectional Encoder Representations from Transformers (BERT)) para categorizar issues no GitHub em pergunta, bug ou melhoria O conjunto de dados utilizado é composto por 800.000 issues de projetos de código aberto, o conjunto de issues foi fornecido pelos organizadores da competição de ferramentas do Workshop Internacional de Engenharia de Software Baseada em Linguagem Natural (NLBSE'22)(KALLIS et al., 2022).

O pré-processamento dos textos envolveu a concatenação dos títulos e descrições das issues, remoção de caracteres de espaço em branco repetidos, tokenização e conversão de letras maiúsculas em minúsculas, e para classificador foi utilizado o modelo pré-treinado

[19] https://www.st.cs.uni-saarland.de/softevo//bugclassify/
[20] https://www.atlassian.com/software/jira

"bert-base-uncased". Os resultados do classificador demonstraram um desempenho robusto, com uma taxa F1 geral de 85,86%. Destacam-se as elevadas taxas F1 para as categorias de bugs (88,66%), melhorias (87,33%) e questões (60,58%), indicando a eficácia do modelo na classificação dessas categorias específicas. Além disso, o classificador alcançou a melhor taxa revocação com 90,15% para categoria Bug, 89,34% melhoria e 54,69% questões, e a melhor taxa de precisão da categoria Bug com 88,31%, melhoria 86,31% e questões 73,92%.

A ferramenta Ticket Tagger, disponível em https://github.com/rafaelkallis/ticket-tagger, é uma aplicação que utiliza estratégias de aprendizado de máquina para classificar issues no GitHub em categorias como questões, bugs ou melhorias, conforme descrito por Kallis et al. (2021) e Kallis et al. (2019). O processo de classificação é realizado com base no título e na descrição das issues. Os autores desenvolveram um aplicativo para realizar a categorização automática de issues presentes no GitHub, utilizando técnicas de aprendizado de máquina. A avaliação do desempenho do classificador foi conduzida em um conjunto de dados composto por 30.000 issues do GitHub. O objetivo principal da ferramenta é incentivar a categorização eficiente de projetos no GitHub, proporcionando uma forma automatizada e precisa de categorização para facilitar o gerenciamento e a compreensão das issues pelos colaboradores do projeto.

2.4 ANÁLISE DE SENTIMENTOS

A análise de sentimentos, as vezes conhecida como mineração de opinião, é uma área de Processamento de Linguagem Natural (PLN) que se dedica a analisar fragmentos textuais para identificar emoções, opiniões, apreciações ou sentimentos expressos pelo autor em relação a um determinado assunto. Essa abordagem é amplamente utilizada para extrair informações sobre a polaridade e a intensidade dos sentimentos associados a diferentes tópicos, como feedbacks sobre serviços, eventos, pessoas, produtos e seus atributos, conforme discutido por Pang e Lee (2008), Liu (2012) e Meena e Prabhakar (2007).

A análise de sentimentos desempenha um papel significativo em diversas áreas. Por exemplo, pode ser aplicada para medir percepções positivas e negativas de consumidores em avaliações de lojas online ou em plataformas de redes sociais (SARLAN; NADAM; BASRI, 2014; PANKAJ et al., 2019). Em situações de emergências e desastres, a análise de sentimentos pode ser utilizada para compreender as emoções e percepções da comunidade envolvida (SINGH; ROY; GANGOPADHYAY, 2018; BEHL et al., 2021). Além disso, é empregada na monitoração de campanhas eleitorais (SANDOVAL-ALMAZAN; VALLE-CRUZ, 2018; SAHU; CHOI, 2021) e na avaliação de sentimentos de alunos em redes sociais relacionadas ao currículo e atividades extracurriculares (IRAM, 2019).

Na área de Engenharia de Software, a análise de sentimentos pode ser aplicada para avaliar as emoções ou sentimentos dos desenvolvedores presentes em textos, como comentários de issues (ORTU et al., 2015a; JURADO; RODRIGUEZ, 2015), fóruns de discussões, listas de e-mails (TOURANI; JIANG; ADAMS, 2014), opiniões sobre evolução de aplicativos (CARREÑO; WINBLADH, 2013; GUZMAN; MAALEJ, 2014; ZHAO; ZHAO, 2019), sites de perguntas e respostas como o *Stack Overflow* (CALEFATO; LANU-

BILE; NOVIELLI, 2018), além de ser utilizada em redes sociais, commits e comentários de código-fonte (SINGH; SINGH, 2017).

A análise de sentimentos é comumente tratada como um problema de classificação textual, buscando automaticamente categorizar os sentimentos ou emoções presentes em trechos de texto (LIU, 2015, 2020). Esses sentimentos podem ser classificados em categorias predefinidas, como positivos, negativos ou neutros. Sentimentos positivos geralmente estão associados a emoções como alegria e amor, enquanto os sentimentos negativos estão relacionados a emoções como raiva e tristeza. Por outro lado, o sentimento neutro indica a ausência de uma emoção específica no texto (LIU, 2020; PANG; LEE; VAITHYANATHAN, 2002). Vale ressaltar que as emoções associadas a cada sentimento podem ter intensidades variadas, incluindo sarcasmo e ironia, o que pode tornar a classificação mais desafiadora.

A classificação de sentimentos pode ser realizada em diferentes níveis, sendo os três principais: nível de documento, nível de frase e nível de aspecto (LIU, 2010). Na classificação a nível de documento, a análise do sentimento é realizada considerando o texto como um todo. Nesse caso, pressupõe-se que o texto possui um único sentimento em relação a um determinado assunto (BENEVENUTO; ARAúJO; RIBEIRO, 2015; TANG, 2015). Na classificação a nível de frase, o foco recai na extração de sentimentos associados a frases ou sentenças específicas do texto (MEENA; PRABHAKAR, 2007). Nesse contexto, assume-se que cada frase do texto pode conter um sentimento distinto, podendo variar das demais frases. Já na classificação a nível de aspecto, os sentimentos são extraídos em relação a diferentes aspectos ou elementos presentes no texto (SINGH et al., 2013; VANAJA; BELWAL, 2018; WANG et al., 2019). Nesse caso, parte-se do pressuposto de que o texto pode abrigar múltiplos sentimentos relacionados a diversas características de uma entidade, como um produto ou serviço. Essa abordagem permite uma análise mais granular, identificando sentimentos específicos associados a diferentes aspectos mencionados no texto.

A implementação da análise de sentimentos pode adotar tanto uma abordagem de aprendizado de máquina quanto uma abordagem baseada em léxico. Exemplos de técnicas de aprendizado de máquina comumente empregadas incluem o classificador Naïve Bayes (AHMED et al., 2017a; PLETEA; VASILESCU; SEREBRENIK, 2014; Singla; Randhawa; Jain, 2017; Liu et al., 2013; DEY et al., 2016; TAN; ZHANG, 2008), SVM (AHMED et al., 2017a; BASARI et al., 2013; Singla; Randhawa; Jain, 2017; TAN; ZHANG, 2008), Árvore de Decisão (AHMED et al., 2017a; Singla; Randhawa; Jain, 2017), k-Nearest Neighbor (k-NN) (DEY et al., 2016; TAN; ZHANG, 2008) e Random Forest (AHMED et al., 2017a). Essas abordagens são aplicadas para atribuir rótulos de sentimentos aos textos com base em padrões extraídos dos dados de treinamento.

Por outro lado, a abordagem baseada em léxico envolve o uso de dicionários ou léxicos de palavras previamente associadas a polaridades sentimentais. Cada palavra no texto é mapeada para seu valor sentimental correspondente, e a polaridade geral do texto é determinada com base na soma ou média desses valores. Essa abordagem é útil quando se dispõe de léxicos abrangentes e contextualmente relevantes para o domínio específico do texto analisado.

2.4.1 Análise de sentimentos baseada em léxicos

No processo de análise de sentimentos baseada em léxicos, a classificação dos sentimentos é realizada com base em um dicionário de palavras que contém polaridades sentimentais, conhecido como léxico de sentimentos ou léxico de opinião (PANG; LEE, 2008; LIU, 2010, 2020). Cada palavra do léxico, que pode ser um adjetivo, substantivo, verbo ou advérbio, é associada a um sentimento específico. Por exemplo, palavras como *ótimo, maravilhoso, excelente e incrível* indicam sentimentos positivos, enquanto palavras como *ruim, terrível, odeio e pior* indicam sentimentos negativos.

Cada palavra no léxico de sentimentos é atribuída a uma pontuação, sendo positiva para palavras com sentimento positivo e negativa para palavras com sentimento negativo (LIU, 2020). Além disso, na abordagem baseada em léxico, é possível ponderar advérbios de intensidade, uma vez que eles caracterizam a força do sentimento. Por exemplo, na frase "Estou bastante preocupada com os testes de usabilidade", o advérbio *bastante* intensifica o sentimento de preocupação. Da mesma forma, é essencial considerar advérbios de negação, pois podem inverter a orientação do sentimento. Por exemplo, na frase "Eu não gosto da maneira como a lista está organizada", o advérbio *não* nega o sentimento de gostar, indicando uma avaliação negativa (LIU, 2020).

Na análise de sentimentos baseada em léxicos, a classificação do sentimento do texto é determinada pela soma da pontuação positiva com a pontuação negativa. Quando o resultado dessa soma é positivo, o texto é classificado como **positivo**; quando é negativo, o texto é classificado como **negativo**; e quando o resultado é igual a zero, o texto é classificado como **neutro**. Essa abordagem simples, mas eficaz, permite avaliar a orientação sentimental de um texto com base nas palavras presentes no léxico de sentimentos e em suas respectivas polaridades (LIU, 2020).

2.4.1.1 SentiStrength-SE

A *SentiStrength-SE* é uma ferramenta de análise de sentimentos baseada em léxicos projetada para textos curtos (THELWALL et al., 2010). A ferramenta atribui pontuações fixas aos tokens (palavras, expressões idiomáticas e emoticons) com sentimentos em um dicionário léxico. Os tokens com sentimentos positivos recebem pontuações no intervalo de [+1, +5], onde +5 corresponde a um sentimento extremamente positivo e +1 a um sentimento ligeiramente positivo. Da mesma forma, os tokens com sentimentos negativos recebem pontuações no intervalo de [-1, -5].

A SentiStrength-SE usa advérbios de intensidade para ajustar a pontuação do token em 1 ou 2 (por exemplo, muito, extremamente) ou diminuir em 1 (por exemplo, alguns). Pontuações são aumentadas para tokens que contêm ponto de exclamação ou letras repetidas (pelo menos duas letras adicionais), tratados como intensificadores. Advérbios de negação são usados para inverter tokens subsequentes (incluindo quaisquer intensificadores anteriores). Por exemplo, se "muito feliz" tivesse uma pontuação positiva +4, então "não muito feliz" teria uma pontuação negativa -4. Palavras com sentimentos negativos em perguntas são ignoradas, por exemplo, a frase *"você não está com raiva?"* seria classificada como não contendo sentimento, apesar da presença da palavra "raiva"

(THELWALL et al., 2010).

A pontuação final positiva e negativa de cada frase é calculada somando todas as pontuações individuais positivas ou negativas dos tokens. A SentiStrength-SE determina a emoção de um texto inteiro com base na maior pontuação entre todas as frases do texto (THELWALL et al., 2010; GUZMAN; BRUEGGE, 2013). A Tabela 2.4 apresenta exemplos de classificação de sentimentos em comentários de issues do GitHub usando a ferramenta SentiStrength-SE.

Tabela 2.4: Exemplos de textos em inglês para classificação de sentimentos em comentários de *issues* no GitHub usando SentiStrength-SE

Comentário	**Cálculo das polaridades**	**Pontuação**		**Polaridade**
		Neg.	**Pos.**	
I am closing this as wont fix because I find it very disturbing that an application on startup would restore full screen. I know Parallels seems to do that and it drives me nuts all the time.	*I am closing this as wont fix because I find it very disturbing* **[-1][-1 Intensidade]** *that an application on startup would restore full screen.* **[-3 1]** *I know Parallels seems to do that and it drives me nuts all the time.* **[-1 1]**	-3	1	Negativo (-1)
Sublime does this. A lot of Mac Applications do this and it's not disturbing to me at all. Potentially one for a configuration?	*Sublime does this.* **[-1 1]** A lot of Mac Applications do this and it's not disturbing **[-1][Negação]** to me at all. **[-1 2]** Potentially one for a configuration? [-1 1]	-1	2	Positivo (+1)
Fair enough.	Fair enough. **[-1 1]**	-1	1	Neutro (0)
these notes are fantastic, great job @rvagg :)	these notes are fantastic[2], great **[2]** **[+1 Várias palavras positivas]** job @rvagg :) **[+1 Emoticon]**	-1	4	Positivo (+1)
Oops Sorry! Closing now.	Oops**[-2]** Sorry**[-2][-1 Várias palavras negativas]**! Closing now.**[-1 1]**	-4	1	Negativo (-1)
Confirmed, thanks. Will fix.	Confirmed, thanks**[1]** **[-1 2]**. Will fix. **[-1 1]**	-1	2	Positivo (+1)
Fixed in joyent/libuv@0971598 and in node in 7e5aeac. It'll be in 0.8.2.	Fixed in joyent/libuv@0971598 and in node in 7e5aeac. It'll be in 0.8.2. **[-1 1]**	-1	1	Neutro (0)

Continua na próxima página

Tabela 2.4 – Continuação

Comentário	**Cálculo das polaridades**	**Pontuação**		**Polaridade**
		Neg.	**Pos.**	
Excellent – thanks for the quick fix!	Excellent[**3**] – thanks[**1**] for the quick fix! [**-1 4**]	-1	4	Positivo (+1)
I can't reproduce. ./test/run.sh works for me. Make sure you installed node dependencies.	I can't reproduce. ./test/-run.sh works for me. Make sure you installed node dependencies. [**-1 1**]	-1	1	Neutro (0)
@santhiya-v Sorry I don't know enough about cmake to be helpful :(	@santhiya-v Sorry [**-2**] I don't know enough about cmake to be helpful :([**-1 Emoticon**] [**-3 1**]	-3	1	Negativo (0)

2.4.2 Aplicação de análise de sentimentos em mineração de repositórios de software

A análise de sentimentos em repositórios de software é uma abordagem valiosa para compreender a dinâmica emocional dos desenvolvedores e suas interações. Sinha, Lazar e Sharif (2016) conduziram uma análise abrangente dos logs de commit em projetos GitHub, abrangendo um extenso período de sete anos e 28.466 projetos fornecidos para o desafio MSR 2016[21]. Utilizando a ferramenta SentiStrength, eles classificaram aproximadamente 2.251.585 logs de commit, estes foram categorizados em grande, médio e pequeno com base no número de commits. Surpreendentemente, a maioria dos commits (74,74%) apresentou sentimentos neutros, enquanto 18,05% foram classificados como negativos e 7,19% como positivos. Além disso, observaram que as terças-feiras foram associadas a um aumento nos sentimentos negativos nos logs de commit.

Já Guzman e Bruegge (2013) propuseram uma abordagem, combinando modelagem de tópicos probabilísticos (LDA) com análise de sentimentos baseada em léxico (SentiStrength) para resumir a consciência emocional em equipes de desenvolvimento de software. Ao analisarem 1.000 artefatos de colaboração produzidos por três equipes em três meses, incluindo mensagens de confirmação, relatórios de bug, e-mails e mensagens do Twitter, eles destacaram a influência do tipo de artefato na expressão de sentimentos pelos desenvolvedores, por exemplo, uma mensagem de e-mail de uma lista de discussão do projeto pode ter mais detalhes técnicos sobre um aspecto do projeto do que um tweet que trata do mesmo tópico. A abordagem foi avaliada em textos de lista de discussões e na ferramenta Confluence[22]. Durante as entrevistas, os autores mostraram às equipes alguns exemplos de textos que foram avaliados pela SentiStrength, gráficos demonstrando a média da pontuação dos sentimentos dos e-mails, resultado do Confluence de suas equipes

[21] <http://seresl.csis.ysu.edu/MSR16challenge>
[22] <https://www.atlassian.com/software/confluence>

e a variação dessa média durante o projeto .

No estudo conduzido por Guzman, Azócar e Li (2014), a ferramenta SentiStrength foi empregada para investigar as emoções presentes em comentários de commit de diversos projetos open source, analisando a relação dessas emoções com diferentes variáveis. O estudo considerou fatores como a linguagem de programação utilizada, a hora e dia da semana em que o commit foi submetido, a distribuição da equipe e a aprovação do projeto. O escopo abrangeu 60.425 comentários de commit, provenientes de 29 projetos de software, com foco nas principais linguagens de programação do GitHub. As análises revelaram que a pontuação média de sentimentos nos comentários de commit tendeu para a neutralidade, variando entre -1 e 1. Uma descoberta significativa foi a prevalência de comentários que se limitavam a aspectos técnicos, carecendo de expressões emocionais ou apresentando sentimentos apenas levemente positivos ou negativos. Entre as conclusões, destaca-se a observação de que projetos desenvolvidos em Java tendiam a apresentar mais comentários de commit com uma carga emocional negativa. Além disso, os projetos que contavam com equipes mais distribuídas mostraram uma tendência a exibir conteúdo emocional mais positivo. Esses resultados fornecem insights valiosos sobre a relação entre as emoções expressas nos comentários de commit e as características específicas dos projetos de software.

O estudo conduzido por Murgia et al. (2014) teve como objetivo investigar se os artefatos de desenvolvimento, como os relatórios de issues, carregam informações emocionais relevantes para o processo de desenvolvimento de software. Para isso, os pesquisadores mineraram 271.416 comentários provenientes de 20.537 usuários, referentes a 81.523 issues de 117 projetos open source na plataforma JIRA, no período de outubro de 2000 a julho de 2013. Dentro desse extenso conjunto de dados, foi selecionado um subconjunto de 792 comentários de desenvolvedores para a realização de um experimento (400 no estudo piloto, 392 no experimento). Esse experimento envolveu um grupo composto por quatro alunos de mestrado, 10 alunos de doutorado e dois pesquisadores associados da *Polytechnique Montréal* e *University of Antwerp*. No processo de análise, as emoções contidas nos comentários das issues foram classificadas em seis categorias primárias de emoções de Parrott, que incluem *amor, alegria, surpresa, raiva, tristeza e medo* (PARROTT, 2001). As emoções amor e alegria foram consideradas indicadores de sentimentos *positivos*, enquanto raiva, tristeza e medo foram indicadores de sentimentos *negativos*. Quanto à emoção surpresa, sua interpretação variou entre os sentimentos negativo e positivo dependendo das expectativas do autor do texto. Os resultados obtidos indicaram que os desenvolvedores expressam emoções, especialmente gratidão, alegria e tristeza, nos comentários associados às issues. Além disso, o estudo revelou discordâncias entre os avaliadores em relação a emoções específicas, sendo que as maiores discordâncias ocorreram quando não havia uma clara indicação de emoção no comentário. Essas descobertas contribuem para uma compreensão mais profunda da dimensão emocional presente na comunicação relacionada ao desenvolvimento de software.

O estudo realizado por Pletea, Vasilescu e Serebrenik (2014) concentrou-se na análise dos sentimentos presentes em discussões relacionadas à segurança no GitHub. Eles examinaram 60.658 commits e 54.892 *pull requests* de projetos de software do *MSR 2014 Mining Challenge*. Utilizando o classificador de sentimento *Natural Language Text Processing To-*

olKits (NLTK) do Python (BIRD; KLEIN; LOPER, 2009), os autores classificaram os sentimentos dos comentários em neutro, negativo ou positivo, além de criar um rótulo agregado que resumia essas três pontuações. Como resultado, descobriram que as discussões sobre segurança compõem cerca de 10% de todas as discussões no GitHub e que há uma maior expressão de emoções negativas nessas discussões em comparação com outros tópicos.

O estudo de Ortu et al. (2015a) analisou a relação entre sentimentos, emoções e polidez dos desenvolvedores com base em mais de 560 mil comentários no Jira. Os autores coletaram issues de 14 projetos principais do Jira, no período de 2002 a dezembro de 2013, selecionando apenas os projetos com o maior número de comentários. Para avaliar os sentimentos, utilizaram a ferramenta *SentiStrength*, enquanto para identificar as emoções dos comentários das issues, desenvolveram um classificador de aprendizado de máquina capaz de reconhecer quatro emoções primárias de Parrott (PARROTT, 2001): *alegria, amor, raiva e tristeza*. Os resultados indicaram que desenvolvedores mais felizes, expressando emoções como alegria e amor em seus comentários, estavam associados a tempos menores de resolução das issues. Em contrapartida, emoções negativas, como tristeza, foram relacionadas a um maior tempo de resolução das issues. Essas descobertas contribuem para a compreensão das complexas interações entre emoções e dinâmicas de desenvolvimento de software.

No estudo de Ortu et al. (2016), foi construída uma base de dados de emoções presentes em comentários de issues. Um experimento envolveu a rotulação manual de 2.000 comentários de issues e 4.000 sentenças escritas por desenvolvedores, categorizando as emoções com base no framework de Parrott (PARROTT, 2001), incluindo categorias como *amor, alegria, surpresa, raiva, tristeza e medo*. Os dados foram coletados de mais de mil projetos, abrangendo mais de 700 mil relatórios de issues e mais de 2 milhões de comentários de issues, provenientes do repositório Jira de quatro comunidades de software open source: *Apache, Spring, JBoss e CodeHaus*.

No trabalho de Destefanis et al. (2018), foram analisadas issues e comentários de projetos GitHub, com a construção de redes de colaboração dividindo os contribuidores em duas categorias: *user* e *commenter*. O grupo *user* compreende usuários especializados, como desenvolvedores e usuários, enquanto *commenter* refere-se a usuários que apenas postam comentários nas issues. A análise empírica envolveu mais de 370 mil comentários de 25 mil colaboradores em 100 mil issues de três projetos de código aberto no GitHub. Utilizou-se a ferramenta *Sentistrength* para calcular os sentimentos dos comentários dos contribuidores e uma ferramenta desenvolvida por Danescu-Niculescu-Mizil et al. (2013) para classificar os comentários como *educados (polite) ou mal-educados (impolite)*. Os resultados apontaram para a existência de diferentes grupos de colaboradores no GitHub, cada um se comportando de maneira distinta. Essa diversidade de comportamentos destaca a complexidade da dinâmica de colaboração presente nesses ambientes de desenvolvimento de software.

No estudo conduzido por Carige e Carneiro (2020), realizou-se um estudo exploratório para caracterizar a polaridade de sentimentos nos comentários associados a issues e tickets das releases do Moby, um projeto mantido pela empresa Docker e hospedado no GitHub. A análise abrangeu 235.295 comentários provenientes de 36.590 issues com status “clo-

sed". A ferramenta SentiStrength-SE foi utilizada para analisar os sentimentos expressos pelos desenvolvedores nos comentários, identificando as polaridades positivas e negativas em cada um deles. Além disso, os autores expandiram o dicionário da ferramenta para incluir emojis, atribuindo polaridades com base na lista de sentimentos de emojis (Emoji List, v1.0) (NOVAK et al., 2015). O estudo revelou a existência de programadores cujas contribuições influenciam na variação da polaridade das releases, apresentando uma inclinação tanto para predominância de polaridade positiva quanto negativa nos comentários postados. Essa análise proporcionou insights valiosos sobre a dinâmica emocional e a recepção das releases por parte da comunidade de desenvolvedores.

2.4.3 Ferramentas de análise de sentimentos desenvolvidas para o contexto de engenharia de software

A ferramenta SentiCR, desenvolvida por Ahmed et al. (2017a), é uma solução de análise de sentimento projetada para o contexto da Engenharia de Software. O processo de construção dessa ferramenta envolveu a rotulação manual de 1.600 comentários de revisão de código, criando assim um oráculo de sentimentos específico para esse domínio. No pré-processamento do texto, a ferramenta calcula o TF-IDF para extrair recursos para classificação. Além disso, foi criada uma lista de *stopwords* a partir de termos comuns em linguagens de programação populares, como Java, C, C++, Python, JavaScript e PHP, para adicioná-la à lista de palavras irrelevantes a serem removidas. Para aprimorar o desempenho da ferramenta, os autores converteram o oráculo de três sentimentos em um conjunto de dados de dois sentimentos: *negativo e não negativo*, sendo este último abrangendo os sentimentos positivos e neutros (AHMED et al., 2017b).

A ferramenta Senti4SD, apresentada por Calefato et al. (2018b), é um classificador desenvolvido especificamente para auxiliar a análise de sentimentos de desenvolvedores de software em canais de comunicação. Seu objetivo principal é retornar o grau de subjetividade (objetivo/neutro ou subjetivo com polaridade positiva/negativa) e o grau de polaridade: positiva e negativa. Os autores conduziram um experimento controlado para criar uma base de dados rotulada com polaridades positivas, neutras e negativas, utilizando perguntas, respostas e comentários do Stack Overflow (STACKOVERFLOW, 2023). O experimento envolveu atividades de treinamento e classificação, onde participantes rotularam posts em positivo, negativo, neutro e misto. A validação adicional foi realizada por meio de um experimento online envolvendo 48 participantes que validaram 559 palavras, *emoticons* e expressões (CALEFATO et al., 2018a).

Os autores conduziram um experimento controlado para construir uma base de dados rotulada com polaridades positivas, neutras e negativas, utilizando perguntas, respostas e comentários provenientes do Stack Overflow (STACKOVERFLOW, 2023). O experimento compreendeu atividades de treinamento e classificação, nos quais os participantes rotularam posts em positivo, negativo, neutro e misto. Cada participante rotulou 25 posts em 30 minutos. Durante o experimento, os participantes foram distribuídos em quatro grupos, cada um composto por três participantes, e cada participante classificou 100 *posts*. No total, cada participante rotulou 500 *posts*, e esses foram posteriormente rotulados por três codificadores, gerando um conjunto de dados com 2000 novos posts

rotulados. O estudo também empregou uma quantidade considerável de termos léxicos para validação, por meio de um experimento com formulários online, com a participação de 48 pessoas validando 559 palavras, *emoticons* e expressões (CALEFATO et al., 2018a).

2.5 CONCLUSÃO DO CAPÍTULO

O presente capítulo forneceu uma base sólida ao apresentar uma variedade de conceitos fundamentais relacionados à mineração de repositórios, destacando elementos cruciais como issues, ITS, reabertura de issues, e categorização automática de issues. Exploramos também as nuances da análise de sentimentos, incluindo métodos tradicionais e ferramentas modernas dedicadas ao contexto da engenharia de software.

Ao compreender esses conceitos, podemos agora direcionar nosso foco para o capítulo 3, onde exploraremos estudos específicos que aplicam análise de sentimentos e examinam a reabertura de issues no contexto do GitHub.

Capítulo

3

ANÁLISE DE SENTIMENTOS EM ISSUES

Este capítulo representa a aplicação prática dos conceitos discutidos nos capítulos anteriores, concentrando-se em experimentos que abordam a análise de sentimentos em issues do GitHub. Com o objetivo de analisar os sentimentos dos desenvolvedores nas discussões do GitHub, este capítulo está estruturado em torno de três objetivos principais.

Primeiramente, realizamos um estudo piloto dedicado à mineração de sentimentos em issues, conforme descrito na Seção 3.1, alinhado ao objetivo de conduzir um estudo inicial sobre ferramentas de análise de sentimentos (Obj 2). As questões de pesquisa relacionadas a essa etapa são: *QP1 Qual é a melhor ferramenta de análise de sentimentos voltada para a área de desenvolvimento de software para classificar os sentimentos de issues em larga escala?*, *QP1.1 Quais são as principais ferramentas de análise de sentimentos voltadas para a área de desenvolvimento de software?*

Em seguida, seção 3.2, abordamos o processo de validação e construção de um dicionário léxico, uma etapa fundamental para garantir a precisão das análises automatizadas, de acordo com o objetivo de validar a ferramenta de análise de sentimentos SentiStrength-SE (Obj 3). As questões de pesquisa para essa seção são: *QP1.2. Como validar a classificação de polaridades do dicionário léxico da ferramenta SentiStrength-SE? QP1.2.1. Os participantes concordam com a classificação das polaridades da ferramenta Sentistrength-SE? QP1.2.2. Qual o comportamento dos resultados e a concordância da classificação do dicionário da ferramenta nos grupos de participantes selecionados pela experiência em desenvolvimento de software, fluência no idioma do dicionário e habilidade em análise de sentimentos? QP1.2.3. O Dicionário léxico gerado pela classificação dos participantes melhora a acurácia na análise de sentimentos de textos reais?*

Posteriormente, exploramos um estudo específico centrado na análise de sentimentos em discussões de issues reabertas, conforme apresentado na Seção 3.3. As questões de pesquisa relevantes para essas seções são: *QP2: Como caracterizar a influência dos sentimentos na reabertura de issues? QP2.1. Existe algum indicativo de que uma issue não será reaberta se ela for fechada com um sentimento positivo? QP2.2. É possível predizer se uma issue será reaberta quando o número de sentimentos negativos adicionados ao*

número de sentimentos neutros for maior que o número de sentimentos positivos presentes nas suas discussões? QP2.3. Os sentimentos dos comentários após o fechamento da issue podem indicar que ela será reaberta?

Além disso, na Seção 3.4, conduzimos uma análise comparativa nas discussões de issues, considerando aquelas com e sem reaberturas, com o objetivo compreender o impacto da presença desses ciclos de reabertura nas interações e dinâmicas emocionais entre os colaboradores. Esses estudos são fundamentais para alcançar o objetivo de caracterizar a influência dos sentimentos nos processos de reabertura de issues, conforme proposto no (Obj 4). As questões de pesquisa relevantes para essas seções são: *QP2.4 Existe algum indicativo de que uma issue fechada será reaberta com base no sentimento, pontuação negativa e pontuação positiva do primeiro comentário após a abertura da issue?*

QP2.5 Existe algum indicativo de que uma issue fechada será reaberta com base no sentimento, pontuação negativa e pontuação positiva do último comentário antes do primeiro fechamento da issue? QP2.6 É possível prever se uma issue será reaberta através dos sentimentos presentes nas suas discussões entre os eventos de abertura (open) e o primeiro fechamento (closed)? QP2.7 Existe uma correlação forte entre os sentimentos das discussões das issues com a probabilidade de reabertura?

3.1 ESTUDO PILOTO DE MINERAÇÃO DE SENTIMENTOS EM ISSUES

Este estudo visa identificar e avaliar ferramentas de análise de sentimentos voltadas para o desenvolvimento de software, aplicadas à classificação de sentimentos em *issues* e *pull requests* do GitHub. O objetivo principal é determinar qual ferramenta se destaca na análise de sentimentos em larga escala para esse domínio. A questão central que orienta esta investigação é: *QP1 Qual é a melhor ferramenta de análise de sentimentos voltada para a área de desenvolvimento de software para classificar os sentimentos de issues em larga escala?* Para responder a essa questão, é considerada também a seguinte questão derivada: *QP1.1 Quais são as principais ferramentas de análise de sentimentos voltadas para a área de desenvolvimento de software?*

3.1.1 Metodologia

Para conduzir o estudo piloto de Mineração de Sentimentos em issues, adotamos uma abordagem em quatro etapas fundamentais: seleção dos repositórios, pré-processamento do texto, classificação manual dos sentimentos e escolha da ferramenta de Análise de Sentimentos. A Figura 3.1 ilustra visualmente as quatro etapas do estudo, e cada uma delas é detalhada a seguir.

3.1.1.1 Seleção do repositório

Inicialmente, selecionamos um projeto de escala pequena para realizar a classificação manual de sentimentos em discussões de *issues*. O projeto selecionado foi o GeoIP2 Java API, pertencente ao repositório maxmind/GeoIP2-Java[1] no GitHub. Realizamos a ex-

[1] <https://github.com/maxmind/GeoIP2-java>, acessado em 02/02/2024

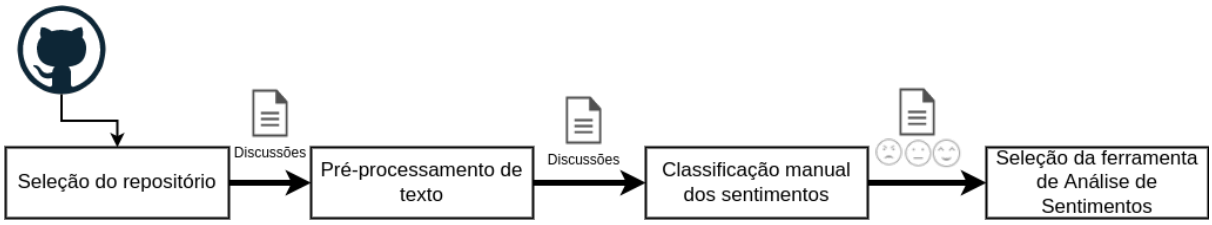


Figura 3.1: Etapas do estudo sobre mineração de sentimentos em issues

tração manual de 66 *issues* (61 fechadas e 5 abertas) criadas no período entre 19/08/2013 e 21/08/2018. A extração manual das *issues* foi concluída em 22/08/2018. Para a fase de classificação, optamos por incluir apenas as *issues* que apresentavam discussões significativas, isto é, aquelas que possuíam pelo menos dois comentários e envolviam dois ou mais participantes. No total, foram selecionadas 39 *issues* para a análise de sentimentos.

3.1.1.2 Pré-processamento de texto

Nesta etapa, realizamos o pré-processamento dos textos presentes nas descrições e comentários das *issues*. Essa etapa visou a limpeza dos textos, removendo trechos irrelevantes que poderiam prejudicar a classificação dos sentimentos. As ações de limpeza incluíram a remoção de URLs, código-fonte, trechos de código, erros de compilação, classes, interfaces, imagens, respostas a comentários, frases de alerta sobre *warnings* e exceções.

3.1.1.3 Classificação manual dos sentimentos

A classificação dos sentimentos dos textos das issues foi realizada por três pesquisadores da área de engenharia de software com conhecimento em análise de sentimentos. A abordagem adotada envolveu a classificação em pares. A classificação foi feita em pares, onde cada comentário ou descrição foi classificado como negativo, positivo ou neutro.

3.1.1.4 Seleção da ferramenta de Análise de Sentimentos

Selecionamos três ferramentas de análise de sentimentos desenvolvidas para o contexto de engenharia de software. A ferramenta *SentiStrength-SE* (ISLAM; ZIBRAN, 2017) utiliza análise de sentimentos baseada em léxico e as ferramentas *SentiCR* (AHMED et al., 2017a) e *Senti4SD* (CALEFATO et al., 2018a) que realizam a análise de sentimentos com aprendizado de máquina, as ferramentas estão descritas na sessão 2.4.3.

No primeiro momento, os resultados da classificação manual são comparados com os resultados da ferramenta *SentiStrength-SE* e em seguida comparamos os resultados da classificação das 10 issues com mais comentários com os resultados das ferramentas *Senti4SD* e *SentiCR*.

3.1.2 Resultados

Neste estudo, inicialmente realizamos a classificação dos sentimentos manualmente de 335 comentários e descrições de 39 issues do projeto GeoIP2 Java API. A classificação dos sentimentos e informações adicionais das issues estão disponíveis em <https://bit.ly/3G1wD8U>.

Para responder a questão de pesquisa *QP1.1: Quais são as principais ferramentas de análise de sentimentos voltadas para a área de desenvolvimento de software?* Nós selecionamos as ferramentas *SentiStrength-SE*, *SentiCR* e *Senti4SD*, ferramentas desenvolvidas para realizar a análise de sentimentos no contexto de engenharia de software. As ferramentas diferem em suas abordagens: *SentiStrength-SE* é baseada em léxico, enquanto *SentiCR* e *Senti4SD* utilizam aprendizado de máquina, o que implica diferentes vantagens e desvantagens para cada uma.

A partir das ferramentas de análise de sentimentos selecionadas podemos responder a pergunta de pesquisa *QP1: Qual é a melhor ferramenta de análise de sentimentos voltada para a área de desenvolvimento de software para classificar os sentimentos de issues em larga escala?*, analisamos os resultados obtidos por três ferramentas: *SentiStrength-SE*, *SentiCR* e *Senti4SD*. A avaliação foi baseada em três critérios principais: precisão na concordância com a classificação manual, eficiência em termos de consumo de recursos, e adequação ao uso em larga escala.

3.1.2.1 Precisão da Classificação Após realizar a classificação manual resolvemos comparar os resultados da classificação com a classificação realizada de forma automatica pela ferramenta *SentiStrength-SE*. A comparação teve como objetivo avaliar a concordância entre a classificação manual e a automática, a fim de mensurar a precisão da ferramenta em relação à percepção humana. A ferramenta *SentiStrength-SE* concordou com a classificação manual em 48,06% dos 335 comentários e descrições das 39 issues analisadas, conforme mostrado na tabela 3.1, a planilha com as classificações dos comentário estão disponíveis <https://bit.ly/2ZHft0L>.

Tabela 3.1: Concordância entre a classificação manual e a ferramenta SentiStrength-SE

Comentários	**Manual** x **SentiStrength-SE**
335	161
%	**48,06%**

Em seguida, selecionamos as 10 issues mais comentadas, totalizando 193 comentários, comparamos a classificação manual com o resultado da classificação das ferramentas *SentiStrength-SE*, *Senti4SD* e *SentiCR*. A planilha com os resulyados das classificações das ferramentas estão disponíveis em <https://bit.ly/3EeJNyW>. <https://bit.ly/3EeJNyW>.

A ferramenta *SentiCR* obteve a maior concordância com a classificação manual (52,33%), seguida por *Senti4SD* (43,01%) e *SentiStrength-SE* (42,49%), conforme ilustrado na ta-

bela 3.2. Esse resultado indica que, do ponto de vista de precisão da classificação, *SentiCR* é a ferramenta que mais se aproxima do julgamento humano, o que pode ser um fator decisivo ao escolher a melhor ferramenta para análise de sentimentos em issues de software.

Tabela 3.2: Concordância entre classificação manual e as ferramentas SentiStrength-SE, SentiCR e Senti4SD

Comentários	**Manual X SentiStrength-SE**	**Manual X SentiCR**	**Manual X Senti4SD**
193	82	101	83
%	42,49%	52,33%	43,01%

Também analisamos a concordância da classificação entre ferramentas *SentiStrength-SE*, *SentiCR* e *Senti4SD*. Identificamos que a maior concordância da classificação foi entre as ferramentas *SentiCR* e *Senti4SD* com uma concordância de 72,02%, seguida da concordância de 70,98% entre *SentiStrength-SE* e *SentiCR* e por fim 59,59 % de concordância entre *SentiStrength-SE* e *Senti4SD*, como mostrado na tabela 3.3.

Tabela 3.3: Concordância entre as ferramentas *SentiStrength-SE*, *SentiCR* e *Senti4SD* na Classificação de Sentimentos

Comentários	**SentiStrength-SE X SentiCR**	**SentiStrength-SE X Senti4SD**	**SentiCR X Senti4SD**
193	137	115	139
%	70,98 %	59,59 %	72,02 %

3.1.2.2 Eficiência de Recursos Além da precisão, a eficiência em termos de uso de recursos também é um critério importante ao considerar o uso dessas ferramentas em larga escala. Conforme ilustrado na tabela 3.4, a ferramenta *SentiStrength-SE* demonstrou ser a ferramenta mais eficiente em termos de espaço em disco, consumo de memória e tempo de classificação. Por ser baseada em léxico, essa ferramenta não requer treinamento de modelos de aprendizado de máquina, o que a torna mais leve e rápida para a análise de grandes volumes de dados. Em contrapartida, tanto *SentiCR* quanto *Senti4SD*, que utilizam técnicas de aprendizado de máquina, exigem mais recursos computacionais e tempo para processar os dados, especialmente quando aplicadas em cenários de larga escala.

3.1.2.3 Adequação ao Uso em Larga Escala Para aplicações que envolvem o processamento de um grande número de issues com comentários, como em projetos de código aberto com milhares de contribuições, a eficiência em termos de recursos pode ser tão importante quanto a precisão. Nesse contexto, *SentiStrength-SE* se destaca por ser

Tabela 3.4: Comparação das Ferramentas de Análise de Sentimentos em Termos de Espaço, Memória e Tempo de Classificação

Ferramenta	**Técnica**	**Espaço em Disco**	**Consumo de Memória**	**Tempo de Classificação**
SentiStrength-SE	Léxico	Menor	Menor	Menor
SentiCR	ML	Maior	Maior	Maior
Senti4SD	ML	Maior	Maior	Maior

mais adequada para uso em larga escala, dado o menor consumo de memória e tempo de execução. Entretanto, *SentiCR*, apesar de exigir mais recursos, pode ser preferida em situações em que a precisão é um fator mais crítico do que a eficiência.

3.1.3 Ameaças à validade

Validade Interna.A escolha de apenas um repositório (*GeoIP2 Java API*) para a análise de sentimentos pode restringir a diversidade das discussões e contextos analisados, uma vez que os resultados podem ser específicos desse projeto.Embora essa escolha possa inicialmente parecer uma limitação, ela foi intencional e adequada para o propósito deste estudo piloto, em que selecionamos um projeto relacionado à área de engenharia de software e de pequena escala, mas que apresenta uma quantidade suficiente de issues com comentários para realizar a análise de sentimentos.

A classificação manual das issues, realizada por três pesquisadores, pode estar sujeita a vieses pessoais ou interpretações subjetivas dos textos. Para mitigar essa ameaça, a abordagem adotada envolveu a classificação em pares, onde cada comentário foi revisado independentemente por dois pesquisadores, e as discrepâncias nas classificações foram discutidas até que se chegasse a um consenso.

Validade Externa. Os resultados obtidos a partir da classificação manual das issues do repositório *GeoIP2 Java API* e a subsequente comparação com as ferramentas *SentiStrength-SE*, *SentiCR* e *Senti4SD* podem ser específicos a esse repositório, devido às suas características únicas, como o tipo de discussões e o estilo de comunicação. No entanto, essa escolha foi adequada para os objetivos do estudo, pois o *GeoIP2 Java API* é representativo de projetos de software de código aberto com discussões técnicas relevantes, tornando os resultados aplicáveis a repositórios com contextos semelhantes dentro da engenharia de software.

Validade de Construto. As ferramentas *SentiStrength-SE*, *SentiCR* e *Senti4SD* podem não captar bem os sentimentos das issues no contexto da engenharia de software, o que pode distorcer os resultados. No entanto, essas ferramentas foram escolhidas por terem sido desenvolvidas especificamente para analisar sentimentos em textos de engenharia de software, o que contribui para a adequação delas ao estudo.

3.1.4 Conclusão

Nesse estudo, avaliamos três ferramentas de análise de sentimentos voltadas para o desenvolvimento de software, *SentiStrength-SE*, *SentiCR* e *Senti4SD* com o objetivo de classificar os sentimentos em comentários de issues no GitHub. A análise comparou a concordância entre a classificação manual de 193 comentários e os resultados obtidos pelas ferramentas.

Os resultados indicaram que a *SentiCR* apresentou a maior concordância com a classificação manual, sendo a mais precisa entre as ferramentas avaliadas. Além disso, a maior concordância entre as ferramentas foi observada entre *SentiCR* e *Senti4SD*, o que reforça a robustez dessas ferramentas para análise de sentimentos. No entanto, a *SentiCR* adota uma categorização binária, classificando os sentimentos como *negativos* ou *não negativos*, o que pode simplificar a análise, mas também limitar a granularidade dos resultados.

Nosso objetivo, entretanto, é a classificação de um grande volume de textos extraídos de issues do GitHub, o que torna a eficiência de recursos e a escalabilidade fatores fundamentais. Nesse sentido, optamos pela *SentiStrength-SE*, que se mostrou mais adequada para lidar com a alta demanda de processamento de grandes quantidades de texto, além de apresentar menor consumo de recursos computacionais. A *SentiStrength-SE* também oferece a vantagem de capturar separadamente as pontuações positivas e negativas, o que agrega mais nuances à análise de sentimentos.

Analisamos a concordância de 193 comentários de issues entre a classificação manual e as ferramentas *SentiStrength-SE*, *SentiCR* e *Senti4SD*, os resultados revelaram que a maior concordância ocorreu entre a classificação manual e a ferramenta *SentiCR*. Entre as ferramentas, a maior concordância foi observada entre *SentiCR* e *Senti4SD*. Se a prioridade da escolha da ferramenta for precisão na classificação, *SentiCR* é a ferramenta recomendada, devido à sua maior concordância com a classificação manual e as ferramentas *SentiStrength-SE* e *Senti4SD*. Acreditamos que esse padrão pode ser atribuído à natureza binária da classificação realizada pela ferramenta *SentiCR*, que categoriza sentimentos como *negativos* ou *não negativos*, levando-nos a agrupar sentimentos positivos e neutros como *não negativos*.

No entanto, nossa prioridade é ter eficiência de recursos e a escalabilidade. A ferramenta *SentiStrength-SE* se apresenta como a melhor escolha, especialmente para projetos que precisam processar grandes volumes de textos com rapidez e baixo consumo de recursos. A ferramenta também permite capturar as pontuações positivas e negativas presentes nas mensagens analisadas.

Como próximo passo, realizamos uma validação dos tokens presentes no dicionário léxico da ferramenta para assegurar sua precisão e adequação ao contexto de engenharia de software, proporcionando maior confiança na utilização da ferramenta em análises futuras em larga escala.

3.2 VALIDAÇÃO E CONSTRUÇÃO DE UM DICIONÁRIO LÉXICO PARA AUXILIAR A ANÁLISE DE SENTIMENTOS EM REPOSITÓRIOS DE PROJETOS DE SOFTWARE

A análise de sentimentos é um método de mineração de texto útil para processar conteúdo textual e filtrar os resultados com métodos de análise para obter informações relevantes e significativas (LIU, 2015). Para a manutenção de software, analisar as emoções ou sentimentos dos desenvolvedores através da mineração de textos de bugs/issues ou commits pode se tornar um método capaz de auxiliar a compreensão no gerenciamento do projeto (MURGIA et al., 2014).

O processo de classificação textual demanda o uso de ferramentas de suporte, para que seja possível alcançar escala. Dentre as ferramentas existentes, destaca-se a SentiStrength[2], que analisa sentimentos em pequenos trechos de texto (THELWALL et al., 2010). Com o propósito de analisar o contexto inerente a Engenharia de Software, foi desenvolvida a ferramenta SentiStrength-SE, que faz uso de uma base de dados composta por palavras comuns em Desenvolvimento de Software (ISLAM; ZIBRAN, 2018). Quando tal conceito é aplicado a ferramentas que desenvolvedores utilizam para troca de textos, é possível relacionar o sentimento do desenvolvedor à sua tarefa.

Neste estudo, analisamos a polaridade de 559 palavras, *emoticons* e expressões idiomáticas contidas no dicionário léxico da ferramenta Sentistrength-SE. Os dados foram obtidos a partir da participação de 48 desenvolvedores de software em um experimento. A alocação dos participantes em grupos considerou os dados do formulário de caracterização, que classificou a experiência em desenvolvimento de software, fluência em inglês e habilidade no tema de análise de sentimentos. A escala de 1 à 5 foi utilizada para descrever os grupos com base nesta caracterização, sendo o *score* 1 atribuído aos participantes com pouca experiência, fluência e habilidade nos grupos apresentados e o *score* 5 aqueles com maior experiência, fluência e habilidade. O estudo foi orientado pelas três perguntas de pesquisa descritas a seguir:

- **QP1.2.1. Os participantes concordam com a classificação das polaridades da ferramenta Sentistrength-SE?**

  O estudo busca mensurar a concordância dos participantes nas rotulações pré-definidas das palavras, *emoticons* e expressões idiomáticas apresentadas no dicionário da ferramenta Sentistrength-SE. Estes termos utilizados na ferramenta foram validados pelos participantes, a partir de uma escala de polaridade variando de -2 a +2. Tal polaridade é comparada com a classificação já apresentada pela ferramenta para encontrar ocorrências de respostas semelhantes. A partir destes resultados, o grau de confiabilidade desta classificação é contabilizado pela concordância do participante com a polaridade rotulada nas respostas. Se houver concordância, então o partipante aceita a rotulação do termo de acordo com a classificação, porém rotulações diferentes resultam na discordância da polaridade desta classificação.

- **QP1.2.2. Qual o comportamento dos resultados e a concordância da**

[2] <http://sentistrength.wlv.ac.uk/>

classificação do dicionário da ferramenta nos grupos de participantes selecionados pela experiência em desenvolvimento de software, fluência no idioma do dicionário e habilidade em análise de sentimentos?

O objetivo é encontrar os valores obtidos por cada grupo de participantes que colaboraram na análise do dicionário léxico da ferramenta. É possível que a tendência nas respostas possa trazer algum padrão para elucidar os resultados de concordância e discordância das palavras, emoticons e expressões idiomáticas.

- **QP1.2.3. O Dicionário léxico gerado pela classificação dos participantes melhora a acurácia na análise de sentimentos de textos reais?**

  O objetivo do experimento é trazer uma validação dos 559 termos léxicos através de uma nova rotulação destes termos por desenvolvedores de software de todos os graus de experiência. Para responder esta pergunta de pesquisa utilizamos a aferição deste novo dicionário rotulado ao final do processo de coleta do experimento com a análise da base de dados *gold standard*[3] com posts extraídos da plataforma de perguntas e respostas *Stack Overflow*(CALEFATO et al., 2018b). As métricas *accuracy*, *precision*, *recall* e *f1-score* do novo dicionário são aferidas.

3.2.1 Metodologia

O estudo foi dividido em três etapas: planejamento, execução e criação/validação do dicionário léxico, conforme descrito a seguir. O estudo piloto obteve respostas de 10 participantes e o experimento contou com a participação de 48 pessoas. Todo o pacote experimental, incluindo os formulários, termos léxicos, tabelas e gráficos sumplementares, bem como os dados obtidos, encontra-se disponível em (MENEZES et al., 2020).

Etapa 1 - Planejamento: Inicialmente, levantamos um conjunto de 560 termos léxicos com palavras, *emoticons* e expressões idiomáticas do dicionário léxico da ferramenta, mas a palavra 'bg' foi removida por apresentar possível interpretação ambígua na validação e classificação. Assim, o conjunto validado consta de 559 palavras, *emoticons* e expressões idiomáticas. Exemplos de classificação, na frase *"It's a good feature"*, a palavra *"good"* possui pontuação 2, então a sentença recebe pontuação positiva igual a 2 (ISLAM; ZIBRAN, 2018). Para o *emoticon* " :) ", a polaridade é positiva com pontuação 1. Na expressão idiomática "Shock horror" a polaridade é negativa com pontuação -2.

Para reduzir a granularidade e evitar ambiguidade na rotulação do experimento, as pontuações -4 e -3 foram mapeadas para a pontuação -2. As pontuações -2 e -1 foram mapeados para a pontuação -1. Analogamente, as pontuações +1 e +2 foram atribuídos à pontuação +1, e as pontuações +3 e +4 foram atribuídos à pontuação +2. Os termos léxicos foram distribuídos em ordem alfabética em 9 grupos para serem validados e classificados. Um grupo foi utilizado para o experimento piloto e os 8 grupos restantes para o experimento, conforme apresentado na Tabela 3.5.

[3]<https://github.com/collab-uniba/Senti4SD/blob/master/Senti4SD_GoldStandard_and_DSM/Senti4SD_GoldStandard_EmotionPolarity.xlsx>

Tabela 3.5: Dados do Experimento e do Estudo Piloto

Grupo	**C**		**D**		**E**		**F**		**G**		**H**		**I**		**J**	
Seção	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
Palavras	28	27	28	27	28	27	28	27	28	27	28	27	28	27	28	27
Emoticons	3	4	3	4	3	4	3	4	3	3	3	3	3	3	3	3
Exp. Idiom.	4	4	4	4	4	4	4	4	4	5	4	5	4	5	4	5
Participantes	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

Cada grupo de participantes realizou as tarefas seguindo uma dada ordem: (i) ***classificação*** **e** ***validação***, ou (ii) ***validação*** **e** ***classificação***, conforme detalha a Tabela 3.6. Na tarefa de **classificação**, o participante rotula o termo léxico, atribuindo-lhe um valor no intervalo de -2 a +2. Na tarefa de **validação**, o termo léxico é rotulado pelo participante que concorda ou não com o rótulo apresentado. Ao discordar, imediatamente é sugerido que o participante apresente uma nova classificação utilizando o intervalo.

Tabela 3.6: Ordem das tarefas realizadas no experimento.

Tarefas	**Grupos**
Classificação e Validação	C1, D1, E1, F1, G1, H1, I1, J1
Validação e Classificação	C2, D2, E2, F2, G2, H2, I2, J2

Um estudo piloto foi executado, com o objetivo de avaliar os instrumentos do estudo experimental. Para a realização do estudo piloto, dois formulários foram criados. No primeiro formulário, o participante começa a fase de classificação e logo após é realizada a fase de validação. O segundo formulário é realizado de forma inversa, o participante começa a validar e depois classificar novos termos léxicos. Cada formulário foi respondido por cinco pessoas. Ambos os formulários incluem 110 palavras, 20 emoticons e 18 expressões idiomáticas. O resultado do piloto apresentou 103 respostas iguais aos da ferramenta e 44 respostas diferentes.

Etapa 2 - Execução: O experimento utilizou os formulários descritos no estudo piloto. Um ajuste significativo diz respeito a quantidade de palavras que o participante deveria classificar e validar. A quantidade de palavras, *emoticons* e expressões idiomáticas presentes no piloto foi reduzida pela metade para o experimento, dado que os participantes do piloto relataram fadiga na execução da tarefa com um número elevado de questões. As 440 palavras, 52 emoticons, 68 expressões idiomáticas do dicionário léxico restantes da ferramenta foram divididos nos oito grupos citados anteriormente.

Etapa 3 - Criação e Validação do Novo Léxico: A partir dos dados obtidos no experimento foi possível classificar as polaridades do dicionário da ferramenta, onde a polaridade final de cada palavra, emoticon ou expressão idiomática foi obtida através da maior votação dos participantes. Em casos de empates, considerou-se o valor de maior pontuação, para ampliar a precisão dos dados. Caso os resultados apresentados tivessem polaridades distintas, optamos por escolher a polaridade encontrada no dicionário da ferramenta previamente extraída. Para validar o léxico obtido do experimento, utilizamos

a base de dados *gold standard*[4] (CALEFATO et al., 2018b) com 4.423 posts com perguntas, respostas e comentários do *Stack Overflow*. Os dados foram pré-processados, onde os elementos do texto que não continham sentimentos foram descartados, como URLs, trechos de códigos e tags. A performance do novo dicionário léxico foi avaliada utilizando a comparação dos resultados da ferramenta SentiStrength-SE. O SentiStrength-SE Alterado também foi avaliado. Este dicionário consiste exatamente nos termos léxicos do SentiStrength-SE apenas com escala de polaridade modificada de -2 a +2. Como critério de avaliação do novo dicionário léxico foram utilizadas as métricas *accuracy*, *precision*, *recall* e f_1-*score*.

3.2.2 Resultados

O experimento coletou 559 respostas de 48 participantes. Os participantes que começaram o estudo na tarefa de ***validação*** do dicionário levaram em média 7 minutos e 10 segundos, enquanto que os participantes que começaram o estudo na tarefa de ***classificação*** levaram em média 3 minutos e 58 segundos para a execução do estudo. Os participantes que começaram com a tarefa de ***classificação*** levaram em média 5 minutos e 15 segundos e 4 minutos e 38 segundos para ***validação*** do dicionário. As 559 palavras, emoticons e expressões idiomáticas combinadas foram validadas pelos participantes. Tais dados foram analisados na **QP1.2.1** e na **QP1.2.2**. Em seguida, o dicionário léxico foi avaliado utilizando uma base de dados do Stack Overflow para avaliação da acurácia e discussão da **QP1.2.3**.

QP1.2.1. Os participantes concordam com a classificação das polaridades da ferramenta Sentistrength-SE?

Os participantes que iniciaram com a tarefa de validar a classificação do dicionário da ferramenta concordaram em média com 55,51% das palavras, 59,72% dos emoticons e 71,88% das expressões idiomáticas. Por outro lado, os participantes que iniciaram com a tarefa de classificação concordaram com 56,94% das palavras, 63,54% dos emoticons e 69,79% das expressões idiomáticas. Assim, as expressões idiomáticas obtiveram mais concordância tanto na fase de classificação quanto na fase de validação. Todos os elementos léxicos obtiveram concordância acima de 50%.

QP1.2.2. Qual o comportamento dos resultados e a concordância da classificação do dicionário da ferramenta nos grupos de participantes selecionados pela experiência em desenvolvimento de software, fluência no idioma do dicionário e habilidade em análise de sentimentos?

Neste estudo, os grupos foram separados considerando-se o formulário de caracterização com as tarefas atribuídas aos participantes de classificação e validação. Todos os resultados dos grupos apresentados no estudo estão representados nas Tabelas 3.7, 3.8 e 3.9.

O grupo que obteve maior concordância com a classificação das palavras apresentadas na ferramenta foi o que realizou a tarefa de validação e possui Experiência de Desenvolvimento de Software grau 1 com 74,25% de acerto. O segundo grupo que alcançou o melhor resultado foi o que realizou a tarefa de classificação e possui maior experiência

Tabela 3.7: Concordância na classificação de palavras por domínio e grau.

	Tarefa	**Grau de experiência**				
		1	**2**	**3**	**4**	**5**
Desenvol. de software	Classificação	43,61%	-	42,83%	36,26%	43,69%
	Validação	***74,25%***	-	56,31%	58,95%	51,67%
Fluência no idioma	Classificação	37,04%	34%	37,15%	40,37%	47,59%
	Validação	***28,57%***	32,54%	***65,86%***	55,39%	56,08%
Experiência no tema	Classificação	44,52%	39,65%	36,74%	37,5%	***66,67%***
	Validação	54,18%	59,46%	59,65%	42,59%	35,71%

com análise de sentimentos. O terceiro grupo que melhor realizou a classificação foi o que possuía fluência intermediária no idioma (Grau 3) e realizou a validação. A maior discordância no experimento foi encontrado no grupo de validação que informou menor fluência no idioma (Grau 1), conforme dados apresentados na Tabela 3.7.

Tabela 3.8: Concordância nos emoticons por domínio e grau.

	Tarefa	**Grau de experiência**				
		1	**2**	**3**	**4**	**5**
Desenvol. de software	Classificação	30,56%	-	***72,22%***	40%	40,48%
	Validação	***100%***	-	67,59%	53,89%	59,13%
Fluência no idioma	Classificação	25%	***88,89%***	46,53%	35,18%	50,60%
	Validação	0%	***50%***	56,25%	68,98%	63,69%
Experiência no tema	Classificação	53,07%	37,5%	41,67%	***50%***	***50%***
	Validação	71,05%	50,69%	63,10%	***25%***	66,67%

Na análise dos emoticons, os três grupos que obtiveram maiores concordâncias com a classificação da ferramenta foram, respectivamente, os grupos que possuíram experiência de desenvolvimento de software grau 1 na tarefa de validação com 100%, fluência no idioma grau 02 na tarefa de classificação com 88,89% e experiência em desenvolvimento de software grau 03 na tarefa de classificação com 72,22%. A maior discordância com a classificação foi encontrada nos grupos de fluência de idioma na validação grau 1 com 0% e classificação de desenvolvimento de software grau 1 com 30,56%.

Ao avaliar as expressões idiomáticas, o grupo com tarefa de validação com fluência no idioma grau 1 obteve maior concordância com a classificação, seguido do grupo com tarefa de validação com experiência em desenvolvimento de software grau 1 com 91,67% de concordância. Os grupos que discordaram das classificações das expressões idiomáticas foram os que possuíam maior experiência em análise de sentimento e realizaram a tarefa de validação, com 12,50%, e classificação, com 25%.

QP1.2.3. O Dicionário léxico gerado pela classificação dos participantes melhora a acurácia na análise de sentimentos de textos reais?

O dicionário léxico com as polaridades obtidas foi submetido à validação utilizando a base de dados de textos dos posts do *Stack Overflow*. Cada palavra, emoticon e expressão

Tabela 3.9: Concordância nas expressões idiomáticas por domínio e grau.

	Tarefa	**Grau de experiência**				
		1	**2**	**3**	**4**	**5**
Desenvol. de software	Classificação	58,33%	-	50%	50%	45,24%
	Validação	***91,67%***	-	77,78%	75%	61,9%
Fluência no idioma	Classificação	50%	33,33%	43,75%	44,44%	60,71%
	Validação	***100%***	41,67%	85.42%	66,67%	67,86%
Experiência no tema	Classificação	46,05%	47,92%	55,36%	***25%***	50%
	Validação	67,11%	66,67%	89,29%	***12,50%***	50%

idiomática foi considerada de acordo com a polaridade mais votada. É importante ressaltar que, nos casos em que os resultados apresentados obtiveram polaridades distintas das selecionadas nos formulários, houve a escolha da polaridade encontrada na base da ferramenta previamente extraída. O novo dicionário léxico apresenta 79% de Acurácia e Precisão, com 78% de *Recall* e F_1-*score*. Isto representa uma acurácia e precisão menor que os resultados apresentados pelo léxico do *SentiStrength-ES*, que utiliza uma escala maior de polaridade do que a utilizada no Novo Dicionário. Para o *SentiStrength-ES* Alterado houve resultados melhores como o aumento de 6% da acurácia, 3% da precisão na classificação, 7% no *recall* e F_1-*score*, conforme apresentado na Tabela 3.10.

Tabela 3.10: Resultados obtidos na validação do dicionário léxico

Léxico	**Accuracy**	**Precision**	**Recall**	**F_1-score**
SentiStrength-ES	81%	81%	82%	81%
SentiStrength-ES Alterado	73%	76 %	71 %	71%
Novo Dicionário	***79%***	***79%***	***78%***	***78%***

Os valores das métricas ficaram próximos dos originais, variando entre 2% e 4%, o que pode ser considerado valores próximos. Em relação com o dicionário da *SentiStrengh* Alterado houve um aumento significativo de acurácia, precisão, *recall* e F_1-*score*. A validação e classificação dos termos trouxe um maior ganho para o Novo Dicionário, sendo o SentiStrength-ES Alterado para operar na mesma escala. A diferença da nova avaliação das palavras, emoticons e expressões idiomáticas foi positiva para ganho na acurácia, precisão e outros indicadores, conforme apresentado na Tabela 3.10.

3.2.3 Ameaças à validade

Validade Interna. Os participantes que realizaram a tarefa de validação antes da tarefa de classificação possam ter sido influenciados pelas pontuações pré-existentes do léxico. Essa influência poderia afetar a independência de suas classificações subsequentes. No entanto, essa ameaça foi mitigada por meio de um design experimental que distribuiu equitativamente os participantes entre as duas ordens de tarefas (validação e classificação). Além disso, o estudo piloto desempenhou um papel crucial ao identificar e controlar

possíveis vieses relacionados à ordem das tarefas. Dessa forma, qualquer diferença observada nos resultados pode ser atribuída ao conteúdo das tarefas em si, e não à sequência em que foram realizadas, fortalecendo a validade interna do experimento.

Validade Externa. O léxico foi validado apenas com base em textos do *Stack Overflow*, o que pode restringir a generalização dos resultados para outros contextos. No entanto, essa preocupação pode ser refutada, considerando que a base de dados utilizada é uma gold standard reconhecida em pesquisas sobre análise de sentimentos na área de desenvolvimento de software. Além disso, o dicionário léxico validado abrange uma diversidade significativa de tokens, incluindo palavras, emoticons e expressões idiomáticas, o que sugere que ele pode ser aplicável a outros contextos similares.

Validade de Construto. As métricas utilizadas no experimento podem não refletir adequadamente os resultados se os participantes não compreenderem bem as tarefas de classificação e validação. No entanto, essa ameaça é mitigada pela escolha de métricas padrão amplamente aceitas na área de análise de sentimentos, como Acurácia, Precisão, *Recall* e *F1-score*.

Validade de Conclusão. Uma interpretação inadequada dos resultados estatísticos, como afirmar que um dicionário é superior ao outro sem considerar variações no tamanho da amostra ou diferenças nos tipos de dados analisados. Essa ameaça é mitigada por uma análise cuidadosa das métricas (acurácia, precisão, *Recall*, *F1-score*) e pela consideração das limitações do estudo, o que fortalece a robustez das conclusões obtidas no experimento.

3.2.4 Conclusão

O presente estudo validou as polaridades e pontuações do dicionário léxico da ferramenta *Sentistrength-SE*. A validação foi realizada através de um experimento com 48 participantes, que possuem diferentes níveis de experiência em desenvolvimento de software, fluência no idioma e conhecimento no tema de análise de sentimentos.

A análise dos artefatos do repositório é importante para a evolução e manutenibilidade do projeto. Em (BOECHAT et al., 2019a), a análise utilizou a classificação das polaridades das *issues* reabertas com apoio fundamental da ferramenta *SentiStrength-SE* para encontrar respostas sobre a previsão de reabertura das *issues* fechadas. Com um dicionário validado e ajustado, a análise de sentimentos dos dados pode trazer resultados mais precisos. O dicionário léxico de Engenharia de Software já existe, mas a construção em cima da validação de termos por desenvolvedores de software é fundamental para melhorar os resultados de classificação da ferramenta. Assim, os termos léxicos validados e construídos neste dicionário podem ajudar na classificação de sentimentos tanto de *issues*, como de outros artefatos relacionados com a manutenibilidade e evolução do repositório de *software*. A validação utilizando a base de dados do *Stack Overflow*, que é uma base relacionada com desenvolvimento de software, foi realizada neste estudo com 79% de acurácia, 78% de *recall*, 79% de precisão, e 78% de *f1 - score*. Com os números apresentados utilizando a base do *Stack Overflow*, o dicionário léxico deste estudo pode ser utilizado como base para analisar textos voltados para desenvolvimento de software,

devido a validação ser realizada diretamente por desenvolvedores que possuem experiências diversas na área. Assim, este dicionário auxilia na análise de outros elementos importantes para a evolução e manutenção de repositórios, como as *issues*, *pull-request*, *commits* e outros artefatos.

Para trabalhos futuros é esperado a expansão deste dicionário léxico com a validação de especialistas na área de análise de sentimentos. Uma outra direção a seguir é a classificação de diferentes bases de dados, contemplando um número maior de textos de *issues* e *commits*.

3.3 ANÁLISE DE SENTIMENTOS EM DISCUSSÕES DE ISSUES REABERTAS DO GITHUB

Uma atividade fundamental da fase de manutenção do software é compreender porque *issues* fechadas são reabertas (CAGLAYAN et al., 2012). As *issues* podem ser reabertas depois de serem fechadas devido ao fechamento incorreto, descoberta do problema real da *issue*, etc. As *issues* reabertas podem aumentar o custo de manutenção, degradar a qualidade geral do produto de software, reduzir a confiança dos usuários e trazer trabalho desnecessário para os desenvolvedores (PAN; MAO, 2014).

Nesse trabalho, optamos por investigar o comportamento das *issues* e *pull requests* que foram reabertas no GitHub. O GitHub é uma plataforma de hospedagem de código-fonte do sistema de controle de versão git, que contém mais de 36 milhões de usuários e 100 milhões de repositórios (GITHUB_INC., 2023a).

No GitHub, os colaboradores dos repositórios, opcionalmente, podem criar e participar de discussões de *issues* para coletar feedback dos usuários nas discussões sobre o projeto, adição de novas features, bugs e outras tarefas de manutenção. Eles podem ainda participar de discussões de *pull request*, que é um tipo de *issue*, para discutir e revisar alterações realizadas no código-fonte antes de serem incorporadas na *branch* principal (ORTU et al., 2016). Os comentários dessas discussões das *issues* postados pelos colaboradores não contêm apenas informações técnicas, mas também informações valiosas sobre sentimentos ou emoções (ORTU et al., 2015b). Os sentimentos podem ser classificados como positivo, negativo ou neutro, que estão associados as emoções como felicidade, tristeza, alegria, raiva, dentre outras (LIU, 2015).

O objetivo do trabalho consiste em investigar os sentimentos dos colaboradores durante as discussões das *issues* que ajudam a prever se uma *issue* fechada tem propensão de ser reaberta. As seguintes questões de pesquisa (QP) direcionam esta investigação:

QP2.1 *Existe algum indicativo de que uma issue não será reaberta se ela for fechada com um sentimento positivo?*

QP2.2 *É possível predizer se uma issue será reaberta quando o número de sentimentos negativos adicionados ao número de sentimentos neutros for maior que o número de sentimentos positivos presentes nas suas discussões?*

QP2.3 *Uma issue com discussões com sentimentos neutros após o fechamento indica que ela será reaberta?*

3.3.1 Metodologia

Inicialmente, foram selecionados os 90 repositórios listados no desafio *MSR Challenge Dataset* da conferênca *Mining Software Repositories*(MSR, 2014) do ano de 2014. Ao valida-los, observamos que oito repositórios não possuiam *issues* reabertas com discussões, e dois repositórios não mais encontravam-se disponíveis. Assim, a lista final incluiu os 80 repositórios disponíveis, que englobavam 506.227 *issues* abertas e fechadas, com uma média de aproximadamente 379 mil linhas de código (LOC), e média de 2.769 arquivos por repositório. As *issues* foram extraídas entre 4 de Junho à 24 Julho de 2019. Selecionamos apenas as *issues* que foram reabertas e que possuíam discussões, resultando em 12.996 issues no *dataset*. A Figura 3.2 apresenta um *workflow* com as etapas seguidas neste trabalho. Cada uma das etapas será discutida a seguir.

Os repositórios selecionados foram implementados nas seguintes linguagens: 8 em C, 8 em C++, 8 em C#, 1 em CSS, 3 em HTML, 7 em Java, 8 em JavaScript, 3 em Markdown, 5 em PHP, 10 em Python, 4 em R, 8 em Ruby e 7 em Scala. Todos os dados

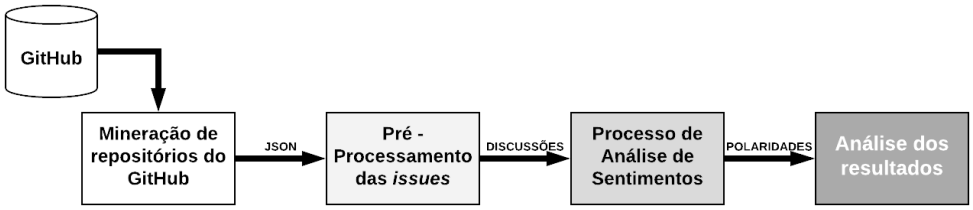


Figura 3.2: *Workflow* de captura e análise de dados.

dos repositórios utilizados neste projeto encontram-se disponíveis em (BOECHAT et al., 2019b).

3.3.1.1 Etapa 1 - Mineração de Repositórios do Github Os dados foram obtidos a partir de um script de mineração desenvolvido utilizando a biblioteca PyGitHub (JACQUES, 2020) para extração dos dados. Este script é responsável por recuperar todas as informações referentes as *issues*: comentários, horário de criação, usuários, reações e outras informações. Processamos todas as respostas das requisições ao GitHub no formato *JavaScript Object Notation* (JSON) e em seguida armazenamos todas as informações em um banco de dados não relacional, MongoDB[4]. O script encontra-se disponível no GitHub (JÚNIOR, 2019). A mineração das *issues* foi realizada em uma estação de trabalho com um processador Core i7-7500U, 8 GB RAM, SSD 240 GB, Sistema Operacional Win 10 x64.

3.3.1.2 Etapa 2 - Pré-processamento das *issues* A etapa de pré-processamento dos dados foi realizada através da limpeza dos textos do título, descrição e comentários

[4]https://www.mongodb.com/

das issues. A limpeza dos textos foi feita através do módulo *RE* (Python-Software-Foundation, 2023) do *Python*, com operações com expressões regulares para excluir trechos indesejados no texto, tais como URLs, código-fonte, trechos de código, erros de compilação, classes, interfaces, imagens, quebra de linhas, excesso de espaços em branco, respostas de comentário, frases de alertas sobre *warning* e exceções.

O pré-processamento considerou o truncamento de palavras, por exemplo *joy** para todas as palavras que começam com *joy*, ao invés de utilizar os processos de lematização e stemização. Não utilizamos a remoção de *stopwords*, pois pode ocorrer de remover palavras que distorcem o verdadeiro sentimento da frase, por exemplo os advérbios de negação ou intensidade podem alterar o sentido da frase (THELWALL et al., 2010).

3.3.1.3 Etapa 3 - Processo de Análise de Sentimentos Durante o processo de análise de sentimentos das discussões das *issues* reabertas, e que tenham pelo menos dois comentários, foi utilizada a versão Java da ferramenta *SentiStrength-SE* (ISLAM; ZIBRAN, 2017; SENTISTRENGTH-SE, 2017) para classificar as polaridades dos textos. A polaridade pode ser *negativa*, *neutra* ou *positiva*. A *SentiStrength-SE* uma versão da ferramenta *SentiStrength* (THELWALL et al., 2010) desenvolvida para aplicar análise de sentimentos baseada em um dicionário léxico no domínio de Engenharia de Software; esse dicionário é um arquivo léxico de emoções, onde palavras com sentimentos negativos estão previamente classificadas com pontuações entre -5 e -1, e palavras com sentimentos positivos possuem pontuações entre +1 e +5. (ISLAM; ZIBRAN, 2017; SENTISTRENGTH-SE, 2017). A ferramenta *SentiStrength-SE* divide a sentença em *tokens* e para cada palavra (*token*) que transmite uma emoção é atribuída uma pontuação. Após pontuar todas as palavras, a ferramenta retorna a pontuação máxima dos sentimentos negativos e a pontuação máxima dos sentimentos positivos. O sentimento final do texto é obtido através da soma da pontuação positiva com a pontuação negativa. Para valores menores que zero, o texto é classificado como negativo; para valores iguais a zero, o texto é classificado como neutro; e para valores maiores que zero, o texto é classificado como positivo.

3.3.1.4 Etapa 4 - Análise dos Resultados A ordem cronológica dos eventos de uma *issue* foi considerada na análise dos resultados. Os eventos do ciclo de vida da issue são ilustrados no diagrama de estado da figura 1.1, que exibe os possíveis status da *issue*: *Aberta*(*Open*) e *Fechada*(*Closed*), bem como as transições responsáveis pelas mudanças de status *Fechada*(*Closed*) e *Reaberta*(*Reopened*). Durante o ciclo de vida de uma *issue*, os colaboradores do repositório e/ou usuários do GitHub podem colabarorar com o repositório através de comentários nas *issues*. Nos comentários, os colaboradores podem expressar seus sentimentos por meio de textos, emoticons e emojis. Neste estudo, foram analisadas as polaridades dos sentimentos encontradas nos comentários no período de tempo entre os status possíveis da *issue*. A Figura 3.3 representa a linha do tempo de uma *issue* reaberta, ilustrando as discussões ao longo de suas diferentes fases.

A linha do tempo mostra o início da *issue* no status *Aberta*, as discussões dos colaboradores entre a abertura e o fechamento, a mudança de status para *Fechada*, as discussões

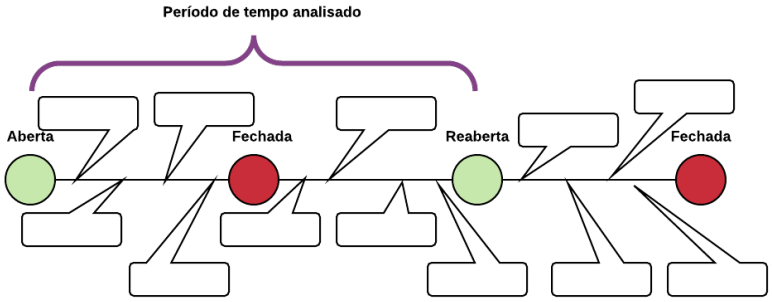


Figura 3.3: Linha do tempo dos eventos da *issue*.

dos colaboradores entre o fechamento e a reabertura da *issue*, a mudança de status para *Reaberta*, as discussões dos colaboradores entre a reabertura e o fechamento da *issue*, e por fim a mudança de status para *Fechada*.

3.3.2 Discussão dos Resultados

Foram analisadas 12.996 *issues* reabertas, que continham discussões, de 80 repositórios do GitHub. A seguir, discutiremos os resultados à luz das QP propostas para este estudo.

QP2.1 - Existe algum indicativo de que uma *issue* não será reaberta se ela for fechada com um sentimento positivo?

Para calcular os resultados, verificou-se o último sentimento antes do fechamento da *issue*. Assim, caso a *issue* possua pelo menos um fechamento, então é contabilizado. O ponto central da análise de dados foi encontrar pelo menos um caso durante o tempo de vida de uma *issue* onde existiu fechamento com sentimento positivo, e reabertura logo em seguida. Verificamos que 2.925 (22,51%) *issues* reabertas foram fechadas com sentimento positivo e em seguida foram reabertas. A Figura 3.4 apresenta a distribuição de issues fechadas com sentimentos positivos com mediana de 14 *issues*. Isso indica que se a *issue* fechar com sentimento positivo, não há garantia de que ela continuará fechada. Entretanto, observa-se uma menor tendência de reaberta dessas *issues*. 10.071 (77,49%) *issues* foram fechadas com sentimento negativo ou neutro, e em seguida foram reabertas.

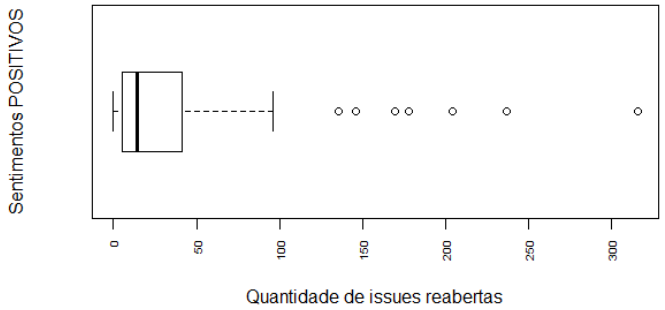


Figura 3.4: Distribuição de *issues* fechadas com sentimentos positivos.

Dentre os repositórios analisados, apenas o `Impress.js` teve 13 de 18 (72.22%) *issues*

reabertas que foram fechadas com sentimento positivo e em seguida foram reabertas. Em contraponto, nenhum caso foi observado nos repositórios `ActionBarSherlock, Kestrel` e `Storm` (BOECHAT et al., 2019b).

QP2.2 - É possível predizer se uma *issue* será reaberta quando o número de sentimentos negativos adicionados ao número de sentimentos neutros for maior que o número de sentimentos positivos presentes nas suas discussões?

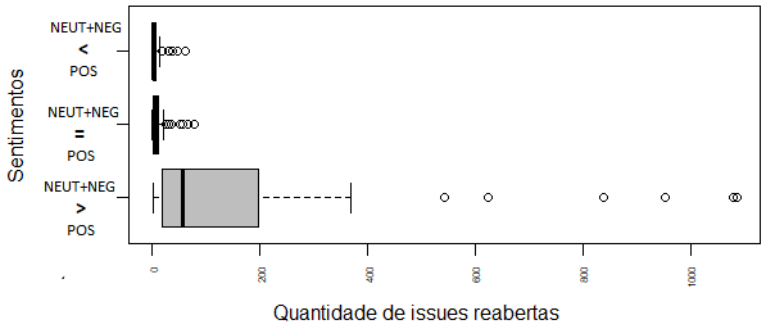


Figura 3.5: Distribuição de *issues* reabertas em função da polaridade de sentimentos.

Verificamos que 11.722 (90.20%) *issues* foram reabertas com a quantidade de sentimentos negativos ou neutros maior que a quantidade de sentimentos positivos. Encontramos 810 (6,23%) *issues* reabertas com quantidade de sentimentos negativos ou neutros igual a quantidade de sentimentos positivos. Em 464 (3,57%) *issues*, a quantidade de sentimentos positivos foi maior que a quantidade de sentimentos negativos ou neutros. A Figura 3.5 apresenta a distribuição de *issues* reabertas com mediana de de 57 *issues* para (Neutros+Negativos) > Positivos, em função da polaridade de sentimentos. Nos repositórios `ActionBarSherlock, Storm, Beanstalkd, Kestrel, Flockdb, Ravendb` e `CCV`, 100% das *issues* foram reabertas com sentimentos negativos ou neutros maior que a quantidade de sentimentos positivos. Os dados permitiram observar que, se houve uma tendência de discussão negativa ou neutra durante o período de atividade da *issue*, então a propensão de reabertura é maior.

QP2.3 - Uma *issue* com discussões com sentimentos neutros após o fechamento indica que ela será reaberta?

Encontramos 5.547 (42,68%) *issues* reabertas que possuem comentários com sentimentos neutros entre o fechamento da *issue* e a sua reabertura. A Figura 3.6 apresenta a distribuição de *issues* que possuem sentimentos neutros entre fechamento e reabertura. 2.079 (16,00%) *issues* reabertas possuem comentários com sentimentos negativos entre o fechamento da *issue* e a sua reabertura. A mediana da Figura 3.6 foi de 24 *issues*. A *issue* pode ser reaberta quando há sentimentos neutros relacionados, mas existe uma propensão maior de reabertura de *issues* com sentimentos neutros do que com sentimentos negativos. Algumas *issues* observadas foram reabertas sem discussões, portanto não foram contabilizadas.

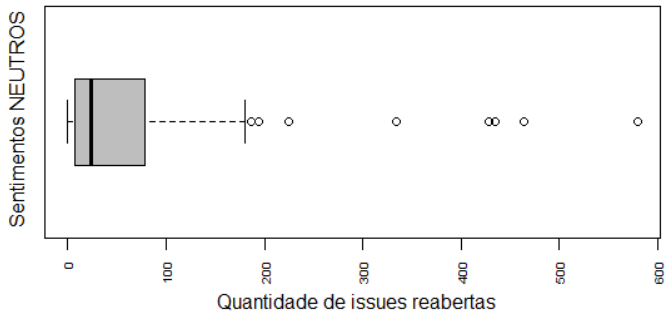


Figura 3.6: Distribuição de *issues* que possuem sentimentos neutros entre fechamento e reabertura.

3.3.3 Ameaças à validade

Validade Interna. A mineração dos dados foi realizada em um período de tempo que começou em 4 de Junho à 24 Julho de 2019. Assim, repositórios ativos podem receber atualizações com novas *issues*, comentários ou mudança de estado (aberta/fechada), alterando a quantidade de dados da nossa base. Como a lista escolhida foi do desafio *MSR Challenge Dataset*, ocorrido em 2014, dentre os repositórios minerados também encontramos alguns que estão arquivados ou são inativos. Entretanto, o cenário utilizado no estudo possui uma quantidade significativa de *issues*.

Validade Externa. Alguns repositórios possuem características que não favorecem o estudo, possuindo poucas *issues*. Resta, portanto, um número menor de *issues* reabertas para serem analisadas. Entretanto, uma menor quantidade propicia uma análise mais criteriosa e assertiva, por serem repositórios verificados e utilizados como referência nos estudos presentes na literatura.

Validade de Construto. Ao passo que havendo uma base de dados com múltiplos domínios distintos, os repositórios possuem tamanhos diferentes entre si. Há uma tendência de alguns repositórios possuírem maior quantidade de dados que os demais. Dessa forma, certos domínios impactam significativamente no resultado. Assim, para encontrar resultados mais homogêneos é preciso expandir a base de dados minerados do GitHub para integrar mais elementos relevantes ao estudo.

3.3.4 Conclusão

O presente estudo analisou o sentimento de cerca de 13 mil *issues* reabertas (que incluíram cerca de 153 mil comentários) em 80 repositórios de projetos hospedados no GitHub, buscando respostas sobre a previsão de reabertura de *issues* através da análise dos sentimentos presentes nas discussões dos colaboradores. A ferramenta SentiStrength foi fundamental para a classificação do grau de polaridade dos textos encontrados.

O impacto dos sentimentos nas discussões em alguns casos pode afetar ou não diretamente o ciclo de vida da *issue*. Identificado na primeira questão de pesquisa, a reabertura de uma *issue* com sentimento positivo existe, mas ela é menos recorrente do que com ou-

tros sentimentos. Também são questionados sobre a tendência da discussão durante a reabertura, que na segunda questão de pesquisa é observado que discussões negativas e neutras podem influenciar diretamente na reabertura de uma *issue*. Durante o período de fechamento e reabertura de uma *issue*, existe um indicativo maior de reabertura se as discussões forem realizadas com sentimento neutro do que se forem realizadas com sentimento negativo. O estudo mostra a importância da análise de sentimentos para o gerenciamento de repositórios de software. As informações extraídas são indicativos que podem ajudar no gerenciamento do projeto, uma vez que *issues* fechadas podem ser identificados para posterior reabetura.

Como trabalhos futuros, planejamos correlacionar linguagens de programação, tipo de domínio e tipos de colaboradores com a análise de sentimentos. Também existe a necessidade de expandir a base de dados com uma maior amostra de repositórios.

3.4 ANÁLISE DE SENTIMENTOS EM DISCUSSÕES DE ISSUES COM E SEM REABERTAS DOS REPOSITÓRIOS DO GITHUB

Este estudo busca aprimorar a análise de sentimentos em *issues* de repositórios do GitHub, com foco em entender a influência dos sentimentos nas reaberturas dessas *issues*. A investigação utiliza dados provenientes dos repositórios listados no desafio *MSR Challenge Dataset* da conferênca MSR no ano de 2014, conforme discutido na seção 3.3.e visa responder à questão principal: *QP2 Como os sentimentos expressos nas discussões de uma issue afetam sua reabertura?*

Para explorar essa questão, o estudo se apoia em um conjunto de perguntas específicas que orientam a análise em diferentes momentos do ciclo de vida das *issues*. Primeiro, investigamos se há indicativos de que uma *issue* fechada será reaberta com base nos sentimentos do primeiro comentário após sua abertura, avaliando a pontuação negativa e positiva desse comentário (QP2.4). Em seguida, a pesquisa examina se o último comentário antes do fechamento inicial da *issue* pode fornecer sinais de uma possível reabertura, considerando as mesmas métricas de sentimentos (QP2.5).

Além disso, o estudo busca determinar se é possível prever a reabertura de uma *issue* por meio da análise dos sentimentos presentes nas discussões que ocorrem entre os eventos de abertura e o primeiro fechamento (QP2.6). Por fim, a investigação se aprofunda ao analisar se existe uma correlação forte entre os sentimentos expressos ao longo das discussões e a probabilidade de reabertura (QP2.7).

Essas perguntas orientam a análise detalhada da influência dos sentimentos nas decisões de reabertura de *issues*, proporcionando uma nova perspectiva sobre como as interações emocionais nos repositórios de software podem impactar o ciclo de vida de uma *issue*.

Na próxima seção, apresentamos detalhes sobre a metodologia empregada neste estudo.

3.4.1 Metodologia

O estudo foi realizado em 6 etapas: seleção dos repositórios, extração das issues, pré-processamento do texto, análise dos sentimentos, subamostragem e análise das issues. A figura 3.7 apresenta as etapas do estudo e cada etapa é descrita a seguir.

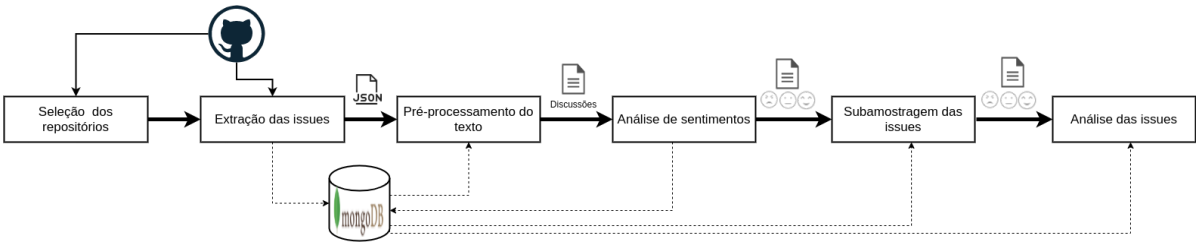


Figura 3.7: Etapas do estudo.

3.4.1.1 Seleção dos repositórios

Inicialmente, foram selecionados os 90 repositórios listados no desafio *MSR Challenge Dataset* da conferênca *Mining Software Repositories* (MSR)[5] do ano de 2014. Os repositórios representavam os top 10 repositórios favoritos dos usuários (*stars*) das principais linguagens de programação do GitHub. Essa abordagem de seleção foi adotada para assegurar uma diversidade de projetos e linguagens, visando aumentar a representatividade dos resultados.

Durante a fase de validação dos repositórios, encontramos alguns repositórios com problemas, como acesso desativado, indisponibilidade no GitHub ou a ausência de *issues* reabertas. Essas questões foram identificadas e tratadas, levando à exclusão de alguns repositórios do estudo. Destacamos que o repositório *Bukkit/Craft-Bukkit* teve seu acesso público desativado, o *facebook/php-sdk* não está mais disponível no GitHub e seis repositórios não apresentaram *issues* reabertas: *clojure/clojure*, *codeguy/Slim*, *joshuaclayton/ blueprint-css*, *mavam/stat-cookbook*, *TTimo/doom3.gpl*, *twitter-archive/gizzard (twitter/gizzard)*. Além disso, os repositórios *vinc/vinc.cc* e *xphere-forks/symfony* não tinham *issues* disponíveis para análise. Adicionalmente, verificamos que o repositório *twitter-archive/kestrel* não possui *issues* reabertas com comentários.

Além disso, observamos que alguns repositórios foram transferidos para novas localizações, e, como resultado, incluímos mais cinco repositórios em nossa análise. O repositório *mxcl/homebrew (Homebrew/legacy-homebrew)* foi dividido em dois projetos distintos: *Homebrew/homebrew-core* e *Home-brew/ brew*. O repositório *joyent/libuv* foi realocado para *libuv/libuv*, o repositório *SamSaffron/MiniProfiler* foi migrado para o *MiniProfiler/ rack-mini-profiler*, e *joyent/node (node-v0.x-archive)* foi transferido para *nodejs/node*.

3.4.1.2 Extração das issues

[5]http://ghtorrent.org/msr14.html

Nessa etapa utilizamos a ferramenta gFetcher (JÚNIOR, 2019), conforme descrito na seção 3.3.1.1, para extrair issues e pull-requests de repositórios públicos no GitHub. Foram coletados os seguintes metadados das issues: ID, título, corpo da issue, autor, status, data de criação, labels, reações, metadados os eventos das issues, eventos, i.e atividades que podem ser realizadas em uma issue como, abrir, fechar, reabrir, adicionar ou remover uma label, data de criação do evento, autor do evento, e metadados dos comentários: comentário, autor do comentário, data de criação e reações. As issues foram armazenadas no banco de dados MongoDB no formato JSON (*JavaScript Object Notation*).

A base de dados MSR14 engloba 490.531 issues fechadas e reabertas, com uma média de aproximadamente 41 mil linhas de código (LOC), 9.524 de *Stargazers*, 4.725 de *commits* e 2.859 de *Forks*. A extração dos dados dos repositórios ocorreu entre os dias 01 e 30 de agosto de 2020. Para este estudo, selecionamos as *issues* que possuem pelo menos um fechamento e que possuíam discussões entre a abertura e o primeiro fechamento.

Os repositórios selecionados abrangem diversas linguagens de programação: 8 em C, 8 em C#, 8 em C++, 1 em CSS, 2 em HTML, 6 em Java, 8 em JavaScript, 5 em PHP, 9 em Python, 4 em R, 8 em Ruby, 5 em Scala e 3 não identificaram a linguagem utilizada. Os dados dos repositórios utilizados neste projeto estão disponíveis em <bit.ly/MSR2014DB> e <https://bit.ly/msr2014info>

3.4.1.3 Pré-processamento de texto

Adotamos os passos de pré-processamento de texto detalhados na Seção 3.3.1.2. Além disso, nesta etapa, incorporamos a conversão de emoji em texto com o suporte da biblioteca de Emoji (VICENZI, 2018). Essa biblioteca decodifica valores de emoji unicode em texto, utilizando a base de dados da biblioteca oficial de emoji do GitHub para Ruby, chamada *gemoji* (GITHUB, 2023).

3.4.1.4 Análise de Sentimentos

Nesta etapa, conduzimos a análise de sentimento nos textos da discussão de issues que possuem pelo menos dois comentários. Utilizamos a versão Java da ferramenta SentiStrength-SE (SENTISTRENGTH-SE, 2017), que conta um léxico de sentimentos adaptado para a área de Engenharia de Software. Essa ferramenta foi empregada para calcular as pontuações negativas e positivas dos textos. O sentimento final de cada texto é determinado adicionando a pontuação positiva à pontuação negativa, conforme a equação (3.1).

$$\text{Sentimento}(x) = \begin{cases} \text{Negativo}, & \text{se PP(x) + PN(x)} < 0 \\ \text{Positivo}, & \text{se PP(x) + PN(x)} > 0 \\ \text{Neutro}, & \text{caso contrário}, \end{cases} \tag{3.1}$$

onde, PP é a pontuação positiva do texto (x) e PN é a pontuação negativa do texto (x). O texto é classificado como negativo quando o resultado da equação PP(x) + PN(x)

retorna valores menores que zero, positivo para valores maiores que zero e neutro para valores iguais a zero.

Ao contrário do estudo apresentado na seção 3.3, as pontuações negativa e positiva, assim como os sentimentos de cada issue e seus comentários são adicionados no banco de dados.

3.4.1.5 Subamostragem das issues

Nesta etapa, realizamos o balanceamentos entre issues com uma reabertura, issues com duas ou mais reabertas e issues que não foram reabertas. Observa-se que o número de issues sem reabertura é substancialmente superior ao das issues reabertas, acarretando, assim, um desequilíbrio nos dados. Em termos mais simples, a classe majoritária ocorre em frequência significativamente maior em comparação com as classes minoritárias (MOHAMED et al., 2018). Esse desequilíbrio nas classes pode impactar a validade estatística dos resultados obtidos.

Para lidar com essa questão, optamos pela técnica de subamostragem, conhecida como *Undersampling*, que consiste na remoção aleatória de instâncias das classes majoritárias. Isso proporciona uma representação equilibrada entre as classes, contribuindo para resultados mais robustos e confiáveis em nossas análises.

3.4.1.6 Análise das issues

Nesta etapa, consideramos a ordem cronológica dos eventos de cada issue reaberta na análise dos resultados, conforme ilustrado na figura 3.8. A análise das discussões de issues será realizada nos seguintes pontos da linha do tempo do issue: 1)último comentário antes do fechamento (evento *Closed*), 2) Primeiro comentário após da abertura (evento *Open*), 3) comentários entre a abertura e o primeiro fechamento (eventos *Open* e *Closed*).

Figura 3.8: Representação da linha do tempo de uma *issue* reaberta

O estudo será avaliado por meio do cálculo de métricas aplicadas aos comentários das issues entre os eventos de abertura *Open* e o primeiro evento de fechamento *Closed*. As métricas são descritas a seguir:

- Primeiro Comentário após a abertura da issue
  - N_{PC}: Pontuação negativa do primeiro comentário após a abertura da issue (evento *Opened*).
  - P_{PC}: Pontuação positiva do primeiro comentário após a abertura da issue (evento *Opened*)

  - SP_{PC}: Valor da intensidade do sentimento, calculado subtraindo a pontuação positiva (P_{PC}) do valor absoluto da pontuação negativa (N_{PC}) do primeiro comentário.

$$SP_{PC} = P_{PC} - |N_{PC}| \tag{3.2}$$

  - S_{PC}: Sentimento do primeiro comentário após a abertura da issue (evento *Opened*), calculado com base na intensidade do sentimento SP_{PC}, conforme a equação (3.3):

$$S_{PC} = \begin{cases} \text{Positivo}, & \text{se } S_{PC} > 0 \\ \text{Negativo}, & \text{se } S_{PC} < 0 \\ \text{Neutro}, & \text{caso contrário.} \end{cases} \tag{3.3}$$

  onde SP_{PC} com valor positivo indica um sentimento positivo (PO), SP_{PC} com valor negativo indica um sentimento negativo (NG) e SP_{PC} com valor igual a zero indica um sentimento neuto (NE), significa que o texto não expressa sentimento positivo ou negativo.

- Último comentário antes do fechamento da issue

  - N: Pontuação negativa do último comentário antes do primeiro fechamento (evento *Closed*)
  - P: Pontuação positiva do último comentário antes do primeiro fechamento (evento *Closed*)
  - SP: Valor da intensidade do sentimento, calculado subtraindo a pontuação positiva (P) do valor absoluto da pontuação negativa (N) do último comentário, conforme a equação (3.4):

$$SP = \mathrm{P} - |\mathrm{N}| \tag{3.4}$$

  - S: Sentimento do último comentário antes do primeiro fechamento (evento *Closed*), calculado com base na intensidade do sentimento SP, conforme a equação (3.5):

$$S = \begin{cases} \text{Positivo}, & \text{se } SP > 0 \\ \text{Negativo}, & \text{se } SP < 0 \\ \text{Neutro}, & \text{caso contrário.} \end{cases} \tag{3.5}$$

  onde SP com valor positivo indica um sentimento positivo (PO), SP com valor negativo indica um sentimento negativo (NG) e SP com valor igual a zero indica um sentimento neuto (NE), significa que o texto não expressa sentimento positivo ou negativo.

- Comentários entre a abertura e o fechamento da issue

– NM: Média das pontuações negativas dos comentários entre os eventos (*Opened* e *Closed*) ou dos comentários entre os eventos *Closed* e *Reopened*, conforme a equação (3.6):

$$NM = \frac{\sum_{i=1}^{nc} N_i}{nc} \tag{3.6}$$

onde N_i é a pontuação negativa do comentário i e nc é número total de comentários da issue entre os eventos.

– PM: Média das pontuações positivas dos comentários entre os eventos (*Opened* e *Closed*) ou dos comentários entre os eventos *Closed* e *Reopened*, conforme a equação (3.7):

$$PM = \frac{\sum_{i=1}^{nc} P_i}{nc} \tag{3.7}$$

onde P_i é a pontuação positiva do comentário i e nc é número total de comentários da issue entre os eventos.

– DCN: Densidade dos comentários com sentimentos negativos entre os eventos (*Opened* e *Closed*) ou dos comentários entre os eventos *Closed* e *Reopened*, conforme a equação (3.8):

$$DCN = \frac{nn}{nc} \tag{3.8}$$

onde nn é número total de comentários classificados como negativo e nc é número total de comentários da issue entre os eventos.

– DCP: Densidade dos comentários com sentimentos positivos entre os eventos (*Opened* e *Closed*) ou dos comentários entre os eventos *Closed* e *Reopened*, conforme a equação (3.9):

$$\text{DCP} = \frac{np}{nc} \tag{3.9}$$

onde np é número total de comentários classificados como positivo e nc é número total de comentários da issue entre os eventos.

3.4.1.6.1 Teste de Normalidade - Shapiro-Wilk

Para avaliar a normalidade dos dados das métricas discutidas anteriormente, considerando diferentes grupos de issues (sem reaberturas, com uma reabertura e com duas ou mais reaberturas), aplicaremos o teste de normalidade Shapiro-Wilk, conforme proposto por Shapiro e Wilk (SHAPIRO; WILK, 1965). As hipóteses nula (H_0) e alternativa (H_1) serão testadas para determinar se os dados seguem uma distribuição normal, utilizando um nível de confiança de $\alpha = 0.05$.

- Hipótese Nula (H_0) : Os dados da métrica (M) para o grupo de issues (G) seguem uma distribuição normal.

- Hipótese Alternativa (H_1) : Os dados da métrica (M) para o grupo de issues (G) não seguem uma distribuição normal.

No contexto deste teste, o conjunto de métricas (M) é composto por N, P, S, SP, N_{PC}, P_{PC}, S_{PC}, SP_{PC}, NM, PM, DCN e DCP. O conjunto de grupos (G) inclui issues sem reaberturas, issues com uma reabertura e issues com duas ou mais reaberturas. O nível de significância adotado é $\alpha = 0.05$. Se o valor de p do teste for maior que α, a hipótese nula será rejeitada, indicando que os dados não seguem uma distribuição normal.

3.4.1.6.2 Teste Wilcoxon

O teste não paramétrico de Wilcoxon (1945), também conhecido como teste dos postos sinalizados de Wilcoxon, será empregado para comparar os resultados das métricas entre os grupos de issues. As comparações serão realizadas de dois em dois, ou seja, entre os grupos de issues sem reaberturas (0 RE) e issues com uma reabertura (1 RE), e entre os grupos de issues sem reaberturas (0 RE) e issues com duas ou mais reaberturas (2+ RE), bem como entre os grupos de issues com uma reabertura (1 RE) e issues com duas ou mais reaberturas (2+ RE). O objetivo é avaliar se há diferenças estatisticamente significativas entre os resultados das métricas em cada par de grupos comparados.

Se o valorde p for menor que o nível de confiança ($\alpha = 0.05$), isso indica que existe uma diferença estatisticamente significativa entre os grupos em relação à métrica correspondente. Por outro lado, se o valor de p for maior que o nível de confiança, sugere que não há evidência estatística suficiente para concluir que existe uma diferença significativa entre os grupos para essa métrica.

3.4.1.6.3 Correlação de Spearman

A correlação de Spearman (1904) será empregada para avaliar a relação entre os resultados das métricas e a quantidade de reaberturas das issues. Quando o valor da correlação se aproxima de $+1$, indica uma relação positiva forte entre o resultado da métrica e a ocorrência de reaberturas de issues. Por outro lado, quando a correlação se aproxima de -1, sugere uma relação negativa forte entre o resultado da métrica e a reabertura de issues. Quando a correlação está próxima de zero, seja positiva ou negativa, isso implica em uma relação fraca entre o resultado da métrica e a ocorrência de reaberturas de issues. O nível de confiança adotado é $\alpha = 0.05$.

3.4.2 Resultados

Identificamos um total de 490.531 issues, distribuídas da seguinte forma: 475.935 issues sem reaberturas, 13.648 issues com uma reabertura e 949 issues com duas ou mais reaberturas. Nesse estudo, dividimos as issues em três grupos, cada um com 949 issues. Os grupos foram nomeados como: Grupo de issues sem reaberturas (*0 RE*), Grupo de issues com uma reabertura (*1 RE*) e Grupo de issues com duas ou mais reaberturas (*+2 RE*).

Inicialmente, realizamos uma análise dos sentimentos, das pontuações positivas e negativas no primeiro comentário após a abertura da issue. A Tabela 3.11 apresenta a frequência de ocorrência de issues nos grupos, considerando a pontuação negativa do primeiro comentário após a abertura da issue (N_{PC}).

Tabela 3.11: Frequência do N_{PC}

N_{PC}	0 RE	1 RE	2 RE
-1	734	698	702
-2	165	190	186
-3	43	50	47
-4	5	8	12
-5	1	2	1

Os resultados apresentados na Tabela 3.11 indicam que a maioria dos primeiros comentários não contém palavras negativas. No entanto, à medida que a pontuação negativa aumenta, observa-se uma diminuição na frequência, sugerindo uma possível relação entre a presença de palavras negativas e o aumento nas reaberturas de issues, especialmente nos grupos com uma reabertura (*1 RE*) e com duas ou mais reaberturas (*+2 RE*). Essa tendência é mais evidente à medida que a magnitude da pontuação negativa aumenta. A frequência de ocorrência de issues em relação à pontuação positiva no primeiro comentário após a abertura da issue é apresentada na Tabela 3.12.

Tabela 3.12: Frequência de P_{PC}

P_{PC}	0 RE	1 RE	2 RE
1	737	668	669
2	126	181	194
3	70	79	71
4	15	17	12
5	0	3	2

Os dados da Tabela 3.12 revelam que, em geral, a maioria dos primeiros comentários, em todos os grupos analisados, não contém palavras ou expressões positivas. No entanto, uma observação interessante surge ao examinarmos os grupos com uma reabertura e dois ou mais episódios de reabertura. Nessas instâncias, notamos que os colaboradores tendem a empregar mais palavras positivas à medida que a pontuação positiva aumenta. Essa correlação sugere que a positividade expressa nos primeiros comentários pode estar vinculada a uma maior incidência de reaberturas.

A frequência de ocorrência da intensidade dos sentimentos do primeiro comentário (S_{PC}) nos grupos sem reabertura (0 RE), uma reabertura (1 RE) e duas ou mais reaberturas (+2 RE) são apresentados na tabela 3.13, o sentimento é obtido a partir da soma das pontuações negativas e positivas para o primeiro comentário após a abertura da issue.

Tabela 3.13: Frequência de S_{PC}

SP_{PC}	0 RE	1 RE	2 RE
-4	0	1	0
-3	1	3	3
-2	25	26	31
-1	122	138	136
0	653	584	580
1	87	126	139
2	51	55	50
3	9	13	8
4	0	2	1

Identificamos na tabela 3.13 que para os valores entre -2 e -4, os grupos com uma reabertura (1 RE) e com duas ou mais reaberturas (+2 RE) possuem mais issues com sentimentos negativos em comparação ao grupo sem reabertura. Identificamos também que nos três grupos, a maioria dos primeiros comentários após a abertura das issues é neutra em termos de soma de pontuações negativas e positivas. Há uma diminuição na frequência à medida que a soma se torna mais negativa ou mais positiva. A partir da intensidade do sentimentos S_{PC} podemos obter o sentimento do primeiro comentário (Métrica S_{PC}) apresentada na tabela 3.13.

Tabela 3.14: Frequência da métrica S_{PC}

S	0 RE	1 RE	2 RE
Negativo	148	168	170
Positivo	147	196	198
Neutro	653	584	580

Ao analisar a Tabela 3.14, notamos que os grupos de issues com reaberturas (1 RE e +2 RE) apresentam maior ocorrência de sentimentos positivos e negativos no primeiro comentário em comparação ao grupo de issues sem reaberturas (0 RE). Em contrapartida, o grupo de issues sem reaberturas (0 RE) exibe uma predominância de sentimentos neutros no primeiro comentário em relação aos grupos com reabertura.

Além disso, examinamos os sentimentos e as pontuações positivas e negativas do último comentário antes do fechamento. A tabela 3.15 apresenta a frequência de ocorrência de issues para os grupos, considerando o resultado da métrica N para o último comentário antes do fechamento.

Tabela 3.15: Frequência da métrica N

	0 RE	1 RE	2 RE
-1	864	807	821
-2	42	86	72
-3	34	47	48
-4	8	8	7

Os resultados na Tabela 3.15 indicam que a maioria dos últimos comentários antes do fechamento das issues não contém palavras negativas. No entanto, à medida que a pontuação negativa aumenta, observamos uma diminuição na frequência, especialmente nos grupos com uma reabertura (*1 RE*) e com duas ou mais reaberturas (*+2 RE*). Esse padrão sugere que um aumento na pontuação negativa do último comentário pode estar associado a uma menor incidência de reaberturas. A tabela 3.16 apresenta a frequência de ocorrência de issues nos grupos para o resultado da métrica P para o último comentário antes do fechamento.

Tabela 3.16: Frequência da métrica P

	0 RE	1 RE	+2 RE
1	613	667	676
2	202	179	190
3	120	91	75
4	12	11	7
5	1	0	0

Os dados na Tabela 3.16 mostram que a maioria dos últimos comentários antes do fechamento das issues não contém palavras ou expressões positivas. Entretanto, em relação aos grupos com uma reabertura (*1 RE*) e com duas ou mais reaberturas (*+2 RE*), observamos que os colaboradores tendem a utilizar mais palavras positivas à medida que a pontuação positiva aumenta. Essa correlação sugere que a positividade expressa nos últimos comentários pode estar associada a uma maior incidência de reaberturas.

Na tabela 3.17 são apresentados as frequências de ocorrência de issues para grupos com e sem reaberturas, considerando a métrica de intensidade do sentimento (S), em relação ao último comentário antes do fechamento da *issue*.

Tabela 3.17: Frequência da métrica SP

SP	0 RE	1 RE	+2 RE
-3	2	7	5
-2	24	36	32
-1	36	60	59
0	580	595	617
1	188	164	166
2	105	79	63
3	12	7	6
4	1	0	0

Ao analisar a tabela 3.17, observamos que nos grupos com reaberturas (1 RE e +2 RE), as intensidades de sentimento com valores negativos entre -1 e -3 ocorrem com mais frequência do que no grupo sem reaberturas (0 RE). Por outro lado, o grupo sem reaberturas (0 RE) apresenta mais issues com intensidade de sentimento com valores positivos. Além disso, notamos que a maioria das issues nos três grupos possui intensidade de sentimento igual a zero, indicando um sentimento neutro. A partir dos resultados da métrica S, que nos permite identificar o sentimento do último comentário, podemos agora explorar a frequência de ocorrência de issues para diferentes grupos de sentimentos no último comentário antes do fechamento. A tabela 3.18 apresenta essa análise detalhada.

Tabela 3.18: Frequência da métrica S

Sentimento	0 RE	1 RE	+2 RE
Negative	62	103	96
Positive	306	250	235
Neutral	580	595	617

Observamos, a partir da tabela 3.18, que o grupo de problemas com uma reabertura (1 RE) possui mais ocorrências com sentimentos negativos no último comentário, seguido do grupo de problemas com duas ou mais reaberturas (+2 RE). Em relação aos sentimentos positivos no último comentário, o grupo de problemas sem reaberturas possui o maior número de ocorrências em comparação aos grupos com reabertura. Também notamos que a maioria dos problemas nos três grupos apresenta mais ocorrências com sentimentos neutros.

Além das análises do primeiro e último sentimento, conduzimos uma avaliação dos sentimentos presentes nos comentários entre a abertura e o fechamento da issue. Nossa atenção concentrou-se na pontuação negativa média (NM), na pontuação positiva média (PM), na densidade de comentários negativos (DCN) e na densidade de comentários positivos (DCP) em diferentes grupos de issues, com base no número de reaberturas. A distribuição das issues nos três grupos em relação à pontuação negativa média (NM) é apresentada na figura 3.9.

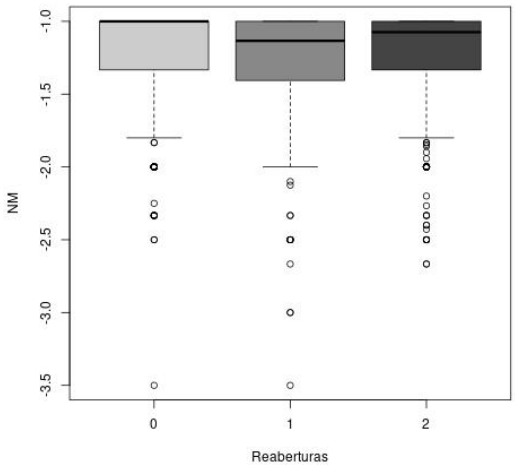


Figura 3.9: Distribuição da métrica NM

A figura 3.9 ilustra a distribuição da pontuação negativa média dos grupos de issues sem reaberturas (0 RE), issues com uma reabertura (1 RE) e issues com duas ou mais reaberturas (+2 RE). Notavelmente, o grupo com uma reabertura (1 RE) destaca-se ao apresentar os maiores valores de mediana e terceiro quartil (Q3), seguido pelo grupo com duas reaberturas ou mais. Os valores próximos a -1 indicam a presença de poucas palavras negativas nos comentários. É interessante observar que os três grupos compartilham valores mínimos e primeiros quartis iguais a -1, sugerindo consistência nesses aspectos entre as categorias de reaberturas.

Em relação à Pontuação Positiva Média (PM), conforme apresentado na figura 3.10, observa-se que todos os grupos exibem valores medianos semelhantes, indicando uma tendência geral de conter relativamente poucas palavras positivas nos comentários. É notável que o grupo de issues com uma reabertura (1 RE) se destaca ao apresentar um terceiro quartil (Q3) ligeiramente superior aos grupos sem reabertura (0 RE) e com duas ou mais reaberturas (+2 RE). Isso sugere que, apesar da tendência geral de poucas palavras positivas, as issues com uma reabertura possuem uma distribuição de pontuações mais elevada, especialmente em seu quartil superior.

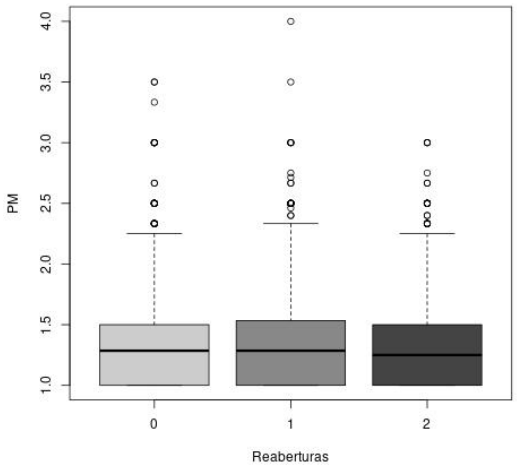


Figura 3.10: Distribuição da métrica PM

Realizamos uma análise detalhada da dinâmica dos sentimentos negativos e positivos nos comentários das issues entre a abertura e o fechamento, considerando o número de reaberturas. A figura 3.11 apresenta os resultados da Densidade de Comentários Negativos (DCN) nos grupos com e sem reaberturas. Em todos os grupos, tanto o valor mediano quanto o primeiro quartil (Q1) são iguais a 0. O grupo de issues com uma reabertura (1 RE) se destaca ao exibir o maior terceiro quartil (Q3), com um valor de 0,250, seguido pelo grupo de issues com duas ou mais reaberturas, que possui um Q3 de 0,200. Por último, o grupo de issues sem reaberturas apresenta um Q3 de 0,167. Esses dados indicam que as issues com reaberturas tendem a ter ligeiramente mais sentimentos negativos do que aquelas sem reaberturas.

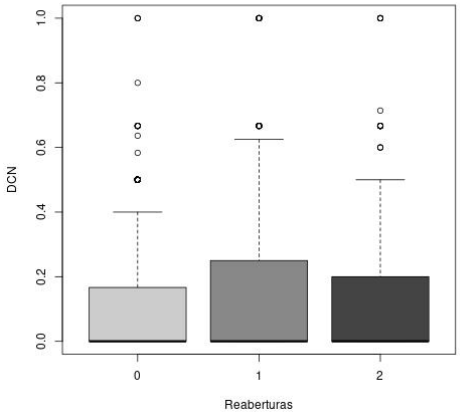


Figura 3.11: Distribuição da métrica DCN

Além disso, os resultados da Densidade de Comentários Positivos (DCP) nos grupos com e sem reaberturas são mostrados na figura 3.12. O grupo de issues com uma reabertura (1 RE) apresenta a mediana ligeiramente maior, com valor 0,1715686, seguido pelo

grupo de issues sem reaberturas (0 RE) com valor 0,1666667, e, por último, o grupo de issues com duas ou mais reaberturas (+2 RE). O grupo de issues sem reabertura (0 RE) possui um terceiro quartil (Q3) maior que os grupos de issues com reaberturas, indicando que até 75% das issues desse grupo possuem comentários positivos entre a abertura e fechamento da issue.

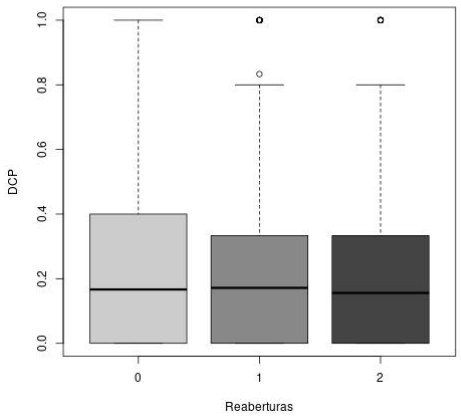


Figura 3.12: Distribuição da métrica DCP

3.4.3 Teste de normalidade Shapiro-Wilk

Realizamos o teste de normalidade Shapiro-Wilk para diferentes métricas em grupos de issues com 0 reabertura, 1 reabertura e 2 ou mais reaberturas. A tabela 3.19 mostra os valores de p obtidos para métricas referentes ao primeiro sentimento, último sentimento e sentimentos entre a abertura e o fechamento. Os resultados do teste de normalidade Shapiro-Wilk, apresentados na Tabela 3.19, revelam informações essenciais sobre a distribuição dos dados em relação às métricas nos diferentes grupos de issues (0 reabertura, 1 reabertura e 2+ reaberturas).

Tabela 3.19: Teste de Normalidade Shapiro-Wilk

Métrica	**0 RE**	**1 RE**	**2+ RE**
N_{PC}	2.374608×10^{-44}	9.208642×10^{-43}	5.957832×10^{-43}
P_{PC}	1.491504×10^{-43}	3.502554×10^{-41}	1.85034×10^{-41}
SP_{PC}	4.907493×10^{-35}	2.462471×10^{-30}	1.820061×10^{-30}
N	1.68965×10^{-49}	2.301562×10^{-47}	4.593668×10^{-48}
P	1.041154×10^{-39}	1.519105×10^{-41}	1.352918×10^{-41}
SP	8.547338×10^{-32}	1.715911×10^{-32}	3.610839×10^{-33}
NM	5.499065×10^{-37}	3.659547×10^{-34}	7.632449×10^{-36}
PM	3.186394×10^{-28}	9.509193×10^{-30}	4.46319×10^{-29}
DCN	1.57804×10^{-40}	1.548992×10^{-36}	3.574652×10^{-37}
DCP	4.617531×10^{-29}	1.87759×10^{-30}	1.070468×10^{-29}

Os resultados indicam que os valores de p são extraordinariamente baixos, variando na ordem de 10^{-43} a 10^{-22}. Esses valores extremamente pequenos são significativamente menores que o nível de significância comum de 0,05 (ou seja, 5%). Essa evidência estatística esmagadora leva à conclusão de que há suporte para rejeitar a hipótese nula, que indica que os dados seguem uma distribuição normal. Em termos mais simples, para todas as métricas e em todos os grupos analisados, não há base estatística para a suposição de normalidade nos dados.

3.4.4 Teste de Wilcoxon

Utilizamos o teste de Wilcoxon, realiza com com um nível de significância (α) de 0,05, para avaliar possíveis diferenças significativas nas métricas entre os grupos com e sem reaberturas. Os resultados são resumidos na Tabela 3.20.

Tabela 3.20: Resultados do Teste de Wilcoxon

Métrica	**0 RE vs 1 RE**	**0 RE vs 2+ RE**	**1 RE vs 2+ RE**
N_{PC}	0.01418626	0.0226751	0.8527919
P_{PC}	0.003369067	0.01147024	0.6476487
SP_{PC}	0.9500049	0.7643696	0.7274183
N	0.004526062	0.05887791	0.3415155
P	0.008853818	0.004270134	0.8312402
SP	0.0002264632	0.0004776055	0.7972919
NM	2.135814e-05	0.003697484	0.1776996
PM	0.1838283	0.1445879	0.926208
DCN	8.091754e-06	0.0001062118	0.5368075
DCP	0.0365391	0.05835107	0.7418242

Ao analisar o primeiro comentário após a abertura da issue, observamos que a pontuação negativa (N_{PC}) apresenta diferenças estatisticamente significativas entre o grupo de issues sem reabertura (0 RE) e o grupo com uma reabertura (1 RE), assim como entre o grupo sem reabertura (0 RE) e o grupo com duas ou mais reaberturas (+2 RE). No entanto, não são encontradas diferenças significativas entre os grupos 1 RE e 2+ RE. Quanto à pontuação positiva (P_{PC}), são identificadas diferenças estatisticamente significativas entre 0 RE e 1 RE, assim como entre 0 RE e 2+ RE, mas não entre 1 RE e 2+ RE. A métrica SP_{PC} não apresenta evidência estatística de diferenças significativas entre nenhum par de grupos.

Ao considerar o último comentário antes do fechamento, são identificadas diferenças estatisticamente significativas entre o grupo sem reaberturas (0 RE) e o grupo com uma reabertura (1 RE) na pontuação negativa (N), assim como entre o grupo sem reabertura (0 RE) e o grupo com duas ou mais reaberturas (+2 RE). No entanto, não são encontradas diferenças significativas entre os grupos com reaberturas. Quanto à pontuação positiva (P), são observadas diferenças estatisticamente significativas entre o grupo sem reabertura (0 RE) e os grupos com reaberturas (1 RE e +2 RE), mas não são encontradas

diferenças significativas entre os grupos com reaberturas. Além disso, existem diferenças estatisticamente significativas entre o grupo sem reabertura e os grupos com reaberturas na métrica SP, mas não entre os grupos com reaberturas.

Na análise entre os sentimentos dos comentários entre a abertura e o fechamento, identificamos diferenças estatisticamente significativas entre o grupo sem reabertura (0 RE) e os grupos com reabertura (1RE e +2RE) nos valores das métricas pontuação negativa média (NM) e pontuação positiva média (PM). Entretanto, não existe evidência de diferenças significativas entre os grupos com reabertura. O mesmo padrão é observado nas métricas densidade de comentários negativos (DCN) e densidade de comentários positivos (DCP), onde encontramos diferenças estatisticamente significativas entre o grupo sem reabertura (0 RE) e os grupos com reabertura (1RE e +2RE), enquanto não há diferenças significativas entre os grupos com reaberturas.

3.4.5 Correlação de Spearman

Os resultados da análise de correlação de Spearman entre as métricas e o número de reaberturas são apresentados na tabela 3.21. O coeficiente de correlação (*rho*) fornece insights sobre a relação monotônica entre as métricas e a quantidade de reaberturas.

Tabela 3.21: Correlação de Spearman entre as métricas e número de reaberturas

Métrica	Correlação (rho)
N_{PC}	-0.03305615
P_{PC}	0.05946764
SP_{PC}	0.01854983
S_{PC}	0.02254068
N	-0.05705047
P	-0.06790966
SP	-0.08549524
S	-0.08040542
NM	-0.06246488
PM	-0.01104148
DCN	0.06899537
DCP	-0.01627266

De acordo com a Tabela 3.21, as métricas de pontuação negativa do primeiro comentário (N_{PC}), pontuação negativa (N), pontuação positiva (P), sentimento (SP), pontuação negativa média (NM) e densidade dos comentários positivos (DCP) exibem uma correlação negativa, sugerindo que, à medida que a quantidade de reaberturas aumenta ou diminui, essas métricas tendem a diminuir. Em outras palavras, um aumento ou diminuição na quantidade de reaberturas está associado a valores menores dessas métricas.

As métricas de pontuação positiva do primeiro comentário (P_{PC}) e densidade de comentários negativos (DCN) mostraram uma correlação positiva, indicando que à medida que a quantidade de reaberturas aumenta ou diminui, essas métricas tendem a aumentar.

Em resumo, mudanças na quantidade de reaberturas estão associadas a valores maiores dessas métrica.

As métricas de sentimento do primeiro comentário (SP_{PC}) e pontuação positiva média (PM) apresentaram correlações próximas de zero, sugerindo que essas métricas não têm uma relação forte ou clara com a quantidade de reaberturas. As variações na quantidade de reaberturas não estão fortemente associadas a mudanças nessas métricas.

3.4.6 Discussões dos resultados

A análise dos sentimentos em issues no GitHub revela insights importantes sobre a relação entre os sentimentos expressos nos comentários e a reabertura das issues. O estudo utilizou o *teste de Wilcoxon* com um nível de significância de $0,05$ para comparar as diferentes métricas que envolvem as pontuações negativas e positivas e os sentimentos entre três grupos de issues: sem reabertura (0 RE), com uma reabertura (1 RE) e com duas ou mais reaberturas (2+ RE). A seguir, são discutidas as respostas para cada questão de pesquisa.

QP2.4: Existe algum indicativo de que uma issue fechada será reaberta com base no sentimento, pontuação negativa e pontuação positiva do primeiro comentário após a abertura da issue? Os resultados mostram que as pontuações negativas do primeiro comentário (N_{PC}) são significativamente maiores para issues que foram reabertas em comparação com issues que não foram reabertas. Para o grupo 0 RE vs 1 RE, o valor de p foi 0,014, e para 0 RE vs 2+ RE, $p = 0{,}022$, sugerindo que sentimentos negativos iniciais podem estar associados a uma maior probabilidade de reabertura. Além disso, as pontuações positivas (P_{PC}) foram menores para issues reabertas, com $p = 0{,}003$ entre 0 RE e 1 RE, e $p = 0{,}011$ entre 0 RE e 2+ RE. Apesar dessas associações, a força dessas correlações foi baixa, indicando que o impacto do sentimento do primeiro comentário sobre a reabertura não é muito forte, mas ainda pode fornecer alguns indicativos iniciais.

QP2.5 Existe algum indicativo de que uma issue fechada será reaberta com base no sentimento, pontuação negativa e pontuação positiva do último comentário antes do primeiro fechamento da issue? Em relação ao último comentário antes do fechamento, as pontuações negativas (N) foram significativamente mais altas nas issues reabertas, com $p = 0{,}004$ entre 0 RE e 1 RE, indicando que a presença de sentimentos negativos antes do fechamento pode estar associada a uma maior probabilidade de reabertura. Da mesma forma, as pontuações positivas (P) foram menores nas issues reabertas, com $p = 0{,}008$ entre 0 RE e 1 RE. Embora esses resultados sejam estatisticamente significativos, assim como no caso do primeiro comentário, a correlação entre os sentimentos finais e a reabertura é relativamente fraca. Isso sugere que, embora sentimentos negativos antes do fechamento possam sinalizar insatisfação ou problemas não resolvidos, eles não são preditores determinantes da reabertura.

QP2.6 É possível prever se uma issue será reaberta através dos sentimentos presentes nas suas discussões entre os eventos de abertura (*open*) e o primeiro fechamento (*closed*)? A análise dos sentimentos ao longo das discussões entre

a abertura e o fechamento da issue revela que as pontuações negativas médias (NM) e o desvio cumulativo de sentimentos negativos (DCN) são maiores nas issues reabertas, com valores de $p < 0{,}001$ entre 0 RE e 1 RE. Isso sugere que a persistência de sentimentos negativos nas discussões está associada à reabertura. No entanto, a correlação entre esses sentimentos e a reabertura foi baixa, indicando que, embora haja uma tendência, ela não é forte o suficiente para que os sentimentos nas discussões sejam considerados um fator preditivo confiável de reabertura.

QP2.7 Existe uma correlação forte entre os sentimentos das discussões das issues com a probabilidade de reabertura? Os resultados indicam que não há uma correlação forte entre os sentimentos das discussões e a probabilidade de reabertura. Embora sentimentos negativos apareçam com mais frequência em issues reabertas, a força dessas correlações foi baixa em todas as métricas analisadas. Isso sugere que os sentimentos, embora relacionados à reabertura de issues em alguns casos, não são um fator determinante. A reabertura de issues parece ser influenciada por uma combinação de fatores além dos sentimentos expressos nas discussões.

QP2: Como caracterizar a influência dos sentimentos na reabertura de issues? A influência dos sentimentos sobre a reabertura de issues é limitada. Embora sentimentos negativos estejam associados a uma maior probabilidade de reabertura, a correlação é fraca. Isso indica que, embora possa haver uma tendência de que issues com discussões mais negativas sejam reabertas, os sentimentos por si só não são um fator preditivo forte. A reabertura de issues parece depender mais de outros aspectos, como a natureza técnica do problema e a complexidade da solução, do que exclusivamente dos sentimentos expressos nos comentários.

3.4.7 Ameaças à validade

Validade Interna. A qualidade e a consistência dos dados coletado das issues podem ser comprometida por informações faltantes, inconsistências nos comentários ou variações na expressão de sentimentos. Além disso, como a extração dos dados foi feita até agosto de 2020, há o risco de desatualizações. Para mitigar essas ameaças, implementamos um rigoroso processo de limpeza e pré-processamento, removendo dados irrelevantes e convertendo emojis em texto. Também excluímos repositórios problemáticos, garantindo que apenas dados de alta qualidade fossem analisados.

Embora os dados sejam de 2020, o estudo foca em padrões históricos, e a estabilidade desses padrões ao longo do tempo apoia a validade das conclusões. Futuras atualizações podem ser consideradas em estudos subsequentes.

Por fim, a subamostragem (*undersampling*) para equilibrar as classes poderia introduzir viés, especialmente ao remover issues sem reaberturas. No entanto, essa técnica foi essencial para permitir comparações estatísticas robustas entre grupos de tamanho semelhante, e foi realizada de maneira completamente aleatória, minimizando o risco de viés.

Validade Externa. O estudo se baseia em dados de repositórios específicos até agosto de 2020, o que pode não refletir práticas ou tendências atuais. No entanto, os repositórios

analisados abrangem uma variedade de linguagens e tipos de projetos, o que aumenta a relevância dos resultados para diferentes cenários de desenvolvimento de software. A metodologia detalhada também permite replicação em outros contextos, ampliando a validade externa.

A análise de sentimentos pode variar entre idiomas e culturas. Como o estudo focou em textos em inglês, os resultados podem ser menos aplicáveis em contextos multilíngues. No entanto, o inglês é amplamente usado em projetos de software internacionais, o que torna os resultados relevantes para a comunidade global de desenvolvimento. O estudo reconhece essas limitações e sugere futuras pesquisas que considerem variações linguísticas e culturais.

Validade de Construto. O uso de uma ferramenta de análise de sentimentos baseada em léxico pode não capturar as nuances dos sentimentos, como sarcasmo ou ironia. Escolhemos essa abordagem pela sua eficiência para processar grandes volumes de dados. Cuidados no pré-processamento, como a conversão de emojis e a remoção de ruídos, ajudaram a melhorar o desempenho da análise. O estudo reconhece as limitações dessa abordagem e sugere o uso de técnicas de aprendizado de máquina em futuras pesquisas para captar sentimentos mais complexos.

As métricas utilizadas no estudo (N_{PC}, P_{PC}, SP_{PC},S_{PC}, N, P, SP , S,NM,PM,DCN e DCP) podem não refletir o impacto emocional real, levando a interpretações simplificadas. No entanto, a aplicação de diversas métricas e testes estatísticos compensou as limitações de cada medida individual, fortalecendo a confiabilidade das conclusões.

Validade de Conclusão. A subamostragem pode comprometer a robustez estatística dos resultados, especialmente com amostras pequenas. No entanto, a subamostragem foi necessária para equilibrar as classes e permitir comparações válidas. Utilizamos o Teste de Wilcoxon, apropriado para distribuições não normais e amostras pequenas, e a correlação de Spearman para avaliar relações monotônicas, o que reforçou a confiabilidade das conclusões, mesmo com as limitações de amostragem.

3.4.8 Conclusões

Neste estudo, exploramos a análise de sentimentos aplicada a comentários em issues de projetos de código aberto. Observamos que muitos comentários são categorizados como neutros, indicando uma predominância de comunicações objetivas e informativas nos registros de issues.

Ao examinarmos os sentimentos nos primeiros e últimos comentários antes do fechamento das issues, notamos uma tendência interessante. Nos três grupos analisados (0 reabertura, 1 reabertura e 2+ reaberturas), observamos uma diminuição na quantidade de sentimentos negativos e um aumento na quantidade de sentimentos positivos. Isso sugere uma transição de abordagens mais críticas e problemáticas nos estágios iniciais para avaliações mais positivas à medida que a resolução da issue avança.

Analisando o último comentário antes do fechamento, identificamos que os grupos com uma ou mais reaberturas apresentam mais issues com sentimentos negativos em comparação ao grupo sem reaberturas. Por outro lado, o grupo sem reaberturas possui

mais issues com sentimentos positivos. Isso aponta para a possibilidade de que a satisfação e a perceção positiva estejam correlacionadas com a ausência de reaberturas.

No entanto, os resultados indicam que a análise de sentimentos, embora valiosa para compreender as emoções e opiniões dos desenvolvedores, não é suficiente para prever de maneira precisa e confiável se uma issue será reaberta. A gestão eficaz de issues em projetos de software é uma tarefa complexa, influenciada por diversos fatores, como a qualidade do código, a urgência na correção e a comunicação eficaz entre os participantes.

Portanto, enquanto a análise de sentimentos desempenha um papel relevante, ela deve ser integrada a um conjunto mais abrangente de ferramentas e técnicas. A qualidade do processo de desenvolvimento de software beneficia-se de uma abordagem holística, considerando vários aspectos para aprimorar a eficiência e eficácia na gestão de issues, além de proporcionar insights mais completos sobre a dinâmica do projeto.

3.5 CONCLUSÃO DO CAPÍTULO

Este capítulo foi marcado por uma abordagem metodológica abrangente e detalhada, iniciando com um estudo piloto dedicado à mineração de sentimentos em issues, conforme descrito na seção 3.1. Esta etapa foi crucial para estabelecer uma base sólida para a análise automatizada, proporcionando insights sobre os desafios e particularidades da linguagem utilizada em discussões de projetos de código aberto.

Em seguida, abordamos o processo de validação e construção de um dicionário léxico na Seção 3.2. A validação do léxico é uma etapa fundamental para garantir a precisão e a confiabilidade das análises automatizadas de sentimentos. Esse esforço contribui para a criação de um recurso robusto e adaptado ao contexto específico do desenvolvimento de software.

Posteriormente, dedicamos nossa atenção a um estudo específico centrado na análise de sentimentos em discussões de issues reabertas, conforme apresentado na Seção 3.3. Essa análise proporcionou uma compreensão aprofundada das dinâmicas emocionais associadas às reaberturas de issues, permitindo-nos identificar padrões e tendências relevantes.

Além disso, conduzimos uma análise comparativa nas discussões de issues, considerando aquelas com e sem reaberturas. Detalhada na Seção 3.4, essa avaliação teve como objetivo investigar o impacto da presença desses ciclos de reabertura nas interações e dinâmicas emocionais entre os colaboradores.

Esses experimentos são cruciais para contextualizar e aprofundar nossa compreensão sobre como os sentimentos influenciam as interações no contexto específico de desenvolvimento de software. A complexidade dessas dinâmicas ressalta a importância de abordagens integradas que considerem não apenas a análise de sentimentos, mas também outros fatores intrínsecos ao ambiente de desenvolvimento colaborativo. O capítulo seguinte explorará mais a fundo os resultados dessas análises, proporcionando uma visão mais ampla sobre o papel dos sentimentos no ciclo de vida das issues em projetos de código aberto.

Capítulo

4

CATEGORIZAÇÃO DE ISSUES

Este capítulo tem com objetivo investigar as diferentes categorias de issues no problema de reabertura, com o intuito de responder à questão de pesquisa *QP3. Quais categorias estão mais associadas à reabertura de issues?*

Inicialmente, na Seção 4.1, exploramos o desenvolvimento de um modelo de categorização de issues, alinhado com o objetivo de criar e desenvolver um categorizador automático de issues do GitHub (Obj 5). Para atingir esse objetivo, investigamos as seguintes questões de pesquisa: *QP3.1 Qual é a melhor técnica de subamostragem de dados combinada com a técnica de aprendizado de máquina para a categorização de issues?* Aqui, avaliamos diferentes abordagens de aprendizado de máquina e técnicas de balanceamento de dados, focando em identificar a estratégia mais eficaz para lidar com a categorização automática de issues.

Em seguida, na Seção 4.2, abordamos um estudo empírico focado na categorização de issues do GitHub, com o objetivo de avaliar como diferentes categorias de issues se relacionam com a reabertura (Obj 6). Neste contexto, duas questões de pesquisa adicionais foram investigadas: *QP3.2. O tempo de duração entre a abertura e o primeiro fechamento de uma issue é um indicativo de reabertura nas diferentes categorias? QP3.3. A quantidade de comentários entre a abertura e o primeiro fechamento de uma issue pode influenciar sua probabilidade de reabertura nas diferentes categorias?*

4.1 MODELO DE CATEGORIZAÇÃO DE ISSUES

Categorizar issues é essencial para uma organização lógica, facilitando a identificação e rastreamento. Isso permite priorizar a resolução das categorias mais críticas, otimizando os esforços da equipe. Neste estudo, o foco está na implementação de um modelo de classificação de issues.

O objetivo principal deste estudo é construir e treinar um modelo capaz de categorizar automaticamente as issues em diferentes categorias predefinidas, com o intuito de melhorar a eficiência no gerenciamento de issuesem repositórios do GitHub. Essa abordagem visa acelerar o processo de atribuição de issues, permitindo que elas sejam endereçadas

rapidamente às equipes responsáveis, reduzindo o tempo de resolução e aumentando a produtividade do projeto. Para orientar essa investigação, levantamos as seguintes questões de pesquisa: *QP3.1 Qual é a melhor técnica de subamostragem de dados combinada com a técnica de aprendizado de máquina para a categorização de issues?*

Essas perguntas guiam o processo de desenvolvimento e avaliação do modelo, permitindo identificar as melhores abordagens tanto em termos de balanceamento de dados quanto de técnicas de aprendizado de máquina para categorização eficiente de issues. A seguir, apresentamos a metodologia utilizada para conduzir o estudo e responder às questões de pesquisa.

4.1.1 Metodologia

O estudo foi realizado em 5 etapas: Base de dados, pré-processamento do texto, balanceamento dos dados, seleção do classificador e análise dos classificadores. A figura 4.1 apresenta uma visão geral das etapas do estudo, as quais são detalhadas a seguir.

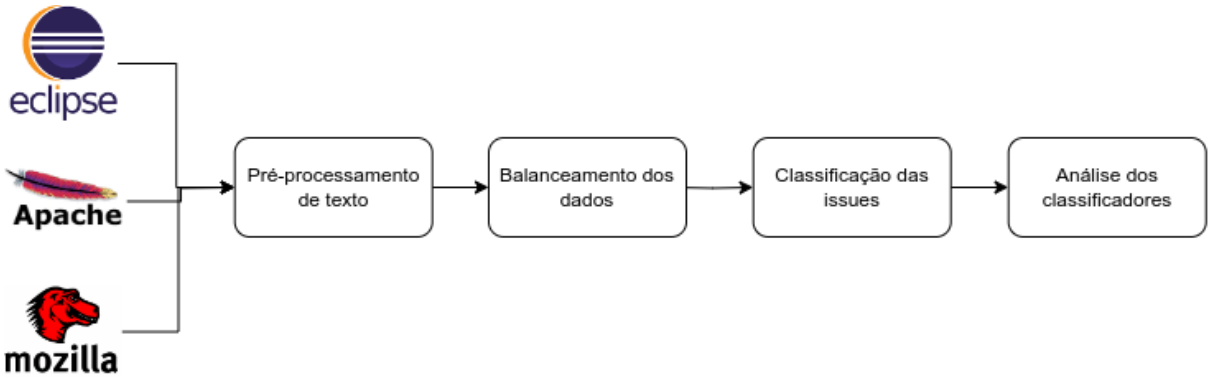


Figura 4.1: Etapas do modelo de classificação de issues

4.1.1.1 Base dados

Para essa etapa, utilizaremos da base de dados (CATOLINO et al., 2023), que contém 1280 issues de projetos dos ecossistemas como Mozilla, Apache e Eclipse. A partir da base de dados original, 126 issues foram excluídas pois não possuem uma categoria de issue específica ou não estavam disponíveis para download. Nas 1154 issues selecionadas, adicionamos a descrição e url correspondente. A base de dados está acessível por meio do link <bit.ly/3OXbGlz>.

As issues selecionadas estão classificadas de acordo com a taxonomia proposta por (CATOLINO et al., 2019) em Banco de dados, Configuração, Desempenho, Funcional, GUI, Info, Permissão/Obsoleto, Redes, Segurança e Testes.

4.1.1.2 Pré-processamento de texto

Nesta etapa realizamos o pré-processamentos dos textos da descrição e título das issues onde fizemos conversão de letras maiúsculas para minúsculas, realizamos correção ortográfica, remoção de trechos indesejados no texto como código-fonte, trechos de código, erros de compilação, urls, tags, entre outros. Em seguida, aplicamos a tokenização, remoção dos *stopwords* realizamos o processo de *stemming*. Por fim, efetuamos a vetorização do texto utilizando a abordagem *CountVectorizer* e a técnica de frequência do termo–inverso da frequência nos documentos (*term frequency–inverse document frequency* - TF-IDF)

4.1.1.3 Balanceamentos dos dados

O desbalanceamento dos dados pode causar o baixo desempenho com modelos tradicionais de aprendizado de máquina e métricas de avaliação que assumem uma distribuição de classe equilibrada. Ou seja, em problemas com categorias desbalanceadas, onde uma categoria significativamente menor do que a outra, os modelos de aprendizado de máquina podem ter dificuldade em aprender padrões representativos da categoria minotária.

Com o objetivo de encontrar um melhor desempenho do classificador, utilizamos três tipos de balaceamento de dados. Estas três técnicas de pré-processamento de dados amplamente utilizadas para lidar com o desequilíbrio das categorias em problemas de classificação. As técnicas utilizadas foram:

- **Subamostragem de dados (*Undersampling*)** visa reduzir a quantidade de amostras da categoria majoritária para equilibrar as proporções entre as categorias. A técnica *NearMiss* seleciona as amostras da categoria majoritária com base em sua proximidade às amostras da categoria minoritária, a fim de evitar o viés do modelo em relação à categoria majoritária (MANI; ZHANG, 2003).

- **Sobreamostragem de dados (*Oversampling*)** visa aumentar o número de amostras das categorias minoritárias. A técnica *Synthetic Minority Over-sampling Technique (SMOTE)* consiste em gerar novas amostras sintéticas a partir de amostras da categoria minoritária na mesma direção dos vizinhos escolhidos aleatoriamente (CHAWLA et al., 2002).

- **Combinação dos métodos de sobre e sub amostragem de dados** visa melhorar o desempenho dos classificadores em situações em que uma categoria é significativamente menor do que a outra. A técnica *SMOTE + ENN (SMOTEENN)* combina a técnica de sobreamostragem (*SMOTE*) com a técnica de subamostragem (*Edited Nearest Neighbors (ENN)*).

  SMOTEENN primeiro gera amostras sintéticas da categoria minoritária para aumentar o número de amostras dessa categoria e em seguida elimina as amostras que são mal classificadas pelo classificador K Vizinhos Mais Próximos (kNN) (BATISTA; PRATI; MONARD, 2004).

4.1.1.4 Classificação das issues Nessa etapa selecionados sete técnicas para realizar a tarefa de classifição:

- **Árvore de Decisão** (*Decision Tree* - DT) método busca construir uma árvore de decisão hierárquica que melhor separe as diferentes categorias (HASTIE et al., 2009).

- **Descida de Gradiente Estocástica** (*Stochastic Gradient Descent* (SGD)) é uma variação do algoritmo Gradiente Descendente que utiliza uma abordagem estocástica para calcular o gradiente do erro em relação aos pesos do modelo e atualiza os pesos em pequenos passos para encontrar os valores que minimizam a função de perda (BOTTOU, 2010).

- **Floresta Aleatória** (RF), método que combina a saída de diferentes árvores de decisão para realizar a classificação (HASTIE et al., 2009).

- **Máquina de Vetores de Suporte** (*Support Vector Machines* (SVM)) modelo que tem como objetivo encontrar o hiperplano que melhor separa as amostras de diferentes categorias no espaço de características (ABE, 2005).

- **Naive Bayes Multinomial** (*Multinomial Naive Bayes* (MNB)) modelo variante de classificação probabilístico Naive Bayes para classificação de texto multinomial. O MNB calcula a probabilidade de um documento pertencer a uma categoria específica usando as frequências de ocorrências das palavras nos documentos dessa categoria (KIBRIYA et al., 2005).

- **Perceptron Multicamadas** (*Multilayer Perceptron* (MLP)) modelo de Rede Neural Artificial com múltiplas camadas conectadas de neurônios, incluindo uma camada de entrada, uma ou mais camadas intermediárias e uma camada de saída (AGGARWAL, 2018).

- **Regressão Logística** (LR) extensão da regressão linear, que permite a modelagem de problemas de classificação. O modelo calcula a probabilidade de uma amostra de pertencer a uma categoria específica usando a função logística (BISHOP; NASRABADI, 2006).

4.1.1.5 Análise dos classificadores

Com o intuito de avaliar a capacidade de generalização dos modelos, nós utilizamos a técnica de validação cruzada k-fold, com k = 10, onde a base de dados é dividida de forma aleatória em k subconjuntos.

A cada iteração do modelo, k-1 subconjuntos são usados na etapa de treinamento e o subconjunto restante é utilizado na etapa de teste. Os modelos são avaliados a partir das seguintes métricas de avaliação (SILVA; PERES; BOSCARIOLI, 2016):

- **Acurácia** (do inglês, *accuracy*) ou taxa de classificações corretas indica um desempenho geral do modelo. Assim, dentre todas as classificações, diz quantas o modelo classificou corretamente, de acordo com a eq. 4.1:

$$\text{Acurácia} = \frac{\text{VP} + \text{VN}}{(\text{VP} + \text{FN} + \text{VN} + \text{FP})} \tag{4.1}$$

onde Verdadeiro Positivo (VP) é número de issues corretamente rotuladas como pertencentes a uma categoria específica e essas issues realmente pertencem a essa categoria. Verdadeiro Negativo (VN) é o número de issues rotuladas como não pertencentes a uma específica categoria e essas issues realmente não pertencem a essa categoria. Falso Positivo (FP) é o número de issues rotuladas erroneamente como pertencentes a uma categoria específica, mas essas issues na verdade não pertence a essa categoria. E Falsos Negativo (FN) é o número de issues erroneamente erroneamente como não pertencente a uma categoria específica, mas essas issues realmente pertencem a essa categoria.

- **Precisão** (do inglês, *precision*) corresponde a porcentagem de acertos ou Verdadeiros Positivos (VP) dentre todas as amostras cuja categoria esperada é a categoria positiva (eq. 4.2):

$$\text{Precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{4.2}$$

- **Revocação** (do inglês, *recall*) corresponde a porcentagem de verdadeiros positivos dentre todos os exemplos cuja categoria esperada é a categoria positiva (eq. 4.3) :

$$\text{Revocação} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Negativos}} \tag{4.3}$$

- $\mathbf{F_1}$ (do inglês *F-score*) Corresponde à média harmônica entre a precisão e a revocação (eq. 4.4):

$$F_1 = \frac{2 \cdot \text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}} \tag{4.4}$$

4.1.2 Resultados

A base de dados selecionada é composta por 391 issues de projetos dos ecossistemas Apache, 392 issues do Eclipse e 371 issues Mozilla classificadas em 10 categorias apresentadas na seção 4.1.1.1. A distribuição das issues pode ser vista na tabela (4.1) :

Tabela 4.1: Categorias de issues vs Tipo de Ecosistema

Categoria	**Apache**	**Eclipse**	**Mozilla**	**Total**	**%**
Base de dados	7	13	15	35	3,03%
Configuração	66	61	60	187	16,20%
Desempenho	17	13	12	42	3,64%
Funcional	147	170	152	469	40,65%
GUI	68	74	55	197	17,07%
Info	5	5	8	18	1,56%
Permissão/Obsoleto	15	8	16	39	3,38%
Redes	17	12	15	44	3,81%
Segurança	26	4	14	44	3,81%
Teste	23	32	24	79	6,85%
Total	**391**	**392**	**371**	**1154**	**100%**

A partir da tabela 4.1 podemos identificar que a categoria *Funcional* possui a maioria das issues com 40,65%, a segunda categoria com mais issues é a *GUI* com 17,7%, a terceira categoria com mais issues é a *Configuração* com 17,7% e o restante das categorias possuem issues abaixo de 7%.

QP3.1 Qual é a melhor técnica de subamostragem de dados combinada com a técnica de aprendizado de máquina para a categorização de issues?

Para responder a questão de pesquisa, avaliamos o desempenho dos classificadores classificadores DT, LR, MLP, MNB, RF, SGD e SVM utilizando diferentes técnicas de subamostragem e sobreamostragem. Os testes iniciais foram realizados sem qualquer técnica de balanceamento e os resultados são mostrados na tabela 4.2.

Tabela 4.2: Resultado dos Classificadores com dados desbalanceados

Classificador	**Acurácia**	**Precisão**	**Recall**	**F1**
DT	47.57%	48.31%	45.74%	46.56%
LR	51.81%	76.25%	27.33%	33.02%
MLP	52.16%	62.64%	39.01%	45.91%
MNB	42.54%	23.24%	11.07%	7.89%
RF	**55.28%**	**72.83%**	**43.73%**	**50.98%**
SGD	52.16%	60.14%	43.59%	49.34%
SVM	49.56%	64.18%	25.40%	29.93%

Todos modelos de classificação apresentados na tabela 4.2 tiveram baixas taxas de acurácia, precisão, revocação e F1. A taxa baixa de classificação pode estar relacionada com os dados desbalanceados como pode ser visto na tabela (4.1). Com o intuito de melhorar o desempenho da classificação, aplicamos técnicas de balanceamento, começando

com a técnica de subamostragem *NearMiss*, cujos resultados estão apresentados na tabela 4.3 :

Tabela 4.3: Resultado dos Classificadores utilizando Subamostragem de dados *NearMiss*

Classificador	**Acurácia**	**Precisão**	**Revocação**	**F1**
DT	50.00%	52.72%	50.00%	50.69%
LR	53.88%	55.23%	53.88%	53.85%
MLP	54.44%	56.24%	54.44%	54.04%
MNB	52.22%	56.47%	52.22%	51.72%
RF	54.44%	56.26%	54.44%	54.03%
SGD	**56.11%**	**54.31%**	**56.11%**	**53.93%**
SVM	51.66%	61.74%	51.66%	54.10%

A tabela 4.3 mostra que a técnica *NearMiss* melhorou a performance geral dos classificadores em relação aos dados desbalanceados (tabela 4.2), com destaque para o classificador SGD, que apresentou a melhor taxa de acurácia (56,11

Em seguida, utilizamos a técnica *SMOTE* para sobreamostragem de dados e os resultados são apresentados na tabela 4.4 :

Tabela 4.4: Resultado dos Classificadores utilizando sobreamostragem de dados SMOTE

Classificador	**Acurácia**	**Precisão**	**Revocação**	**F1**
DT	87.91%	87.78%	87.91%	87.83%
LR	93.29%	93.09%	93.29%	93.09%
MLP	95.22%	95.20%	95.22%	94.95%
MNB	90.36%	89.87%	90.36%	89.46%
RF	94.45%	94.62%	94.45%	94.50%
SGD	95.11%	95.04%	95.11%	94.95%
SVM	**96.43%**	**96.81%**	**96.43%**	**96.51%**

A partir na tabela 4.4 podemos identificar que a técnica SMOTE proporcionou uma melhoria significativa no desempenho, com o classificador SVM obtendo a melhor acurácia (96,43%). Por fim, utilizamos o método *SMOTEENN*, que combina subamostragem e sobreamostragem, e os resultados são mostrados na tabela 4.5.

Tabela 4.5: Avaliação dos classificadores utilizando o método SMOTEENN

Classificador	Acurácia	Precisão	*Revocação*	F1
DT	95,06%	85,27%	85,43%	85,35%
LR	98,89%	88,91%	89,10%	89,00%
MLP	**99,75%**	**99,76%**	**93,22%**	**94,82%**
MNB	96,39%	86,97%	86,45%	86,51%
RF	99,23%	89,25%	89,47%	89,35%
SGD	99,68%	99,66%	93,15%	94,74%
SVM	99,68%	89,67%	89,89%	89,78%

A partir dos resultados mostrados na tabela 4.5 podemos identificar que todos os classificadores apresentam acurária acima de 95%. O melhor resultado foi alcançado pelo classificador MLP com 99,75% de acurácia seguido por SGD, SVM e RF, que também apresentaram altos desempenhos.

Portanto, a combinação da técnica *SMOTEENN* com o classificador MLP apresentou o melhor desempenho geral para a categorização de issues, oferecendo uma alta acurácia e desempenho robusto nas métricas de precisão, revocação e F1. Os resultados detalhados das métricas podem ser acessados através do link <https://bit.ly/SACI>.

4.1.3 Ameaças à validade

Validade Interna. A exclusão de issues da base de dados pode ter causado um viés e afetado a validade interna do estudo. No entanto, essas exclusões foram feitas com base em critérios objetivos, como a ausência de categorias específicas ou a indisponibilidade das issues para download. Esses critérios foram aplicados de forma consistente para assegurar a qualidade e a integridade dos dados analisados.

No processo de pré-processamento, a remoção de elementos como código-fonte, URLs e tags poderia ter eliminado informações relevantes que influenciam a classificação das issues. Contudo, esses elementos foram retirados porque não contribuem diretamente para a análise textual.

A seleção de um conjunto específico de classificadores, como SVM e Random Forest, poderia ter enviesado os resultados, uma vez que outros modelos não explorados poderiam apresentar melhor performance. Para mitigar essa ameaça, os classificadores foram escolhidos com base em sua ampla aceitação na comunidade científica e em sua eficácia comprovada em tarefas de categorização de texto. A seleção incluiu uma variedade de paradigmas de aprendizado de máquina (baseados em árvores, redes neurais, e modelos probabilísticos), assegurando uma cobertura abrangente.

Diferenças entre os projetos dos ecossistemas Mozilla, Apache e Eclipse poderiam ter introduzido vieses na categorização das issues. Para mitigar esse risco, foi aplicação a técnica de validação cruzada *k-fold*, que assegura que o modelo foi treinado e testado de forma consistente em diferentes partes da base de dados, reduzindo o impacto dessas variações e assegurando uma avaliação justa do modelo.

Validade Externa. A base de dados utilizada no estudo inclui issues de apenas três ecossistemas (Mozilla, Apache, Eclipse), o que pode limitar a generalização dos resultados para outros projetos de software. No entanto, essa limitação é mitigada pelo fato de que esses ecossistemas representam uma diversidade significativa de projetos com diferentes características e desafios, o que sugere uma boa aplicabilidade dos resultados em contextos semelhantes.

As categorias de issues utilizadas no estudo foram baseadas em uma taxonomia específica proposta por Catolino et al. (2019) , o pode não cobrir todas as possíveis categorias de issues que existem em outros projetos ou ecossistemas. Entretanto, essa ameaça é mitigada, pois a taxonomia aplicada é suficientemente abrangente para capturar as principais classes de issues em projetos de software tradicionais, mesmo que outras categorias possam existir em contextos específicos.

O estudo fez uso de ferramentas e técnicas específicas (e.g., CountVectorizer, TF-IDF, SVM, Random Forest), o que pode limitar a aplicabilidade dos resultados em outros contextos onde técnicas diferentes sejam utilizadas. Contudo, essa ameaça é refutada pela ampla aceitação e eficácia comprovada dessas ferramentas e técnicas em tarefas de categorização de texto, o que indica que os resultados obtidos são robustos e aplicáveis a diferentes contextos, especialmente aqueles semelhantes aos abordados no estudo.

Validade de Construto. A taxonomia de categorias de issue utilizada neste estudo foi baseada no trabalho de Catolino et al. (2019) e, embora seja abrangente, pode não capturar todas as nuances das diferentes categorias de issues presentes em outros contextos ou ecossistemas de software. A ameaça de validade de construto associada a essa limitação é mitigada pelo fato de que o estudo foi publicado em um jornal altamente reconhecido, o que confere maior credibilidade à taxonomia proposta.

A escolha das métricas de avaliação (acurácia, precisão, revocação e F1-score) apesar de serem amplamente utilizadas, podem não capturar todas as dimensões da performance do categorizador em contextos específicos. Para mitigar essa ameaça, o estudo empregou múltiplas métricas de avaliação, oferecendo uma visão mais completa e equilibrada do desempenho do modelo.

Validade de Conclusão. A escolha das técnicas de balanceamento de dados, como subamostragem, sobreamostragem ou a combinação de ambas, pode alterar a distribuição original dos dados e influenciar as conclusões sobre o desempenho do categorizador. Para mitigar essa ameaça, realizamos uma análise comparativa entre as técnicas (subamostragem *NearMiss*, sobreamostragem *SMOTE* e a combinação *SMOTEENN*). Além disso, os modelos foram avaliados com dados originais e balanceados, permitindo a identificação de possíveis vieses e garantindo uma interpretação mais precisa dos resultados.

4.1.4 Conclusão

Neste estudo, avaliamos diversas técnicas de balanceamento de dados e classificadores para aprimorar a categorização de issues em repositórios de software. Os resultados indicaram que técnicas como *NearMiss* e SMOTE melhoraram o desempenho dos classificadores. No entanto, a melhor performance foi alcançada pela combinação da técnica

SMOTEENN, que integra subamostragem e sobreamostragem, com o classificador MLP, que obteve uma acurácia de 99,75%. Esses resultados demonstram que a combinação de um classificador eficaz com técnicas avançadas de balanceamento pode ser extremamente eficiente na categorização de issues, mesmo em datasets desbalanceados.

Para próximos experimentos, utilizaremos o categorizador otimizado (*SMOTEENN + MLP*) para validar sua eficácia na análise de reabertura de issues do GitHub.

4.2 CATEGORIZAÇÃO DE ISSUES DO GITHUB

O estudo tem como objetivo identificar as categorias de issues presentes nas issues do GitHub e como as categorias podem influenciar a reabertura das issues. Para alcançar esse objetivo, procuramos responder às seguintes questões de pesquisa: *QP3. Quais categorias estão mais associadas à reabertura de issues?*. Além disso, investigaremos duas questões adicionais: *QP3.2. O tempo de duração entre a abertura e o primeiro fechamento de uma issue é um indicativo de reabertura nas diferentes categorias? QP3.3. A quantidade de comentários entre a abertura e o primeiro fechamento de uma issue pode influenciar sua probabilidade de reabertura nas diferentes categorias?*. A metodologia utilizada para responder a essas questões é apresentada na próxima seção.

4.2.1 Metodologia

O estudo será conduzido em cinco etapas: seleção dos repositórios, extração das issues, pré-processamento de texto, classificação das issues e análise. A figura 4.2 ilustra essas etapas em detalhes.

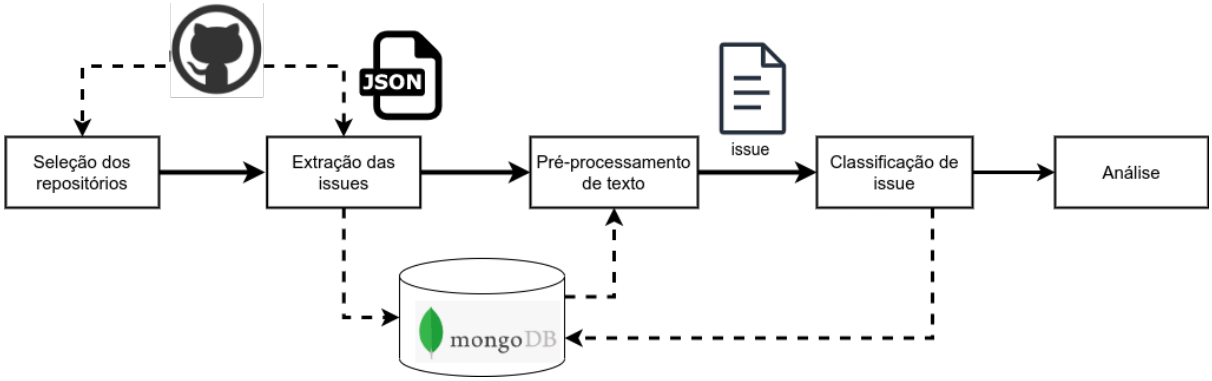


Figura 4.2: Etapas do estudo sobre classificação de issues do GitHub

4.2.1.1 Seleção da base de dados

Selecionamos a base de dados MRS2014 composta por issues do GitHub dos repositórios listados no desafio MRS Challenge Dataset da conferência do ano de 2014, onde as etapas para criação da base de dados são descritas nas seções 3.4.1.1 e 3.4.1.2.

4.2.1.2 Pré-processamento

A etapa de pré-processamento dos textos presentes no título e descrição da issues são descritos na Seção 4.1.1.2.

4.2.1.3 Classificação

Nessa etapa realizamos a classificações das issues da base de dados MSR14 através do classificador MLP apresentado na seção 4.1.1.4 treinado com dados balanceados utilizando a técnica de combinação de subamostragem e sobreamostragem SMOTEENN, descrita na seção 4.1.1.3.

4.2.1.4 Análise

Nessa etapa, consideramos a ordem cronológica dos eventos de cada issue com ou sem reabertas, conforme ilustrado na figura 3.8.O objetivo é analisar a duração em horas e a quantidade de comentários das issues, observando os eventos entre a abertura (*Open*) e o primeiro fechamento (*Closed*) nas seguintes categorias:*Base de dados, Configuração, Desempenho, Funcional, GUI, Info, Permissão/obsoleto, Redes, Segurança, e Testes*

Para a análise dos grupos de issues sem e com reaberturas em cada uma dessas categorias, utilizamos as seguintes estatísticas descritivas:

- Medidas de localização relativa: tempo mínimo (Min), tempo máximo (Max), primeiro quartil (primeiro quartil (Q1)) e terceiro quartil (terceiro quartil (Q3));
- Medidas de tendência central: tempo médio e mediana; e
- Medida de dispersão: desvio padrão.

Essas estatísticas nos permitirão identificar diferenças significativas no comportamento das issues com e sem reaberturas em cada categoria, oferecendo uma visão detalhada sobre as dinâmicas de cada grupo.

4.2.1.5 Remoção de Outliers

Após calcular as estatísticas descritivas, realizaremos a remoção de outliers a partir dos dados de duração média dos grupos com e sem reaberturas de cada categoria, utilizando a técnica da amplitude interquartis (*Interquartile Range* (IQR)). IQR é uma medida estatística usada para identificar e remover valores atípicos (outliers) de um conjunto de dados. Outliers são pontos de dados que se afastam significativamente da maioria dos outros pontos e podem distorcer a análise ou modelagem estatística, conforme discutido por Mendenhall, Beaver e Beaver (2012).

A IQR é calculada como a diferença entre o terceiro quartil (Q3) e o primeiro quartil (Q1) de um conjunto de dados, conforme mostrado na equação 4.5.

$$IQR = Q3 - Q1 \tag{4.5}$$

Para identificar os *outliers*, definimos um *Limite Inferior* (LI) e um *Limite Superior* (LS) com base na IQR:

$$LI = Q1 - k \cdot IQR \tag{4.6}$$

$$LS = Q3 + k \cdot IQR \tag{4.7}$$

Onde k é um fator que controla a sensibilidade à detecção de *outliers*. Utilizaremos o k igual à $1,5$. Qualquer issue com tempo de duração entre a abertura e o fechamento que esteja abaixo do LI ou acima do LS será considerada um *outlier* e será removida da análise.

4.2.2 Resultados

A base MSR2014 abrange um total de 490.531 issues, cada uma acompanhada por comentários, dos quais pelo menos um fechamento está registrado. Dentro desse conjunto, 475.935 issues não apresentam reaberturas, enquanto 14.596 delas (representando 2.98% do total) possuem uma ou mais reaberturas. Das issues que foram reabertas, 13.648 tiveram apenas uma reabertura, 815 foram reabertas duas vezes, 91 tiveram três reaberturas, 31 tiveram quatro reaberturas, 6 tiveram cinco reaberturas, 2 foram reabertas seis vezes, 1 teve sete reaberturas, 1 teve 10 reaberturas e 1 teve 13 reaberturas.

Para a análise, as issues foram divididas em dois grupos principais: 0) issues sem reaberturas e 1) issues com reaberturas. Além disso, foram classificadas em categorias funcionais, a saber: *Banco de Dados, Configuração, Desempenho, Funcional, Interface Gráfica do Usuário (GUI), Informação (Info), Permissão/Obsoleto, Segurança e Testes*, conforme descrito na tabela 4.6

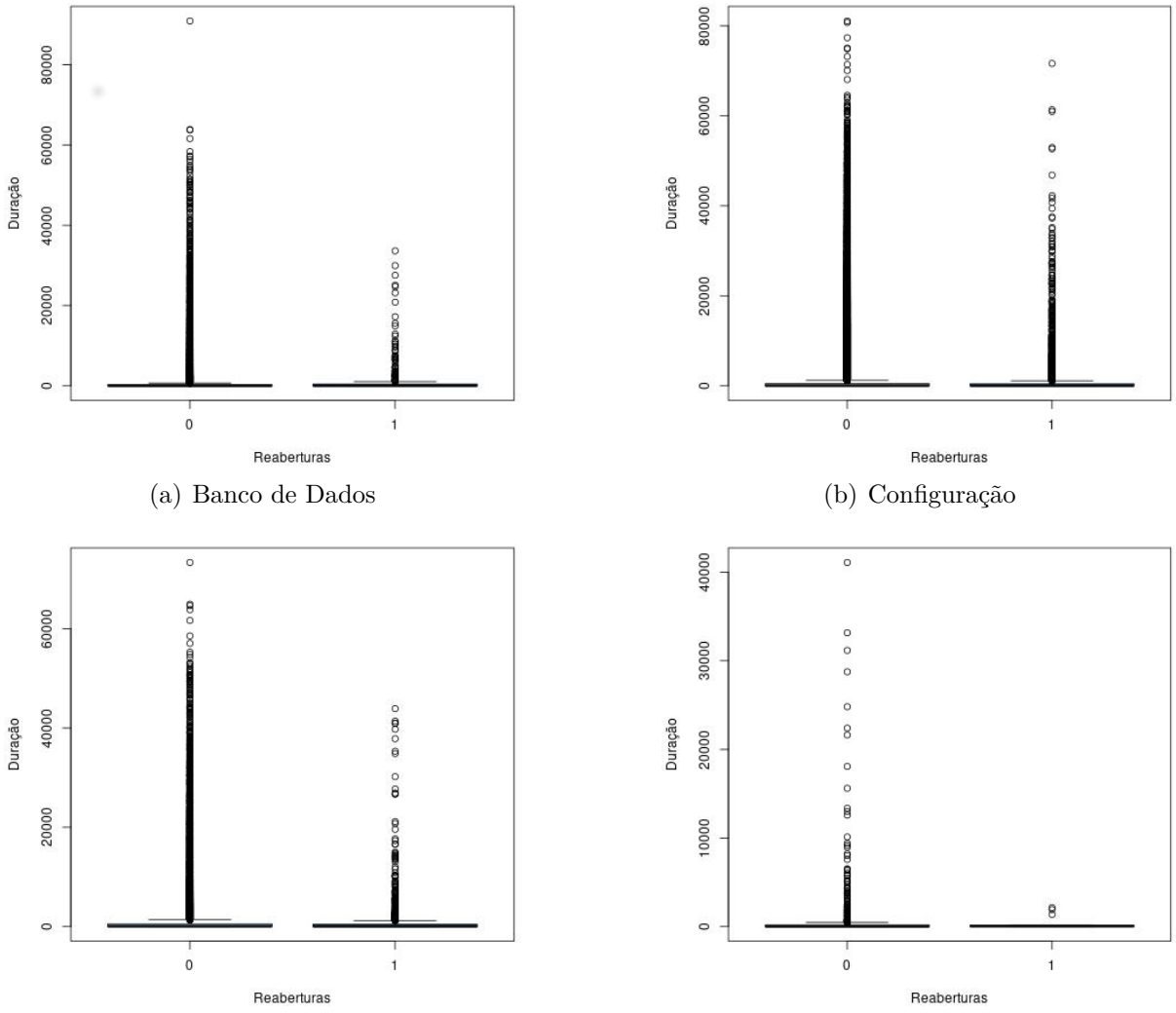
Tabela 4.6: Análise da Taxa de Reabertura por Categoria de Issues

Categoria	**0) Issues sem reabertura**	**1) Issues com reabertura**	**% issues com reabertura**
Base de dados	16980	454	2,60%
Configuração	163263	5365	3,18%
Desempenho	34537	1128	3,16%
Funcional	935	13	1,37%
GUI	102608	3094	2,93%
Info	8339	183	2,15%
Permissão/obsoleto	25671	680	2,58%
Redes	28318	969	3,31%
Segurança	22118	716	3,14%
Testes	73166	1994	2,65%

A tabela 4.6 apresenta a quantidade de issues sem reabertura, a quantidade de issues com reabertura e a porcentagem de issues com reabertura. Observa-se que a porcentagem de reabertura varia entre 1,37% e 3,31% nas diferentes categorias, sendo a categoria

Funcional a de menor porcentagem e *Redes* a de maior. A categoria *Configuração* possui o maior número de issues reabertas, com 5.365 issues (3,18%), enquanto *Funcional* apresenta o menor número, com 13 issues (1,37%).

Em seguida, analisamos o tempo de duração em horas para os grupos de issues com e sem reaberturas de todas as categorias. A distribuição da duração em horas entre a abertura e o fechamento das issues dos grupos com e sem reaberturas é apresentada nas figuras 4.3 e 4.4.Para ambos os grupos, foram calculados o tempo mínimo (Min), o tempo máximo (Max), o primeiro quartil (Q1), o terceiro quartil (Q3), o tempo médio, a mediana e o desvio padrão.



(a) Banco de Dados

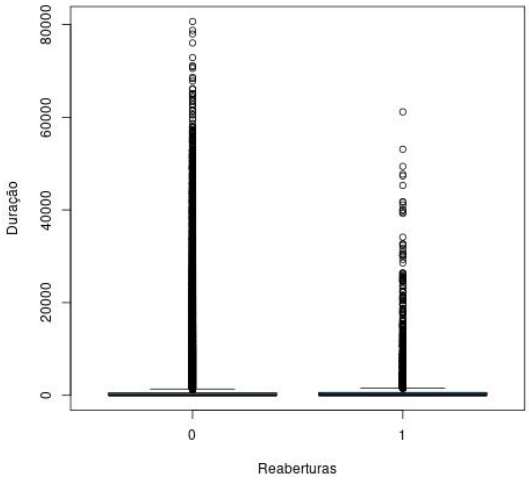
(b) Configuração

(c) Desempenho

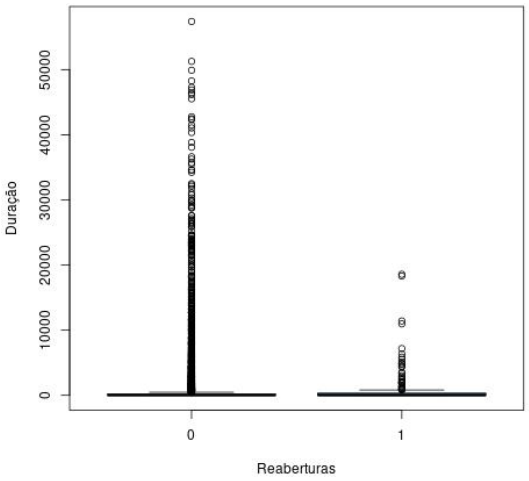
(d) Funcional

Figura 4.3: Distribuição da duração em horas entre a abertura e o fechamento de issues com e sem reaberturas das categorias *Banco de Dados*, *Configuração*, *Desempenho* e *Funcional*.

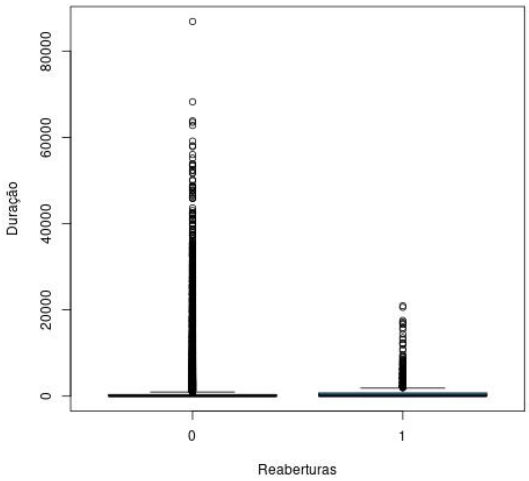
A análise das figuras 4.3 e 4.4 revela variações significativas no tempo de duração



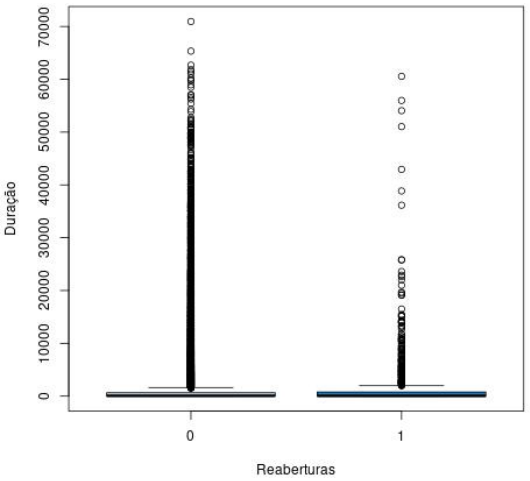
(a) GUI



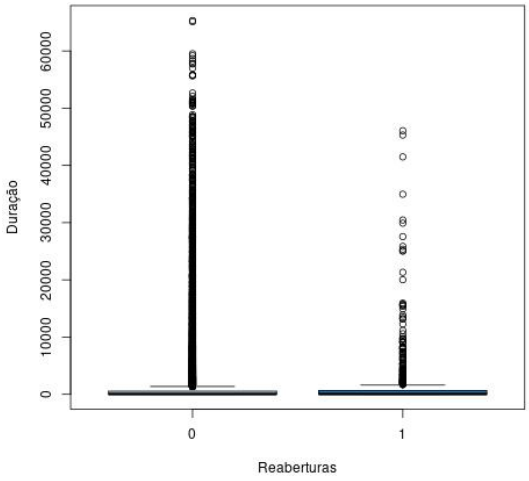
(b) Info



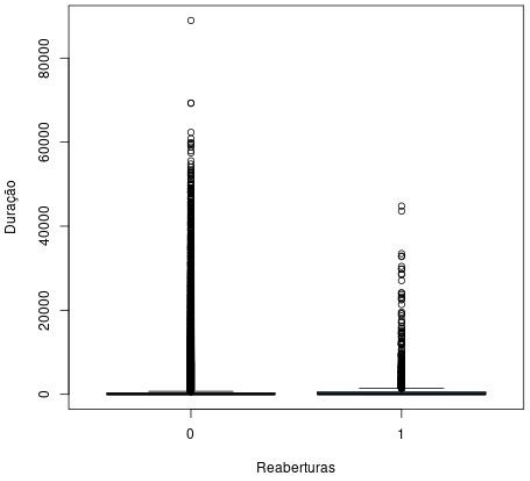
(c) Permissões/Obsoleto



(d) Redes



(e) Segurança



(f) Testes

Figura 4.4: Distribuição da duração em horas entre a abertura e o fechamento de issues com e sem reaberturas das categorias *GUI*, *Info*,*Permissões/Obsoleto*, *Redes*, *Segurança* e *Testes*.

das issues, tanto para os grupos com reaberturas quanto para os sem reaberturas, nas diferentes categorias.

Para as issues sem reabertura, a duração mediana variou de 20,98833 horas na categoria *Informação (Info)* e 63,41846 horas em *Redes*. O tempo médio, por sua vez, oscila entre 691,5928 horas categoria *Funcional* e 1834,15 horas na categoria *Redes*. A categoria *Configuração* apresenta o tempo mínimo mais curto para fechamento, com apenas 0,000278 horas,, enquanto a categoria *Banco de Dados* tem o tempo máximo mais longo, com 71.619,59 horas.

Para as issues com reabertura, o tempo mediano de duração variou de 26,12944 horas na categoria *Informação (Info)* a 69,20056 na categoria *Permissão/Obsoleto*. A categoria *Funcional* mostra o menor tempo médio de fechamento, com 453,2405 horas e um tempo mediano de 40,69028 horas. As issues que foram fechadas mais rapidamente e aquelas que demoraram mais tempo pertencem à categoria *Configuração*, com tempos de 0,00056 horas e 71.619.59 horas, respectivamente.

Ao comparar o tempo de duração das issues com e sem reabertura para todas as categorias, observamos que o tempo mediano é maior para as issues com reaberturas nas categorias *Banco de Dados*, *Funcional*, *Informação (Info)*, *Permissão/Obsoleto*, *Segurança* e *Redes*. Por outro lado, nas categorias *Configuração*, *Desempenho*, *GUI* e *Redes*, o tempo mediano é maior para as issues sem reabertura.

Após essa análise inicial, realizamos a remoção de *outliers* para refinar os resultados.Utilizamos a técnica da amplitude interquartis (IQR) para identificar e excluir valores atípicos, conforme descrito pelas equações (equação 4.5), 4.6 e 4.7. As issues com tempo de duração inferior ao LI ou superior ao LS foram excluídas. A tabela 4.7 apresenta a quantidade de issues sem reaberturas, issues com reaberturas e a porcentagem de reaberturas de cada categoria.

Tabela 4.7: Análise da Taxa de Reabertura por Categoria de Issues sem *outliers*

Categoria	**Issues sem reabertura**	**Issues com reabertura**	**% issues com reabertura**
Banco de dados	13759	372	2.63%
Configuração	134158	4442	3.20%
Desempenho	28455	937	3.19%
Funcional	776	10	1.27%
GUI	83761	2541	2.94%
info	6840	149	2.13%
Permissão/obsoleto	21160	563	2.59%
Redes	23282	800	3.32%
Segurança	18108	593	3.17%
Testes	60459	1664	2.68%

A tabela 4.7 apresenta uma análise da taxa de reabertura por categoria de issues, considerando a remoção de *outliers*. Após esse processo, foram removidas 81.958 (17,86%) issues sem reaberturas e 2.444 (17,28%) issues com reaberturas. A seguir, são destacadas

as alterações nas quantidades nas diferentes categorias em relação à tabela original 4.6:

- Na categoria Banco de Dados, foram removidas 3.221 issues (18,97%) sem reaberturas e 82 (18,06%) issues com reaberturas.
- Na categoria *Configuração*, foram removidas 29.105 issues (17,83%) sem reaberturas e 923 (17,20%) issues com reaberturas.
- Na categoria *Desempenho*, foram removidas 6.083 issues (17,61%) sem reaberturas e 191 (16,93%) issues com reaberturas.
- Na categoria *Funcional*, foram removidas 159 issues (17,01%) sem reaberturas e 3 (23,08%) issues com reaberturas.
- Na categoria GUI, foram removidas 18.848 issues (18,37%) sem reaberturas e 554 (17,91%)issues com reaberturas.
- Na categoria *Informação (Info)*, foram removidas 1.499 issues (17,98%) sem reaberturas e 34 (18,58%) issues com reaberturas.
- Na categoria *Permissão/Obsoleto*, foram removidas 4.511 issues (17,57%) sem reaberturas e 117 (17,21%)issues com reaberturas.
- Na categoria *Redes*, foram removidas 5.036 issues (17,78%) sem reaberturas e 169 (17,44%) issues com reaberturas.
- Na categoria *Segurança*, foram removidas 4.010 issues (18,13%) sem reaberturas e 123 (17,18%) issues com reaberturas
- Na categoria *Testes*, foram removidas 12.707 issues (17,37%) sem reaberturas e 330 (16,55%) issues com reaberturas.

Ao analisar o grupo de issues sem reaberturas, observamos que as categorias *Configuração*, *GUI* e *Teste* se destacam com o maior número de issues, indicando que essas áreas podem enfrentar uma quantidade significativa de problemas ou solicitações. Por outro lado, as categorias *Funcional*, *Banco de Dados* e *Redes* possuem uma menor quantidade de issues, sugerindo uma eficiência maior na resolução desses problemas ou uma menor incidência de questões nessas áreas.

Quando nos deparamos com o grupo de issues com reaberturas, notamos que a porcentagem de issues com reaberturas varia de 2,23% a 4,42%. Surpreendentemente, a categoria *Banco de Dados* destaca-se com a maior porcentagem de issues reabertas (4,42%), enquanto a categoria *Funcional* apresenta a menor (2,23%). Essa variação nas taxas de reabertura sugere que algumas categorias podem enfrentar desafios persistentes, apontando para a necessidade de uma abordagem mais cautelosa ou a implementação de melhorias no processo de resolução de problemas.

Aprofundando nossa análise, categorias como *Banco de Dados*, *Desempenho* e *Segurança* destacam-se com porcentagens mais elevadas de issues reabertas. Essa observação

sugere que essas áreas específicas podem demandar uma atenção adicional em termos de qualidade e eficácia na resolução de issues. Diante desse cenário, é possível direcionar esforços de melhoria contínua e realizar uma alocação estratégica de recursos. Essa abordagem visa otimizar o processo de gerenciamento de issues, buscando promover uma resolução mais eficiente e eficaz em todas as categorias. As figuras 4.5 e 4.6 mostra a distribuição da duração em horas entre a abertura e o fechamento de issues com e sem reaberturas, considerando a exclusão de *outliers*. Após a remoção dos *outliers* conseguimos observar uma redução significativa no tempo mediano da duração das issues com e sem reaberturas em todas as categorias.

Para as issues sem reabertura, a categoria *Desempenho* apresentou uma mediana mais alta, indicando um tempo médio mais longo para a resolução dessas issues. Em contrapartida, a categoria *Banco de Dados* exibe uma mediana mais baixa, sugerindo uma resolução mais rápida das issues nessa categoria. Embora a duração mínima em todas as categorias não tenha sido alterada, as variações nas durações máximas, que variam de 471,2739 horas na categoria *Funcional* a 1627,322 horas na categoria *Redes*, oferecem insights valiosos sobre a amplitude dos tempos de resolução em diferentes áreas. Ao considerar o terceiro quartil (Q3), que varia entre 49,77618 horas na categoria *Funcional* e 163,6639 horas na categoria *Redes*, percebemos que 75% das issues dentro de cada categoria são resolvidas dentro desse intervalo. Essa análise aprofundada por categoria fornece informações cruciais para orientar estratégias de otimização e aprimoramento contínuo nos processos de resolução de issues.

Nas issues com reaberturas, as categorias *Testes* e *Permissão/Obsoleto* apresentam os maiores valores medianos, com 29,29347 e 26,30583 horas, respectivamente. Em contrapartida a categoria *Info* possui o menor valor mediano com 9,85333 horas. Embora o valor mínimo não tenha sido alterado, o valor máximo teve uma grande redução, variando entre 820,1219 e 1976,571 horas nas categorias *Info* e *Redes*.

Identificamos que as issues com reaberturas das categorias *Banco de dados*, *GUI*, *Segurança* e *Testes* possuem a duração mediana maior do que no grupo de issues sem reaberturas. Por outro lado, nas categorias *Configuração*, *Desempenho*, *Funcional*, *Info*, *Permissão/Obsoleto* e *Redes*, a duração mediana foi menor.

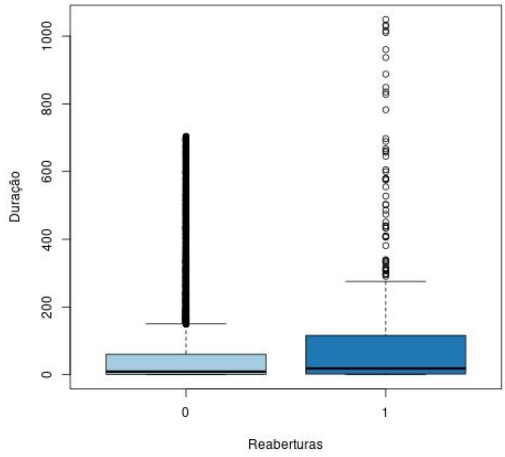
Em seguida, realizamos uma análise do número de comentários entre a abertura e o fechamento das issues com e sem reaberturas em todas as categorias, como são apresentadas nas figuras 4.7 e 4.8.

Observamos que as issues com e sem reaberturas, em todas as categorias apresentaram o valor mínimo e o primeiro quartil (Q1) com 2 comentários, que o número mínimo para selecionar as issues para o experimento.

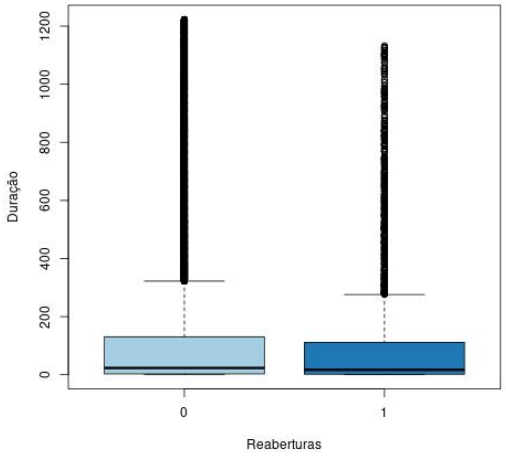
A maioria das categorias possui a mediana com 3 comentários, exceto as categorias *Banco de dados* e *Info*, que possuem 2 comentários, indicando um número limitado de comentários entre a abertura e o fechamento.

Em relação ao terceiro quartil (Q3), percebemos que a maioria das categorias possui 5 comentários, com exceção das categorias *Banco de dados*, *Funcional* e *Info*, que possuem 4 comentários. Notamos também que a categoria *Configuração* apresenta o maior número de comentários em uma única issue, com 521.

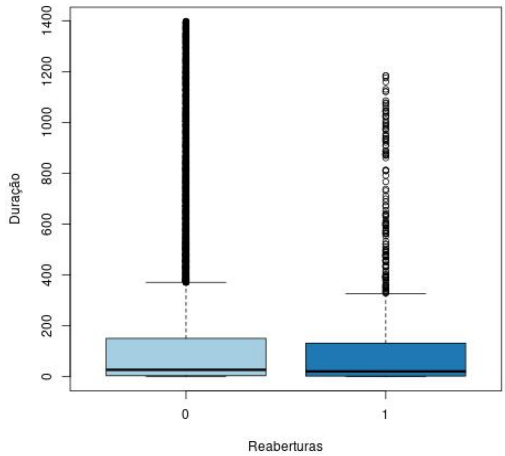
A mediana na maioria das categorias é 3 comentários, exceto a categoria *Funcional*,



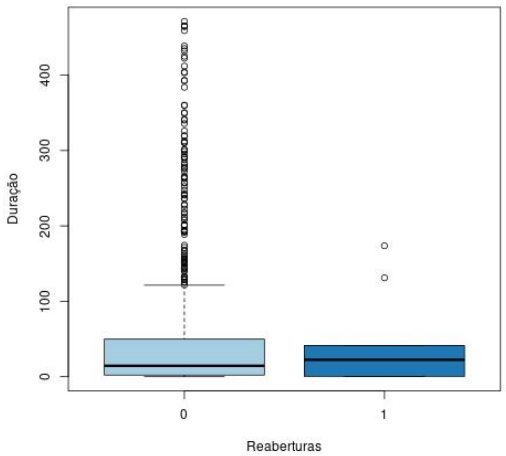
(a) Banco de Dados



(b) Configuração

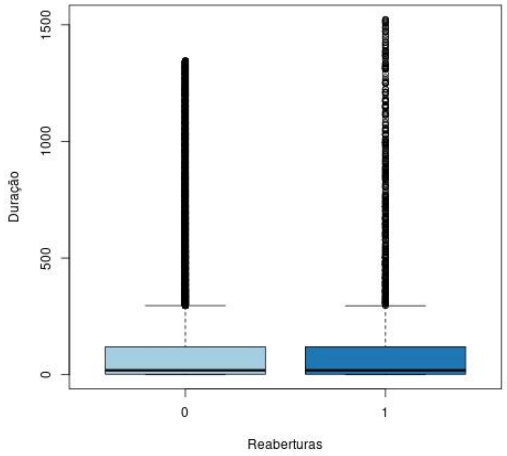


(c) Desempenho

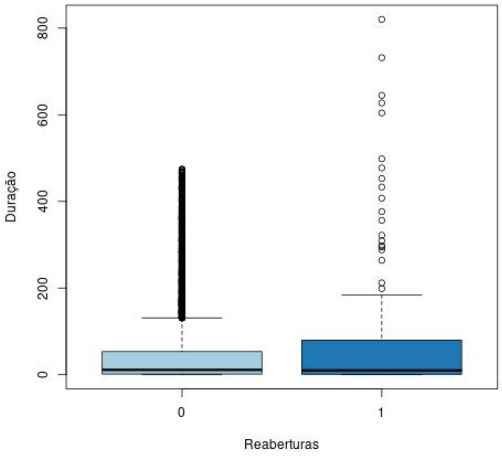


(d) Funcional

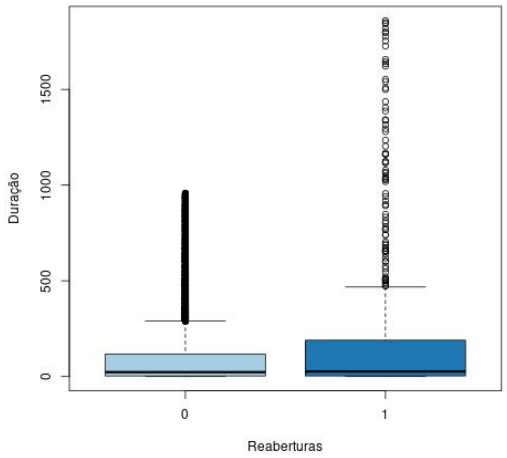
Figura 4.5: Distribuição da duração em horas entre a abertura e o fechamento de issues com e sem reaberturas das categorias *Banco de Dados*, *Configuração*, *Desempenho* e *Funcional* após a remoção de *outliers*.



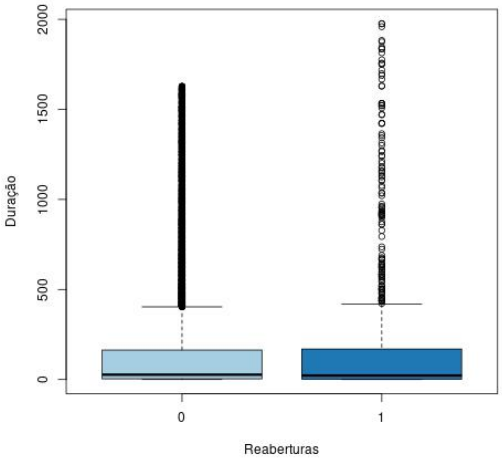
(a) GUI



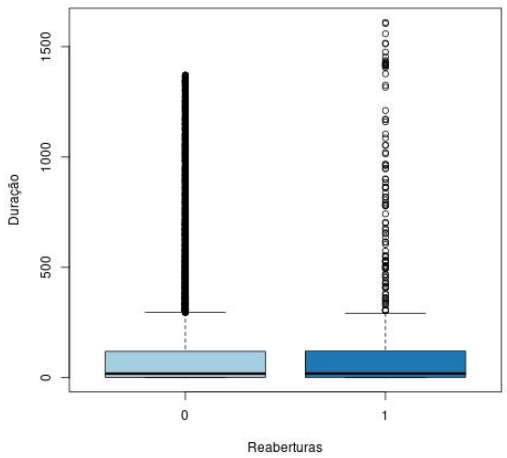
(b) Info



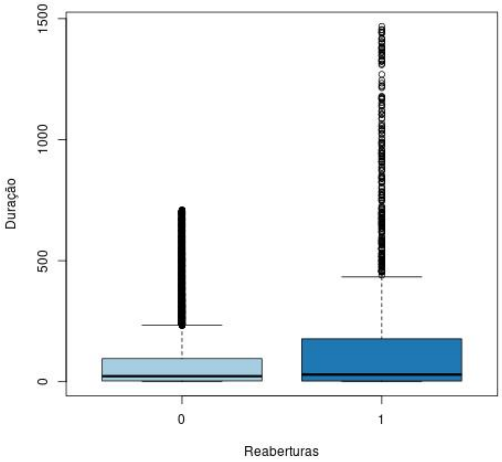
(c) Permissões/Obsoleto



(d) Redes



(e) Segurança



(f) Testes

Figura 4.6: Distribuição da duração em horas entre a abertura e o fechamento de issues com e sem reaberturas das categorias *GUI*, *Info*,*Permissões/Obsoleto*, *Redes*, *Segurança* e *Testes* após a remoção de outliers.

que possui 2,5, indicando poucos comentários entre a abertura e o fechamento.

O número de comentários no terceiro quartil varia entre 3 e 6, sendo a categoria *Funcional* a com o menor valor, e as categorias *Configuração*, *Desempenho*, *Permissão/Obsoleto* e *Testes* com o maior valor. Em comparação com a tabela B.5, observamos que o valor mínimo e o primeiro quartil nos dois grupos de issues, sem e com reaberturas, são idênticos em todas as categorias. O valor mediano nas categorias *Banco de dados*, *Funcional* e *Info* no grupo de issues com reaberturas teve um leve aumento, enquanto nas demais permaneceu inalterado. No terceiro quartil do grupo de issues com reaberturas, as categorias *Banco de dados*, *Configuração*, *Desempenho*, *Funcional*, *Info*, *Permissão/Obsoleto* e *Testes* experimentaram um pequeno aumento no número de comentários, enquanto as categorias *GUI*, *Redes* e *Segurança* mantiveram a mesma quantidade. Para valores descritivos mais detalhados, referentes ao tempo de duração e ao número de comentários entre a abertura e o fechamento das issues com e sem reaberturas nas diferentes categorias, consulte as tabelas no Apêndice B.

4.2.3 Análise e Discussão dos Resultados

Nesta seção, discutiremos os resultados em função da questão de pesquisa QP3 e suas questões de pesquisa derivadas QP3.2 e QP3.3.

QP3. Quais categorias estão mais associadas à reabertura de issues? Realizamos uma análise comparativa das categorias de issues com e sem reaberturas. Observamos que a frequência de reaberturas varia entre as categorias, sugerindo que algumas são mais propensas a reaberturas do que outras.

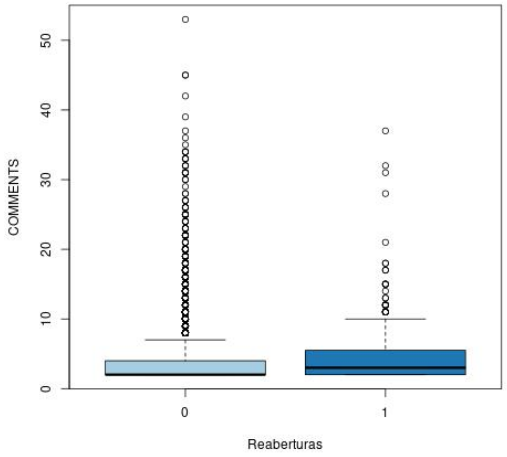
A categoria *Configuração* se destaca por apresentar um número relativamente alto de issues reabertas. Isso pode ser atribuído à complexidade e à necessidade contínua de ajustes nas configurações, exigindo mais ciclos de revisão. As categorias *Desempenho* e *Segurança* também mostram uma tendência significativa de reabertura. Issues de *Desempenho* frequentemente requerem múltiplas iterações para otimização, enquanto questões de *Segurança* podem ser reabertas devido à necessidade de revisões e correções contínuas.

Essas observações indicam que categorias que envolvem complexidade técnica ou requisitos críticos podem estar mais associadas à reabertura de issues. As características específicas dessas categorias, como a complexidade das tarefas ou a criticidade das funções, podem contribuir para uma maior probabilidade de reabertura.

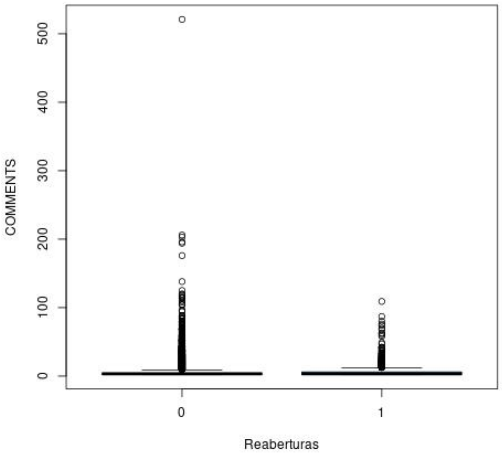
QP3.2. O tempo de duração entre a abertura e o primeiro fechamento de uma issue é um indicativo de reabertura nas diferentes categorias? A análise do tempo entre a abertura e o primeiro fechamento das issues revelou variações significativas nas medianas entre categorias e entre issues com e sem reabertura.

As categorias *Desempenho* e *Redes* apresentam medianas de duração mais altas para issues reabertas, sugerindo que essas questões podem ser mais complexas e levar mais tempo para serem resolvidas inicialmente. A maior complexidade pode estar associada a uma maior probabilidade de reabertura, indicando que questões mais complexas podem não ser completamente resolvidas no primeiro ciclo.

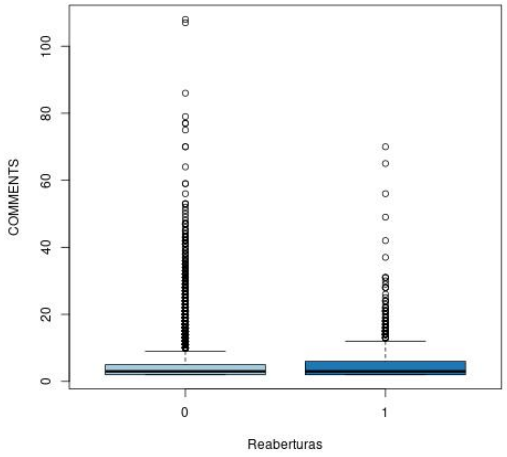
Por outro lado, categorias como *Banco de Dados* e *Info* mostram medianas de duração



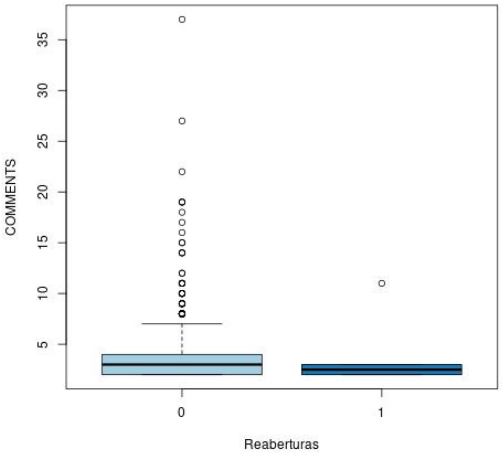
(a) Banco de Dados



(b) Configuração

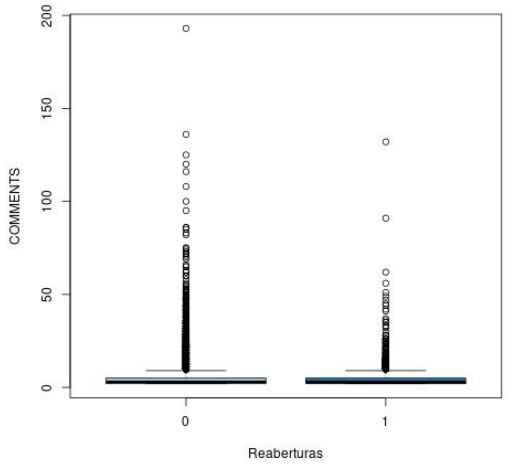


(c) Desempenho

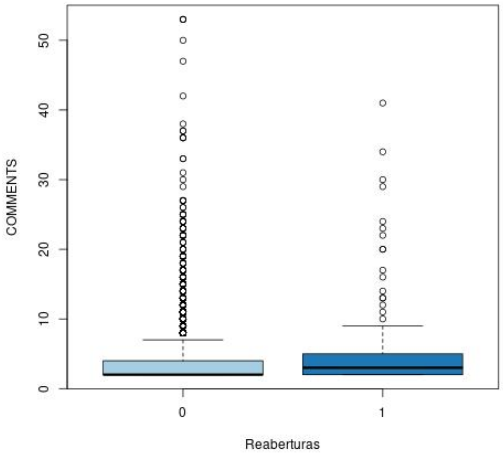


(d) Funcional

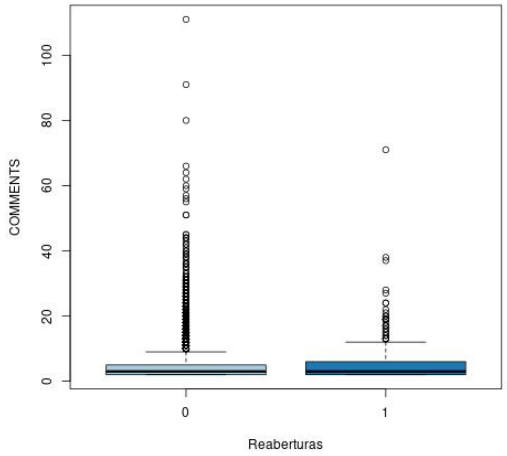
Figura 4.7: Distribuição da duração em horas entre a abertura e o fechamento de issues com e sem reaberturas das categorias *Banco de Dados*, *Configuração*, *Desempenho* e *Funcional* após a remoção de outliers.



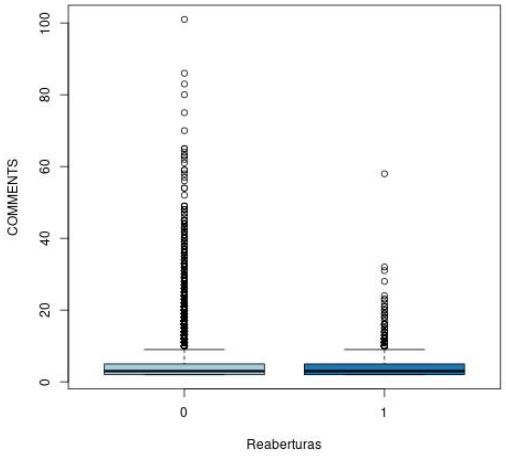
(a) GUI



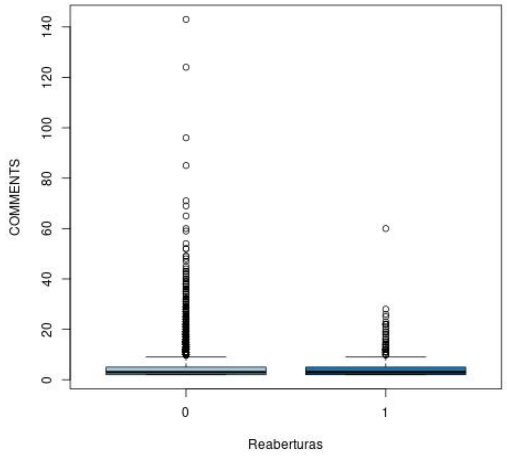
(b) Info



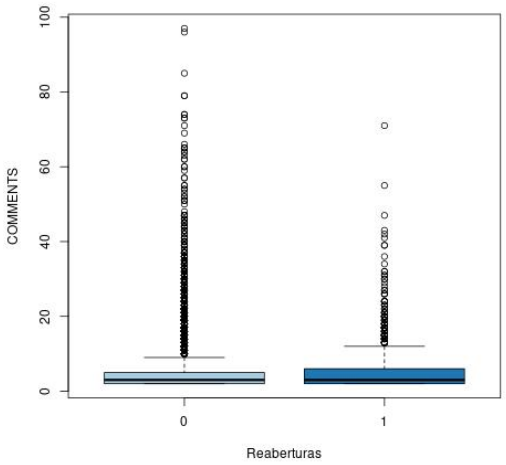
(c) Permissões/Obsoleto



(d) Redes



(e) Segurança



(f) Testes

Figura 4.8: Distribuição da duração em horas entre a abertura e o fechamento de issues com e sem reaberturas das categorias *GUI*, *Info*,*Permissões/Obsoleto*, *Redes*, *Segurança* e *Testes* após a remoção de outliers.

mais baixas, indicando que issues nessas categorias são resolvidas mais rapidamente e, portanto, têm menor probabilidade de reabertura.

Esses resultados sugerem que o tempo de duração entre a abertura e o primeiro fechamento pode indicar a probabilidade de reabertura, especialmente em categorias mais complexas. Issues que permanecem abertas por períodos mais longos antes do fechamento podem ser mais propensas a reaberturas, refletindo a necessidade de uma solução mais robusta ou ajustes adicionais.

QP3.3. A quantidade de comentários entre a abertura e o primeiro fechamento de uma issue pode influenciar sua probabilidade de reabertura nas diferentes categorias? A análise da quantidade de comentários entre a abertura e o primeiro fechamento das issues fornece uma perspectiva sobre a interação e o nível de discussão necessário para resolver as issues.

Nas categorias *Configuração*, *Desempenho* e *Testes*, que apresentam um maior número de comentários em issues reabertas, a alta interação pode refletir discussões extensas e tentativas de resolver problemas complexos. Essa alta quantidade de comentários pode estar associada a uma maior probabilidade de reabertura, indicando que mais interações sugerem uma maior complexidade ou necessidade de refinamento adicional.

Em contraste, as categorias *Funcional* e *Info* apresentam medianas menores de comentários, o que pode sugerir uma resolução mais direta e menos complexa, resultando em menor necessidade de reabertura.

Esses achados sugerem que a quantidade de comentários pode influenciar a probabilidade de reabertura, especialmente nas categorias onde um maior número de interações é necessário para resolver a issue. Issues com mais comentários podem estar associadas a problemas mais complexos e, portanto, podem ser mais propensas a serem reabertas.

4.2.4 Ameaças à validade

Validade Interna. O processo de pré-processamento dos dados pode ter introduzido vieses, especialmente na remoção de ruído ou na filtragem de informações, o que poderia impactar a categorização das issues. No entanto, essa ameaça é mitigada, pois o pré-processamento foi padronizado e seguiu os procedimentos que foram previamente bem-sucedidos em experimentos de categorização similares, minimizando o risco de vieses ou perda de dados relevantes.

A remoção de *outliers* baseada na duração entre a abertura e o primeiro fechamento das issues pode ter eliminado dados potencialmente relevantes que poderiam influenciar a análise. Essa ameaça é refutada pelo uso da técnica de Amplitude Interquartil (IQR), que é uma abordagem estatisticamente robusta para a detecção e remoção de *outliers*. Essa técnica minimiza o impacto de valores extremos sem distorcer significativamente a amostra.

Validade Externa. A base de dados MSR2014 pode não refletir completamente o comportamento de issues em outros repositórios ou em domínios diferentes, e a coleta de dados foi limitada até o ano de 2020, o que pode não refletir práticas mais recentes. No entanto, essa ameaça é mitigada pela robustez e diversidade da base MSR2014, que

inclui um conjunto variado de issues de diferentes tipos de projetos. A seleção dessa base foi feita com base em sua abrangência e representatividade de múltiplos domínios de software, o que fortalece a generalização dos resultados e reduz o potencial de viés devido a limitações temporais ou à seleção dos repositórios.

Validade Construto. O classificador MLP pode não capturar toda a complexidade envolvida na categorização das issues, especialmente na previsão de reabertura. Essa ameaça é mitigada, pois o MLP foi selecionado com base em seu desempenho superior em experimentos anteriores, destacando-se em métricas de acurácia, precisão, revocação e F1-score. Apesar do balanceamento aplicado durante a criação do classificador, a escolha do MLP é justificada pela sua alta performance.

Além disso, a categorização das issues pode ser considerada simplista e não capturar nuances detalhadas, como tipos específicos de reabertura ou razões subjacentes. No entanto, essa ameaça é mitigada pela definição clara da categorização, que se baseia em uma taxonomia específica e foi aplicada de maneira consistente em ambas as análises dos grupos com e sem reabertura. A taxonomia de Catolino et al. (2019) inclui categorias como Banco de Dados, Configuração, Desempenho, Funcional, GUI, Informação, Permissão/Obsoleto, Redes, Segurança e Testes, e foi utilizada para classificar as issues. Analisamos os grupos de issues com e sem reabertura dentro dessas categorias, considerando o tempo de duração entre abertura e fechamento e o número de comentários entre esses eventos. Embora a categorização possa não capturar todas as nuances possíveis, a utilização de uma taxonomia bem definida e a análise detalhada dos fatores temporais e quantitativos proporcionam uma base sólida e consistente para a avaliação das issues.

Validade de Conclusão. As análises de tempo de duração e quantidade de comentários podem não capturar todos os fatores que influenciam a reabertura de uma issue. Para mitigar essa ameaça, foram utilizadas métricas estatísticas robustas (mínimo, máximo, média, mediana, desvio padrão, Q1 e Q3), adequadas para capturar diferenças entre os grupos de issues. Além disso, a remoção de *outliers* assegura que os resultados não sejam distorcidos por valores extremos, proporcionando uma visão mais precisa dos padrões de comportamento das issues.

A diferença no número de issues entre os grupos com e sem reaberturas pode influenciar os resultados. No entanto, essa ameaça foi mitigada ao utilizar todos os dados disponíveis, refletindo a realidade dos repositórios estudados sem a necessidade de balanceamento artificial. Esse enfoque permite uma análise mais fiel dos comportamentos naturais das issues e evita distorções nos resultados.

4.2.5 Conclusão

Este experimento forneceu uma análise aprofundada das associações entre categorias de issues, o tempo de duração entre a abertura e o primeiro fechamento, e a quantidade de comentários com a probabilidade de reabertura de issues. Utilizando um categorizador MLP treinado com dados balanceado com a técnica *SMOTEENN* para realizar a análise.

Os resultados mostraram que as categorias *Configuração*, *Desempenho* e *Segurança* apresentam uma maior probabilidade de reabertura em comparação com outras catego-

rias. Isso pode ser atribuído à sua complexidade técnica e criticidade, indicando que essas issues frequentemente precisam de mais ciclos de revisão e ajustes. Um gerenciamento mais cuidadoso pode ser necessário para reduzir as reaberturas nessas áreas.

Além disso, observamos que issues com maior tempo de duração antes do fechamento inicial, especialmente nas categorias *Desempenho* e *Redes*, são mais propensas a serem reabertas. Isso sugere que problemas complexos que demoram mais para serem resolvidos inicialmente têm mais chances de aparecer novamente, o que destaca a necessidade de uma solução mais completa.

A quantidade de comentários também influenciou a probabilidade de reabertura. Issues com mais comentários, como nas categorias *Configuração* e *Testes*, mostraram maior probabilidade de reabertura, indicando que uma maior interação e discussão podem refletir a necessidade de ajustes adicionais. Em resumo, nosso estudo identificou que a categoria, o tempo de duração e a quantidade de comentários são fatores importantes que afetam a probabilidade de reabertura de issues.

4.3 CONCLUSÃO DO CAPÍTULO

A categorização de issues apresenta-se como uma promissora abordagem para analisar reabertura de issues. Ademais, analisamos os resultados da seção 4.2.2, identificamos que a categoria que mais possui issues reabertas foi a categoria redes com 3,32%, essa categoria possui mais que o dobro da categoria funcional que possui 1,27% issues reabertas, entretanto essa categoria possui apenas 10 issues reabertas que um número muito pequeno em relação as demais categorias.

No capítulo 5 apresentamos por tanto uma análise de reabertura de issues em relação aos sentimentos dos colaboradores presentes nas discussões e o sua categoria de issue.

Capítulo 5

CARACTERIZAÇÃO DE REABERTURA DE ISSUES

Neste capítulo, examinaremos como os sentimentos presentes nas discussões de issues desempenham um papel na decisão de reabrir uma issue e como as diferentes categorias de issues podem afetar essa decisão. O objetivo principal é conduzir um estudo empírico para avaliar a influência dos sentimentos em issues reabertas em projetos do GitHub. Este estudo visa explorar a relação entre os sentimentos expressos em diferentes categorias de issues e a probabilidade de reabertura dessas issues. O objetivo principal deste capítulo é conduzir um estudo empírico para avaliar os sentimentos de acordo com as categorias de issues reabertas em projetos do GitHub (Obj 7). Para atingir esse objetivo, abordamos as seguintes questões de pesquisa: QP4. Como caracterizar os sentimentos nas diferentes categorias no contexto de reaberturas de issues? QP4.1. Como o sentimento, a pontuação negativa e a pontuação positiva do primeiro comentário após a abertura da issue influenciam a reabertura nas diferentes categorias? QP4.2. Como o sentimento, a pontuação negativa e a pontuação positiva do último comentário antes do primeiro fechamento da issue influenciam a reabertura nas diferentes categorias? QP4.3. Como os sentimentos, pontuações negativas e positivas entre a abertura e o primeiro fechamento da issue influenciam a reabertura nas diferentes categorias?

A metodologia empregada para este estudo é detalhada na Seção 5.1, enquanto a Seção 5.2 apresenta os resultados e considerações finais. Este capítulo visa fornecer uma compreensão aprofundada de como as emoções expressas em diferentes momentos e categorias de issues podem impactar a decisão de reabertura, oferecendo insights valiosos para a gestão e análise de projetos de software.

5.1 METODOLOGIA

O estudo possue quatro etapas: seleção dos repositórios e extração das issues, análise de sentimentos, categorização de issues e análise. A figura 5.1 apresenta as etapas do estudo e cada etapa é descrita a seguir.

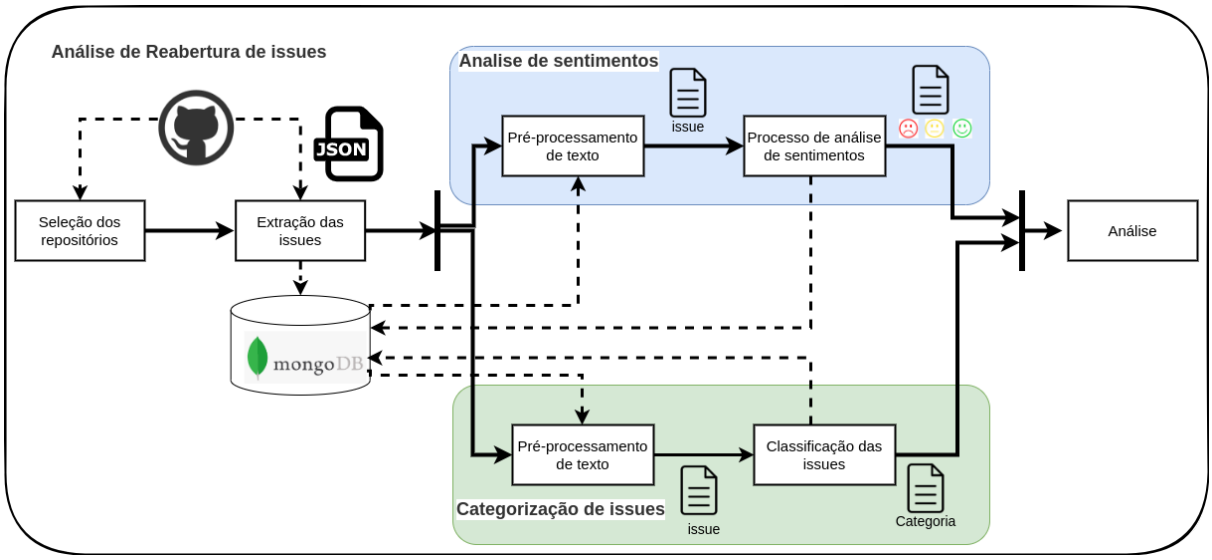


Figura 5.1: Etapas do estudo.

5.1.1 Seleção e Extração das issues

Selecionamos a base de dados MRS2014 composta por issues do GitHub dos repositórios listados no desafio MRS Challenge Dataset da conferência MSR do ano de 2014, as etapas para criação da base de dados são descritas nas seções 3.4.1.1 e 3.4.1.2.

5.1.2 Análise de Sentimentos

O processo de análise de sentimento é composto pelas etapas de pré-processamento e análise de sentimentos dos textos contidos no título, descrição e comentários das issues. Os passos da etapa de pré-processamento dos textos estão descritos nas seções 3.3.1.2 e 3.4.1.3. E nos passos da análise de sentimentos utilizaremos a ferramenta SentiStrength-SE para calcular pontuações negativas, pontuações positivas e o sentimento como descrito na seção 3.4.1.4.

5.1.3 Categorização de issues

O processo de categorização das issues é realizado a partir dos textos do seu título e da sua descrição, para isso são realizados as etapas de pré-processamento de texto descrito na Seção 4.1.1.2 e de categorização das issues utilizando o modelo de classificação MLP apresentado nas seções 4.1.1.4 e 4.2.1.3.

5.1.4 Análise de Reabertura de issues

Nessa etapa, consideramos a ordem cronológica dos eventos de abertura (*Open*) e o primeiro fechamento (*Closed*) das issues, como descrito na seção 3.4.1.6. Nossa análise se concentra na avaliação dos sentimentos e nas pontuações negativas e positivas das discussões das issues (conforme descrito na seção 5.1.2). Além disso, classificamos as

issues nas categorias: Banco de Dados, Configuração, Desempenho, Funcional, Interface Gráfica do Usuário (GUI), Informação (Info), Permissão/Obsoleto (Perm/Obs), Redes, Segurança e Teste, conforme discutido na seção 5.1.3.

Para realizar essa análise, faremos uso das medidas de tendência central, incluindo o valor mínimo (Min), o valor máximo (Max), a média, a mediana (Med), o primeiro quartil (Q1) e o terceiro quartil (Q3), das seguintes métricas descritas na seção 3.4.1.6, que são as seguintes:

- Primeiro comentário após a abertura da issue (conforme ilustrado na figura 5.2) : pontuação negativa (N_{PC}), pontuação positiva (P_{PC}), intensidade do sentimento (S_{PC}) e o sentimento SP_{PC}.

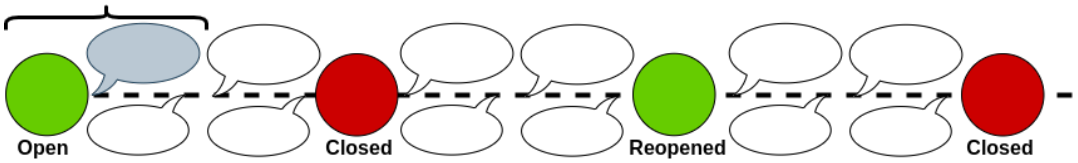


Figura 5.2: Primeiro comentário após a abertura de uma issue reaberta.

- Último comentário antes do fechamento da issue (conforme mostrado na figura 5.3): pontuação negativa (N), pontuação positiva (P), intensidade do sentimento (S) e o sentimento (SP)

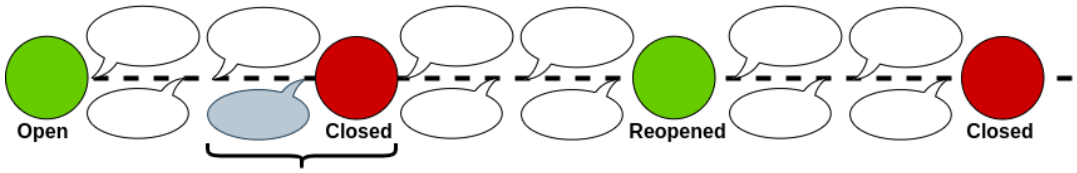


Figura 5.3: Último comentário antes do primeiro fechamento de uma issue reaberta.

- Comentários realizados entre a abertura e o fechamento da issue (conforme evidenciado na figura 5.4): pontuação negativa média (NM), pontuação positiva média (PM), densidade negativa (DCN), densidade positiva (DCP) e densidade neutra ($DCNEU$).

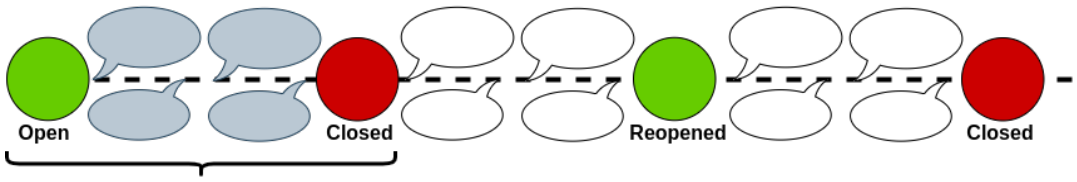


Figura 5.4: Comentários entre a abertura e o primeiro fechamento de uma issue reaberta.

5.2 RESULTADOS

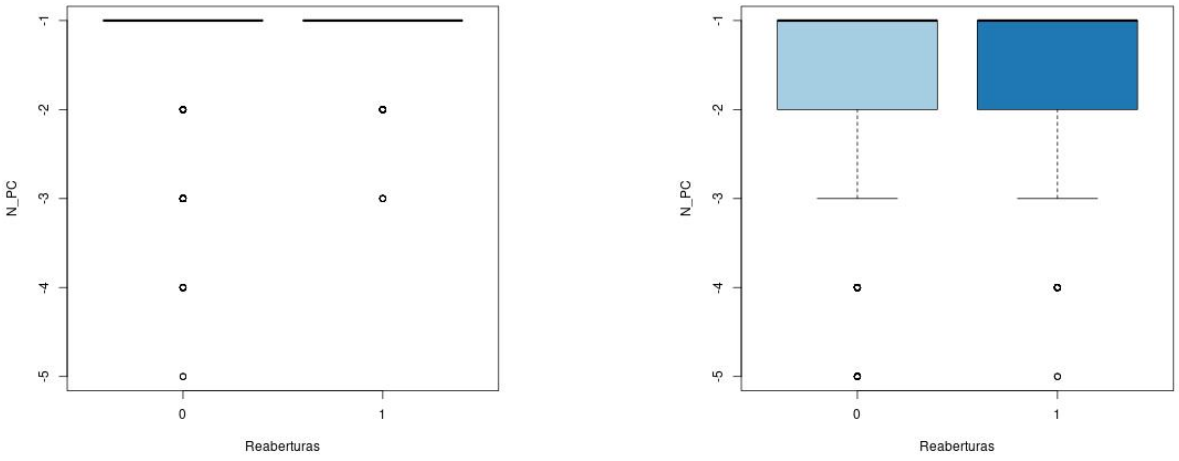
Analisamos um total de 390.757 issues sem reaberturas e 12.071 issues com uma ou mais reaberturas, categorizadas em *Banco de Dados*, *Configuração*, *Desempenho*, *Funcional*, *Interface Gráfica do Usuário (GUI)*, *Informação (Info)*, *Permissão/Obsoleto*, *Redes*, *Segurança* e *Teste*, conforme discutido na seção 4.2.2 e apresentada na tabela 4.7. A seguir, detalhamos os resultados obtidos:

5.2.1 Primeiro comentário

Extraímos as pontuações negativa (N_{PC}), positiva (P_{PC}), intensidade do sentimento (SP_{PC}) e o sentimento geral (S_{PC}) a partir do texto do primeiro comentário feito após a abertura das issues, considerando os grupos com e sem reaberturas nas categorias *Banco de Dados*, *Configuração*, *Desempenho*, *Funcional*, *Interface Gráfica do Usuário (GUI)*, *Informação (Info)*, *Permissão/Obsoleto*, *Redes*, *Segurança* e *Teste*.

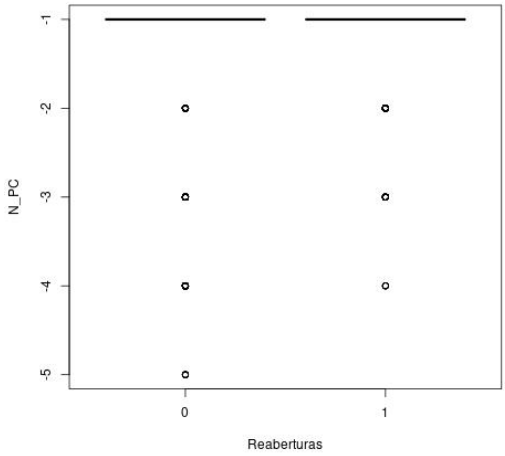
As distribuições dos resultados da métrica N_{PC} estão ilustradas nas nas figuras 5.5, 5.6 e 5.7. Observa-se que, em todas categorias de issues com e sem reabertura, a mediana, o terceiro quartil (Q3) e o valor máximo da pontuação negativa do primeiro comentário após a abertura (N_{PC}) são predominantemente iguais a -1. Esse padrão sugere uma baixa ocorrência de palavras ou expressões com conotação negativa nos comentários iniciais.

Um destaque relevante é a categoria *Configuração*, onde, tanto para issues com reaberturas quanto nas issues sem reaberturas, o Q1 apresenta a pontuação -2. Isso indica uma presença maior de termos negativos nessa categoria, em contraste com as demais. Na categoria *Funcional*, para o grupo de issues com reaberturas, todas as medidas apresentam valores iguais a -1, sugerindo que essas issues tendem a receber pontuações menos negativas no primeiro comentário em comparação às outras categorias.

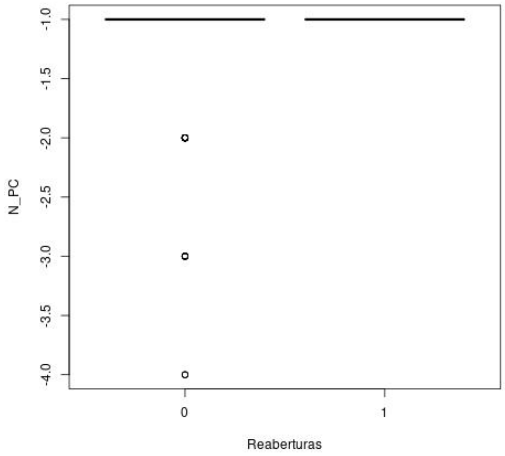


(a) Banco de Dados (b) Configuração

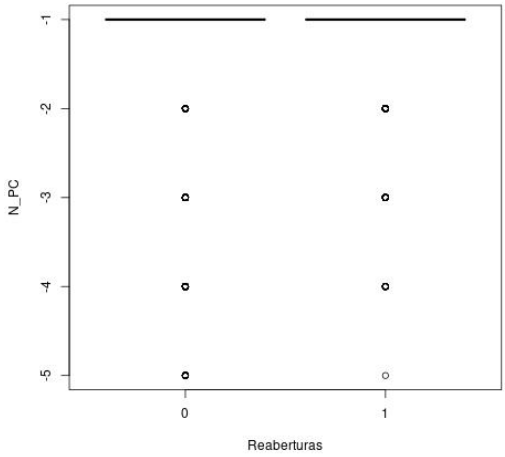
Figura 5.5: Distribuição de N_{PC} das categorias *Banco de Dados* e *Configuração*.



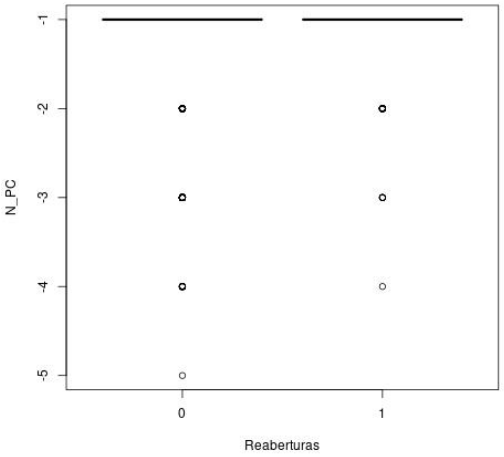
(a) Desempenho



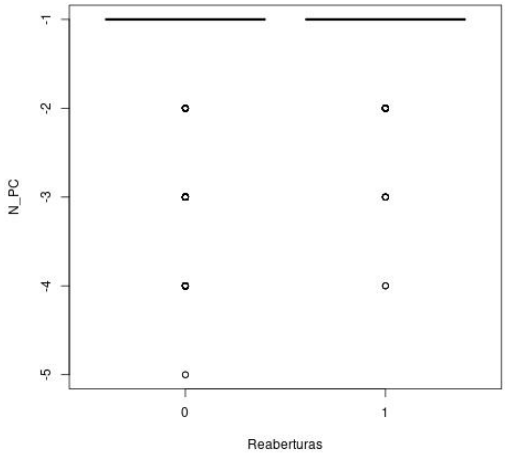
(b) Funcional



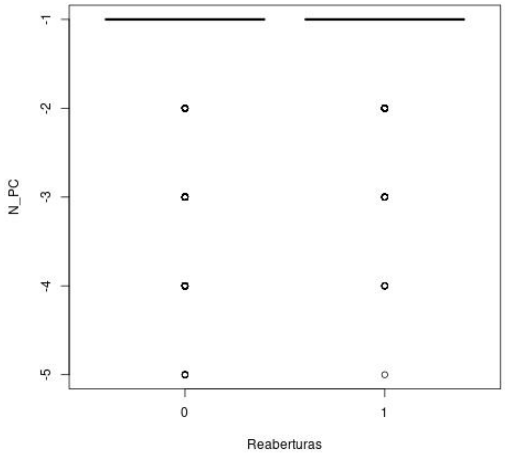
(c) GUI



(d) Info

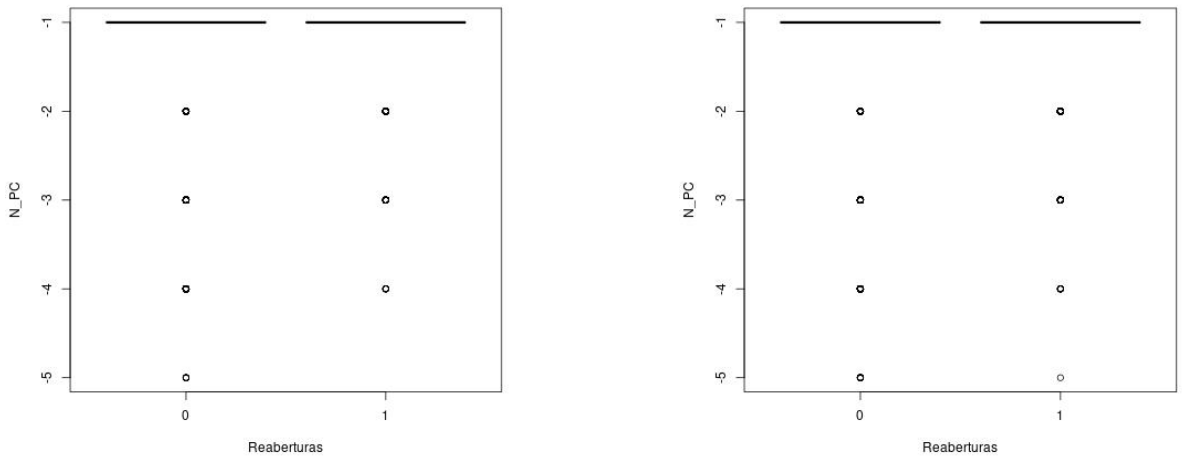


(e) Permissões/Obsoleto



(f) Redes

Figura 5.6: Distribuição de N_{PC} das categorias *Desempenho* e *Funcional*, *GUI*, *Info*,*Permissões/Obsoleto* e *Redes*.



(a) Segurança (b) Testes

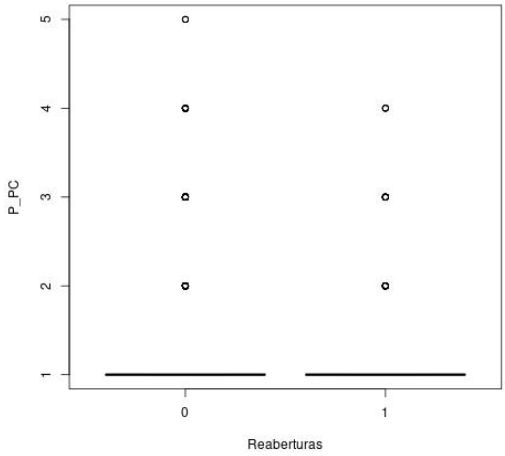
Figura 5.7: Distribuição de N_{PC} das categorias *Segurança* e *Testes*.

Analisamos também a pontuação positiva do primeiro comentário (P_{PC}) para cada categoria de issues, tanto nos grupos com quanto sem reaberturas. Os resultados são apresentados nas figuras 5.8 e 5.9.

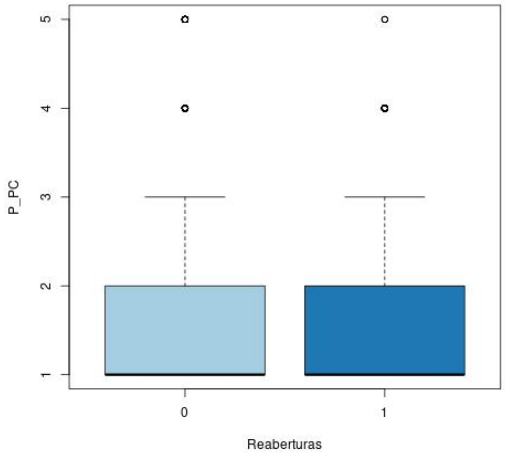
Para o grupo de issues sem reaberturas, todas as categorias apresentaram valores mínimos, medianas e Q1 iguais a 1, indicando a ausência de palavras ou tokens positivos no primeiro comentário dessas issues. A média das pontuações variou de 1,067 na categoria *Funcional* a 1,366 na categoria *Interface Gráfica do Usuário (GUI)*. Nas categorias *Configuração* e *GUI*, o Q3 atingiu o valor 2, sugerindo que, embora 75% das issues apresentem pouca ou nenhuma positividade, algumas alcançam uma moderada presença de termos positivos. As pontuações máximas em algumas categorias chegaram a 5.

No grupo de issues com reaberturas, o padrão é semelhante: as pontuações mínimas, medianas e Q1 também são 1, reforçando a ausência de palavras positivas nos primeiros comentários. No entanto, categorias como *Configuração*, *GUI*, *Redes* e *Segurança* se destacam por apresentar valores de $Q3$ iguais a 2, indicando uma leve presença de positividade em 25

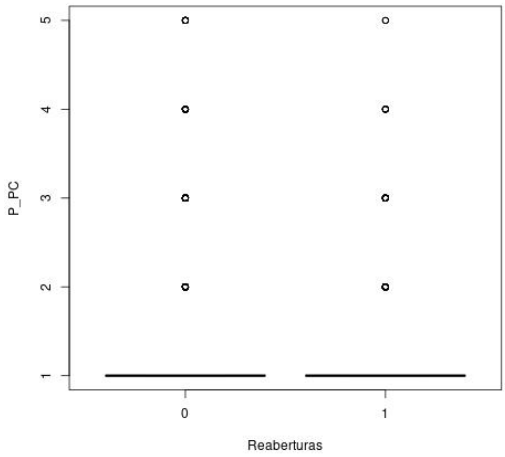
Esses resultados mostram que, em ambos os grupos (com e sem reaberturas), o primeiro comentário geralmente carece de positividade, embora uma pequena fração das issues apresente comentários iniciais com maior presença de termos positivos.



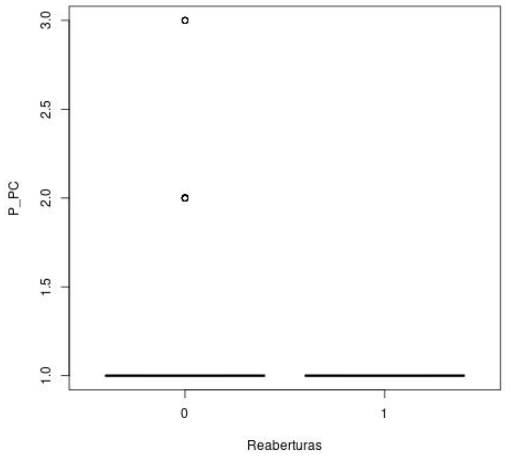
(a) Banco de Dados



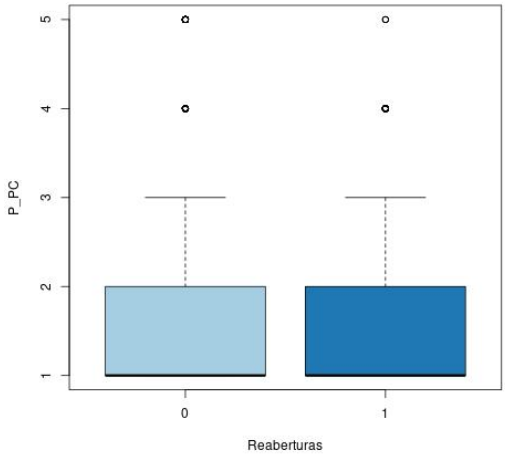
(b) Configuração



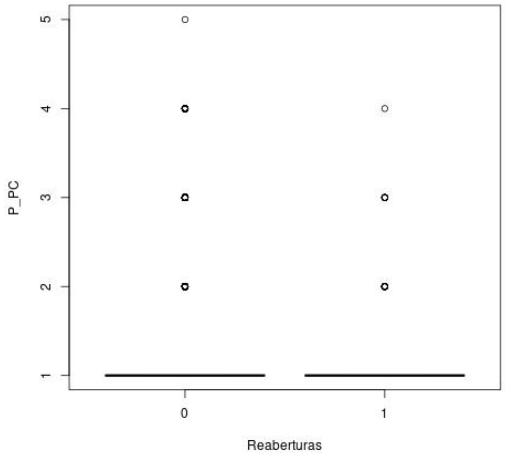
(c) Desempenho



(d) Funcional

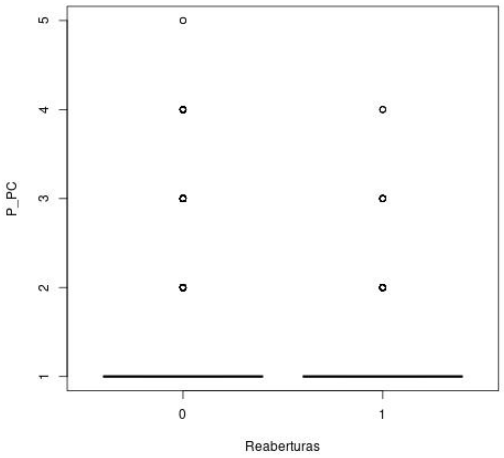


(e) GUI

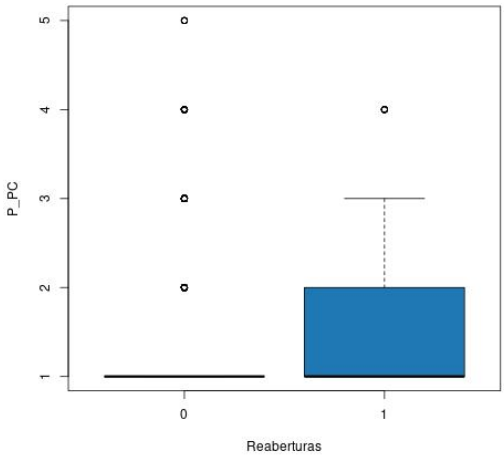


(f) Info

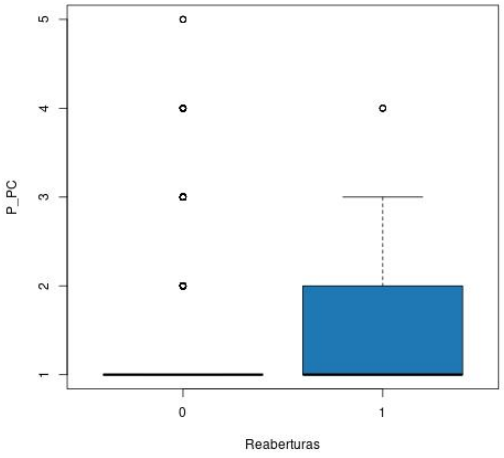
Figura 5.8: Distribuição de P_{PC} das categorias *Banco de Dados*, *Configuração*, *Desempenho*, *Funcional*, *GUI* e *Info*.



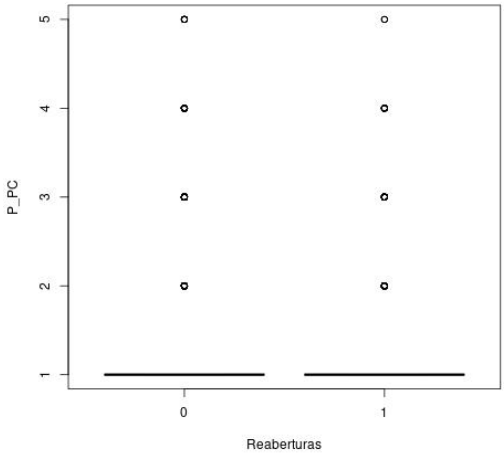
(a) Permissões/Obsoleto



(b) Redes



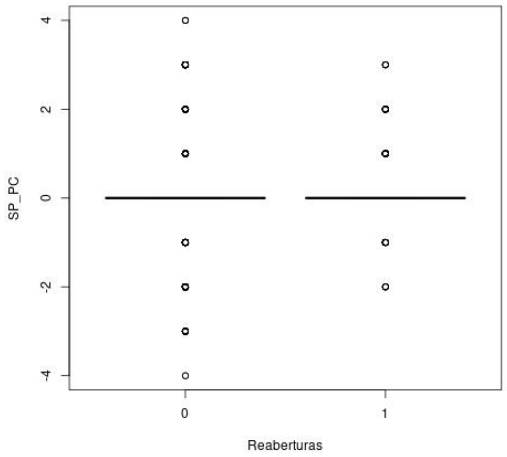
(c) Segurança



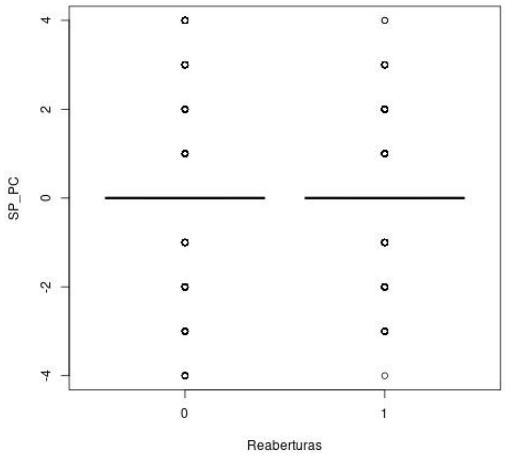
(d) Testes

Figura 5.9: Distribuição de P_{PC} das categorias *Permissões/Obsoleto*, *Redes*, *Segurança* e *Testes*.

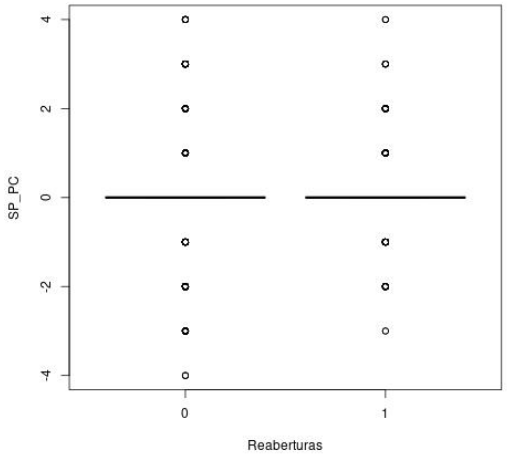
Com base nas métricas P_{PC} (pontuação positiva do primeiro comentário) e N_{PC} (pontuação negativa do primeiro comentário), calculamos a intensidade do sentimento (S_{PC}). Os resultados são apresentados nas figuras 5.10 e 5.11. Intensidade do Sentimento (S_{PC}) reflete o grau de positividade ou negatividade expressa nos primeiros comentários após a abertura das issues. Valores positivos indicam um sentimento mais positivo, enquanto valores negativos refletem uma percepção mais negativa.



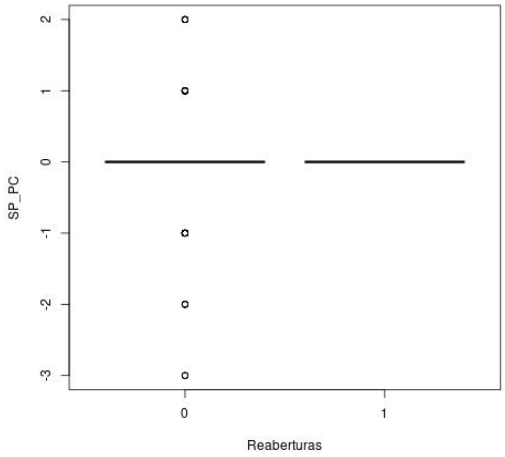
(a) Banco de Dados



(b) Configuração

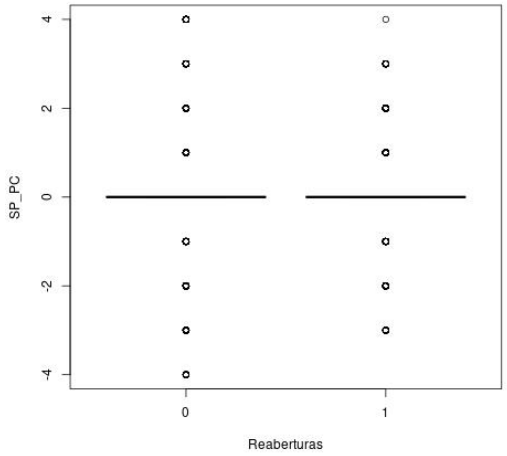


(c) Desempenho

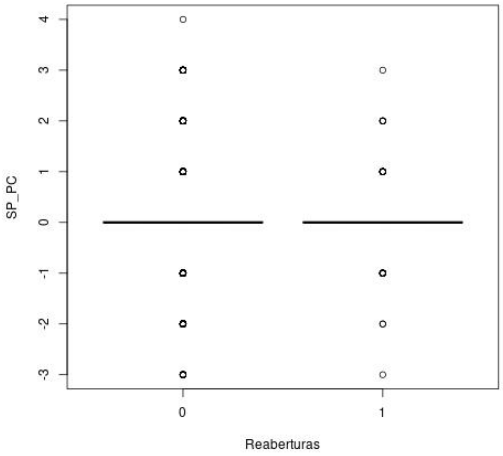


(d) Funcional

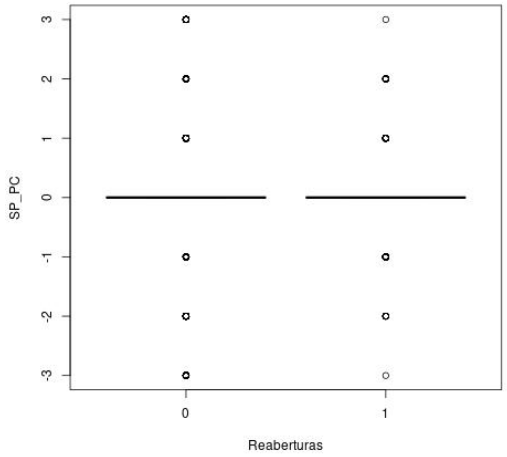
Figura 5.10: Distribuição de SP_{PC} das categorias *Banco de Dados*, *Configuração*, *Desempenho* e *Funcional* .



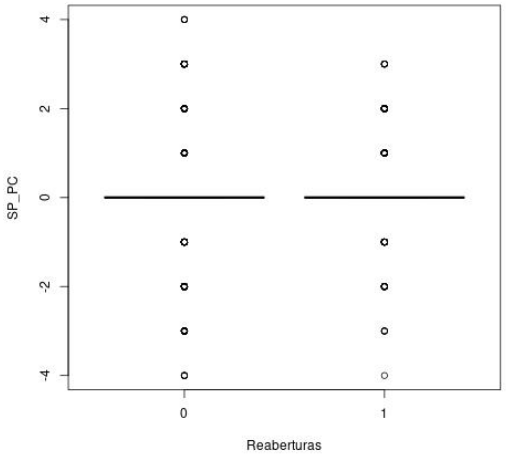
(a) GUI



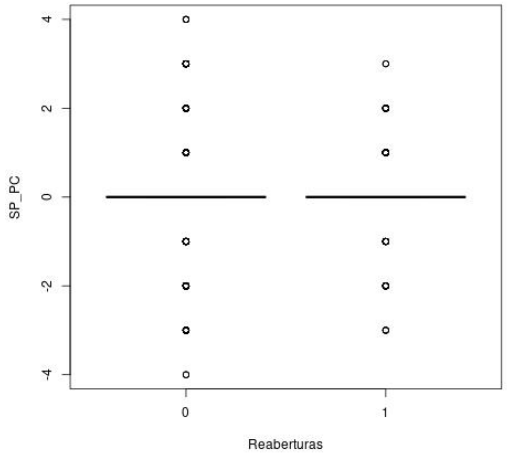
(b) Info



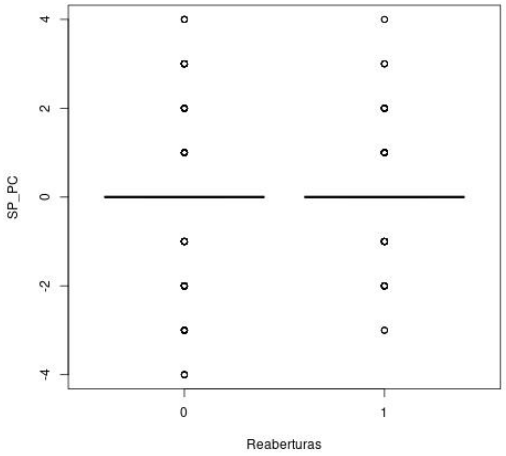
(c) Permissões/Obsoleto



(d) Redes



(e) Segurança



(f) Testes

Figura 5.11: Distribuição de SP_{PC} das categorias *Configuração*, *Desempenho*,*Funcional*, *Segurança* e *Testes*.

Nas figuras 5.10 e 5.11, observamos que, em todas as categorias de ambos os grupos, as métricas mediana, Q1 e Q3 apresentaram valor igual a zero. Esses resultados sugerem que a ausência de sentimento é predominante nos primeiros comentários, tanto em issues com reaberturas quanto sem reaberturas. Essa neutralidade inicial pode indicar uma tendência nas discussões iniciais de serem mais objetivas ou desprovidas de emoção clara.

No entanto, uma análise mais detalhada revela algumas variações entre as categorias. Notavelmente, a categoria *Funcional* no grupo de issues com reabertura mostrou todas as métricas com valor igual a zero, o que reforça a ideia de que essas issues tendem a receber comentários iniciais sem uma avaliação emocional significativa. Além disso, em ambas as tabelas, verificamos que poucas issues, em qualquer categoria, registraram valores de S_{PC} fora da faixa neutra (menores ou maiores que zero), evidenciando que a expressão emocional intensa é relativamente rara nos primeiros comentários, independentemente da presença de reaberturas.

A partir da métrica SP_{PC}, que reflete a intensidade do sentimento nos primeiros comentários após a abertura das issues, foi possível identificar o sentimento predominante nas issues (métrica S_{PC}), utilizamos a equação (3.3), analisamos as frequências dos sentimentos expressos nas issues e apresentamos a distribuição desses sentimentos, bem como suas respectivas porcentagens para cada categoria nas tabelas 5.1 e 5.2:

Tabela 5.1: S_{PC} de issues sem reaberturas

Categoria	Negativo(%)	Neutro(%)	Positivo(%)	Total
Banco de dados	1007 (7,32%)	11177 (81,23%)	1575 (11,45%)	13759
Configuração	23572 (17,57%)	86326 (64,35%)	24260 (18,08%)	134158
Desempenho	4061 (14,27%)	19934 (70,06%)	4459 (15,67%)	28454
Funcional	41 (5,28%)	699 (90,08%)	36 (4,64%)	776
GUI	12979 (15,5%)	53686 (64,1%)	17095 (20,41%)	83760
Info	559 (8,17%)	5539 (80,98%)	742 (10,85%)	6840
Permissões/Obsoleto	2183 (10,32%)	16556 (78,24%)	2421 (11,44%)	21160
Redes	3053 (13,11%)	15760 (67,69%)	4470 (19,2%)	23283
Segurança	2181 (12,04%)	12639 (69,8%)	3288 (18,16%)	18108
Testes	8035 (13,29%)	45591 (75,41%)	6833 (11,3%)	60459

As tabelas 5.1 e 5.2, em ambos os grupos (com e sem reaberturas), o sentimento predominante nas issues tende a ser neutro na maioria das categorias. Essa prevalência de sentimentos neutros pode indicar que, nos estágios iniciais das discussões nas issues, há uma comunicação mais técnica e objetiva, com menor expressão de emoções positivas ou negativas.

No grupo de issues sem reaberturas, a maioria das categorias apresenta uma porcentagem maior de issues com sentimentos positivos em comparação aos sentimentos negativos, como observado nas categorias *Banco de Dados*, *Configuração*, e *GUI*. No entanto, as categorias *Funcional* e *Testes* são exceções, com uma ligeira predominância de sentimentos negativos sobre os positivos, mesmo que a diferença seja pequena. Esse fato pode sugerir que issues relacionadas a funcionalidades e testes tendem a provocar discussões um pouco

Tabela 5.2: S_{PC} de issues com reaberturas

Categoria	Negativo(%)	Neutro(%)	Positivo(%)	Total
Banco de dados	49 (13,17%)	264 (70,97%)	59 (15,86%)	372
Configuração	850 (19,14%)	2573 (57,92%)	1019 (22,94%)	4442
Desempenho	150 (16,01%)	609 (64,99%)	178 (19%)	937
Funcional	0 (0%)	10 (100%)	0 (0%)	10
GUI	455 (17,91%)	1476 (58,09%)	610 (24,01%)	2541
Info	21 (14,09%)	107 (71,81%)	21 (14,09%)	149
Permissões/Obsoleto	68 (12,08%)	417 (74,07%)	78 (13,85%)	563
Redes	115 (14,38%)	500 (62,5%)	185 (23,13%)	800
Segurança	89 (15,01%)	372 (62,73%)	132 (22,26%)	593
Testes	231 (13,88%)	1168 (70,19%)	265 (15,93%)	1664

mais críticas.

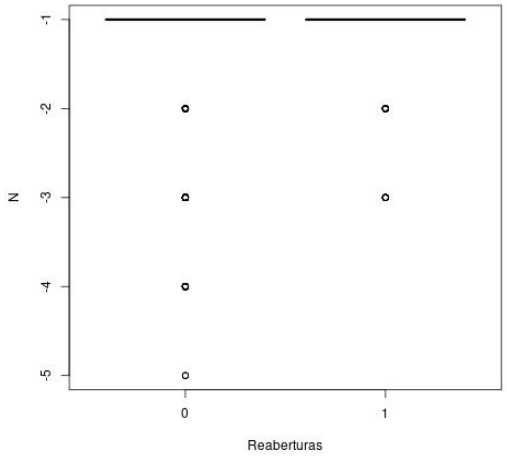
No grupo de issues com reaberturas, também é possível observar uma predominância de sentimentos neutros, com a maioria das categorias apresentando uma distribuição equilibrada entre sentimentos positivos e negativos. No entanto, a categoria *Info* chama a atenção por ter uma divisão igual entre sentimentos positivos e negativos, enquanto a categoria *Funcional* se destaca por apresentar exclusivamente sentimentos neutros, sem qualquer manifestação positiva ou negativa. Essa ausência de variação de sentimentos na categoria *Funcional* pode refletir uma abordagem mais neutra ou objetiva por parte dos colaboradores quando lidam com reaberturas nesse tipo de issue.

Ao comparar os dois grupos, percebe-se que as issues com reaberturas tendem a ter uma porcentagem ligeiramente maior de sentimentos negativos em diversas categorias, como *Banco de Dados*, *Configuração*, e *Desempenho*. Isso sugere que as issues que são reabertas podem gerar discussões mais negativas, potencialmente relacionadas à frustração com a necessidade de reabertura e resolução contínua de problemas. A categoria *Funcional* é uma exceção, com todas as issues mantendo um sentimento neutro.

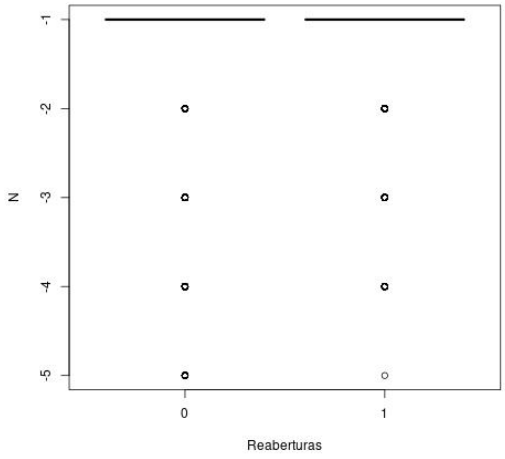
Após essa análise inicial com base nos primeiros comentários, aplicamos métricas semelhantes ao último comentário antes do primeiro fechamento da issue, a fim de explorar se o sentimento final antes do fechamento pode indicar a possibilidade de uma reabertura.

5.2.2 Último comentário

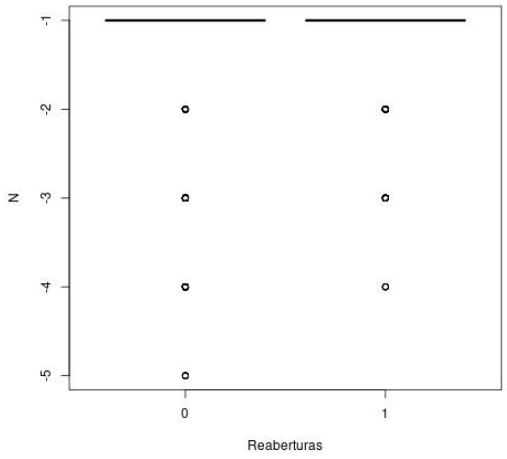
A partir do último comentário antes do primeiro fechamento da issue, extraímos as métricas: pontuação negativa (N), pontuação positiva (P), intensidade do sentimento (S) e o sentimento (S) dos grupos de issues com e sem reaberturas nas categorias *Banco de Dados*, *Configuração*, *Desempenho*, *Funcional*, *GUI*, *Info*, *ermissões/Obsoleto*, *Redes*, *Segurança* e *Testes*. Os resultados da métrica pontuação negativa (N) estão apresentados nas figuras 5.12 e 5.13



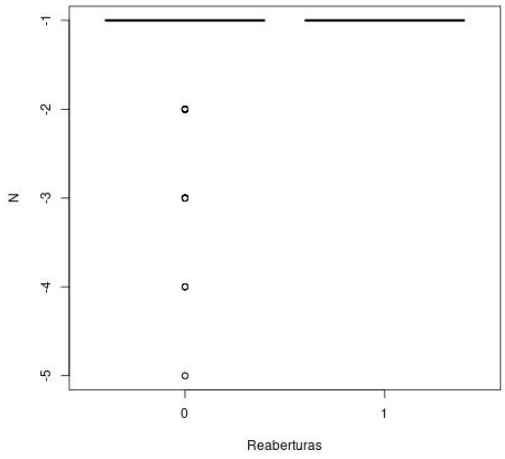
(a) Banco de Dados



(b) Configuração

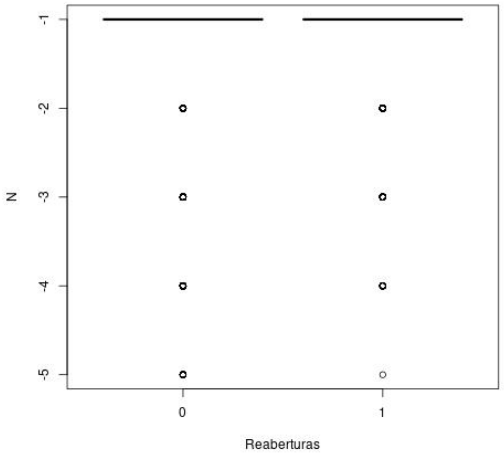


(c) Desempenho

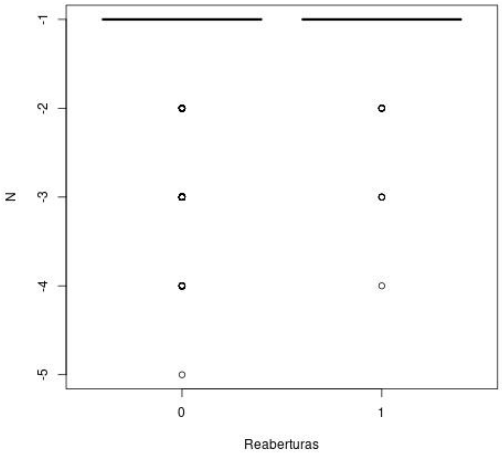


(d) Funcional

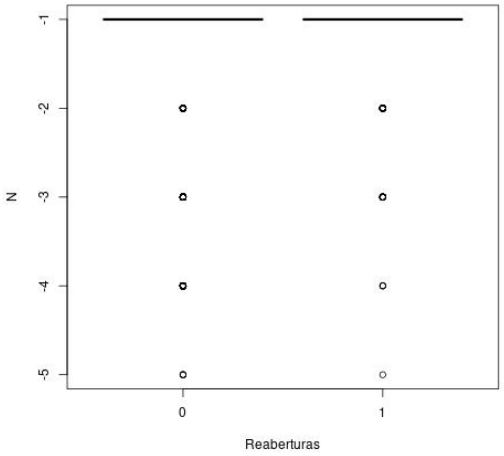
Figura 5.12: Distribuição de N das categorias *Banco de Dados*, *Configuração*, *Desempenho* e *Funcional*.



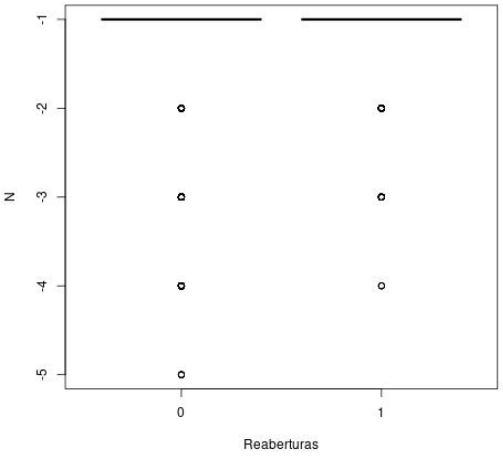
(a) GUI



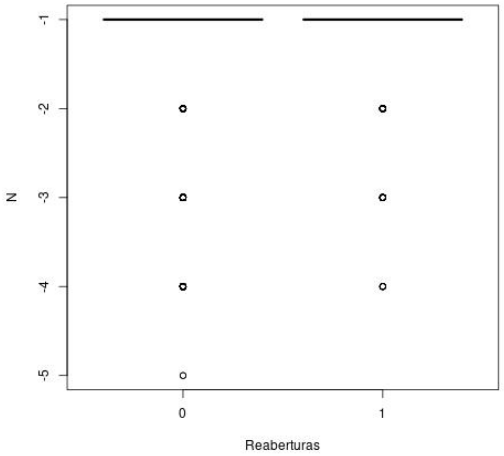
(b) Info



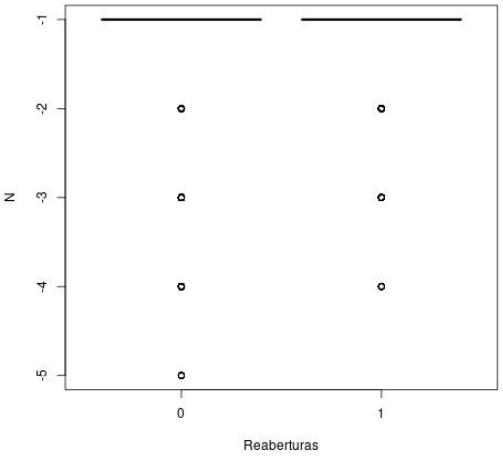
(c) Permissões/Obsoleto



(d) Redes



(e) Segurança



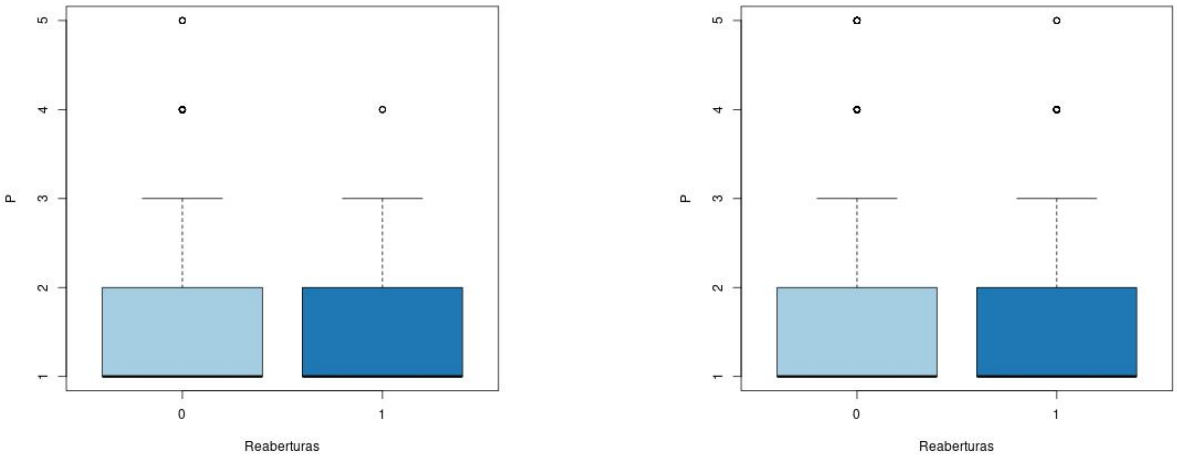
(f) Testes

Figura 5.13: Distribuição de N das categorias *GUI*, *Info*,*Permissões/Obsoleto*, *Redes*, *Segurança* e *Testes*.

Observando as figuras 5.12 e 5.13 identificamos que, em todas as categorias tanto no grupo de issues com reaberturas quanto no grupo de issues sem reaberturas, Q1, a mediana e o Q3 são consistentemente -1. Isso indica que a maioria das interações não apresenta sentimentos negativos, sugerindo que, em geral, as discussões nas issues são neutras ou possuem uma carga negativa baixa.

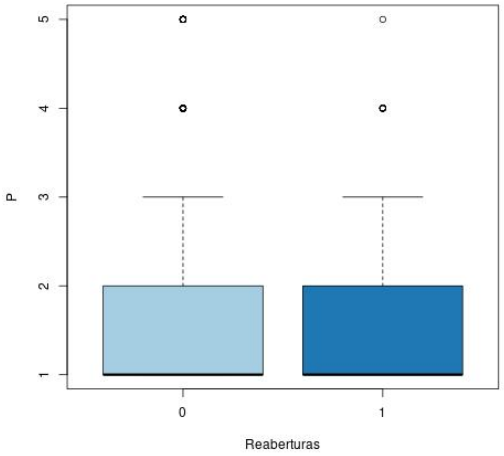
No entanto, as categorias *Configuração*, *GUI* e *Permissões/Obsoleto* são as únicas que apresentam o valor mínimo de -5 nas issues com reaberturas. Isso pode indicar que a presença de sentimentos negativos intensos nessas categorias específicas está relacionada à necessidade de revisitar ou reabrir a issue.

A pontuação positiva (P) do último comentário também foi extraída e os resultados estão apresentados nas figuras 5.14, 5.15 e 5.16. Observamos que, em todas as categorias, tanto para issues com reaberturas quanto para issues sem reaberturas, o valor mínimo, o Q1 e a mediana possuem pontuação positiva igual a 1. Isso indica que, na maioria dos casos, o último comentário antes do fechamento da issue contém poucas ou nenhuma palavra com conotação positiva significativa.

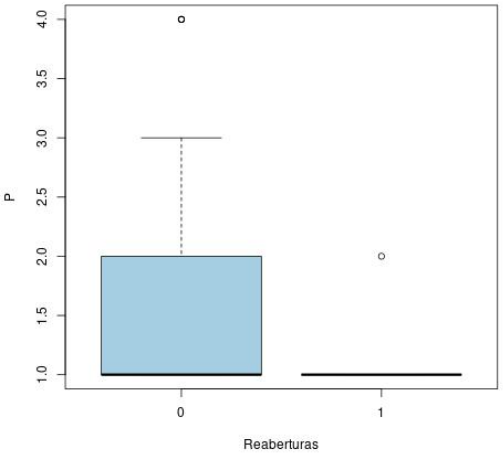


(a) Banco de Dados (b) Configuração

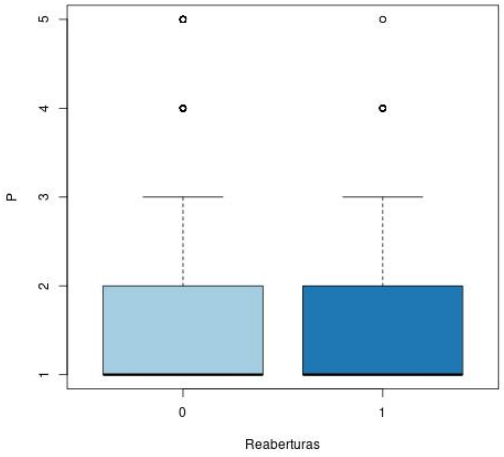
Figura 5.14: Distribuição de P das categorias *Banco de Dados* e *Configuração*.



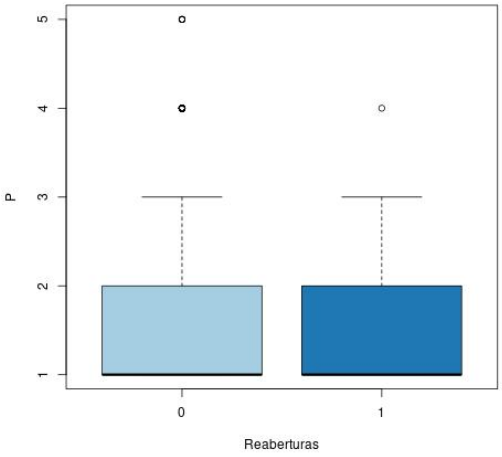
(a) Desempenho



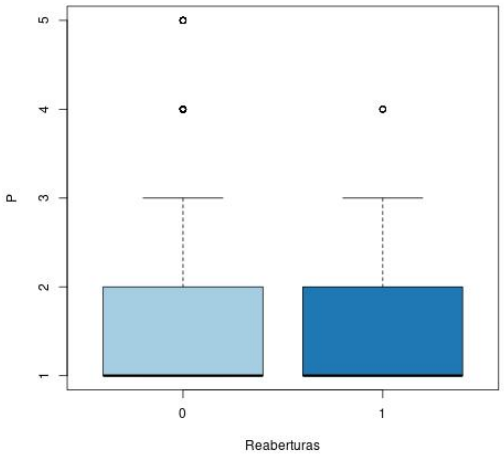
(b) Funcional



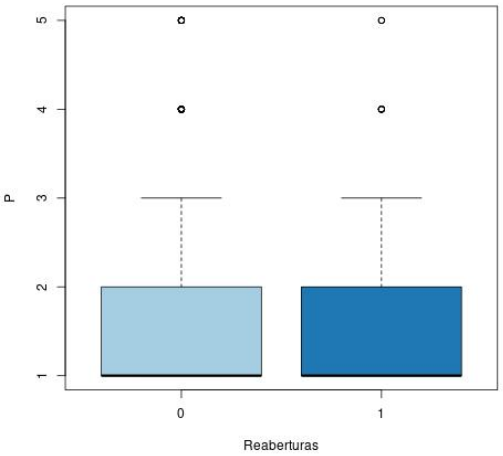
(c) GUI



(d) Info

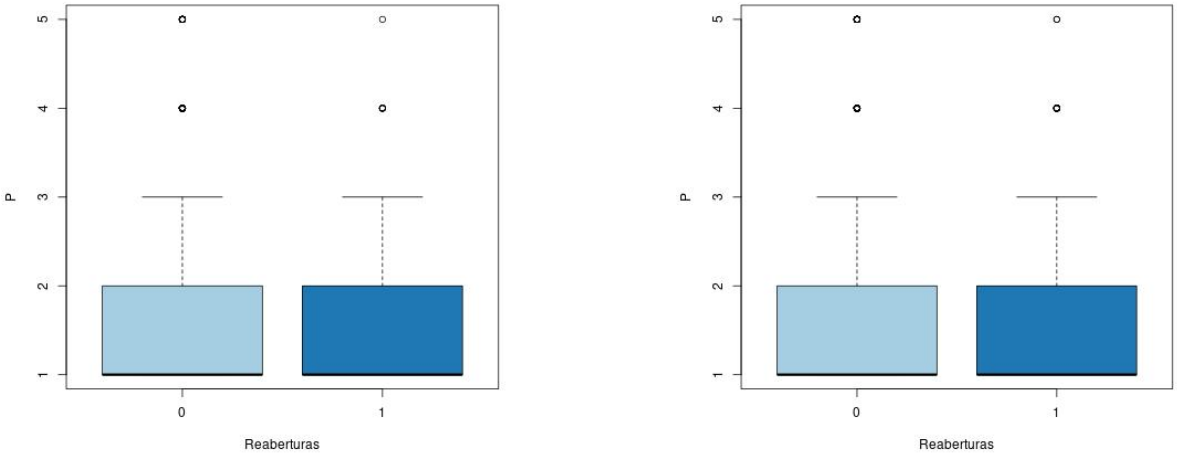


(e) Permissões/Obsoleto



(f) Redes

Figura 5.15: Distribuição de P das categorias *Desempenho*, *Funcional* , *GUI*, *Info*,*Permissões/Obsoleto* e *Redes*.



(a) Segurança (b) Testes

Figura 5.16: Distribuição de P das categorias *Segurança* e *Testes*.

Em ambos os grupos, a maioria das categorias apresenta o Q3 com valor igual a 2, sugerindo que há uma presença moderada de palavras positivas no último comentário. A única exceção significativa é a categoria *Funcional* no grupo de issues com reaberturas, cujo Q3 é 1, sugerindo uma ausência ainda mais pronunciada de palavras positivas nessa categoria.

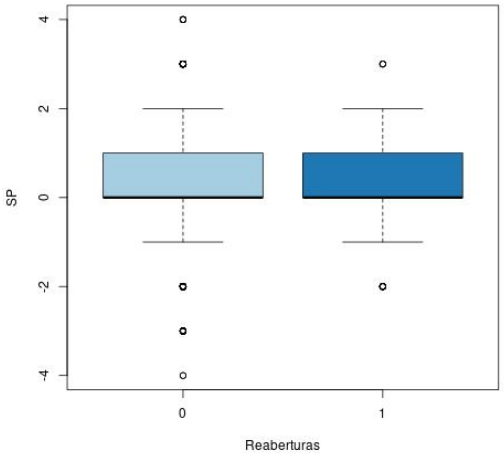
Quanto à pontuação máxima, a maioria das categorias nos dois grupos de issues (com e sem reaberturas) apresenta valores máximos entre 4 e 5, o que demonstra que alguns últimos comentários possuem uma carga expressivamente positiva. No entanto, a categoria *Funcional* no grupo de issues com reaberturas se destaca, com uma pontuação máxima de apenas 2, sugerindo que as interações nessas issues tendem a ser menos positivas em comparação com as demais categorias.

A intensidade do sentimento (SP), calculada a partir das métricas N e P, está representada nas figuras 5.17 e 5.18. A análise dos resultados mostra que, em todas as categorias de ambos os grupos (issues com e sem reaberturas), a mediana e o Q1 são iguais a zero. Isso sugere a predominância de comentários neutros, sem sentimentos negativos ou positivos significativos.

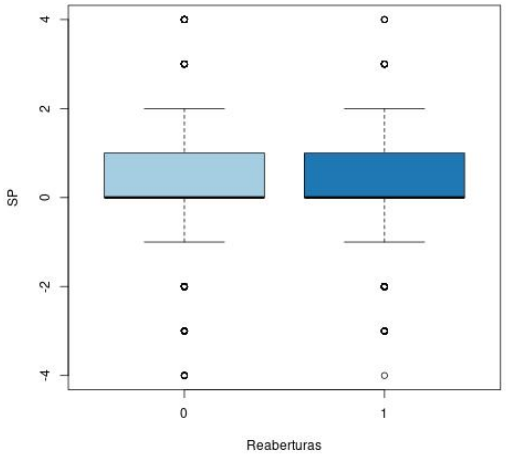
No grupo de issues sem reaberturas, o Q3 é majoritariamente igual a 1 em várias categorias, indicando uma leve presença de sentimentos positivos. A única exceção é a categoria *Funcional*, cujo Q3 é igual a 0, sugerindo uma predominância de sentimentos neutros. Os valores mínimos nas categorias variam entre -3 e -4, refletindo a presença de sentimentos negativos mais intensos, enquanto os valores máximos oscilam entre 3 e 4, indicando a existência de sentimentos positivos moderadamente intensos.

Já no grupo de issues com reaberturas, o Q3 também tende a ser igual a 1, sugerindo a presença de sentimentos positivos, exceto nas categorias *Desempenho*, *Funcional* e *Testes*, onde o Q3 permanece em 0, indicando neutralidade nas interações. Os valores mínimos variam de -2 a -4, evidenciando sentimentos negativos mais intensos nas categorias *Con-*

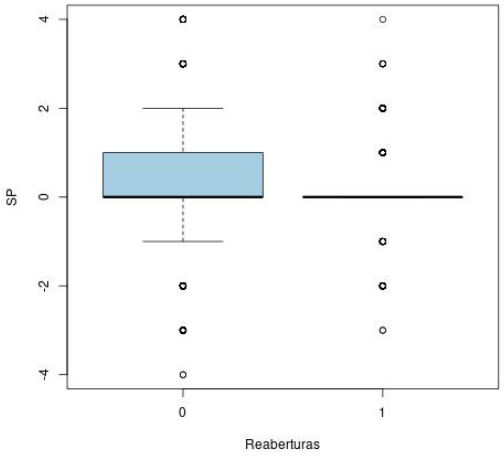
figuração e *GUI*, com exceção da categoria *Funcional*, que possui um valor mínimo de 0. Quanto aos valores máximos, a maioria das categorias apresenta uma pontuação entre 3 e 4, sugerindo a presença de sentimentos positivos mais intensos, com exceção novamente da categoria *Funcional*, que atinge no máximo 1, indicando uma menor intensidade de sentimentos positivos.



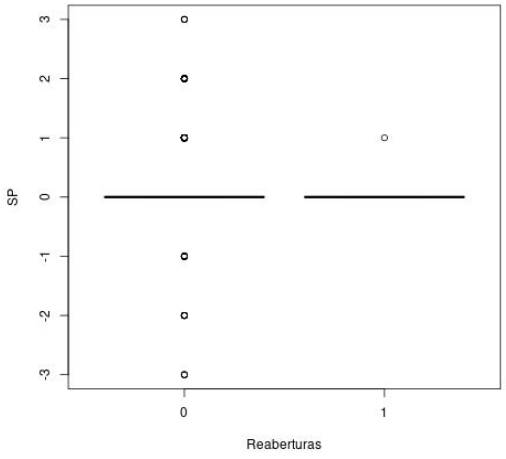
(a) Banco de Dados



(b) Configuração

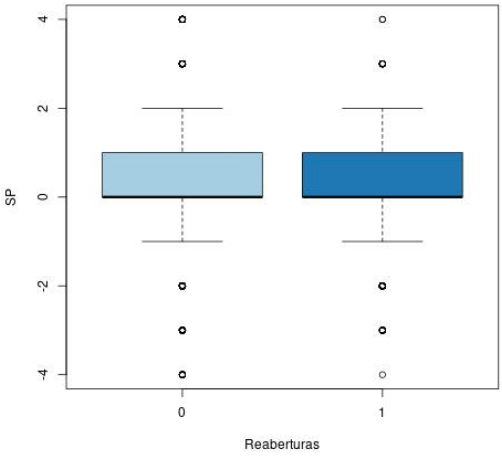


(c) Desempenho

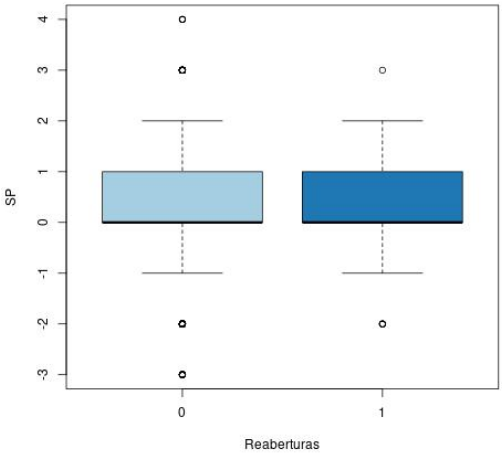


(d) Funcional

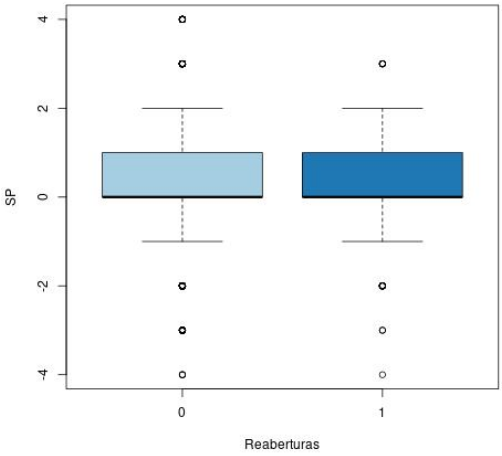
Figura 5.17: Distribuição de *SP* das categorias *Banco de Dados*, *Configuração*, *Desempenho* e *Funcional*.



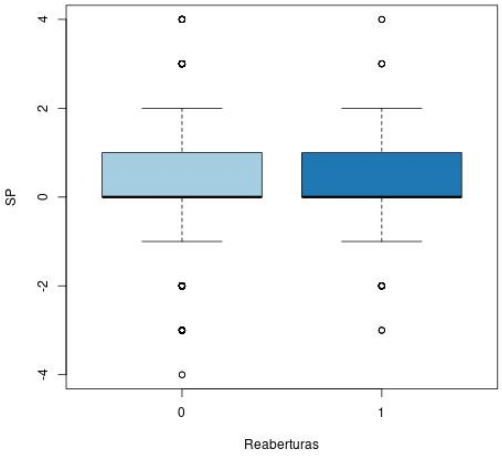
(a) GUI



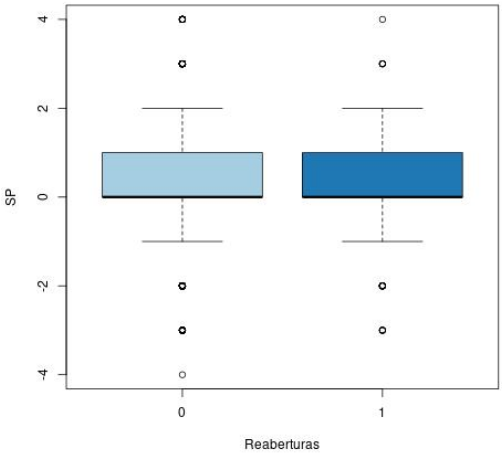
(b) Info



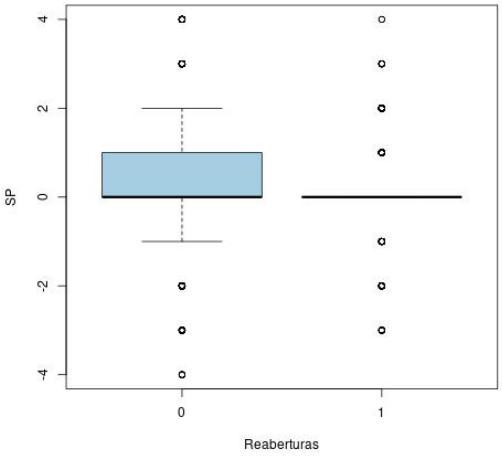
(c) Permissões/Obsoleto



(d) Redes



(e) Segurança



(f) Testes

Figura 5.18: Distribuição de *SP* das categorias *GUI*, *Info*,*Permissões/Obsoleto*, *Redes*, *Segurança* e *Testes*.

A partir dos resultados da métrica SP, foi possível calcular o sentimento predominante nos últimos comentários das issues, gerando a métrica S. A análise das frequências dos sentimentos expressos nas issues, sem e com reaberturas, está apresentada nas tabelas 5.3 e 5.4 , que detalham a distribuição dos sentimentos (Negativo, Neutro, Positivo) em cada categoria, com suas respectivas porcentagens.

Tabela 5.3: S - issue sem reaberturas

Categoria	Negativo(%)	Neutro(%)	Positivo(%)	Total
Banco de dados	874 (6,35%)	8128 (59,07%)	4757 (34,57%)	13759
Configuração	10021 (7,47%)	80820 (60,24%)	43317 (32,29%)	134158
Desempenho	2108 (7,41%)	17849 (62,73%)	8497 (29,86%)	28454
Funcional	29 (3,74%)	559 (72,04%)	188 (24,23%)	776
GUI	6063 (7,24%)	48553 (57,97%)	29144 (34,79%)	83760
Info	411 (6,01%)	3676 (53,74%)	2753 (40,25%)	6840
Permissões/Obsoleto	1263 (5,97%)	13319 (62,94%)	6578 (31,09%)	21160
Redes	1822 (7,83%)	14273 (61,3%)	7188 (30,87%)	23283
Segurança	1281 (7,07%)	10788 (59,58%)	6039 (33,35%)	18108
Testes	3466 (5,73%)	35718 (59,08%)	21275 (35,19%)	60459

Tabela 5.4: S - issues com reaberturas

Categoria	Negativo(%)	Neutro(%)	Positivo(%)	Total
Banco de dados	31 (8,33%)	243 (65,32%)	98 (26,34%)	372
Configuração	516 (11,62%)	2780 (62,58%)	1146 (25,8%)	4442
Desempenho	118 (12,59%)	585 (62,43%)	234 (24,97%)	937
Funcional	0 (0%)	9 (90%)	1 (10%)	10
GUI	296 (11,65%)	1581 (62,22%)	664 (26,13%)	2541
Info	12 (8,05%)	94 (63,09%)	43 (28,86%)	149
Permissões/Obsoleto	53 (9,41%)	362 (64,3%)	148 (26,29%)	563
Redes	79 (9,88%)	496 (62%)	225 (28,13%)	800
Segurança	54 (9,11%)	387 (65,26%)	152 (25,63%)	593
Testes	162 (9,74%)	1124 (67,55%)	378 (22,72%)	1664

A análise das tabelas 5.3 e 5.4 revela padrões interessantes em ambos os grupos. O sentimento Neutro é predominante nas duas situações, porém, a porcentagem de issues com esse sentimento é mais alta no grupo com reaberturas, em todas as categorias.

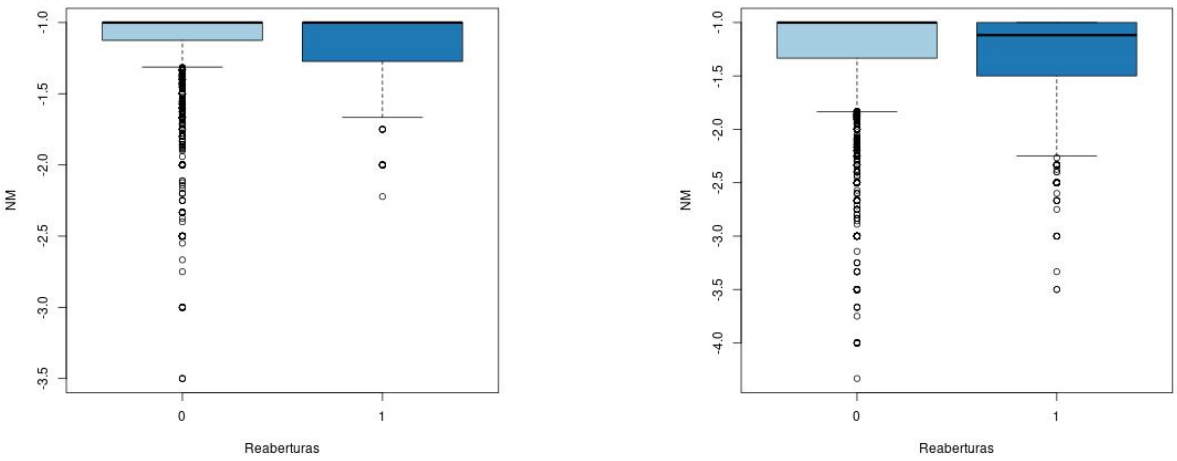
Além disso, observamos que as issues com reaberturas tendem a expressar menos sentimentos positivos, como evidenciado pela menor porcentagem de sentimentos positivos nesse grupo em comparação ao grupo sem reaberturas. Esse dado sugere uma correlação entre a menor expressão de sentimentos positivos e a maior probabilidade de reabertura de uma issue.

Apesar de os sentimentos negativos serem menos frequentes em ambos os grupos, é notável que a maioria das categorias nas issues com reaberturas apresenta uma porcentagem maior de sentimentos negativos em comparação às issues sem reaberturas. Uma exceção a essa regra é a categoria *Funcional*, que não apresentou nenhum comentário classificado como negativo, possivelmente devido ao baixo número de issues nessa categoria.

Esses resultados indicam que o comportamento emocional nos últimos comentários pode estar relacionado à dinâmica de reabertura de issues, especialmente com a redução de sentimentos positivos e o aumento da neutralidade ou negatividade antes do fechamento.

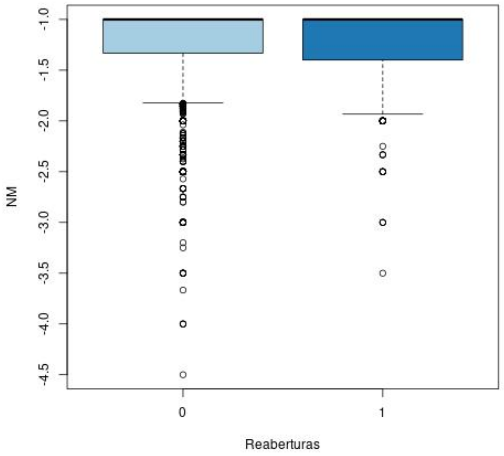
5.2.3 Comentários entre a abertura e o fechamento

Para analisar os comentários feitos entre a abertura e o fechamento das issues, foram calculadas as seguintes métricas: pontuação negativa média (NM), pontuação positiva média (PM), densidade de comentários negativos (DCN) e densidade de comentários positivos (DCP). Essas métricas foram aplicadas os grupos de issues com e sem reaberturas de diferentes categorias, incluindo *Banco de Dados*, *Configuração*, *Desempenho*, *Funcional*, *Interface Gráfica do Usuário (GUI)*, *Informação (Info)*, *Permissão/Obsoleto*, *Redes*, *Segurança* e *Testes*. Os resultados da pontuação negativa média (NM) são apresentados nas As figuras 5.19, 5.20 e 5.21:

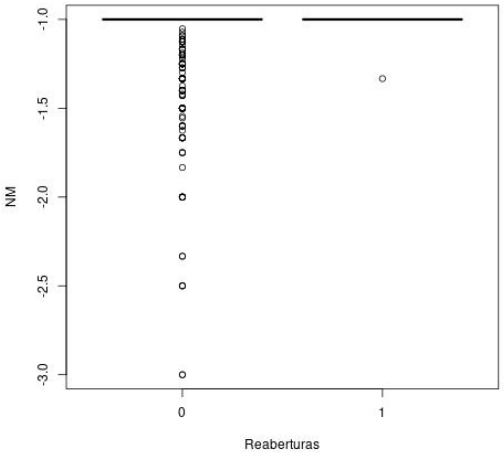


(a) Banco de Dados (b) Configuração

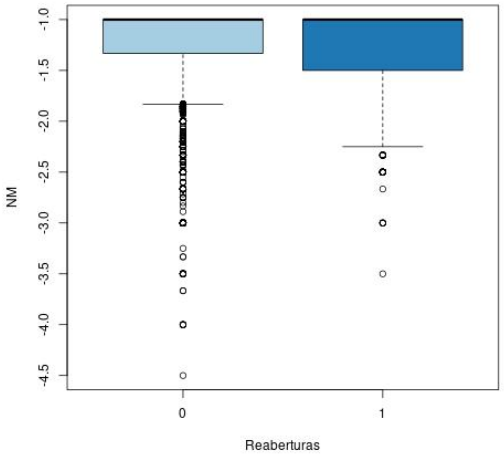
Figura 5.19: Distribuição de NM das categorias *Banco de Dados* e *Configuração*



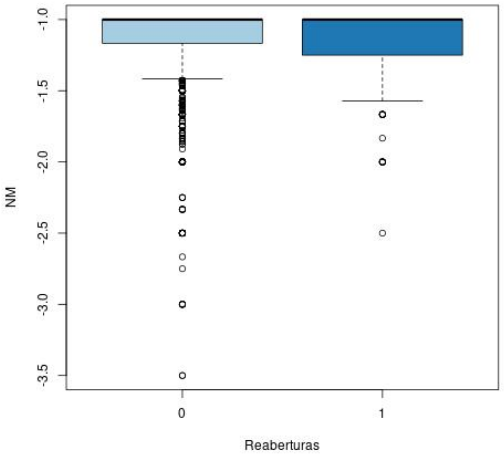
(a) Desempenho



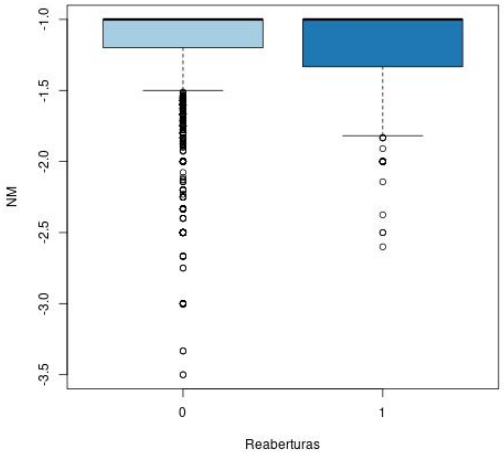
(b) Funcional



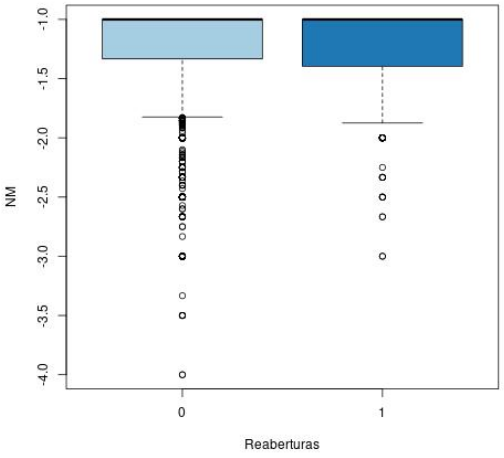
(c) GUI



(d) Info

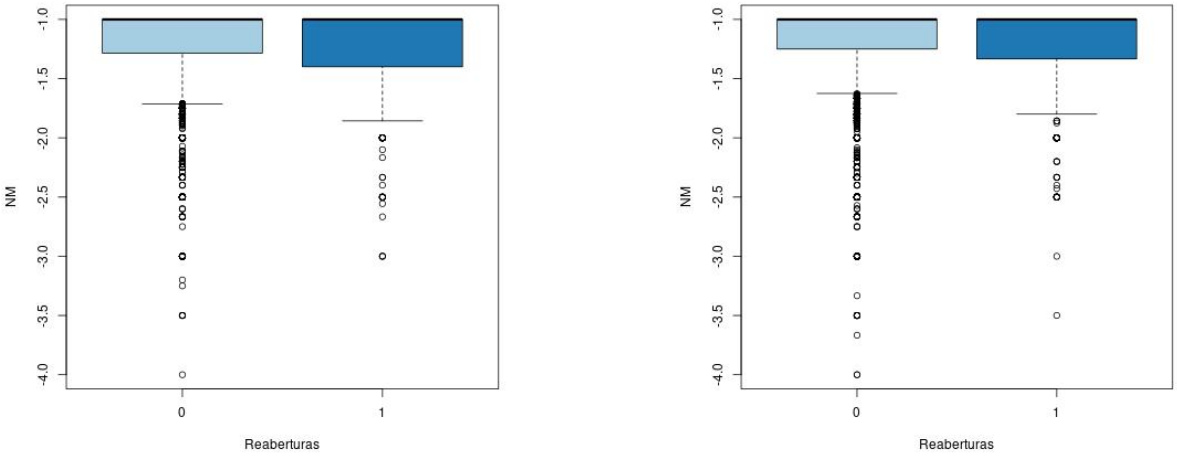


(e) Permissões/Obsoleto



(f) Redes

Figura 5.20: Distribuição de NM das categorias *Desempenho*, *Funcional*, *GUI*, *Info*,*Permissões/Obsoleto* e *Redes*



(a) Segurança (b) Testes

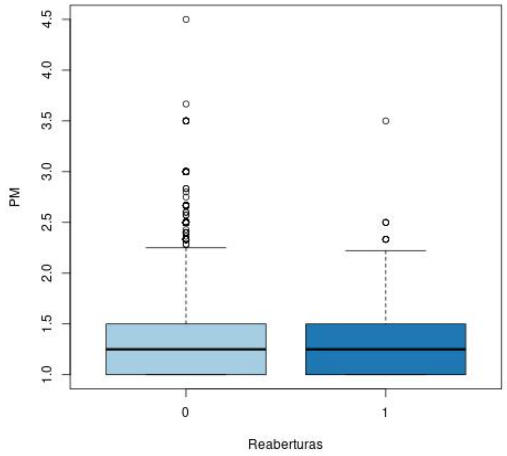
Figura 5.21: Distribuição de NM das categorias *Segurança* e *Testes*

A partir dos valores de NM apresentados nasfiguras 5.19, 5.20 e 5.21, observamos que, em todas as categorias de ambos os grupos de issues (com e sem reaberturas), a pontuação máxima e o Q3 mantêm-se constantes em -1. Isso sugere a ausência de termos extremamente negativos nos comentários entre a abertura e o fechamento das issues. Além disso, a maioria das categorias apresentou uma mediana de -1, com exceção da categoria *Configuração* no grupo de issues com reaberturas, onde a mediana foi ligeiramente superior, atingindo -1,118. Esse aumento discreto indica uma maior frequência de palavras negativas nessa categoria específica.

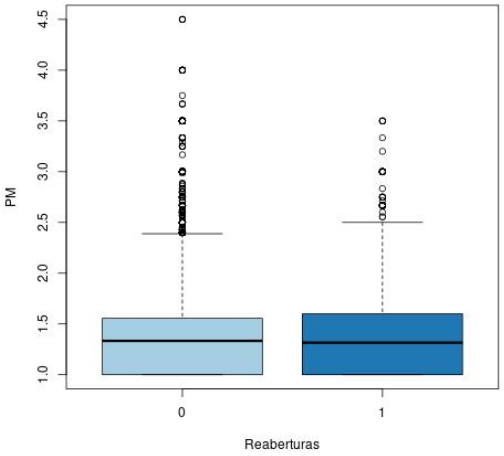
Nos grupos de issues sem reaberturas, os valores de Q1 variaram entre -1 e -1,333, com as categorias *Configuração*, *Desempenho*, *GUI* e *Redes* apresentando os valores mais elevados. A categoria *Funcional*, no entanto, apresentou um Q1 ligeiramente inferior (-1). Esses resultados sugerem uma baixa intensidade de palavras negativas nos comentários entre a abertura e o fechamento. Os valores mínimos para este grupo oscilaram entre -3,5 e -4,5, com a categoria *Desempenho* exibindo o valor mais elevado e *Funcional* mostrando o valor mais baixo. A variação dos valores de Q1 e mínimos diferentes de -1 sugere a presença de palavras negativas em algumas das categorias analisadas.

Nos grupos de issues com reaberturas, o Q1 variou de -1 a -1,4, com destaque para as categorias *Desempenho* e *Segurança*, que apresentaram os valores mais elevados. Apenas a categoria *Funcional* manteve um valor de Q1 igual a -1, indicando uma quantidade consideravelmente baixa de termos negativos. Os valores mínimos neste grupo variaram de -1,333 a -3,5, com as categorias *Configuração*, *Desempenho*, *GUI* e *Testes* apresentando uma maior incidência de palavras negativas. Esses dados reforçam a hipótese de que a frequência e a intensidade de palavras negativas são mais pronunciadas em alguns tipos de issues reabertas.

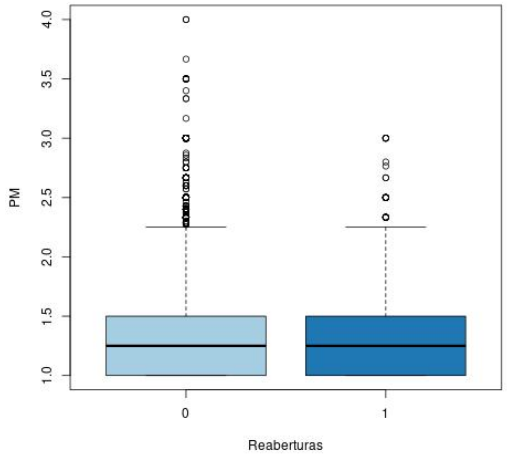
A pontuação positiva média PM foi extraída dos comentários entre a abertura e o fechamento, e os resultados são ilustrados nas figuras 5.22 e 5.23:



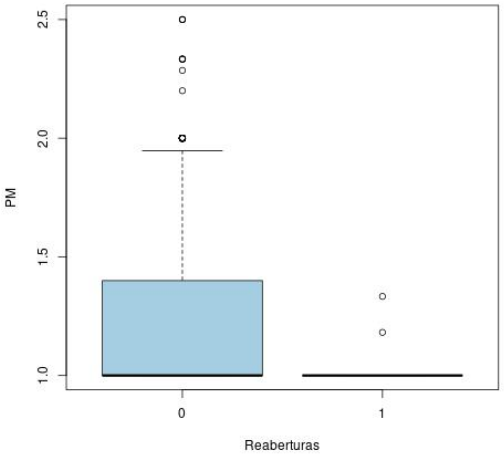
(a) Banco de Dados



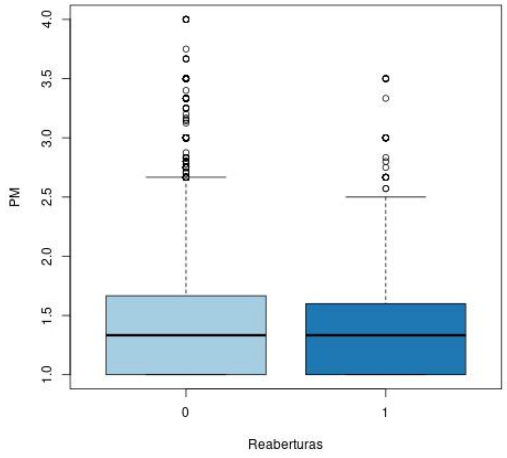
(b) Configuração



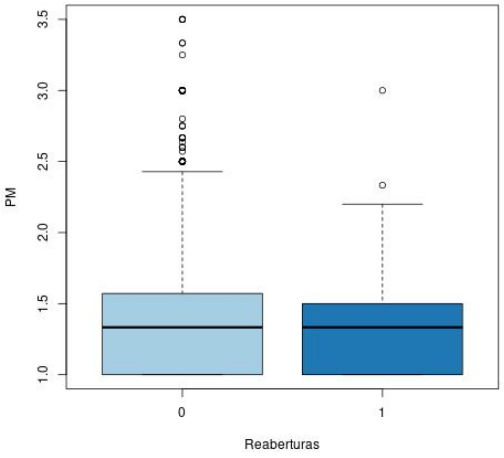
(c) Desempenho



(d) Funcional

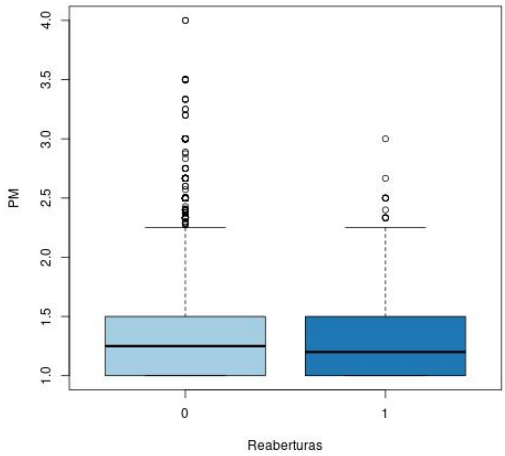


(e) GUI

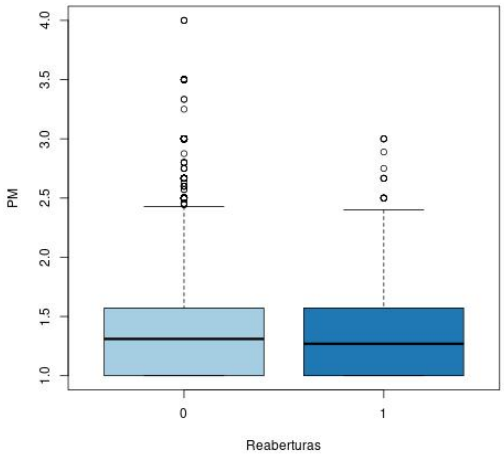


(f) Info

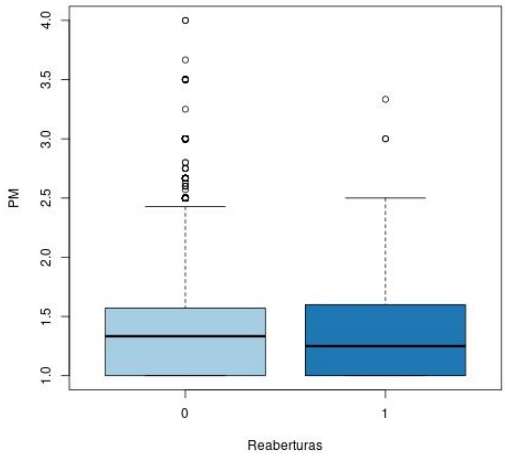
Figura 5.22: Distribuição de PM das categorias *Banco de Dados*, *Configuração*, *Desempenho*, *Funcional*, *GUI* e *Info*.



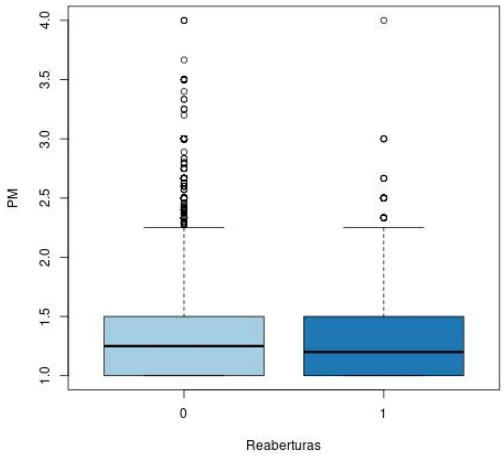
(a) Permissões/Obsoleto



(b) Redes



(c) Segurança



(d) Testes

Figura 5.23: Distribuição de PM das categorias *Permissões/Obsoleto*, *Redes*, *Segurança* e *Testes*.

De acordo com os resultados da métrica PM , em todas as categorias de ambos os grupos de issues (com e sem reaberturas), os valores mínimos e o Q1 são iguais a 1. Isso indica uma ausência de pontuações positivas nos comentários, ou seja, esses comentários não contêm palavras ou expressões positivas.

A mediana para todas as categorias variou entre 1 e 1,333, com a maioria das categorias apresentando uma leve predominância de comentários com alguma positividade, já que suas medianas superaram ligeiramente o valor de 1. No entanto, a categoria *Funcional* em ambos os grupos teve uma mediana igual a 1, indicando uma ausência significativa de palavras positivas nos comentários dessa categoria.

No grupo de issues sem reaberturas, as médias variaram entre 1,221 e 1,421, com o Q3 oscilando entre 1,5 e 1,667. A categoria *GUI* destacou-se com os maiores valores, sugerindo uma maior presença de termos positivos. Os valores máximos para esse grupo variaram entre 2,5 e 4,5, com as categorias *Banco de Dados* e *Configuração* atingindo os maiores valores. Isso indica que, embora a maioria dos comentários não seja particularmente positiva, algumas categorias possuem comentários com pontuações extremamente altas, sugerindo uma presença ocasional de palavras muito positivas.

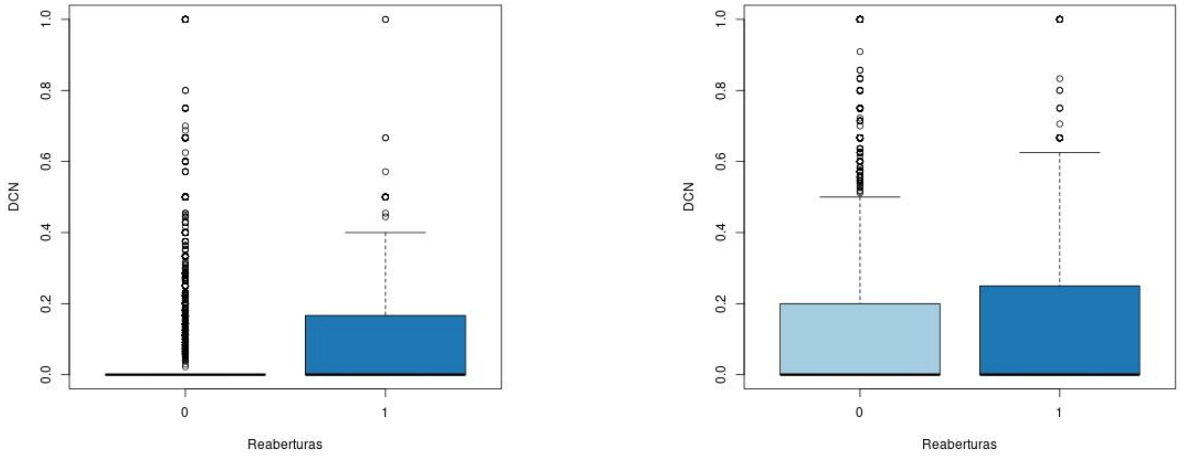
Por outro lado, no grupo de issues com reaberturas, a mediana variou entre 1 e 1,333, com as categorias *GUI* e *Info* mostrando os maiores valores. O Q3 variou entre 1 e 1,6, com as categorias *Configuração*, *GUI* e *Segurança* liderando nos valores mais altos. Os valores máximos variaram de 1,333 a 3,5, com *Banco de Dados* e *GUI* apresentando os maiores valores, o que sugere a presença de palavras positivas mais intensas nessas categorias. Notavelmente, a categoria *Funcional* teve valores de mínimo, Q1, mediana e Q3 iguais a 1, indicando uma ausência quase completa de comentários com palavras positivas.

Após analisar as pontuações negativas e positivas, realizamos a análise dos sentimentos dos comentários, focando nos sentimentos negativos e positivos entre a abertura e o fechamento das issues, tanto nas issues com quanto nas sem reaberturas. Foram consideradas as categorias *Banco de Dados*, *Configuração*, *Desempenho*, *Funcional*, *Interface Gráfica do Usuário (GUI)*, *Informação (Info)*, *Permissão/Obsoleto*, *Redes*, *Segurança* e *Teste*, utilizando as métricas de densidade de comentários com sentimentos negativos (DCN) e densidade de comentários com sentimentos positivos (DCP). Os resultados da métrica DCN são apresentados nas figuras 5.25 e 5.26.

Para as issues sem reaberturas, observamos que todas as categorias apresentam valores mínimos e o Q1 iguais a 0, indicando a ausência de sentimentos negativos na maioria dos comentários. A mediana também é 0 em todas as categorias, confirmando que, na maioria dos casos, os comentários não contêm expressões negativas. No entanto, as médias variaram entre 0,03333 e 0,1129, sugerindo que, embora raros, alguns comentários com sentimentos negativos estão presentes em certas categorias. As categorias com maiores médias foram *Configuração* (0,1129), *GUI* (0,1081) e *Desempenho* (0,09893), indicando uma maior frequência de comentários negativos nessas categorias em comparação com as demais. O terceiro quartil (Q3) variou entre 0 e 0,2, com destaque para as categorias *Configuração* e *GUI*, que apresentaram os maiores valores de Q3, sugerindo que uma proporção significativa das issues nessas categorias contém comentários com sentimentos negativos. Os valores máximos foram 1 para a maioria das categorias, exceto para *Banco*

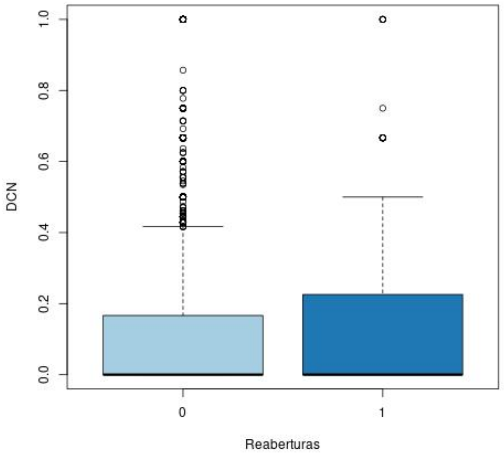
de Dados, *Info*, e *Permissão/Obsoleto*, que também apresentaram valores máximos de 1, indicando que algumas issues têm discussões altamente negativas, embora sejam relativamente raras.

Para as issues com reaberturas, os padrões observados foram semelhantes, com valores mínimos e Q1 iguais a 0 em todas as categorias, denotando a ausência de sentimentos negativos na maioria dos comentários. A mediana variou entre 0 e 0,33333, com a categoria *GUI* apresentando o maior valor mediano (0,2222), sugerindo uma tendência maior de comentários negativos nessa categoria. A categoria *Funcional* mostrou o menor valor mediano, indicando uma baixa densidade de sentimentos negativos nessa categoria. As médias variaram entre 0,03333 e 0,1351, com destaque para as categorias *Configuração* (0,1332) e *GUI* (0,1351), que tiveram as maiores médias, sugerindo uma maior recorrência de sentimentos negativos. O Q3 variou entre 0,13793 e 0,25, com as categorias *Banco de Dados*, *GUI*, *Info* e *Segurança* apresentando os maiores valores, indicando uma maior concentração de comentários negativos. Os valores máximos foram 1 para todas as categorias, exceto para *Funcional* (0,33333) e *Info* (0,66667), refletindo uma menor intensidade de sentimentos negativos nessas categorias comparadas às outras. Tanto nas issues com quanto nas sem reaberturas, a maioria das categorias apresentou predominância de comentários sem sentimentos negativos. Entretanto, algumas categorias, especialmente *Configuração*, *GUI* e *Desempenho*, mostraram uma maior densidade de comentários negativos, sugerindo que as discussões nessas áreas tendem a gerar mais frustração ou descontentamento entre os usuários. Isso é particularmente notável nas issues com reaberturas, onde a presença de comentários negativos pode estar associada a problemas recorrentes ou mal resolvidos.

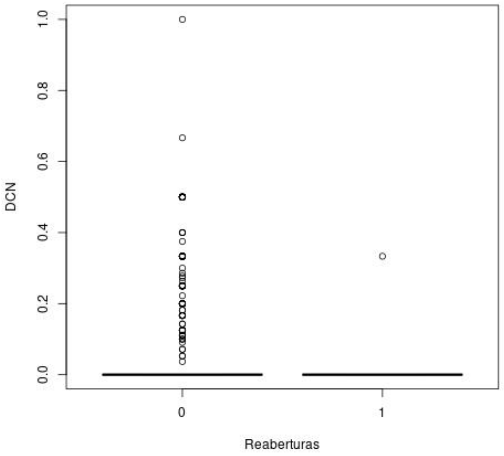


(a) Banco de Dados (b) Configuração

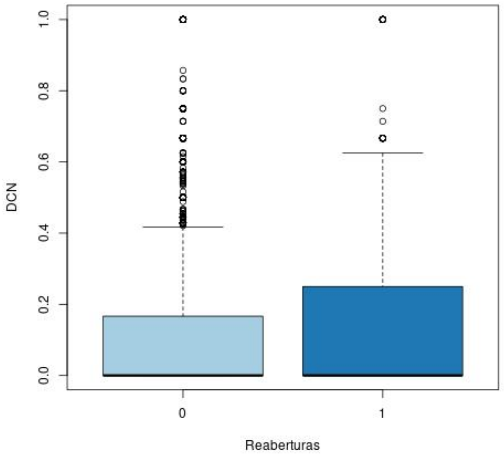
Figura 5.24: Distribuição de DCN das categorias *Banco de Dados* e *Configuração*.



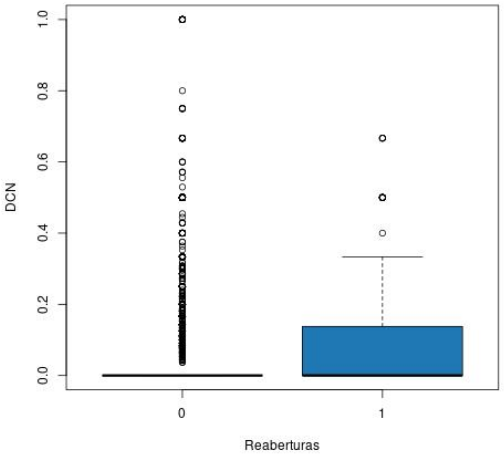
(a) Desempenho



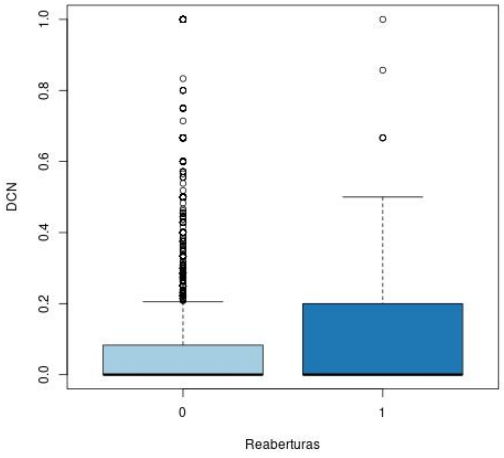
(b) Funcional



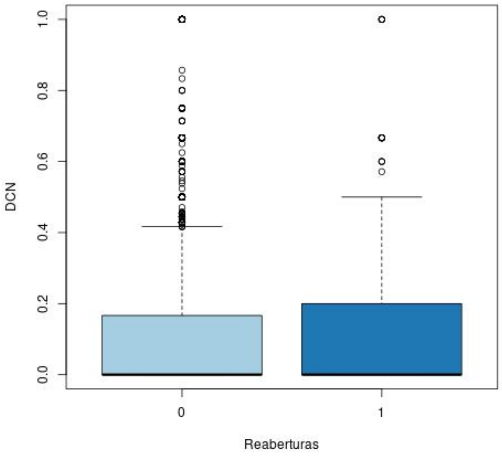
(c) GUI



(d) Info

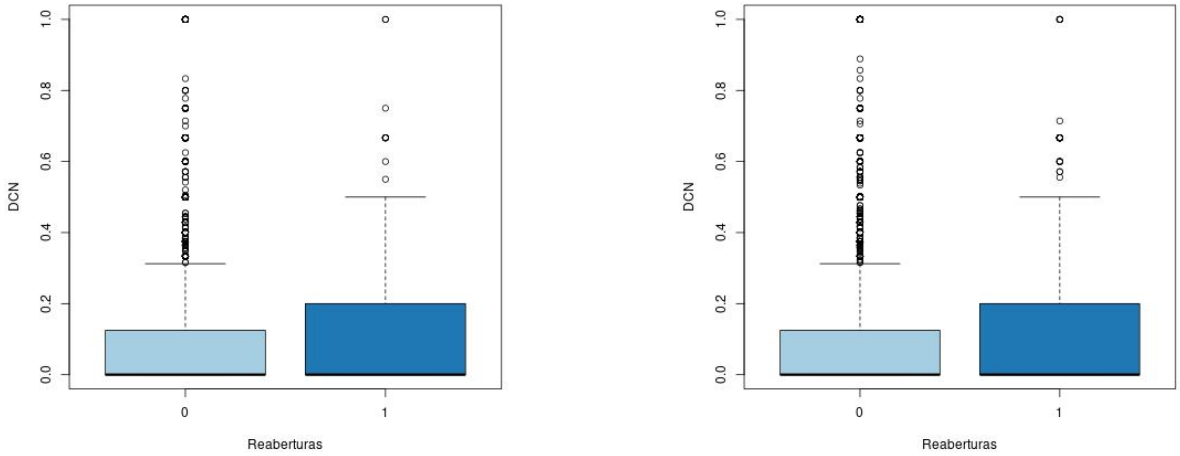


(e) Permissões/Obsoleto



(f) Redes

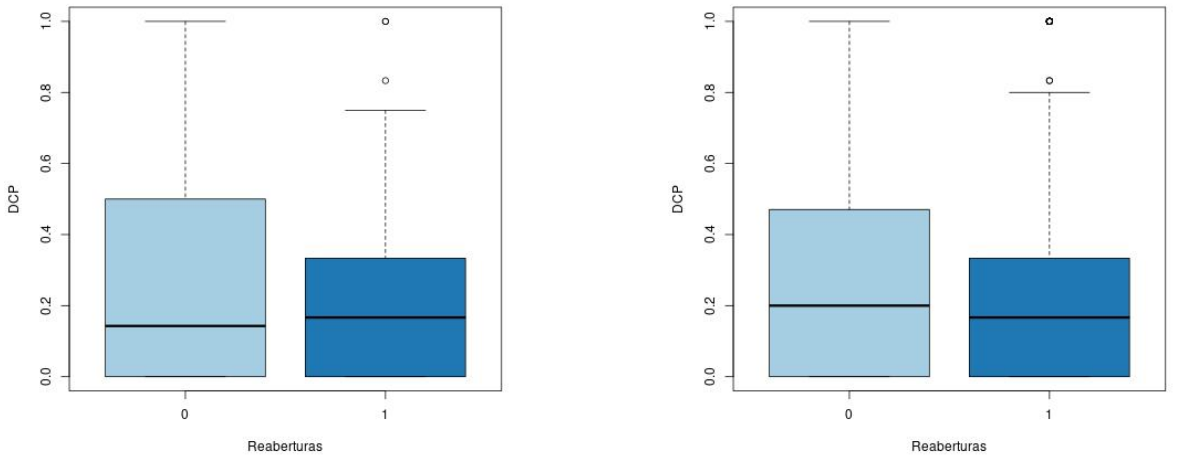
Figura 5.25: Distribuição de DCN das categorias *Desempenho* e *Funcional*, *GUI*, *Info*, *Permissões/Obsoleto* e *Redes*.



(a) Segurança (b) Testes

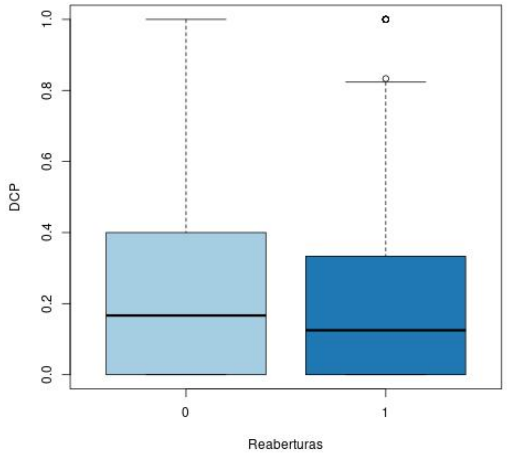
Figura 5.26: Distribuição de DCN das categorias *Segurança* e *Testes*.

Além de avaliar a densidade de sentimentos negativos, é igualmente essencial examinar a densidade de comentários com sentimentos positivos (DCP) em relação às discussões das issues. A métrica DCP desempenha um papel significativo na análise das interações nos grupos de issues, fornecendo uma perspectiva sobre o quanto as discussões nas issues são marcadas por expressões de apoio, satisfação ou concordância. As figuras 5.27, 5.28 e 5.29 apresentam os resultados da métrica DCP para todas as categorias dos grupos de issues com e sem reaberturas.

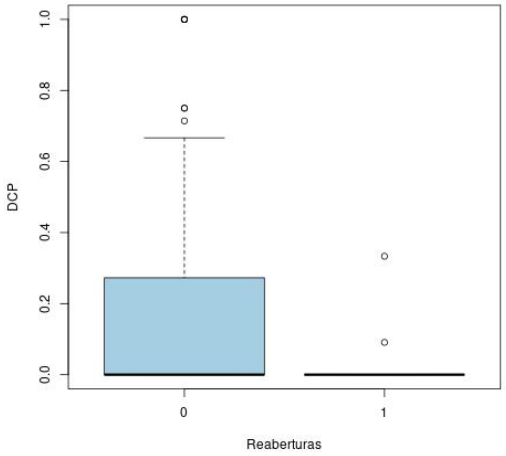


(a) Banco de Dados (b) Configuração

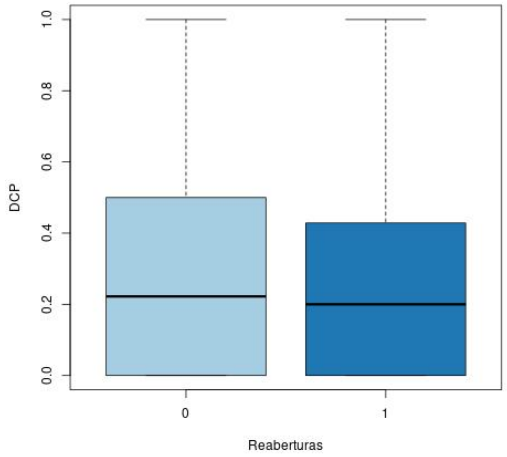
Figura 5.27: Distribuição de DCP das categorias *Banco de Dados* e *Configuração*, *Desempenho* e *Funcional*.



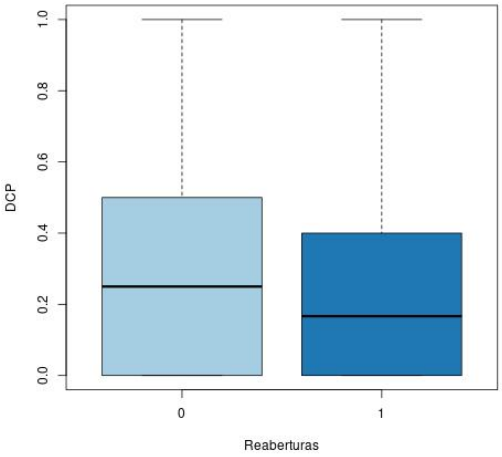
(a) Desempenho



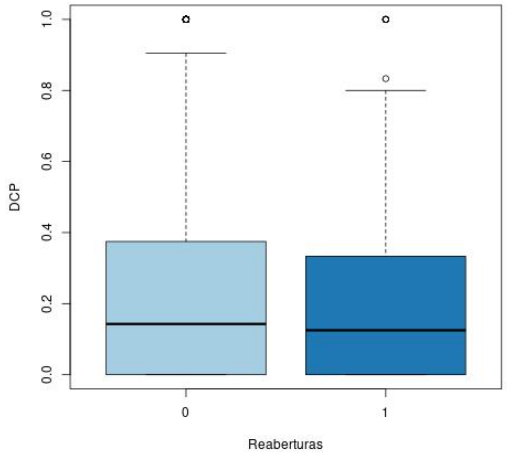
(b) Funcional



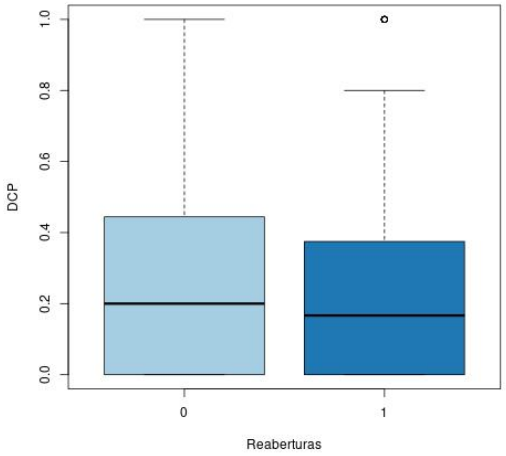
(c) GUI



(d) Info

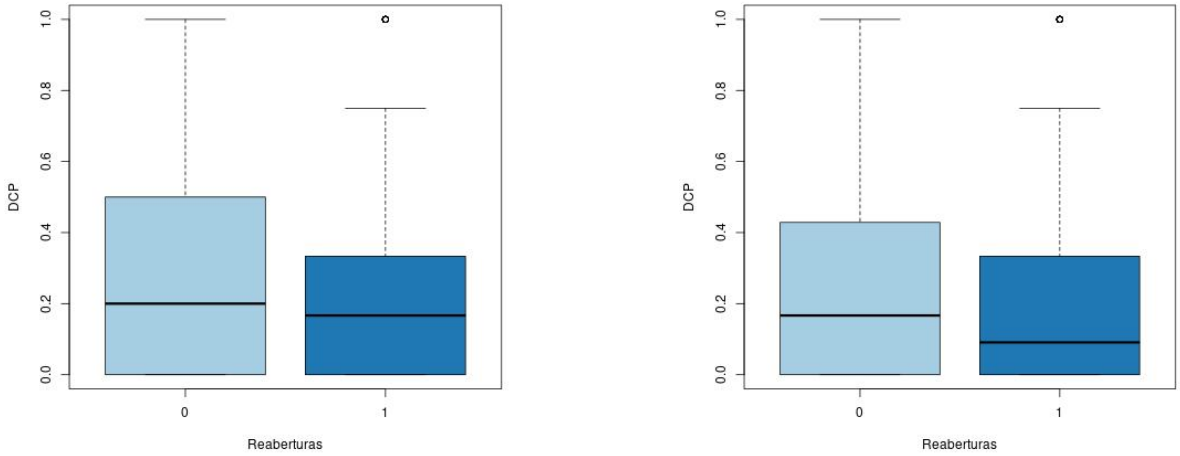


(e) Permissões/Obsoleto



(f) Redes

Figura 5.28: Distribuição de DCP das categorias *Desempenho*, *Funcional*, *GUI*, *Info*,*Permissões/Obsoleto* e *Redes*.



(a) Segurança (b) Testes

Figura 5.29: Distribuição de DCP das categorias *Segurança* e *Testes*.

Observamos que, em todas as categorias, nos dois grupos (com e sem reaberturas), o valor mínimo e o Q1 são consistentemente iguais a zero. Isso sugere que, em muitas issues, não há evidência de comentários positivos notáveis. Para as issues sem reaberturas, a mediana varia de 0 a 0,25, indicando que, em geral, as discussões neste grupo apresentam uma densidade relativamente baixa de comentários positivos. A categoria Info tem a maior mediana, enquanto a categoria Funcional possui mediana igual a zero, o que sugere que, em metade das discussões nessa categoria, não há comentários positivos significativos.

O terceiro quartil Q3 varia entre 0,1426 a 0,2573, sendo a categoria *GUI*a que apresenta o maior valor, enquanto a categoria *Funcional* tem o menor. Em todas as categorias, o valor máximo é igual a 1, o que sugere que, em algumas issues, há uma presença densa de sentimentos positivos, com a maioria dos comentários refletindo essa positividade.

No grupo de issues com reaberturas, a mediana varia de 0 a 0,1667. As categorias*Banco de Dados*, *Configuração*, *Info*, *Redes* e *Segurança* apresentam a mediana mais alta, enquanto a categoria *Funcional* continua com mediana igual a zero, indicando que uma fração considerável das discussões nesse grupo tem uma densidade muito baixa de comentários positivos. Além disso, o valor máximo varia entre 0,3333 e 1. A maioria das categorias alcançou o valor máximo de 1, com exceção da categoria *Funcional*, cujo valor máximo foi de 0,3333. Isso indica que, embora as issues com reaberturas possam conter casos com densidade elevada de comentários positivos, essas situações ainda são raras.

Esses resultados levantam questões importantes sobre o papel dos sentimentos nas reaberturas de issues e como esses sentimentos variam entre as diferentes categorias, o que será explorado na seção seguinte. Para uma análise mais detalhada, incluindo valores descritivos das métricas de análise de sentimentos para as diferentes categorias de issues com e sem reaberturas, consulte o Apêndice C.

5.3 DISCUSSÕES DOS RESULTADOS

Nesta seção, discutimos os resultados relacionados às questões de pesquisa QP4.1, QP4.2 e QP4.3, que compõem a resposta à questão de pesquisa **QP4: Como caracterizar os sentimentos nas diferentes categorias no contexto de reaberturas de issues?**.

QP4.1. Como o sentimento, a pontuação negativa e a pontuação positiva do primeiro comentário após a abertura da issue influenciam a reabertura nas diferentes categorias?

No primeiro comentário após a abertura das issues, observamos que o sentimento predominante é neutro na maioria das categorias. No entanto, nas issues que foram reabertas, há uma tendência de se observar uma maior ocorrência de sentimentos tanto positivos quanto negativos.

A categoria *Configuração* se destaca por apresentar níveis mais altos de sentimentos negativos, sugerindo que frustrações ou problemas identificados logo no início podem estar associados a uma maior chance de reabertura. Por outro lado, nas categorias *GUI* e *Redes*, sentimentos positivos são mais comuns, indicando que, nessas áreas, os comentários iniciais tendem a ser mais otimistas, apesar das reaberturas posteriores.

Já a categoria *Funcional* apresenta um equilíbrio entre sentimentos positivos e negativos, com ambos os tipos de sentimento ocorrendo com menor frequência. Esse equilíbrio pode sugerir que as issues dessa categoria são tratadas de maneira mais pragmática desde o início, resultando em menos variações emocionais.

QP4.2. Como o sentimento, a pontuação negativa e a pontuação positiva do último comentário antes do fechamento da issue influenciam a reabertura nas diferentes categorias?

Ao analisar o último comentário antes do fechamento das issues, vimos que, na maioria das categorias, os sentimentos negativos são pouco frequentes. No entanto, as categorias Configuração, GUI e Permissões/Obsoleto se destacam, pois nas issues com reaberturas, os comentários chegam a ter sentimentos negativos intensos, o que pode indicar uma ligação com a reabertura.

Sobre os sentimentos positivos, vimos que eles são moderados em várias categorias. A categoria *Funcional*, no entanto, tem menos sentimentos positivos nos últimos comentários de issues com reaberturas. Além disso, essa categoria também apresenta uma menor intensidade de sentimentos positivos em geral.

Em termos de intensidade emocional (positiva e negativa), os comentários tendem a ser neutros em quase todas as categorias. No grupo de issues com reaberturas, algumas categorias, como *Desempenho*, *Funcional* e *Testes*, mostram menos sentimentos positivos antes do fechamento, o que sugere que a falta de emoções positivas pode influenciar na reabertura da issue.

QP4.3. Como os sentimentos, pontuações negativas e positivas entre a abertura e o primeiro fechamento da issue influenciam a reabertura nas diferentes categorias?

Para analisar a influência dos sentimentos nas reaberturas de issues, foram exami-

nadas as métricas de pontuação negativa média (NM), pontuação positiva média (PM), densidade de comentários negativos (DCN) e densidade de comentários positivos (DCP) entre a abertura e o fechamento das issues.

Observou-se que, em todas as categorias, tanto para issues com quanto sem reaberturas, a pontuação negativa máxima e o terceiro quartil (Q3) foram constantes em -1. Isso indica que não houve termos extremamente negativos nos comentários. No entanto, na categoria *Configuração*, as issues reabertas apresentaram uma mediana de -1,118, sugerindo uma presença ligeiramente maior de palavras negativas. A categoria *Desempenho* destacou-se com o valor mínimo mais negativo (-4,5), o que indica uma maior intensidade de sentimentos negativos.

Em relação à pontuação positiva, todos os valores mínimos e Q1 foram iguais a 1, sugerindo uma ausência de expressões positivas na maioria das categorias. Algumas categorias, como *GUI* e *Configuração*, mostraram medianas e máximas ligeiramente positivas, enquanto a categoria *Funcional* apresentou uma ausência quase completa de palavras positivas.

A densidade de comentários negativos nas issues sem reaberturas revelou baixos níveis de sentimentos negativos, com a mediana igual a zero na maioria das categorias. No entanto, categorias como *Configuração*, *GUI* e *Desempenho* apresentaram maiores frequências de comentários negativos. Para as issues com reaberturas, a tendência foi semelhante, mas categorias como *Configuração* e *GUI* mostraram médias mais altas, sugerindo uma maior recorrência de sentimentos negativos.

Quanto à densidade de comentários positivos, a maioria das categorias revelou uma baixa presença de sentimentos positivos em ambos os grupos. As categorias *Info* e *GUI* foram as que apresentaram maior densidade de sentimentos positivos. Por outro lado, a categoria *Funcional* continuou com uma ausência significativa de comentários positivos em ambos os grupos.

Em resumo, categorias como *Configuração*, *Desempenho* e *GUI* tendem a exibir mais sentimentos negativos, especialmente nas issues reabertas. Já a categoria *Funcional* mostra uma baixa densidade de sentimentos, tanto negativos quanto positivos, indicando uma presença geral reduzida de sentimentos nas discussões.

QP4: Como caracterizar os sentimentos nas diferentes categorias no contexto de reaberturas de issues?

Para entender como os sentimentos variam nas diferentes categorias de issues e como isso influencia as reaberturas, analisamos as pontuações negativas e positivas, e a densidade de comentários negativos e positivos entre a abertura e o fechamento das issues.

Nos resultados, tanto para issues com reaberturas quanto para aquelas sem reaberturas, a pontuação negativa média (NM) não apresentou termos extremamente negativos, com a maioria das categorias mostrando valores consistentes em -1. No entanto, a categoria *Configuração* teve uma mediana um pouco mais negativa em issues reabertas, indicando mais palavras negativas. Para issues sem reaberturas, a categoria *Desempenho* teve a pontuação negativa mais baixa, sugerindo sentimentos negativos mais intensos.

Quanto à pontuação positiva média (PM), a maioria dos comentários não tinha palavras positivas, com valores mínimos e o primeiro quartil (Q1) iguais a 1. Algumas

categorias, como *GUI* e *Configuração*, mostraram um pouco mais de positividade, mas a categoria *Funcional* teve quase nenhuma expressão positiva.

Analisando a densidade de comentários negativos (DCN), para issues sem reaberturas, a maioria das categorias não mostrou muitos sentimentos negativos, mas categorias como *Configuração* e *GUI* tiveram uma maior frequência de comentários negativos. Para issues reabertas, a densidade de sentimentos negativos foi mais alta em *Configuração* e *GUI*.

Na densidade de comentários positivos (DCP), os resultados foram semelhantes, com a maioria das categorias mostrando baixa presença de sentimentos positivos. No entanto, *Info* e *GUI* tiveram uma maior quantidade de positividade, enquanto *Funcional* continuou a ter uma densidade muito baixa.

A análise dos sentimentos nas diferentes categorias revela que *Configuração*, *Desempenho* e *GUI* estão associadas a uma maior frequência de sentimentos negativos, especialmente em issues reabertas. Por outro lado, a categoria *Funcional* tende a ter uma presença muito baixa de sentimentos, tanto negativos quanto positivos. Essas observações indicam que discussões sobre *Configuração* e *GUI* podem refletir maior frustração ou insatisfação, o que pode estar relacionado às reaberturas de issues. A relação entre sentimentos e reaberturas pode ser complexa e merece uma análise mais aprofundada. Na próxima seção, serão discutidas as possíveis ameaças à validade do estudo.

5.4 AMEAÇAS À VALIDADE

Validade Interna. O pré-processamento dos textos e a análise com SentiStrength-SE podem ter introduzido viés, caso o o pré-processamento não tenha sido suficientemente rigoroso ou a ferramenta não tenha capturado adequadamente os sentimentos em discussões técnicas. No entanto, essa ameaça foi mitigada, pois o pré-processamento foi realizado de forma criteriosa, utilizando técnicas como remoção de dados irrelevantes e conversão de emojis em texto. Além disso, ao *SentiStrength-SE* foi escolhida por ser especializado no domínio de desenvolvimento de software, minimizando a possibilidade de interpretações incorretas dos sentimentos. Outro ponto crítico é o impacto do pré-processamento na categorização das issues. Existe a possibilidade de que informações importantes tenham sido removidas ou ruídos introduzidos durante essa etapa, mas essa ameaça foi mitigada pelo uso de técnicas bem estabelecidas, garantindo que as informações essenciais das issues fossem preservadas para que o classificador MLP operasse com dados limpos e relevantes.

Além disso, há o risco de que as pontuações de sentimentos geradas pela *SentiStrength-SE* não reflitam com precisão o sentimento real dos desenvolvedores. Essa ameaça foi refutada pelo fato de a ferramenta ter sido desenvolvida especificamente para o contexto de desenvolvimento de software. Ademais, o uso de métricas estatísticas adequadas, como mínimo, máximo, média, mediana, Q1 e Q3, proporcionou uma visão clara das distribuições de sentimentos. A remoção de *outliers* também pode ter afetado a distribuição dos dados e, consequentemente, os resultados da análise. Contudo, essa ameaça foi mitigada pela aplicação da técnica de amplitude interquartil (IQR) para remover valores extremos, o que garantiu que a análise fosse focada nos padrões centrais dos dados, evitando distorções causadas por *outliers*.

Validade Externa. A base de dados MSR14 pode não refletir completamente o comportamento de issues em outros repositórios ou domínios de software. No entanto, essa ameaça foi mitigada, já que a MSR14 abrange repositórios de diferentes domínios, o que aumenta a confiança na generalização dos resultados. Outra preocupação é que a coleta de dados tenha sido limitada até 2020, não refletindo práticas de desenvolvimento ou sentimentos mais recentes. Essa limitação foi refutada pelo fato de que a base de dados capturou um conjunto representativo e diversificado de issues e projetos, oferecendo uma visão robusta das práticas de desenvolvimento do período analisado.

Validade de Construto. Uma possível ameaça está na categorização das issues, que pode ser considerada simplista e incapaz de capturar nuances mais detalhadas. No entanto, essa ameaça foi mitigada pela utilização da taxonomia de Catolino et al. (2019), publicada em uma revista amplamente reconhecida, além do uso de métricas de sentimentos que permitiram explorar padrões de comportamento nas diferentes categorias de issues. Além disso, o uso do classificador MLP pode não capturar toda a complexidade associada à categorização de issues. Essa ameaça foi refutada, já que o MLP foi escolhido com base em seu desempenho superior em experimentos anteriores, onde obteve as melhores métricas de acurácia, precisão, revocação e F1-score após o balanceamento da base de treinamento com a combinação de sobre e subamostragem de dados utilizando o método SMOTEENN. O pré-processamento dos textos também foi feito de maneira criteriosa, garantindo que o MLP trabalhasse com dados adequados para realizar uma categorização precisa.

Por fim, a ferramenta *SentiStrength-SE* pode não capturar todos os aspectos subjetivos dos sentimentos expressos em discussões técnicas. No entanto, essa ameaça foi refutada pelo fato de o *SentiStrength-SE* ter sido desenvolvido especificamente para o domínio de desenvolvimento de software, o que garante sua capacidade de lidar com sentimentos expressos em discussões técnicas. Além disso, o pré-processamento dos textos foi realizado de forma rigorosa, assegurando a qualidade dos dados e minimizando ruídos, o que fortalece a validade dos resultados.

Validade de Conclusão. As métricas estatísticas utilizadas no estudo, como mínimo, máximo, média, mediana, Q1 e Q3, podem não capturar completamente os aspectos subjetivos e mais complexos dos sentimentos expressos nas issues. No entanto, essa ameaça foi refutada, pois essas métricas fornecem uma visão descritiva robusta das distribuições de sentimentos e permitem a identificação de padrões gerais entre os grupos de issues com e sem reabertura. Elas são amplamente reconhecidas como ferramentas eficazes para descrever a variabilidade dos dados e detectar tendências.

Além disso, a análise incluiu métricas detalhadas que abordam aspectos específicos dos sentimentos ao longo do ciclo de vida das issues. Foram consideradas a pontuação negativa e positiva, o sentimento e a intensidade do sentimento no primeiro comentário, assim como essas mesmas métricas no último comentário antes do primeiro fechamento. Também foram analisadas a pontuação positiva média, a pontuação negativa média, a densidade dos sentimentos negativos e positivos nos comentários entre a abertura e o primeiro fechamento das issues.

Embora essas métricas forneçam uma base sólida para a análise dos padrões senti-

mentais, podem ter limitações na captura de aspectos qualitativos mais profundos, como motivações subjacentes ou nuances emocionais mais sutis. Contudo, seu uso é justificado para estabelecer uma base quantitativa robusta, que permitindo comparar diferenças e similaridades nos padrões de sentimentos entre issues reabertas e não reabertas.

Além disso, a diferença no número de issues entre os grupos com e sem reaberturas pode representar uma ameaça à validade dos resultados. No entanto, optamos por utilizar todos os dados disponíveis, evitando balanceamentos artificiais, para refletir os comportamentos naturais das issues. Esse enfoque garante que os resultados sejam representativos da realidade dos repositórios estudados, sem introduzir possíveis distorções decorrentes de tentativas de equivalência entre os grupos.

5.5 CONCLUSÃO DO CAPÍTULO

Neste capítulo, realizamos uma análise abrangente de diversas métricas destinadas a avaliar os sentimentos presentes nas discussões ocorridas em grupos de issues, explorando tanto as categorias de issues quanto os diferentes estágios do ciclo de vida de uma issue.

A análise revelou que as categorias das issues desempenham um papel mais significativo do que os sentimentos na determinação da probabilidade de reabertura. A categoria *Configuração* frequentemente apresentou sentimentos negativos mais intensos, sugerindo que desafios nesta área estão mais propensos a levar à reabertura de issues. Em contraste, a categoria *Informação* demonstrou uma maior pontuação positiva, indicando discussões mais construtivas e uma menor probabilidade de reabertura. No próximo capítulo, apresentaremos as conclusões finais da tese.

Capítulo

6

CONCLUSÃO E PERSPECTIVAS FUTURAS

Nesta teste, nós buscamos investigar a eficácia da análise de sentimentos nos textos contidos em discussões de uma issue e sua categoria como um meio de prever a reabertura dessas issues em repositórios de software do GitHub. O ponto de partida foi um estudo piloto para conhecer três ferramentas de análise de sentimentos desenvolvidas no contexto de engenharia de software, dentre as ferramentas a SentiStrength-SE foi escolhida devido a sua facilidade de uso e limitações de hardware. Em seguida foi realizada uma validação do léxico da ferramenta escolhida.

A partir da ferramenta de análise de sentimentos escolhida, nos realizamos um estudo sobre a análise dos sentimentos em issues com reaberturas dos projetos listados na Mining Changeng da conferência MSR 2014 e outro estudo estatístico sobre issues com e sem reaberturas sobre o ponto de vista dos sentimentos presentes nas discussões entre a abertura e o primeiro fechamento das issues.

Após as análises dos sentimentos decidimos investigar os tipos de categorias de issues e sua relação com reabertura de issues, e para isso foi preciso construir um categorizador automático de issues a partir dos textos contidos no título e descrição da issue e aplicar nas issues da base de dados MSR2014. Por último, investigamos os sentimentos contidos nas discussões das issues em cada tipo de categoria nas reaberturas das issues.

Os resultados demonstraram que, embora a análise de sentimentos seja uma ferramenta valiosa para entender as opiniões e sentimentos dos desenvolvedores e usuários envolvidos em projetos, ela não se revelou suficiente para identificar de forma precisa e confiável se uma issue será reaberta. No entanto, a categorização de issues emergiu como uma abordagem mais eficaz na identificação da probabilidade de reabertura de issues. Isso sugere que a compreensão mais profunda das características e contextos das issues em si é crucial para prever seu ciclo de vida, enquanto a análise de sentimentos pode complementar essa análise, mas não deve ser considerada como a única métrica para essa finalidade. Portanto, esta tese destaca a importância de adotar abordagens multifacetadas e contextuais para melhorar a qualidade do gerenciamento de issues em projetos de desenvolvimento de software, contribuindo para uma maior eficiência e eficácia no processo de desenvolvimento de software.

6.1 CONSTRIBUIÇÕES

Esta tese apresenta as seguintes contribuições:

- Estudo sobre Análise de Sentimentos em issues do Github.
  - Boechat, Gláucya; Mota Jr, Joselito; Machado, Ivan; Mendonça, Manoel. Análise de Sentimentos em Discussões de Issues Reabertas do Github. In: WORKSHOP DE VISUALIZAÇÃO, EVOLUÇÃO E MANUTENÇÃO DE SOFTWARE (VEM), 1. , 2019, Salvador. Porto Alegre: SBC, 2019 . p. 1-8. (Prêmio de melhor artigo)
- Validação do dicionário Léxico da ferramenta *SentiStrength-SE*
  - Menezes, Hiolanda; Boechat, Gláucya; Mota Jr, Joselito; Machado, Ivan. Validação e construção de um dicionário léxico para auxiliar a análise de sentimentos em repositórios de projetos de software. In: WORKSHOP DE VISUALIZAÇÃO, EVOLUÇÃO E MANUTENÇÃO DE SOFTWARE (VEM), 8. , 2020, Evento Online. Porto Alegre: SBC, 2020 . p. 41-48.
- Concepção e desenvolvimento de uma categorização automática de issues.
- Criação da base de dados de issues do GitHub categorizadas pelo tipo de issues e classificação dos sentimentos da issues e dos seus comentários.
- Outras contribuições
  - Júnior, Joselito; Boechat, Gláucya; Machado, Ivan. Label it be! A large-scale study of issue labeling in modern open-source repositories. In: XXIV Ibero-American Conference on Software Engineering (CIbSE). San Jose, Costa Rica, 2021
  - Amorim, A., Boechat, G., Novais, R., Vieira, V. and Villela, K. Quality Attributes Analysis in a Crowdsourcing-based Emergency Management System. In Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS), Volume 2, pages 501-509. Porto, Portugal 2017.

6.2 LIMITAÇÕES

Podemos identificar algumas limitações do trabalho quanto ao uso de uma ferramenta para analise de sentimento baseada em léxico. Embora essa ferramenta seja eficiente em termos de tempo de processamento e consumo de memória, é possível que o uso de técnicas de aprendizado de máquina, como os modelos de aprendizado profundo, proporcionasse resultados mais precisos e robustos. No entanto, essas técnicas exigem significativamente mais recursos computacionais e tempo para realizar o processo de análise de sentimentos, o que pode limitar sua aplicabilidade em grandes volumes de dados. Além disso, nos estudos não foram consideradas as reações (+1, -1, *laugh*, *confused*, *heart*, *hooray*, *rocket* e *eyes*) dos comentários e da descrição da issue, estes podem indicar que outros colaboradores

concordam ou discortam com o conteúdo texto. A inclusão dessas reações em análises futuras poderia enriquecer a compreensão dos sentimentos expressos nas discussões.

Outro ponto a ser destacado é que a coleta das issues foi finalizada em 2020. Desde então, novas issues podem ter sido criadas, outras fechadas ou reabertas, o que limita a atualização e a relevância dos dados analisados. Contudo, é importante mencionar que as tarefas de mineração, categorização e análise de sentimentos demandam um tempo considerável, o que justifica o período de coleta utilizado.

Finalmente, utilizamos técnicas tradicionais de aprendizado de máquina para construir o categorizador automático de issues. No futuro, pretendemos explorar técnicas de aprendizado profundo, que têm demonstrado um desempenho superior na classificação de texto devido à sua capacidade de aprender representações complexas e hierárquicas diretamente dos dados brutos (DEVLIN et al., 2019). Essa abordagem pode superar algumas das limitações identificadas neste estudo.

6.3 TRABALHOS FUTUROS

O trabalho foi realizado apenas sobre os textos das issues, sem fazer análises dos commits, releases, os tipos de usuários, localização, entre outros fatores. Essas informações adicionais podem ffornecer uma compreensão mais completa do problema das issues reabertas.Com base nisso, sugerimos os seguintes trabalhos futuros:

1. **Análise da Conexão entre Commits e Reabertura de Issues**
    - Realizar um estudo para explorar a relação entre as mensagens e frequência dos commits associadas a issues reabertas, com o objetivo de identificar padrões de desenvolvimento que possam contribuir para a reabertura.
2. **Integração de Dados de Releases e Impacto na Reabertura de Issues**
    - Investigar como as releases do projeto impactam a reabertura de issues, analisando se determinadas releases estão associadas a um aumento na taxa de reabertura e quais características dessas releases podem influenciar esse fenômeno.
3. **Análise Geográfica e Temporal de Issues Reabertas**
    - Estudar como a localização geográfica dos colaboradores e o fuso horário influenciam a reabertura de issues, considerando diferenças culturais e temporais que possam afetar a comunicação e a resolução de problemas.
4. **Caracterização de Tipos de Usuários e sua Relação com Reabertura de Issues**
    - Realizar um survey ou entrevistas com os colaboradores do projeto para investigar os tipos de colaboradores que mais frequentemente reabrem issues, analisando aspectos como o nível de experiência, o papel no projeto (mantenedor, colaborador ocasional, etc.), e o comportamento de interação dentro do repositório.

5. **Desenvolvimento de Ferramentas de Previsão de Reabertura de Issues**

   - Desenvolver modelos preditivos utilizando *Large Language Models* (LLMs) para antecipar a reabertura de issues com base em características como a categoria da issue, os sentimentos e conteúdo das discussões, o histórico de commits, o comportamento dos colaboradores, e os dados das releases.

6. **Exploração de Padrões de Comunicação em Issues Reabertas**

   - Realizar um estudo para identificar os padrões de comunicação e colaboração entre os usuários em issues que foram reabertas, identificando se há uma linguagem ou comportamento específico que pode estar relacionado com a reabertura.

7. **Análise da Influência de Pull Requests na Reabertura de Issues**

   - Realizar um estudo para investigar como as interações e decisões tomadas em pull requests podem influenciar a reabertura de issues. Isso pode incluir a análise do conteúdo das discussões em pull requests, o tempo de revisão, as alterações solicitadas, e como essas dinâmicas se correlacionam com a probabilidade de uma issue ser reaberta.

REFERÊNCIAS BIBLIOGRÁFICAS

ABE, S. *Support vector machines for pattern classification*. Loughborough, UK: Springer, 2005.

AGGARWAL, C. C. *Neural networks and deep learning: a textbook*. Yorktown Heights, NY, USA: Spinger, 2018.

AHMED, T.; BOSU, A.; IQBAL, A.; RAHIMI, S. Senticr: A customized sentiment analysis tool for code review interactions. In: *Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering*. Piscataway, NJ, USA: IEEE Press, 2017. (ASE 2017), p. 106–111. ISBN 9781538626849.

AHMED, T.; BOSU, A.; IQBAL, A.; RAHIMI, S. *senticr/SentiCR GitHub repository*. 2017. Commit 8f774cc45c48d649d2c95816c1adba1e26db83ee on 5 Nov 2017. Disponível em: <https://github.com/senticr/SentiCR>. Acesso em: 12/07/2023.

ANTONIOL, G.; AYARI, K.; PENTA, M. D.; KHOMH, F.; GUÉHÉNEUC, Y.-G. Is it a bug or an enhancement? In: *Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research Meeting of Minds - CASCON '08*. Ontario,Canada: ACM, 2008. p. 304–318.

ANVIK, J.; HIEW, L.; MURPHY, G. C. Who should fix this bug? In: *Proceedings of the 28th International Conference on Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2006. (ICSE '06), p. 361–370. ISBN 1595933751.

BASARI, A. S. H.; HUSSIN, B.; ANANTA, I. G. P.; ZENIARJA, J. Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization. *Procedia Engineering*, v. 53, p. 453 – 462, 2013. Malaysian Technical Universities Conference on Engineering & Technology 2012, MUCET 2012.

BATISTA, G. E. d. A. P.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, ACM, v. 6, n. 1, p. 20–29, 2004.

BEHL, S.; RAO, A.; AGGARWAL, S.; CHADHA, S.; PANNU, H. Twitter for disaster relief through sentiment analysis for covid-19 and natural hazard crises. *International Journal of Disaster Risk Reduction*, v. 55, p. 102101, 2021. ISSN 2212-4209.

BENEVENUTO, F.; ARAúJO, M.; RIBEIRO, F. Sentiment analysis methods for social media. In: *Proceedings of the 21st Brazilian Symposium on Multimedia and the Web*. New York, NY, USA: Association for Computing Machinery, 2015. (WebMedia '15), p. 11. ISBN 9781450339599.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. Beijing: O'Reilly, 2009.

BISHOP, C. M.; NASRABADI, N. M. *Pattern recognition and machine learning*. [S.l.]: Springer, 2006.

BOECHAT, G.; JÚNIOR, J. M.; MACHADO, I.; MENDONÇA, M. Análise de sentimentos em discussões de issues reabertas do github. In: *Anais do VII Workshop on Software Visualization, Evolution and Maintenance (VEM)*. Porto Alegre, RS, Brasil: SBC, 2019. p. 13–20. Disponível em: <https://sol.sbc.org.br/index.php/vem/article/view/7579>.

BOECHAT, G.; JÚNIOR, J. M.; MACHADO, I.; MENDONÇA, M. *Análise de Sentimentos em Discussões de Issues Reabertas do Github (Material Suplementar)*. Salvador, BA: Zenodo, 2019. Zenodo. <http://doi.org/10.5281/zenodo.3376175>.

BORGES, H.; BRITO, R.; VALENTE, M. T. Beyond textual issues: Understanding the usage and impact of github reactions. In: *Anais do XXXIII Brazilian Symposium on Software Engineering*. Porto Alegre, RS, Brasil: SBC, 2019.

BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In: LECHEVALLIER, Y.; SAPORTA, G. (Ed.). *Proceedings of COMPSTAT'2010*. Heidelberg: Physica-Verlag HD, 2010. p. 177–186. ISBN 978-3-7908-2604-3.

CAGLAYAN, B.; MISIRLI, A. T.; MIRANSKYY, A.; TURHAN, B.; BENER, A. Factors characterizing reopened issues: A case study. In: *Proceedings of the 8th Int. Conf. on Predictive Models in Soft. Engineering*. New York, USA: ACM, 2012. p. 1–10.

CALEFATO, F.; LANUBILE, F.; MAIORANO, F.; NOVIELLI, N. *Senti4SD GitHub repository*. 2018. Release v1.0 n 20 Jun 2018, commit 8a3467e9e8dbaa0af9343862aadc83a56a4668bc on 24 Jun 2019. Disponível em: <https://github.com/collab-uniba/Senti4SD/>. Acesso em: 12/07/2023.

CALEFATO, F.; LANUBILE, F.; MAIORANO, F.; NOVIELLI, N. Sentiment polarity detection for software development. In: *Proceedings of the 40th International Conference on Software Engineering*. New York, NY, USA: ACM, 2018. (ICSE '18), p. 128–128.

CALEFATO, F.; LANUBILE, F.; NOVIELLI, N. How to ask for technical help? evidence-based guidelines for writing questions on stack overflow. *Information and Software Technology*, v. 94, p. 186 – 207, 2018.

CARIGE, R.; CARNEIRO, G. Sentiment polarity of programmers in an open source software project: An exploratory study. In: *Proceedings of the XXXIV Brazilian Symposium on Software Engineering (SBES 2020)*. New York, NY, USA: ACM, 2020. (SBES 2020).

CARREÑO, L. V. G.; WINBLADH, K. Analysis of user comments: An approach for software requirements evolution. In: *35th International Conference on Software Engineering (ICSE)*. San Francisco, CA, USA: IEEE COMPUTER SOCIETY, 2013. p. 582–591.

CATOLINO, G.; PALOMBA, F.; ZAIDMAN, A.; FERRUCCI, F. Not all bugs are the same: Understanding, characterizing, and classifying bug types. *The Journal of Systems and Software*, v. 152, p. 165–181, 2019. ISSN 0164-1212.

CATOLINO, G.; PALOMBA, F.; ZAIDMAN, A.; FERRUCCI, F. *Not All Bugs Are the Same: Understanding, Characterizing, and Classifying the Root Cause of Bugs*. 2023. Dataset modified on 2023-02-15, 06:03. Disponível em: <https://figshare.com/s/dcb95c70c4472b2ac935>. Acesso em: 18/07/2023.

CHACON, B. S. S. *Pro Git*. 2. ed. New York, NY: Springer, 2014.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, v. 16, p. 321–357, 2002.

CHERUVELIL, J.; SILVA, B. C. d. Developers' sentiment and issue reopening. In: *Proceedings of the 4th International Workshop on Emotion Awareness in Software Engineering*. IEEE Press, 2019. (SEmotion '19), p. 29–33. Disponível em: <https://doi.org/10.1109/SEmotion.2019.00013>.

CHILLAREGE, R.; BHANDARI, I. S.; CHAAR, J. K.; HALLIDAY, M. J.; MOEBUS, D. S.; RAY BANERJEE K.AND WONG, M.-Y. Orthogonal defect classification-a concept for in-process measurements. *IEEE Transactions on Software Engineering*, v. 18, n. 11, p. 943–956, 1992.

DANESCU-NICULESCU-MIZIL, C.; SUDHOF, M.; JURAFSKY, D.; LESKOVEC, J.; POTTS, C. A computational approach to politeness with application to social factors. *CoRR*, abs/1306.6078, 2013.

DESTEFANIS, G.; ORTU, M.; BOWES, D.; MARCHESI, M.; TONELLI, R. On measuring affects of github issues' commenters. In: *Proceedings of the 3rd International Workshop on Emotion Awareness in Software Engineering*. New York, NY, USA: ACM, 2018. (SEmotion '18), p. 14–19.

DEVLIN, J.; CHANG, M.; LEE, K.; TOUTANOVA, K. BERT: pre-training of deep bidirectional transformers for language understanding. Association for Computational Linguistics, Minneapolis, Minnesota, p. 4171–4186, jun 2019.

DEY, L.; CHAKRABORTY, S.; BISWAS, A.; BOSE, B.; TIWARI, S. Sentiment analysis of review datasets using naïve bayes' and k-nn classifier. *International Journal of Information Engineering and Electronic Business*, MECS Publisher, v. 8, n. 4, p. 54–62, Jul 2016.

FARIAS, M. A. de F.; NOVAIS, R.; JÚNIOR, M. C.; CARVALHO, L. P. da S.; MENDONCA, M.; SPÍNOLA, R. O. A systematic mapping study on mining software repositories. In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2016. (SAC '16), p. 1472–1479.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Knowledge discovery and data mining: Towards a unifying framework. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon, USA: AAAI Press, 1996. (KDD'96), p. 82–88.

GIGER, E.; PINZGER, M.; GALL, H. Predicting the fix time of bugs. In: *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*. New York, NY, USA: ACM, 2010. (RSSE '10), p. 52–56.

GITHUB. *gemoji*. GitHub, 2023. Release gemoji 4.1.0 on Mar 29, 2023. Disponível em: <https://github.com/github/gemoji>. Acesso em: 25/09/2023.

GITHUB-TOPICS. *Topics on GitHub*. 2020. Accessed september 10, 2023. Disponível em: <https://github.com/topics/>.

GITHUB_DOCS. *Hello World*. 2023. Disponível em: <https://docs.github.com/en/get-started/quickstart/hello-world>. Acesso em: 12/07/2023.

GITHUB_INC. *About GitHub*. 2023. Accessed Janury 5, 2021. Disponível em: <https://github.com/about>. Acesso em: 12/07/2023.

GITHUB_INC. *GitHub Issues documentation*. 2023. Disponível em: <https://docs.github.com/en/issues>. Acesso em: 02/08/2023.

GITHUB_INC. *Issue event types - GitHub Docs*. 2023. Disponível em: <https://docs.github.com/en/webhooks-and-events/events/issue-event-types>. Acesso em: 02/08/2023.

GUZMAN, E.; AZóCAR, D.; LI, Y. Sentiment analysis of commit comments in github: An empirical study. In: *Proceedings of the 11th Working Conference on Mining Software Repositories*. New York, NY, USA: ACM, 2014. (MSR 2014), p. 352–355.

GUZMAN, E.; BRUEGGE, B. Towards emotional awareness in software development teams. In: *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. New York, NY, USA: ACM, 2013. (ESEC/FSE 2013), p. 671–674.

GUZMAN, E.; MAALEJ, W. How do users like this feature? a fine grained sentiment analysis of app reviews. In: *2014 IEEE 22nd International Requirements Engineering Conference (RE)*. Karlskrona, Sweden: IEEE, 2014. p. 153–162.

HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques*. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 0123814790.

Hassan, A. E. The road ahead for mining software repositories. In: *2008 Frontiers of Software Maintenance*. Beijing, China: IEEE, 2008. p. 48–57.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H.; FRIEDMAN, J. H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer, 2009.

IRAM, A. Sentiment analysis of student's facebook posts. In: BAJWA, I. S.; KAMA-REDDINE, F.; COSTA, A. (Ed.). *Intelligent Technologies and Applications*. Singapore: Springer Singapore, 2019. p. 86–97.

ISLAM, M. R.; ZIBRAN, M. F. Leveraging automated sentiment analysis in software engineering. In: *14th Int. Conf. on Min. Soft. Repositories(MSR)*. Buenos Aires Argentina: IEEE Press, 2017. p. 203–214.

ISLAM, M. R.; ZIBRAN, M. F. SentiStrength-SE: Exploiting domain specificity for improved sentiment analysis in software engineering text. *J. of Systems and Software*, v. 145, p. 125 – 146, 2018.

IZQUIERDO, J. L. C.; COSENTINO, V.; ROLANDI, B.; BERGEL, A.; CABOT, J. Gila: Github label analyzer. In: *Proceedings of the International Conference on Software Analysis, Evolution, and Reengineering (SANER)*. [S.l.: s.n.], 2015. p. 479–483.

JACQUES, V. *PyGithub*. 2020. Revision 14ae2cad. Disponível em: <https://pygithub.readthedocs.io>. Acesso em: 01/02/2024.

JÚNIOR, J. M.; BOECHAT, G.; MACHADO, I. Label it be! A large-scale study of issue labeling in modern opensource repositories. In: *24th Iberoamerican Conference on Software Engineering (CIbSE 2021)*. San Jose, Costa Rica: Curran Associates, 2021. p. 262–275.

JURADO, F.; RODRIGUEZ, P. Sentiment analysis in monitoring software development processes: An exploratory case study on github's project issues. *Journal of Systems and Software*, v. 104, p. 82–89, 2015. ISSN 0164-1212. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0164121215000485>.

JÚNIOR, J. M. *joselitojunior94/gfetcher GitHub repository*. 2019. Commit 4988b4029d36dca9667fd7a9f431cc338b9f8c01 on 18 Oct 2020. Disponível em: <https://github.com/joselitojunior94/gfetcher>. Acesso em: 12/07/2023.

KALLIS, R.; CHAPARRO, O.; SORBO, A. D.; PANICHELLA, S. NLBSE'22 tool competition. In: *1st International Workshop on Natural Language-Based Software Engineering (NLBSE)*. Pittsburgh, PA, USA: IEEE, 2022. p. 25–28.

KALLIS, R.; Di Sorbo, A.; CANFORA, G.; PANICHELLA, S. Ticket tagger: Machine learning driven issue classification. In: *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. Cleveland, OH, USA: IEEE, 2019. p. 406–409.

KALLIS, R.; Di Sorbo, A.; CANFORA, G.; PANICHELLA, S. Predicting issue types on github. *Science of Computer Programming*, v. 205, p. 102598, 2021. ISSN 0167-6423. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167642320302069>.

KIBRIYA, A. M.; FRANK, E.; PFAHRINGER, B.; HOLMES, G. Multinomial naive bayes for text categorization revisited. In: SPRINGER. *AI 2004: Advances in Artificial*

Intelligence: 17th Australian Joint Conference on Artificial Intelligence. Cairns, Australia, 2005. p. 488–499.

KSHIRSAGAR, A. P.; CHANDRE, P. R. Issue tracking system with duplicate issue detection. In: *Proceedings of the Sixth International Conference on Computer and Communication Technology 2015*. New York, NY, USA: ACM, 2015. (ICCCT '15), p. 41–45.

LIU, B. Sentiment analysis and subjectivity. In: INDURKHYA, N.; DAMERAU, F. J. (Ed.). *Handbook of Natural Language Processing*. Second edition. Boca Ration, FL: CRC Press, 2010. cap. 25, p. 627–666. ISBN 978-1-4200-8592-1.

LIU, B. *Sentiment Analysis and Opinion Mining*. 1. ed. Switzerland: Morgan and Claypool Publishers, 2012. (Synthesis Lectures on Human Language Technologies (SLHLT)).

LIU, B. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge: Cambridge University Press, 2015.

LIU, B. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. 2. ed. Cambridge: Cambridge University Press, 2020. (Studies in Natural Language Processing).

Liu, B.; Blasch, E.; Chen, Y.; Shen, D.; Chen, G. Scalable sentiment classification for big data analysis using naïve bayes classifier. In: *2013 IEEE International Conference on Big Data*. Silicon Valley, CA, USA: IEEE, 2013. p. 99–104.

MANI, I.; ZHANG, I. KNN approach to unbalanced data distributions: a case study involving information extraction. In: *Proceedings of workshop on learning from imbalanced datasets*. Washington DC: ICML, 2003. v. 126.

MEENA, A.; PRABHAKAR, T. V. Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In: AMATI, G.; CARPINETO, C.; ROMANO, G. (Ed.). *Advances in Information Retrieval*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 573–580.

MENDENHALL, W.; BEAVER, R. J.; BEAVER, B. M. *Introduction to probability and statistics*. USAf: Cengage Learning, 2012.

MENDONCA, M. Mineração de dados. In: *Anais da VI Escola Reginal de Informática de São Paulo*. São Carlos, SP, Brasil: ICMC - Universidade de São Paulo, 2001. v. 1, p. 189–218.

MENEZES, H.; BOECHAT, G.; JúNIOR, J. M.; MACHADO, I. *Validação e construção de um dicionário léxico para auxiliar a análise de sentimentos em repositórios de projetos de software (Material Suplementar)*. (Online): Zenodo, 2020. Zenodo <http://doi.org/10.5281/zenodo.4029777>.

MERTEN, T.; KRÄMER, D.; MAGER, B.; SCHELL, P.; BÜRSNER, S.; PAECH, B. Do information retrieval algorithms for automated traceability perform effectively on issue tracking system data? In: DANEVA, M.; PASTOR, O. (Ed.). *Requirements Engineering:*

Foundation for Software Quality. Cham: Springer International Publishing, 2016. p. 45–62. ISBN 978-3-319-30282-9.

MI, Q.; KEUNG, J. An empirical analysis of reopened bugs based on open source projects. In: *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2016. (EASE '16), p. 1–10. ISBN 9781450336918.

MOHAMED, A.; ZHANG, L.; JIANG, J.; KTOB, A. Predicting which pull requests will get reopened in github. In: *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*. Nara, Japan: IEEE, 2018. p. 375–385.

MSR. *2014 Mining Challenge Dataset*. 2014. Disponível em: <https://2014.msrconf.org/challenge.php>. Acesso em: 22/01/2021.

MSR '19: Proceedings of the 16th International Conference on Mining Software Repositories. IEEE Press, 2019. Accessed November 2, 2020. Disponível em: <https://2019.msrconf.org/>.

MSR '21: Proceedings of the 18th International Conference on Mining Software Repositories. New York, NY, USA: Association for Computing Machinery, 2021. Disponível em: <https://2021.msrconf.org/>.

MURGIA, A.; TOURANI, P.; ADAMS, B.; ORTU, M. Do Developers Feel Emotions? An Exploratory Analysis of Emotions in Software Artifacts. In: *Proceedings of the 11th Working Conference on Mining Software Repositories*. New York, NY, USA: ACM, 2014. (MSR 2014), p. 262–271.

NOVAK, P. K.; SMAILOVIć, J.; SLUBAN, B.; MOZETIč, I. *Emoji sentiment data*. Kaggle, 2015. Slovenian language resource repository CLARIN.SI. Disponível em: <https://www.kaggle.com/thomasseleck/emoji-sentiment-data>. Acesso em: 12/07/2023.

OCTOVERSE-GITHUB. *Site*. 2023. Disponível em: <https://octoverse.github.com/>. Acesso em: 12/07/2023.

ORTU, M.; ADAMS, B.; DESTEFANIS, G.; TOURANI, P.; MARCHESI, M.; TONELLI, R. Are bullies more productive? empirical study of affectiveness vs. issue fixing time. In: *Proceedings of the 12th Working Conference on Mining Software Repositories*. [S.l.]: IEEE Press, 2015. (MSR '15), p. 303–313. ISBN 9780769555942.

ORTU, M.; DESTEFANIS, G.; ADAMS, B.; MURGIA, A.; MARCHESI, M.; TONELLI, R. The jira repository dataset: Understanding social aspects of software development. In: *Proceedings of the 11th Int. Conf. on Predictive Models and Data Analytics in Software Engineering*. New York, NY, USA: ACM, 2015. (PROMISE '15), p. 1–4.

ORTU, M.; MURGIA, A.; DESTEFANIS, G.; TOURANI, P.; TONELLI, R.; MARCHESI, M.; ADAMS, B. The emotional side of software developers in jira. In: *Proceedings of the 13th Int. Conf. on Mining Soft. Repositories(MSR)*. NY, USA: ACM, 2016. p. 480–483. ISBN 978-1-4503-4186-8.

PAN, J.; MAO, X. An empirical study on interaction factors influencing bug reopenings. In: *21st Asia-Pacific Soft. Engineering Conf.* Jeju, South Korea: IEEE, 2014. v. 2, p. 39–42.

PANDEY, N.; HUDAIT, A.; SANYAL, D. K.; SEN, A. Automated classification of issue reports from a software issue tracker. In: SA, P. K.; SAHOO, M. N.; MURUGAPPAN, M.; WU, Y.; MAJHI, B. (Ed.). *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*. Singapore: Springer Singapore, 2018. p. 423–430. ISBN 978-981-10-3373-5.

PANDEY, N.; SANYAL, D.; HUDAIT, A. et al. Automated classification of software issue reports using machine learning techniques: an empirical study. *Innovations in Systems and Software Engineering*, Springer, v. 13, n. 4, p. 279–297, 2017.

PANG, B.; LEE, L. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, Now Publishers Inc., v. 2, n. 1-2, p. 1–135, 2008.

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up? sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*. USA: Association for Computational Linguistics, 2002. (EMNLP '02), p. 79–86.

PANKAJ; PANDEY, P.; MUSKAN; SONI, N. Sentiment analysis on customer feedback data: Amazon product reviews. In: *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*. Faridabad, India: IEEE, 2019. p. 320–322.

PARROTT, W. G. *Emotions in Social Psychology - Essential Readings*. Sebastopol, CA, USA: Psychology Press, 2001.

PLETEA, D.; VASILESCU, B.; SEREBRENIK, A. Security and emotion: Sentiment analysis of security discussions on github. In: *Proceedings of the 11th Working Conference on Mining Software Repositories*. New York, NY, USA: ACM, 2014. (MSR 2014), p. 348–351.

Python-Software-Foundation. *re - Regular expression operations*. 2023. Disponível em: <https://docs.python.org/3/library/re.html>. Acesso em: 12/07/2023.

RAATIKAINEN, M.; MOTGER, Q.; LÜDERS, C. M.; FRANCH, X.; MYLLYAHO, L.; KETTUNEN, E.; MARCO, J.; TIIHONEN, J.; HALONEN, M.; MÄNNISTÖ, T. Improved management of issue dependencies in issue trackers of large collaborative projects. *IEEE Transactions on Software Engineering*, IEEE Computer Society, Los Alamitos, CA, USA, v. 49, n. 04, p. 2128–2148, April 2023. ISSN 1939-3520.

ROBBES, R.; HILL, E.; BIRD, C. Guest Editorial: Special Section on Mining Software Repositories. *Empirical Software Engineering*, v. 23, n. 2, p. 833–834, Apr 2018.

SAHU, K.; CHOI, Y. Sentiment analysis of the united states senate twitter feeds in election year 2020. In: *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. NV, USA: IEEE, 2021. p. 0129–0133.

SANDOVAL-ALMAZAN, R.; VALLE-CRUZ, D. Facebook impact and sentiment analysis on political campaigns. In: *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*. New York, NY, USA: Association for Computing Machinery, 2018. (dg.o '18). ISBN 9781450365260.

SARLAN, A.; NADAM, C.; BASRI, S. Twitter sentiment analysis. In: *Proceedings of the 6th International Conference on Information Technology and Multimedia*. Putrajaya, Malaysia: IEEE, 2014. p. 212–216.

SENTISTRENGTH-SE. *Sentiment detection tool in Software Engineering domain*. 2017. Disponível em: <https://laser.cs.uno.edu/Projects/Projects.html>. Acesso em: 22/01/2021.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). *Biometrika*, v. 52, n. 3-4, p. 591–611, 1965.

Shihab, E.; Ihara, A.; Kamei, Y.; Ibrahim, W. M.; Ohira, M.; Adams, B.; Hassan, A. E.; Matsumoto, K. Predicting re-opened bugs: A case study on the eclipse project. In: *2010 17th Working Conference on Reverse Engineering*. Beverly, MA, USA: IEEE, 2010. p. 249–258.

SHIHAB, E.; IHARA, A.; KAMEI, Y.; IBRAHIM, W. M.; OHIRA, M.; ADAMS, B.; HASSAN, A. E.; MATSUMOTO, K.-i. Studying re-opened bugs in open source software. *Empirical Software Engineering*, v. 18, n. 5, p. 1005–1042, Oct 2013.

SIDDIQ, M. L.; SANTOS, J. C. S. Bert-based github issue report classification. In: *2022 IEEE/ACM 1st International Workshop on Natural Language-Based Software Engineering (NLBSE)*. Pittsburgh, PA, USA: IEEE, 2022. p. 33–36.

SILVA, L. A.; PERES, S. M.; BOSCARIOLI, C. *Introdução à Mineração de Dados - Com Aplicações em R*. BARUERI, SP: GEN LTC, 2016. ISBN 9788535284461.

SINGH, N.; ROY, N.; GANGOPADHYAY, A. Analyzing the sentiment of crowd for improving the emergency response services. In: *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*. Taormina, Sicily, Italy: IEEE, 2018. p. 1–8.

SINGH, N.; SINGH, P. How do code refactoring activities impact software developers' sentiments? - an empirical investigation into github commits. In: *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*. Nanjing, China: IEEE, 2017. p. 648–653.

SINGH, V. K.; PIRYANI, R.; UDDIN, A.; WAILA, P. Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In: *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*. Kottayam, Kerala, India: IEEE, 2013. p. 712–717.

Singla, Z.; Randhawa, S.; Jain, S. Sentiment analysis of customer product reviews using machine learning. In: *2017 International Conference on Intelligent Computing and Control (I2C2)*. Coimbatore, India: IEEE, 2017. p. 1–5.

SINHA, V.; LAZAR, A.; SHARIF, B. Analyzing developer sentiment in commit logs. In: *2016 IEEE/ACM 13th Working Conference on Mining Software Repositories (MSR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2016. p. 520–523.

SOMMERVILLE, I. *Software Engineering*. 9. ed. Harlow, England: Addison-Wesley, 2010.

SOUZA, R. R.; CHAVEZ, C. F.; BITTENCOURT, R. A. Patch rejection in Firefox: negative reviews, backouts, and issue reopening. *J. of Soft. Eng. Res. and Dev.*, v. 3, n. 1, Jun 2015.

SPEARMAN, C. The proof and measurement of association between two things. *American Journal of Psychology*, v. 15, n. 1, p. 72–101, 1904. Disponível em: <https://www.jstor.org/stable/1412159>.

STACKOVERFLOW. *Site*. 2023. Disponível em: <https://stackoverflow.com/>. Acesso em: 12/07/2023.

TAN, L.; LIU, C.; LI, Z.; WANG, X.; ZHOU, Y.; ZHAI, C. Bug characteristics in open source software. *Empirical Softw. Engg.*, Kluwer Academic Publishers, USA, v. 19, n. 6, p. 1665–1705, dec 2014. ISSN 1382-3256. Disponível em: <https://doi.org/10.1007/s10664-013-9258-8>.

TAN, S.; ZHANG, J. An empirical study of sentiment analysis for chinese documents. *Expert Systems with Applications*, v. 34, n. 4, p. 2622 – 2629, 2008. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417407001534>.

TANG, D. Sentiment-specific representation learning for document-level sentiment analysis. In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2015. (WSDM '15), p. 447–452. ISBN 9781450333177.

THELWALL, M.; BUCKLEY, K.; PALTOGLOU, G.; CAI, D.; KAPPAS, A. Sentiment strength detection in short informal text. *J. Am. Soc. Inf. Sci. Technol.*, John Wiley & Sons, Inc., New York, NY, USA, v. 61, n. 12, p. 2544–2558, dez. 2010.

T.MERTEN; MAGER, B.; HÜBNER, P.; QUIRCHMAYR, T.; BÜRSNER, S.; PAECH, B. Requirements communication in issue tracking systems in four open-source projects. In: *6th International Workshop on Requirements Prioritization and Communication (RePriCo)*. Essen, Germany: CEUR Workshop Proceedings, 2015. p. 114–125.

TOURANI, P.; JIANG, Y.; ADAMS, B. Monitoring sentiment in open source mailing lists -– exploratory study on the apache ecosystem. In: *Proceedings of the 2014 Conference of*

the Center for Advanced Studies on Collaborative Research. Markham, Ontario, Canada: IBM Corp., 2014. (CASCON '14), p. 34–44.

VANAJA, S.; BELWAL, M. Aspect-level sentiment analysis on e-commerce data. In: *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. [S.l.: s.n.], 2018. p. 1275–1279.

VICENZI, A. *Emojis for Python*. 2018. Disponível em: <https://emojis.readthedocs.io/>. Acesso em: 12/07/2023.

WAGNER, S.; FERNáNDEZ, D. M. Chapter 3 - analyzing text in software projects. In: BIRD, C.; MENZIES, T.; ZIMMERMANN, T. (Ed.). *The Art and Science of Analyzing Software Data*. Boston: Morgan Kaufmann, 2015. p. 39 – 72.

WANG, Y.; SUN, A.; HUANG, M.; ZHU, X. Aspect-level sentiment analysis using as-capsules. In: *The World Wide Web Conference*. New York, NY, USA: Association for Computing Machinery, 2019. (WWW '19), p. 2033–2044. ISBN 9781450366748.

WILCOXON, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, v. 1, n. 6, p. 80–83, 1945. Disponível em: <https://www.jstor.org/stable/3001968>.

XIA, X.; LO, D.; SHIHAB, E.; WANG, X.; ZHOU, B. Automatic, high accuracy prediction of reopened bugs. *Automated Software Engg.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 22, n. 1, p. 75–109, mar. 2015. ISSN 0928-8910. Disponível em: <http://dx.doi.org/10.1007/s10515-014-0162-2>.

XIA, X.; LO, D.; WANG, X.; YANG, X.; LI, S.; SUN, J. A comparative study of supervised learning algorithms for re-opened bug prediction. In: *17th European Conference on Software Maintenance and Reengineering (CSMR 2013)*. Genova, Italy: IEEE, 2013. p. 331–334.

ZHAO, L.; ZHAO, A. Sentiment analysis based requirement evolution prediction. *Future Internet*, v. 11, n. 2, 2019.

ZIMMERMANN, T.; NAGAPPAN, N.; GUO, P. J.; MURPHY, B. Characterizing and predicting which bugs get reopened. In: *34th Int. Conf. on Software Engineering (ICSE)*. Zurich, Switzerland: IEEE, 2012. p. 1074–1083.

Apêndice A

MRS2014

Repositórios extraídas a partir dos projetos listados no desafio de mineração (*Mining Challenge*) da conferência *The International Working Conference on Mining Software Repositories (MSR)* disponível em <https://2014.msrconf.org/challenge.php>

Tabela A.1: Tabela de Dados

Repositório	**Issues sem Reab.**	**Issues c/1 Reab**	**Issues c/ 2 ou mais Reab**
akka/akka	14030	321	11
ariya/phantomjs	3220	92	9
AutoMapper/AutoMapper	2750	81	4
bcit-ci/CodeIgniter	4379	109	7
beanstalkd/beanstalkd	463	10	0
bitcoin/bitcoin	15828	606	70
boto/boto	1629	34	1
cakephp/cakephp	11363	367	15
Compass/compass	1140	58	5
d3/d3	2562	39	2
diaspora/diaspora	5553	186	10
divio/django-cms	4399	113	4
django/django	9118	119	9
elastic/elasticsearch	49944	986	75
facebook/facebook-android-sdk	554	9	0
facebook/folly	840	28	2
facebook/hhvm	6891	255	14
foundation/foundation-sites	8936	257	10
FortAwesome/Font-Awesome	10481	304	12
gitlabhq/gitlabhq	2855	75	10
h5bp/html5-boilerplate	1370	51	6
harvesthq/chosen	2493	85	3
hbons/SparkleShare	1386	86	11
heartcombo/devise	3458	108	4

Tabela A.2: Tabela de Dados

Repositório	**Issues sem Reab.**	**Issues c/1 Reab**	**Issues c/ 2 ou mais Reab**
Homebrew/brew	4995	168	14
Homebrew/homebrew-core	24926	422	16
Homebrew/legacy-homebrew	34829	895	49
httpie/httpie	586	26	1
imathis/octopress	1193	36	2
impress/impress.js	565	20	0
JakeWharton/ActionBarSherlock	811	34	2
jazzband/django-debug-toolbar	880	27	1
jekyll/jekyll	6809	211	12
joyent/libuv	1368	44	3
jquery/jquery	3647	152	10
KentonWhite/ProjectTemplate	173	12	2
libgit2/libgit2	3972	61	3
libuv/libuv	2521	69	8
liuliu/ccv	119	4	0
mangos/MaNGOS	71	8	0
memcached/memcached	580	8	0
midgetspy/Sick-Beard	378	11	0
MiniProfiler/rack-mini-profiler	377	6	0
mobile-shell/mosh	672	40	3
mongodb/mongo	876	18	0
mono/mono	11719	254	28
moxiecode/plupload	1045	38	1
mrdoob/three.js	12872	371	18
NancyFx/Nancy	1613	65	4
nathanmarz/storm	182	3	0
netty/netty	8717	253	12
nodejs/http-parser	358	4	0
nodejs/node	30644	816	78
nodejs/node-v0.x-archive	8217	329	25
openframeworks/openFrameworks	3573	149	7
openzipkin/zipkin	2222	42	0
pallets/flask	2565	86	2
pockethub/PocketHub	747	20	1
psf/requests	4263	196	33
r-lib/devtools	1681	46	2
rails/rails	28471	1144	74
ravendb/ravendb	1020	32	0
reddit-archive/reddit	758	15	0
redis/redis	3571	75	2
restsharp/RestSharp	1060	36	1
rstudio/shiny	1732	49	1
SamSaffron/MiniProfiler	89	2	0
sbt/sbt	3652	160	15

Tabela A.3: Tabela de Dados

Repositório	**Issues sem Reab.**	**Issues c/1 Reab**	**Issues c/ 2 ou mais Reab**
scala/scala	7007	233	24
scalatra/scalatra	358	10	1
sebastianbergmann/phpunit	3302	81	1
ServiceStack/ServiceStack	261	5	
SignalR/SignalR	3534	130	7
symfony/symfony	30351	653	29
ThinkUpLLC/ThinkUp	941	18	2
thoughtbot/paperclip	2111	66	3
tornadoweb/tornado	1849	35	
TrinityCore/TrinityCore	17610	1047	109
twitter/finagle	735	20	2
twitter-archive/flockdb	25	1	
vmg/redcarpet	487	14	
xbmc/xbmc	13933	256	25
yihui/knitr	1259	39	1
zendframework/zendframework	5326	204	20

Apêndice

B

RESULTADOS DESCRITIVOS DA CATEGORIZAÇÃO DE REABERTURA DE ISSUES

As tabelas a seguir apresentam os valores descritivos (mínimo, máximo, mediana, média, Q1 e Q3) das diferentes categorias de issues com e sem reaberturas, referente ao tempo de duração e número de comentários entre a abertura e o fechamento apresentados nos gráficos do Capítulo 4.

Tabela B.1: Análise da Duração em Horas entre a abertura e o fechamento de issues sem reaberturas

Categoria	Min	Max	Média	Mediana	Desvio Padrão	Q1 (25%)	Q3 (75%)
Banco de dados	0.000833	90891.51	1462.786	21.52319	5033.201	1.65875	283.1338
Configuração	0.000278	81064.03	1534.060	49.88417	4865.066	4.873333	492.2476
Desempenho	0.000556	73320	1688.894	58.7025	5131.089	6.560833	563.0389
Funcional	0.001667	41090.62	691.5928	21.50806	2961.781	2.700972	190.2932
GUI	0.000278	80672.66	1878.795	42.87097	5829.181	3.780069	540.3305
Info	0.000833	57388.52	925.0218	20.98833	3687.223	2.470833	191.2681
Permissão/Obsoleto	0.000556	86854.58	1335.560	47.22083	4434.496	5.366528	386.4442
Redes	0.000278	70937.75	1834.150	63.41847	5469.548	6.128889	655.0094
Segurança	0.000556	65328.43	1730.185	41.64222	5358.944	2.961111	550.29
Testes	0.000556	88947.65	1121.915	46.55361	4024.293	5.323681	286.4263

Tabela B.2: Análise da Duração em Horas entre a abertura e o fechamento de issues com reaberturas

Tipo	Min	Max	Média	Mediana	Desvio Padrão	Q1 (25%)	Q3 (75%)
Banco de dados	0.00222	33625.34	1299.491	45.13347	3960.707	3.336181	439.7966
Configuração	0.00056	71619.59	1221.979	33.75278	4103.03	2.742778	455.0089
Desempenho	0.00361	43886.59	1326.62	47.65986	4477.153	3.597431	478.7094
Funcional	0.09722	2139.852	453.2405	40.69028	794.2859	17.43611	173.6397
GUI	0.00083	61171.43	1602.04	40.44667	4886.507	3.205556	610.4937
Info	0.01472	18581.34	871.6407	26.12944	2516.697	1.225417	339.1074
Permissão/Obsoleto	0.01	20945.48	1212.14	69.20056	2825.749	4.702431	747.204
Redes	0.00417	60550.82	1759.842	50.95056	5346.817	3.743889	793.9797
Segurança	0.00722	46067.23	1561.18	45.58986	4814.577	3.294722	653.6695
Testes	0.00222	44794.68	1240.31	63.70208	3689.255	5.151319	591.5561

Tabela B.3: Análise da Duração em Horas entre a Abertura e o Fechamento de Issues sem Reaberturas (Sem *Outliers*)

Categoria	Min	Max	Média	Mediana	Desvio Padrão	Q1 (25%)	Q3 (75%)
Banco de dados	0.000833	703.9406	66.01572	8.9875	128.6321	1.046111	60.66375
Configuração	0.000278	1223.248	130.5022	23.17139	236.6393	2.811597	130.5965
Desempenho	0.000556	1397.521	152.9046	27.04208	275.1042	3.838472	150.7915
Funcional	0.001667	471.2739	52.86241	14.50736	92.2673	2.134167	49.77618
GUI	0.000278	1345.085	132.3339	18.88153	255.7591	2.184097	119.8142
Info	0.000833	474.3342	51.54092	11.04958	89.74099	1.51375	53.34792
Permissão/Obsoleto	0.000556	957.5647	108.9938	22.96528	187.3278	3.144028	117.9586
Redes	0.000278	1627.322	172.3247	26.72417	318.6584	3.483681	163.6639
Segurança	0.000556	1370.975	136.1308	18.34083	262.7078	1.620208	119.4453
Testes	0.000556	707.9367	84.48788	22.76944	137.4107	3.232222	95.5525

Tabela B.4: Análise da Duração em Horas entre a Abertura e o Fechamento de Issues com Reaberturas (Sem *Outliers*)

Categoria	Min	Max	Média	Mediana	Desvio Padrão	Q1 (25%)	Q3 (75%)
Banco de dados	0.002222	1049.445	117.2409	18.37167	212.4852	2.070417	115.4933
Configuração	0.000556	1132.831	117.7525	16.91056	221.5423	1.564514	111.2874
Desempenho	0.003611	1183.775	138.4781	20.845	250.0069	2.230833	131.9811
Funcional	0.097222	173.6397	44.96506	22.41736	59.48118	4.573403	41.17403
GUI	0.000833	1520.375	149.0611	18.69042	297.9605	1.812361	119.6313
Info	0.014722	820.1219	82.13697	9.853333	156.9537	0.8755556	79.50472
Permissão/Obsoleto	0.010000	1859.754	208.0476	26.30583	389.8804	3.034306	189.9368
Redes	0.004167	1976.571	201.7984	22.16847	398.7725	1.763194	169.4433
Segurança	0.007222	1609.905	170.6095	18.67444	337.124	1.711944	120.7964
Testes	0.002222	1467.861	172.596	29.29347	304.3073	2.842083	177.0108

Tabela B.5: Número de comentários entre a abertura e o fechamento de issues sem reaberturas

Categoria	Min	1ºQu	Mediana	Média	3ºQu	Max
Banco de dados	2	2	2	3.628	4	53
Configuração	2	2	3	4.697	5	521
Desempenho	2	2	3	4.751	5	108
Funcional	2	2	3	3.679	4	37
GUI	2	2	3	4.231	5	193
Info	2	2	2	3.766	4	53
Permissão/Obsoleto	2	2	3	4.394	5	111
Redes	2	2	3	4.801	5	101
Segurança	2	2	3	4.333	5	143
Testes	2	2	3	4.419	5	97

Tabela B.6: Número de comentários entre a abertura e o fechamento de issues com reaberturas

Categoria	Min	1ºQu	Mediana	Média	3ºQu	Max
Banco de dados	2	2	3	4.737	5.25	37
Configuração	2	2	3	5.126	6	109
Desempenho	2	2	3	5.528	6	70
Funcional	2	2	2.5	3.3	3	11
GUI	2	2	3	4.852	5	132
Info	2	2	3	5.356	5	41
Permissão/Obsoleto	2	2	3	5.023	6	71
Redes	2	2	3	4.794	5	58
Segurança	2	2	3	4.671	5	60
Testes	2	2	3	5.348	6	71

Apêndice

C

RESULTADOS DESCRITIVOS DA CARACTERIZAÇÃO DE REABERTURA DE ISSUES

As tabelas a seguir apresentam os valores descritivos (mínimo, máximo, mediana, média, Q1 e Q3) das diferentes categorias de issues com e sem reaberturas, referentes às métricas de análise de sentimentos, complementando os gráficos apresentados no Capítulo 5

Tabela C.1: N_{PC} - issues sem reaberturas

Categoria	Min	Q1	Mediana	Média	Q3	Max
Banco de dados	-5	-1	-1	-1,119	-1	-1
Configuração	-5	-2	-1	-1,317	-1	-1
Desempenho	-5	-1	-1	-1,267	-1	-1
Funcional	-4	-1	-1	-1,076	-1	-1
GUI	-5	-1	-1	-1,255	-1	-1
Info	-5	-1	-1	-1,126	-1	-1
Permissões/Obsoleto	-5	-1	-1	-1,163	-1	-1
Redes	-5	-1	-1	-1,235	-1	-1
Segurança	-5	-1	-1	-1,232	-1	-1
Testes	-5	-1	-1	-1,265	-1	-1

Tabela C.2: N_{PC} - issues com reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max.
Banco de dados	-3	-1	-1	-1,194	-1	-1
Configuração	-5	-2	-1	-1,37	-1	-1
Desempenho	-4	-1	-1	-1,288	-1	-1
Funcional	-1	-1	-1	-1	-1	-1
GUI	-5	-1	-1	-1,32	-1	-1
Info	-4	-1	-1	-1,235	-1	-1
Permissões/Obsoleto	-4	-1	-1	-1,201	-1	-1
Redes	-5	-1	-1	-1,282	-1	-1
Segurança	-4	-1	-1	-1,339	-1	-1
Testes	-5	-1	-1	-1,276	-1	-1

Tabela C.3: P_{PC} - issues sem reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max.
Banco de dados	1	1	1	1,187	1	5
Configuração	1	1	1	1,359	2	5
Desempenho	1	1	1	1,287	1	5
Funcional	1	1	1	1,067	1	3
GUI	1	1	1	1,366	2	5
Info	1	1	1	1,179	1	5
Permissões/Obsoleto	1	1	1	1,181	1	5
Redes	1	1	1	1,347	1	5
Segurança	1	1	1	1,319	1	5
Testes	1	1	1	1,258	1	5

Tabela C.4: P_{PC} - issues com reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max.
Banco de dados	1	1	1	1,28	1	4
Configuração	1	1	1	1,458	2	5
Desempenho	1	1	1	1,35	1	5
Funcional	1	1	1	1	1	1
GUI	1	1	1	1,438	2	5
Info	1	1	1	1,255	1	4
Permissões/Obsoleto	1	1	1	1,229	1	4
Redes	1	1	1	1,434	2	4
Segurança	1	1	1	1,406	2	4
Testes	1	1	1	1,309	1	5

Tabela C.5: SP_{PC} de issues sem reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max.
Banco de dados	-4	0	0	0,06759	0	4
Configuração	-4	0	0	0,042	0	4
Desempenho	-4	0	0	0,02028	0	4
Funcional	-3	0	0	-0,009021	0	2
GUI	-4	0	0	0,1109	0	4
Info	-3	0	0	0,05307	0	4
Permissões/Obsoleto	-3	0	0	0,01763	0	3
Redes	-4	0	0	0,1115	0	4
Segurança	-4	0	0	0,08725	0	4
Testes	-4	0	0	-0,006947	0	4

Tabela C.6: SP_{PC} de issues com reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	-2	0	0	0,08602	0	3
Configuração	-4	0	0	0,08735	0	4
Desempenho	-3	0	0	0,0619	0	4
Funcional	0	0	0	0	0	0
GUI	-3	0	0	0,1181	0	4
Info	-3	0	0	0,02013	0	3
Permissões/Obsoleto	-3	0	0	0,02842	0	3
Redes	-4	0	0	0,1512	0	3
Segurança	-3	0	0	0,06745	0	3
Testes	-3	0	0	0,03305	0	4

Tabela C.7: N de issues sem reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	-5	-1	-1	-1,13	-1	-1
Configuração	-5	-1	-1	-1,158	-1	-1
Desempenho	-5	-1	-1	-1,151	-1	-1
Funcional	-5	-1	-1	-1,093	-1	-1
GUI	-5	-1	-1	-1,157	-1	-1
Info	-5	-1	-1	-1,135	-1	-1
Permissões/Obsoleto	-5	-1	-1	-1,126	-1	-1
Redes	-5	-1	-1	-1,158	-1	-1
Segurança	-5	-1	-1	-1,149	-1	-1
Testes	-5	-1	-1	-1,117	-1	-1

Tabela C.8: N de issues com reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	-3	-1	-1	-1,151	-1	-1
Configuração	-5	-1	-1	-1,24	-1	-1
Desempenho	-4	-1	-1	-1,228	-1	-1
Funcional	-1	-1	-1	-1	-1	-1
GUI	-5	-1	-1	-1,239	-1	-1
Info	-4	-1	-1	-1,154	-1	-1
Permissões/Obsoleto	-5	-1	-1	-1,185	-1	-1
Redes	-4	-1	-1	-1,2	-1	-1
Segurança	-4	-1	-1	-1,204	-1	-1
Testes	-4	-1	-1	-1,187	-1	-1

Tabela C.9: P de issues sem reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Configuração	1	1	1	1,503	2	5
Banco de dados	1	1	1	1,506	2	5
Funcional	1	1	1	1,379	2	4
GUI	1	1	1	1,548	2	5
Info	1	1	1	1,618	2	5
Redes	1	1	1	1,489	2	5
Desempenho	1	1	1	1,47	2	5
Permissões/Obsoleto	1	1	1	1,48	2	5
Segurança	1	1	1	1,518	2	5
Testes	1	1	1	1,532	2	5

Tabela C.10: P de issues com reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Configuração	1	1	1	1,436	2	5
Banco de dados	1	1	1	1,39	2	4
Funcional	1	1	1	1,1	1	2
GUI	1	1	1	1,434	2	5
Info	1	1	1	1,45	2	4
Redes	1	1	1	1,444	2	5
Desempenho	1	1	1	1,392	2	5
Permissões/Obsoleto	1	1	1	1,414	2	4
Segurança	1	1	1	1,422	2	5
Testes	1	1	1	1,359	2	5

Tabela C.11: SP de issues sem reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	-4	0	0	0,3757	1	4
Configuração	-4	0	0	0,3442	1	4
Desempenho	-4	0	0	0,3189	1	4
Funcional	-3	0	0	0,2861	0	3
GUI	-4	0	0	0,3912	1	4
Info	-3	0	0	0,4832	1	4
Permissões/Obsoleto	-4	0	0	0,3534	1	4
Redes	-4	0	0	0,33	1	4,00
Segurança	-4	0	0	0,3683	1	4
Testes	-4	0	0	0,4149	1	4

Tabela C.12: SP de issues com reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	-2	0	0	0,2392	1	3
Configuração	-4	0	0	0,1959	1	4
Desempenho	-3	0	0	0,1633	0	4
Funcional	0	0	0	0,1	0,0	1
GUI	-4	0	0	0,1956	1	4
Info	-2	0	0	0,2953	1	3
Permissões/Obsoleto	-4	0	0	0,2291	1	3
Redes	-3	0	0	0,2437	1	4
Segurança	-3	0	0	0,2175	1	4
Testes	-3	0	0	0,1719	0	4

Tabela C.13: NM de issues sem reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	-3,5	-1,125	-1	-1,125	-1	-1
Configuração	-4,333	-1,333	-1	-1,209	-1	-1
Desempenho	-4,5	-1,333	-1	-1,186	-1	-1
Funcional	-3,000	-1	-1	-1,088	-1	-1
GUI	-4,5	-1,333	-1	-1,192	-1	-1
Info	-3,5	-1,167	-1	-1,132	-1	-1
Permissões/Obsoleto	-3,5	-1,2	-1	-1,141	-1	-1
Redes	-4	-1,333	-1	-1,185	-1	-1
Segurança	-4	-1,286	-1	-1,178	-1	-1
Testes	-4	-1,250	-1	-1,158	-1	-1

Tabela C.14: NM de issues com reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	-2,222	-1,273	-1	-1,162	-1	-1
Configuração	-3,5	-1,5	-1,118	-1,256	-1	-1
Desempenho	-3,5	-1,400	-1	-1,233	-1	-1
Funcional	-1,333	-1	-1	-1,033	-1	-1
GUI	-3,5	-1,5	-1	-1,245	-1	-1
Info	-2,500	-1,250	-1	-1,176	-1	-1
Permissões/Obsoleto	-2,600	-1,333	-1	-1,192	-1	-1
Redes	-3,000	-1,393	-1	-1,225	-1	-1
Segurança	-3,000	-1,400	-1	-1,244	-1	-1
Testes	-3,5	-1,333	-1	-1,202	-1	-1

Tabela C.15: Pontuação Positiva Média (PM) para categorias de *issues* sem reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	1	1	1,250	1,334	1,5	4,500
Configuração	1	1	1,333	1,384	1,556	4,500
Desempenho	1	1	1,250	1,351	1,5	4
Funcional	1	1	1	1,221	1,400	2,500
GUI	1	1	1,333	1,421	1,667	4
Info	1	1	1,333	1,392	1,571	3,5
Permissões/Obsoleto	1	1	1,250	1,321	1,5	4
Redes	1	1	1,310	1,376	1,571	4
Segurança	1	1	1,333	1,386	1,571	4
Testes	1	1	1,250	1,351	1,5	4

Tabela C.16: Pontuação Positiva Média (PM) para categorias de *issues* com reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	1	1	1,25	1,32	1,50	3,50
Configuração	1	1	1,316	1,382	1,6	3,5
Desempenho	1	1	1,250	1,339	1,5	3
Funcional	1	1	1	1,052	1	1,333
GUI	1	1	1,333	1,392	1,6	3,5
Info	1	1	1,333	1,338	1,5	3
Permissões/Obsoleto	1	1	1,2	1,305	1,5	3
Redes	1	1	1,270	1,375	1,571	3
Segurança	1	1	1,250	1,367	1,6	3,333
Testes	1	1	1,2	1,297	1,5	4

Tabela C.17: DCNde issues sem reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Configuração	0	0	0	0,1129	0,20	1
Banco de dados	0	0	0	0,06878	0	1
Funcional	0	0	0	0,0451	0	1
GUI	0	0	0	0,1081	0,1667	1
Info	0	0	0	0,07205	0	1
Redes	0	0	0	0,0995	0,1667	1
Desempenho	0	0	0	0,09893	0,16667	1
Permissões/Obsoleto	0	0	0	0,07894	0,08333	1
Segurança	0	0	0	0,09131	0,12500	1
Testes	0	0	0	0,0838	0,1250	1

Tabela C.18: DCN de issues com reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Configuração	0	0	0	0,1332	0,25	1
Banco de dados	0	0	0	0,09721	0,16667	1
Funcional	0	0	0	0,03333	0	0,33333
GUI	0	0	0	0,1351	0,25	1
Info	0	0	0	0,09903	0,13793	0,66667
Redes	0	0	0	0,1168	0,20	1
Desempenho	0	0	0	0,1299	0,2258	1
Permissões/Obsoleto	0	0	0	0,1072	0,20	1
Segurança	0	0	0	0,1151	0,20	1
Testes	0,000	0,000	0,000	0,111	0,2	1,000

Tabela C.19: DCP de issues sem reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	0	0	0,1429	0,2234	0,5	1
Configuração	0	0	0,2	0,2319	0,4706	1
Desempenho	0	0	0,1667	0,2163	0,4	1
Funcional	0	0	0	0,1426	0,2727	1
GUI	0	0	0,2222	0,2573	0,5	1
Info	0	0	0,25	0,2516	0,5	1
Permissões/Obsoleto	0	0	0,1429	0,2089	0,3750	1
Redes	0	0	0,2	0,2292	0,4444	1
Segurança	0	0	0,2	0,2406	0,5	1
Testes	0	0	0,1667	0,2175	0,4286	1

Tabela C.20: DCP de issues com reaberturas

Categoria	Min.	Q1	Mediana	Média	Q3	Max
Banco de dados	0	0	0,1667	0,2033	0,3333	1
Configuração	0	0	0,1667	0,2176	0,3333	1
Desempenho	0	0	0,1250	0,2069	0,3333	1
Funcional	0	0	0	0,04242	0	0,33333
GUI	0	0	0,2	0,2326	0,4286	1
Info	0	0	0,1667	0,2160	0,4000	1
Permissões/Obsoleto	0	0	0,1250	0,1933	0,3333	1
Redes	0	0	0,1667	0,2211	0,3750	1
Segurança	0	0	0,1667	0,2184	0,3333	1
Testes	0	0	0,09091	0,17841	0,33333	1