

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

JOÃO VITOR SOUZA ROCHA

**INVESTIGANDO O POTENCIAL DO USO DE EXEMPLOS TRABALHADOS NO
AUXÍLIO À SOLUÇÃO DE ISSUES EM UM PROJETO NO GITHUB**

CAMPO MOURÃO

2025

JOÃO VITOR SOUZA ROCHA

**INVESTIGANDO O POTENCIAL DO USO DE EXEMPLOS TRABALHADOS NO
AUXÍLIO À SOLUÇÃO DE ISSUES EM UM PROJETO NO GITHUB**

**Investigating the potential of using worked examples to help resolve issues
in a GitHub project**

Trabalho de Conclusão de Curso de Graduação
apresentado (a) como requisito para obtenção
do título de Bacharel em Ciência da
Computação do Curso de Bacharelado em
Ciência da Computação da Universidade
Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Walter Takashi
Nakamura

Coorientador: Prof. Dr. Igor Scaliante Wiese

CAMPO MOURÃO

2025



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

JOÃO VITOR SOUZA ROCHA

**INVESTIGANDO O POTENCIAL DO USO DE EXEMPLOS TRABALHADOS NO
AUXÍLIO À SOLUÇÃO DE ISSUES EM UM PROJETO NO GITHUB**

Trabalho de Conclusão de Curso de Graduação
apresentado (a) como requisito para obtenção
do título de Bacharel em Ciência da
Computação do Curso de Bacharelado em
Ciência da Computação da Universidade
Tecnológica Federal do Paraná.

Data de aprovação: 18 / junho / 2025

Igor Scaliante Wiese
Doutorado
Universidade Tecnológica Federal do Paraná

Ivanilton Polato
Doutorado
Universidade Tecnológica Federal do Paraná

Marco Aurélio Graciotto Silva
Doutorado
Universidade Tecnológica Federal do Paraná

Walter Takashi Nakamura
Doutorado
Universidade Tecnológica Federal do Paraná

CAMPO MOURÃO
2025

AGRADECIMENTOS

Gostaria de iniciar meus agradecimentos expressando minha profunda gratidão ao meu orientador, Prof. Dr. Walter Takashi Nakamura, e ao meu coorientador Prof. Dr. Igor Scaliante Wiese. Agradeço por toda a orientação, conselhos, auxílio e dedicação ao longo deste projeto, fundamentais para meu desenvolvimento como aluno e profissional. Se hoje me interessar tão profundamente pela pesquisa e desejo seguir no mundo acadêmico, saibam que isso se deve, em grande parte, à inspiração e ao exemplo de vocês. Espero ter a oportunidade de continuar trabalhando com ambos nas próximas etapas da minha formação. Meu mais sincero e profundo obrigado!

É claro que não poderia deixar de fazer um agradecimento especial aos professores Prof. Dr. Ivanilton Polato, ao Prof. Dr. Marco Aurélio Graciotto Silva e ao Prof. Dr. Igor Fábio Steinmacher, pela orientação ao longo de todo este projeto. Suas observações, que sempre fizeram transparecer sua extensa e admirável bagagem de conhecimento e experiência como profissionais e pesquisadores, foram de valor inestimável para o pleno desenvolvimento deste trabalho.

Gostaria de demonstrar minha gratidão à minha família, que tanto me fortaleceu e incentivou ao longo desses anos. Agradeço aos meus pais, Claudinea Francisca de Souza Rocha e Jaime Ferreira Rocha, exemplos de força inigualável e que, mesmo longe, nunca falharam em demonstrar seu amor. À minha namorada, Maria Fernanda de Oliveira Ribeiro — meu amor, companheira para todas as horas e minha maior inspiração — não tenho palavras para expressar o quanto sincera é minha gratidão por seu amor infinito e apoio incomensurável, que me fortaleceram imensamente ao longo dessa jornada.

Por fim, mas de forma nenhuma menos importante, agradeço aos meus amigos Guilherme Shibuya, Arthur Akira, Jonathan Martins, Luiz Gustavo, Luiz Vargas, Felipe Matos e Marcos Paulo. Vocês estiveram presentes na maior parte desta trajetória, e com certeza foram responsáveis por alguns dos melhores momentos vividos ao longo dela. Apesar da distância, tenho certeza de que nossa amizade continuará se fortalecendo cada vez mais.

O presente trabalho foi realizado com apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Brasil. Processos nº 2022/04234-5 e nº 2021/0666-2.

RESUMO

O recente aumento de contribuições em projetos de software livre, especialmente em repositórios no GitHub, levanta questionamentos sobre as dificuldades enfrentadas por programadores iniciantes durante o processo de contribuição, que enfrentam insegurança devido à falta de familiaridade com o projeto, sua comunidade e os padrões de desenvolvimento adotados. Mesmo com tentativas de sanar as dificuldades enfrentadas por desenvolvedores inexperientes por meio de mentorias, boas práticas de comunidade e documentação, contudo ainda há uma série de desafios particulares a serem superados. Nesse contexto, a utilização de Worked Examples surge como alternativa para auxiliar desenvolvedores durante o processo de contribuição em projetos de software livre. O objetivo deste estudo é investigar como o uso de Worked Examples de diferentes similaridades, apresentados como issues fechadas e aprovadas, recomendados automaticamente por um bot a partir de uma issue inicial do GitHub, pode influenciar a resolução de issues de um projeto do projeto JabRef por alunos do curso de Bacharelado em Ciência da Computação, além dos benefícios e desafios da adoção dessa prática. Conjuntamente, por meio do monitoramento da visão dos participantes com um rastreador ocular, a pesquisa visa avaliar o esforço cognitivo realizado durante a resolução das issues por meio de métricas consolidadas na literatura, a fim de identificar critérios relevantes para a seleção de Exemplos Trabalhados de maior similaridade com a tarefa proposta. Os resultados indicam que Worked Examples de alta similaridade, especialmente quando apresentados em grupos de 3 recomendações, auxiliaram os participantes do estudo na formulação de uma solução para as issues propostas, servindo como base teórica para a compreensão do problema. Isso é evidenciado pela análise dos dados emitidos pelo rastreador ocular, que indicou uma grande quantidade de Áreas de Interesse nas recomendações que chamaram a atenção dos participantes e foram úteis de alguma forma na formulação de uma solução, podendo, porém, indicar uma elevada dispersão de atenção durante a análise das recomendações. Além disso, a complexidade das recomendações em conjunto com a similaridade entre o Worked Example e a issue proposta ainda se mostram desafios importantes a serem superados. Os resultados deste estudo fornecem diretrizes para aprimorar as recomendações de exemplos trabalhados, ajudando na formação de desenvolvedores e inspirando novas abordagens em comunidades de software livre.

Palavras-chave: exemplos trabalhados; software livre; github; rastreamento ocular.

ABSTRACT

The recent increase in contributions to open-source software projects, especially on GitHub repositories, raises questions about the difficulties faced by beginner programmers during the contribution process, which often leads to insecurity due to unfamiliarity with the project, its community, and adopted development standards. Even with attempts to alleviate these challenges for inexperienced developers through mentorship, good community and documentation practices, several specific hurdles remain. In this context, the use of Worked Examples emerges as a potential alternative to assist developers during the contribution process in open-source projects. The objective of this study is to investigate how the use of Worked Examples of different similarities, presented as closed and approved issues and automatically recommended by a bot based on an initial GitHub issue, can influence the resolution of issues from JabRef by Bachelor of Computer Science students. The study also aims to identify the benefits and challenges resulting from adopting this practice. Concurrently, by monitoring participants' eye gaze with an eye-tracker, the research seeks to evaluate the cognitive effort expended during issue resolution using established literature metrics and to identify relevant criteria for selecting more similar Worked Examples. The results indicate that high-similarity Worked Examples, especially when presented in groups of three recommendations, helped study participants formulate solutions for the proposed issues, serving as a theoretical basis for understanding the problem. This is evidenced by the analysis of eye-tracker data, which showed a significant number of Areas of Interest (AOIs) in the recommendations that captured participants' attention and were useful in some way for formulating a solution. However, this may also indicate a high dispersion of attention during the analysis of recommendations. Furthermore, the complexity of recommendations combined with the similarity between the Worked Example and the proposed issue remain important challenges to overcome. The findings of this study provide guidelines to enhance Worked Example recommendations, aiding in developer training and inspiring new approaches in open-source software communities.

Keywords: worked examples; open source software; github; eye-tracking.

LISTA DE FIGURAS

Figura 1 – Distribuição de Fixations.....	23
Figura 2 – Boxplot do número de Fixations e duração dos Fixations nas Conversations de acordo com a utilidade do WE.....	24
Figura 3 – Boxplot do número de Fixations e duração dos Fixations nas Conversations de acordo com a similaridade do WE.....	24
Figura 4 – Boxplot do número de Fixations e Fixation Duration no código do PR de acordo com a utilidade do WE	25
Figura 5 – Boxplot do número de Fixations e Fixation Duration no código do PR de acordo com a similaridade do WE.....	25
Figura 6 – Distribuição dos códigos gerados na análise qualitativa.....	29
Figura 7 – Distribuição de Worked Examples que Ajudaram (H) e Não Ajudaram (DNH) de acordo com a experiência do participante em Java.....	30
Figura 8 – Resultado do questionário TAM: (a) por item avaliado e (b) por dimensão avaliada.	31
Figura 9 – Resultado do questionário TAM de acordo com a percepção dos participantes sobre a utilidade do WE.....	31

LISTA DE TABELAS

Tabela 1 – Issues selecionadas e índice de similaridade das recomendações.....	15
Tabela 2 – Percepção dos participantes sobre cada recomendação.....	27

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Objetivos	11
1.2	Justificativa	11
2	TRABALHOS RELACIONADOS	12
3	METODOLOGIA	14
3.1	Seleção do projeto e busca por issues	14
3.2	Ferramentas Utilizadas	15
3.3	Seleção de Participantes	16
3.4	Perfil dos Participantes	16
3.5	Condução do Estudo	17
3.6	Análise Quantitativa	18
3.7	Análise Qualitativa	19
4	RESULTADOS	20
4.1	RQ1: Quais os benefícios do uso de worked examples no auxílio à construção da solução de issues?	20
4.2	RQ2: Quais os desafios do uso de worked examples no auxílio à construção da solução de issues?	21
4.3	RQ3: Quais métricas do EyeTracker evidenciam os benefícios e desafios identificados na construção da solução?	22
4.3.1	Fixation	22
4.3.2	Fixation in Conversation	23
4.3.3	Fixation in PR Code	24
4.3.4	Average Fixation Duration (AFD)	25
4.3.5	Re-review frequency	26
4.4	RQ4: Qual a percepção das pessoas na construção da solução de acordo com o grau de similaridade do Worked Example?	27
4.5	RQ5: Qual a percepção das pessoas na construção da solução de acordo com o grau de similaridade do Worked Example?	30
5	DISCUSSÕES	32
6	AMEAÇAS À VALIDADE	35
7	CONCLUSÃO	36
	REFERÊNCIAS	37

1 INTRODUÇÃO

A popularidade dos projetos de Software Livre em plataformas como o GitHub tem sido impulsionada pela quantidade significativa de novos projetos que surgem e a importância desse tipo de prática para manter desde projetos independentes a projetos específicos de grandes corporações com milhares de contribuidores pelo mundo. Com essa popularização, surgiram diversas questões sobre os desafios enfrentados pelos programadores, sejam novatos ou experientes, durante o processo de contribuição.

Pesquisas anteriores estudaram os efeitos da aplicação de boas práticas em projetos de software livre por parte dos responsáveis por manter os projetos, dando atenção à experiência de novatos nestas comunidades (Alderliesten; Zaidman, 2021; Bhasin; Murray; Storey, 2021; Tan; Zhou; Sun, 2020). Muitas vezes, novatos têm dificuldade em colaborar com esses projetos devido à falta de orientação adequada para realizar suas contribuições de forma correta e se integrem ao projeto (Steinmacher *et al.*, 2016), inexperiência na seleção de tarefas (Santos *et al.*, 2022), baixa autoconfiança (Bhasin; Murray; Storey, 2021) e falta de familiaridade com o código e práticas comuns da comunidade de um projeto (Tan *et al.*, 2023).

Diferentes abordagens têm sido utilizadas para mitigar este problema, como a elaboração de diretrizes para a padronização da descrição de issues, commits e Pull Requests (PR) nos projetos (Tan *et al.*, 2023) e a categorização de issues por meio de rótulos, o que facilita a busca de issues que se enquadram em um certo perfil ou área de interesse, principalmente para programadores novatos ou sem qualquer experiência (Alderliesten; Zaidman, 2021; Tan; Zhou; Sun, 2020). Uma das propostas encorajadas pela própria plataforma GitHub é a utilização de tags *good first issue* em tarefas apropriadas para novatos a fim de integrar novos colaboradores aos projetos.

Mais recentemente, Tan *et al.* (2023) também investigaram o uso dessa tag, mas em conjunto com a mentoria por desenvolvedores experientes e há muito tempo engajados no projeto. Os resultados revelaram que a orientação, suporte prático e clarificação de issues por meio da mentoria aumentaram em 71% a finalização bem-sucedida de tarefas por parte dos novatos, mas que essa abordagem também não é responsável por mantê-los engajados no projeto por muito mais tempo. Embora os resultados indiquem os benefícios da mentoria, existem desafios particulares como a falta de habilidades interpessoais dos mentores, dificuldades com o idioma (inglês), falta de reconhecimento e benefícios (monetários ou não) e dificuldades na recomendação de tarefas (Steinmacher *et al.*, 2021), o que pode resultar na indisponibilidade de mentores que auxiliem e orientem novatos na resolução de tarefas. Além disso, resolver certas issues sem um amplo conhecimento de programação ou familiaridade com o projeto pode ser desafiador.

Nesse contexto, a utilização de Worked Examples (WE) surge como uma alternativa para auxiliar desenvolvedores novatos na resolução de issues em aberto, trazendo casos de issues já resolvidas e que servem de base para o desenvolvimento de soluções eficientes de acordo com os padrões esperados pelo projeto, tornando o processo de aprendizado mais eficaz (Chen;

Kalyuga; Sweller, 2015; Chen; Retnowati; Kalyuga, 2019; Chen *et al.*, 2023). De acordo com Chen *et al.* (2023), o aprendizado realizado com exemplos trabalhados traz maior capacidade de fixação para o estudante, além de aumentar a tendência de generalização do conteúdo aprendido, permitindo a aplicação do conhecimento obtido em tarefas distintas dos exemplos trabalhados apresentados.

Estudos anteriores vem indicando a eficácia do uso de WE no processo de aprendizagem em diversas áreas e níveis de ensino, como geometria (Chen; Kalyuga; Sweller, 2015), álgebra (Booth *et al.*, 2013), matemática (Chen *et al.*, 2023) e na estimativa da magnitude numérica em crianças (Fitzsimmons *et al.*, 2023). Na área da computação, estudos envolvendo o uso de WEs para o ensino vem indicando resultados positivos, melhorando a eficiência na resolução de problemas de programação (Ericson; Margulieux; Rick, 2017; Zhi *et al.*, 2019), proporcionando uma melhor assimilação do conteúdo (Garces *et al.*, 2022) e menos problemas de compilação em novatos (Gaweda *et al.*, 2020). Contudo, um ponto pouco estudado é a aplicabilidade de WEs em situações práticas, por exemplo, no auxílio a desenvolvedores novatos na resolução de issues abertas em projetos de Software Livre. Considerando as dificuldades que novatos enfrentam para contribuir com projetos de Software Livre apresentadas anteriormente, é importante que alternativas sejam estudadas para possibilitar que desenvolvedores novatos compreendam a estrutura e os padrões esperados pelo projeto, bem como auxiliar na resolução de issues, de forma a facilitar o onboarding e a sua integração com a comunidade.

O objetivo deste estudo é observar os efeitos gerados pela recomendação automática de issues fechadas, que fazem o papel de WEs, por meio de um bot treinado com issues fechadas do JabRef, projeto de software livre com atividade frequente e documentação ampla. A pesquisa foi conduzida com 20 alunos do curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná, que relataram suas experiências e percepções durante a resolução de Issues propostas com o auxílio das recomendações indicadas pelo bot. Para compreender o processo de desenvolvimento da solução por parte dos participantes, foi utilizado um software de gravação de tela para monitorar a interação do participante, bem como um rastreador ocular para avaliar a carga cognitiva e identificar os pontos de interesse.

As principais contribuições deste estudo são:

- Evidências sobre a aplicabilidade de WEs (Worked Examples) para ajudar a resolver issues abertas no GitHub;
- Evidências sobre a relação entre o grau de similaridade do WE recomendado e sua utilidade na resolução da issue aberta;
- Um conjunto de benefícios e desafios no uso de WEs como auxílio para a formulação de soluções de issues;
- Um conjunto de métricas de eye-tracking destacando os benefícios e desafios do uso de WEs;
- Perspectivas futuras de pesquisa sobre o uso de WEs em projetos de Software Livre.

1.1 Objetivos

O estudo tem como objetivo responder à seguinte questão de pesquisa: “Como o uso de exemplos trabalhados pode auxiliar desenvolvedores novatos em direção à resolução de uma issue em projetos de software livre?”. Para responder a essa questão de pesquisa, cinco subquestões foram definidas:

- RQ1: Quais os benefícios do uso de worked examples no auxílio à construção da solução de issues?
- RQ2: Quais os desafios do uso de worked examples no auxílio à construção da solução de issues?
- RQ3: Quais métricas do EyeTracker evidenciam os benefícios e desafios identificados na construção da solução?
- RQ4: Qual a percepção das pessoas na construção da solução de acordo com o grau de similaridade do Worked Example?
- RQ5: Qual a percepção sobre a utilidade, facilidade de uso e intenção de uso dos participantes em relação a uma ferramenta de sugestão de Worked Examples?

1.2 Justificativa

A proposta deste estudo é observar os efeitos da recomendação automática de issues fechadas, atuando como Worked Examples, na resolução de issues em aberto no GitHub por alunos do Bacharelado em Ciência da Computação a partir do terceiro período, utilizando o projeto de software livre JabRef como base para a experimentação. O estudo empregou um bot treinado para recomendar issues fechadas similares às issues abertas, visando avaliar como a exposição a esses exemplos trabalhados influencia a capacidade dos participantes em resolver as issues abertas propostas de maneira eficiente. Além disso, foram captadas métricas de rastreamento ocular para analisar os benefícios e desafios dessa abordagem, com o objetivo de compreender melhor o impacto dos WE no processo de aprendizagem e na integração de novos contribuidores em comunidades de software livre.

2 TRABALHOS RELACIONADOS

O termo Worked Example (WE) refere-se à demonstração detalhada de como resolver um problema, incluindo a descrição do problema e todas as etapas envolvidas na sua solução (Ericson; Margulieux; Rick, 2017). O uso de WEs é uma estratégia instrucional efetiva que auxilia novatos a resolverem problemas complexos (Zhi *et al.*, 2019), sendo utilizada há anos em diversas áreas como geometria (Chen; Kalyuga; Sweller, 2015), álgebra (Booth *et al.*, 2013) e matemática (Chen *et al.*, 2023). Na área da computação, o uso de WEs é comum nas atividades de programação, sendo frequentemente utilizados para auxiliar novatos a progredirem em suas atividades por meio do uso, adaptação e expansão de códigos oriundos de projetos de programação semelhantes elaborados e documentados (Hüsing *et al.*, 2024). Nesse contexto, diferentes formas de utilização de WEs têm sido relatadas na literatura ao longo dos últimos anos.

Zhi *et al.* (2019) conduziram um estudo com 17 estudantes do ensino médio para avaliar o impacto dos WEs usando um sistema denominado Peer Code Helper (PCH) na programação. Os autores utilizaram três problemas de programação progressivamente difíceis e dividiram os alunos em dois grupos em cross-design, cada um recebendo WEs sobre problemas diferentes. Os resultados indicam que os WEs podem melhorar a eficiência da aprendizagem e poupar tempo nas tarefas iniciais. Os alunos que usaram WEs também completaram mais tarefas num período de tempo fixo, embora a diferença não tenha sido significativa.

Expandindo a análise para contextos mais específicos, Hüsing *et al.* (2024) recentemente investigaram o uso de WEs por novatos na prática de programação epistêmica, que consiste no uso de programação para entender fenômenos e contextos do mundo real. Os participantes foram solicitados a analisarem dados meteorológicos de três locais diferentes e informar sobre qual local prefeririam passar as férias de verão. Para ajudá-los, os pesquisadores forneceram um WE contendo explicações, trechos de código e interpretações relacionadas à leitura e preparação de dados, visualização de dados, filtragem e comparação de dados. Como resultado, os WEs ajudaram os participantes a encontrarem soluções corretas para os problemas propostos mais facilmente e, conseqüentemente, com menos esforço mental do que se o fizessem sem qualquer auxílio ou exemplo a ser usado como inspiração.

Embora investiguem aspectos distintos — Zhi *et al.* (2019) focaram no efeito de Worked Examples na resolução de tarefas de código, enquanto Hüsing *et al.* (2024) analisaram seu impacto em cenários de codificação do mundo real — ambos os estudos visam avaliar os efeitos dos Worked Examples na aprendizagem de programação. Seus resultados sugerem, coletivamente, que a adoção de Worked Examples em tarefas de código pode melhorar o desempenho, mesmo em diferentes contextos educacionais.

Morrison, Margulieux e Guzdial (2015) avaliaram a eficácia da rotulagem de sub-objetivos em materiais instrucionais para o ensino de programação introdutória, especificamente através do uso de WEs. Os pesquisadores dividiram os participantes em três grupos de acordo com a forma de rotulagem de sub-objetivos, cada grupo utilizando três WEs e problemas práti-

cos, intercalados e variados em contexto. Os resultados revelaram que a transferência isomórfica (problemas idênticos) levou a um melhor desempenho em comparação com a transferência contextual (diferentes contextos), enfatizando a importância de manter pontos em comum entre WEs e problemas práticos para reduzir a carga cognitiva e melhorar os resultados de aprendizagem.

Gaweda *et al.* (2020) avaliaram a eficácia de exercícios de digitação, apresentados como WEs, na melhoria do desempenho dos alunos nos cursos de Ciência da Computação. Os autores implementaram uma plataforma chamada TYPOS, onde os alunos realizavam exercícios de codificação que exigiam que eles redigitassem códigos de exemplo exibidos como imagens, permitindo um aprendizado ativo e feedback imediato sobre erros de sintaxe. Os resultados indicaram que os alunos que se envolveram ativamente com os exercícios de digitação demonstraram maiores ganhos de aprendizagem, melhor desempenho e menor quantidade de erros de sintaxe.

Os estudos apresentados revelam o potencial dos WE no auxílio à aprendizagem de programação e resolução de problemas, contudo, os experimentos conduzidos e problemas de código desenvolvidos foram idealizados em contextos exclusivamente acadêmicos. Assim, a aplicabilidade dos WE em situações reais, como no auxílio a desenvolvedores novatos na resolução de issues abertas em projetos de Software Livre, ainda é uma questão em aberto.

Para este fim, a adoção de um rastreador ocular para mensurar a carga cognitiva dos participantes durante os experimentos mostra-se uma técnica útil. Por meio desta técnica, é possível capturar padrões de carga cognitiva visual em tarefas complexas, como observado no estudo de Cheng *et al.* (2022) que utilizaram rastreadores oculares para analisar grupos de alunos de ensino superior de Ciência da Computação durante a identificação de bugs em programfas, adotando métricas como a duração de Fixations por linha de código e a Frequência de Re-revisão em Áreas de Interesse. Conjuntamente, foram adotadas as boas práticas metodológicas estabelecidas por Sharafi *et al.* (2020), que padronizam as definições de Fixation e Áreas de Interesse, fundamentais para a plena condução deste estudo.

3 METODOLOGIA

A ênfase deste capítulo está em reportar o que e como será feito para alcançar o objetivo do trabalho. Este capítulo pode ser subdividido, inicialmente, em duas seções, sendo uma para os materiais e outra para os métodos.

3.1 Seleção do projeto e busca por issues

O repositório do JabRef foi selecionado para conduzir a busca por issues, um projeto de software livre (open-source software - OSS) para gerenciamento de referências. Este repositório foi escolhido devido à sua documentação robusta e comunidade ativa, evidenciados por métricas recentes do GitHub. O projeto possui mais de 3,9 mil estrelas e 2,8 mil *forks*, refletindo sua popularidade e relevância no domínio de gerenciamento de referências bibliográficas. Adicionalmente, exibe um alto volume de pull requests, demonstrando contribuições ativas e colaboração eficiente. Com mais de 150 commits mensais e 4.083 issues fechadas, o projeto mostra manutenção consistente e dinamismo. Além disso, sua alta disponibilidade de issues próprias para iniciantes — como correções de pequenos bugs e melhorias simples de funcionalidade — tornou-o particularmente adequado para conduzir o estudo.

A seleção das issues foi feita por meio de filtragem pela tag “good first issue”, utilizada para sinalizar issues indicadas para novatos e programadores inexperientes, as quais costumam apresentar problemas de complexidade moderada e não exigem elevado conhecimento de código ou da linguagem de programação empregada no projeto. Após a filtragem foram selecionadas as 20 issues mais recentes e fechadas, de forma que as issues escolhidas contivessem uma solução aceita disponível e pudesse ser comparada com a solução proposta pelos participantes.

As 20 issues selecionadas foram analisadas por uma ferramenta desenvolvida pelos autores para recomendar issues similares dentro do mesmo projeto no GitHub. Esta ferramenta consiste em um bot que extrai o título e o conteúdo da issue e verifica a similaridade com as demais issues fechadas presentes no repositório por meio do cálculo da distância de cosseno. As issues mais similares são, então, apresentadas na issue original em forma de comentário contendo o título da issue e o grau de similaridade identificado. Devido ao tempo e à quantidade limitada de participantes disponíveis para a condução do estudo, foram selecionadas seis issues fechadas analisadas com a ferramenta, sendo três delas associadas a recomendações de issues com baixa similaridade (abaixo de 50%) e três com alta similaridade (acima de 50%).

Para que o bot pudesse ser executado para analisar issues sem afetar o repositório original do JabRef, criou-se um repositório próprio para o estudo¹. As 6 issues selecionadas foram importadas por meio da extensão Kamino para o Google Chrome². Essa extensão clona issues selecionadas ou repositórios inteiros, mantendo a referência à issue original. A Tabela 1 lista

¹ <https://github.com/Jaovitosr/Chatbot>

² <https://chromewebstore.google.com/detail/kamino/ffdebockfdjileaojbbccofhgncmioaf>

Tabela 1 – Issues selecionadas e índice de similaridade das recomendações.

ID	Título	Similaridade		
		R1	R2	R3
Issues com alta similaridade				
15	Save/save as: file type shows BIBTEX_DB https://github.com/Jaovitosr/Chatbot/issues/15	79,30	78,23	70,97
12	JabRef error if keyword group has no name https://github.com/Jaovitosr/Chatbot/issues/12	69,07	68,07	67,83
18	"Remove group" options are presented for clicked group, but actions are taken on all selected groups https://github.com/Jaovitosr/Chatbot/issues/18	69,76	67,98	56,26
Issues com baixa similaridade				
14	Double-click on preview style should add it to "Selected" https://github.com/Jaovitosr/Chatbot/issues/14	34,07	33,88	33,46
9	Editing preference of external application should offer existing value for editing https://github.com/Jaovitosr/Chatbot/issues/9	30,77	28,21	25,77
7	Unprotect selection https://github.com/Jaovitosr/Chatbot/issues/7	28,00	23,14	21,84

Fonte: Autoria própria.

as issues selecionadas para os participantes resolverem durante o estudo, juntamente com o respectivo índice de similaridade determinado pelo bot para cada uma das três recomendações.

3.2 Ferramentas Utilizadas

Para monitorar as respostas cognitivas dos participantes e compreender o rationale por trás da elaboração da solução, adotou-se a técnica de Eye Tracking. Para isso, foi utilizado um Eye Tracker Tobii Pro Fusion para identificar o local onde o participante observa na tela em determinado instante, elemento essencial para identificar os pontos de foco e compreender melhor o processo de elaboração da solução.

Para capturar a visão do usuário no GitHub ao longo do estudo utilizou-se o software iTrace ³(Guarnera *et al.*, 2018), projeto de software livre de uma ferramenta compatível com diversos modelos de Eye-Trackers e que torna possível a captação da visão do usuário em diversas plataformas. A ferramenta foi desenvolvida com foco em auxiliar estudos e pesquisas focadas em Engenharia de Software e Eye Tracking, oferecendo suporte a diversas softwares de uso comum na área da computação, como IDEs (Eclipse e Visual Studio Code) e plataformas como GitHub e StackOverflow por meio de plugins. A calibração do software foi realizada individualmente ao início de cada entrevista, e os participantes se mantiveram a uma distância de 50cm da tela.

Como o estudo envolve interação em uma página Web (repositório do GitHub) por meio de um navegador, utilizou-se neste estudo o plugin iTrace-Chrome ⁴(Guarnera *et al.*, 2018), desenvolvido para ser utilizado no navegador Google Chrome. Vale ressaltar que algumas das funcionalidades deste plugin estavam incompletas (como por exemplo, o retorno da linha de código visualizada pelo usuário). Portanto, realizou-se a implementação dessas funcionalidades a

³ <https://www.i-trace.org/>

⁴ <https://github.com/iTrace-Dev/iTrace-Chrome>

fim de que a ferramenta retornasse os dados que não estavam sendo registrados originalmente. A implementação está disponível em um repositório no GitHub, próprio para o projeto ⁵.

Além destas ferramentas, utilizou-se o software aTube Catcher ⁶ para gravar a tela e a voz do participante durante o estudo para análise posterior.

3.3 Seleção de Participantes

Para conduzir o estudo, foram selecionados 20 participantes, todos estudantes de graduação do terceiro período do Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná (UTFPR), campus Campo Mourão. A delimitação quanto ao período deveu-se à necessidade de uma base teórica e prática suficiente para compreender e contribuir com os temas abordados pelo estudo. Além disso, a limitação de selecionar alunos da mesma universidade deve-se à disponibilidade de apenas um Eye Tracker para medir a carga cognitiva dos participantes, o que tornou inviável transportar o equipamento para outras universidades próximas. Adicionalmente, essa abordagem garantiu maior controle da base teórica dos participantes sobre os temas tratados pelas issues. Os participantes foram selecionados por conveniência, considerando sua disponibilidade e interesse em participar do estudo.

3.4 Perfil dos Participantes

O perfil dos participantes foi definido por dois principais critérios: seu nível de experiência com a linguagem de programação Java e com contribuições em projetos de software livre, ambos informados por eles por meio de um formulário de caracterização preenchido antes do início do experimento. Cada participante informou seu nível de experiência com Java numa escala de 0 a 3, sendo:

- 0: Nunca programei em Java
- 1: Tenho noções básicas sobre Java
- 2: Utilizei Java em algum projeto ou curso
- 3: Trabalhei com Java na indústria

Dentre os 20 participantes, 5 indicaram que nunca programaram em Java (Nível 0), 7 afirmaram ter noções básicas (Nível 1), 4 relataram ter utilizado Java em algum projeto ou curso (Nível 2) e 4 já trabalharam com Java na indústria (Nível 3). Já sobre a experiência com contribuições de software livre, 14 participantes afirmaram já terem feito contribuições em projetos de software livre, enquanto 6 nunca o fizeram.

⁵ <https://github.com/Jaovitosr/iTrace-Chrome.git>

⁶ <https://www.atube.me/>

3.5 Condução do Estudo

O estudo foi conduzido de forma individual em uma sala reservada onde estavam apenas um dos autores e o participante. Primeiramente, os participantes foram informados sobre o objetivo do estudo. Em seguida, os participantes preencheram o Termo de Consentimento Livre e Esclarecido (TCLE) informando sobre a voluntariedade da participação, a forma de participação, a possibilidade de abandonar o estudo a qualquer momento sem ônus e a forma como os dados coletados serão tratados e utilizados. Os participantes foram informados de que: (i) todos os dados coletados serão utilizados apenas para fins de pesquisa, não sendo fornecidos para terceiros; (ii) as respostas serão apresentadas de forma agregada para fortalecer ainda mais o anonimato, evitando que seja possível vincular comportamentos/respostas a indivíduos; (iii) qualquer eventual citação apresentada em publicações será feita em total anonimato, sem associação com o nome dos envolvidos. Após preencher o TCLE, os participantes preencheram um questionário de caracterização contendo questões para identificar o nível de conhecimento sobre Java (linguagem utilizada no JabRef) e se o participante já contribuiu alguma vez em projetos de SL. Após preencher o questionário, o estudo foi iniciado.

Cada participante foi atribuído a duas dentre seis issues, de forma aleatória, para serem analisadas. Para uma das issues, o participante era apresentado a somente uma recomendação de issue similar e para a outra três. Uma das issues possuía recomendações de issues com alta similaridade e a outra recomendações de baixa similaridade. Sendo assim, cada participante foi atribuído a uma das condições: (1) Uma issue com 3 recomendações de alta similaridade e uma issue com 1 recomendação de baixa similaridade; (2) Uma issue com 3 recomendações de baixa similaridade e uma issue com 1 recomendação de alta similaridade. O objetivo foi garantir que todos os participantes trabalhassem com issues de alta e baixa similaridade, intercalando entre três de baixa similaridade e uma de alta, e vice-versa.

O participante recebeu uma issue por vez, analisando sua descrição e acessando as recomendações feitas pelo bot, conforme atribuído pelo procedimento descrito anteriormente. Cada participante foi instruído a tomar o tempo necessário para analisar as issues e recomendações do bot, então indicar — com base nas informações observadas e exemplos de código fornecidos — se os Worked Examples o ajudaram a formular uma solução possível para a issue proposta, e explicar como.

Após formularem a solução, lhes foi concedido acesso à solução real da issue proposta, podendo analisá-la pelo tempo necessário. Em seguida, foi solicitado que comparassem com sua própria solução formulada na etapa anterior da entrevista, fornecendo feedback sobre se os Worked Examples auxiliaram ou prejudicaram seu processo de formulação da solução. Os procedimentos supracitados foram repetidos para a segunda issue.

Todas as interações dos participantes, voz e movimentos oculares foram registrados usando as ferramentas descritas na Seção 3.2. Após cada issue ser analisada, o rastreamento ocular era interrompido para que o arquivo de log pudesse ser gerado e verificado, garantindo que eventuais problemas evitados e os dados da próxima entrevista não fossem comprometidas.

Após concluir a análise das duas issues, o participante preencheu o questionário TAM (Technology Acceptance Model) (Davis; Bagozzi; Warshaw, 1989), para avaliar sua percepção sobre utilidade, facilidade de uso e intenção de uso em relação à abordagem proposta.

3.6 Análise Quantitativa

Cada AOI (*Área de Interesse*) capturada pelo plugin em conjunto com o Eye Tracker foi registrada em um arquivo de log, com cada instância de AOI associada a um timestamp e listada sequencialmente. As AOIs representam estímulos visuais fixados pelos participantes durante o experimento, servindo como base para determinar quais elementos atraíram mais atenção. Os arquivos foram analisados para extrair métricas padrão de eye-tracking (Cheng *et al.*, 2022; Sharafi *et al.*, 2020).

- Fixation Count: quantidade de vezes em que uma determinada área de interesse (AOI) foi focada. Grandes quantidades de fixation em uma mesma AOI indicam áreas relevantes e de interesse do participante;
- Fixation Duration: duração do foco do participante em determinada área de interesse (AOI). Longos tempos de fixação indicam áreas relevantes e de interesse do participante;
- Frequência de re-revisão: quantidade de vezes que o participante retorna para um ponto de foco anterior após um determinado período de tempo, indicando uma AOI relevante em potencial;
- Média de Duração da Fixation (AFD): calculada pela soma de todas as fixations em uma determinada AOI dividida pela quantidade de fixations dessa AOI.

Considerando que o objetivo é identificar elementos do código (classes, variáveis, métodos, funções, estruturas de controle e comentários) e das issues (textos, listas, imagens e comentários) que possam chamar a atenção dos participantes durante a análise para formular a solução, estes elementos foram extraídos do código-fonte HTML do GitHub, e cada um foi definido como uma AOI. Para diferenciar atenção consciente de uma relance, foi adotado um limiar de 1 segundo para determinar uma fixation, conforme trabalhos anteriores de revisão de código (Cheng *et al.*, 2015; Cheng *et al.*, 2022). Com essas definições, foi desenvolvido um algoritmo para automatizar o processo de extração ⁷, que itera pelas entradas dos arquivos e retorna dados quantitativos para análise posterior.

⁷ <https://github.com/Jaovitosr/Fixation-Extractor.git>

3.7 Análise Qualitativa

As respostas dos participantes foram transcritas em um documento de texto e analisadas por um dos autores do artigo utilizando os procedimentos de codificação aberta do Grounded Theory (Corbin e Strauss, 2014). As respostas foram analisadas de forma a identificar os benefícios e desafios (QP1 e QP2) no uso de Worked Examples para auxiliar na elaboração da solução de issues. Em seguida, códigos que representam a ideia ou tópico central do trecho identificado foram criados e atribuídos. Por fim, os códigos foram agrupados de acordo com a similaridade, de forma a abstrair conceitos que envolvam diferentes códigos. Todo o processo de codificação foi realizado por um dos autores em forma de um CodeBook em uma planilha do Google Sheets e revisado por outros dois pesquisadores, especialistas em análise qualitativa, disponível abertamente no repositório Figshare, em <https://doi.org/10.6084/m9.figshare.27936096>.

4 RESULTADOS

Os resultados foram divididos em cinco subseções, as quais apresentam os resultados de acordo com as questões de pesquisa apresentadas.

4.1 RQ1: Quais os benefícios do uso de worked examples no auxílio à construção da solução de issues?

A análise resultou em seis códigos que destacam os benefícios do uso de Worked Examples para auxiliar na construção de uma solução para issues, e que são descritos a seguir.

O primeiro benefício observado é que Worked Examples podem servir como ponto de partida para formular a solução de uma issue. Por meio de recomendações de issues similares, os participantes puderam analisar estruturas de diretórios do projeto e elementos de código úteis como funções e métodos, fornecendo referências para iniciar a formulação da solução. O participante P13, por exemplo, relatou interesse em analisar os três Worked Examples recomendados para buscar inspiração: *Eu acho que eu ia primeiro tentar ver a solução do primeiro link, que parece ser mais simples, tem menos código pelo menos... mas daí se não desse certo eu tentaria dos outros dois, que inclusive foram solucionados pelo mesmo pull request*. Os participantes P9 e P15 demonstraram a utilidade dos WE na identificação de comportamentos de funções do projeto, enquanto o participante P7 destacou auxílio na compreensão e navegação de diretórios do projeto: *Eu sei como a organização de arquivos deles é confusa, então isso, em particular, seria muito útil. Não o código, mas a especificação do caminho do arquivo para me localizar*. Isso evidencia a capacidade dos WE de reduzir o tempo de orientação inicial em projetos complexos ao funcionarem como mapas do projeto por meio de trechos de código fornecidos nos WE recomendados — particularmente valioso para contribuidores novatos.

Além do benefício anterior, WE também permitiram identificar trechos de código úteis como classes, métodos ou outros elementos que auxiliam diretamente na resolução da issue proposta. O participante P10, por exemplo, relatou: *existem algumas funções que em algumas partes do problema que resolveram... do Pull Request que resolveu o problema, existem algumas funções similares que são usadas, então, tipo, a recomendação seria útil*. Conjuntamente, é percebida a capacidade de Worked Examples de complementar a descrição da issue aberta e compreender melhor o cenário em que o participante estava trabalhando, especialmente se confusa ou incompleta. Conforme destacado pelo participante P20 destacou a possibilidade de obter informações que reduzem o tempo necessário para compreender o problema — que seria maior sem a recomendação — como funções e classes usadas em diferentes partes do projeto, padrões de código, etc.

Portanto, nota-se que ao apresentar descrições que apontam problemas similares à open issue, os WE permitem que desenvolvedores identifiquem padrões de solução reutilizáveis, esclareçam o escopo do problema e compreendam comportamentos do sistema. Isso foi evidenciado também pelo reconhecimento de relações temáticas pelo participante P18 : *Eu*

consegui acessar todas as três sugestões... Todas as sugestões estavam relacionadas a grupos de alguma forma". Adicionalmente, o P19 relatou melhoria na compreensão de requisitos por meio de descrições de issues, e o P17 identificou paralelos funcionais entre os WE e a issue proposta. Coletivamente, esses casos demonstram a capacidade dos WE como bases contextuais que fornecem elementos de código reutilizáveis e insights específicos de domínio para projetos complexos.

Complementando os benefícios supracitados, observamos outra vantagem dos Worked Examples: sua capacidade de recomendar Pull Requests que resolveram problemas relacionados ao proposto. Como relatou o participante P7: *"...eu tentaria usar este PR aqui como base, que ele usa para armazenar os tipos. Eu provavelmente procuraria por 'store bibentry types' ou algo que envolva 'store' dentro do código para tentar resolver este problema"*. Por meio destas recomendações, os participantes recebem soluções completas e aprovadas para issues semelhantes à proposta durante a condução do estudo, permitindo que compreendam melhor o contexto técnico com uma solução real.

4.2 RQ2: Quais os desafios do uso de worked examples no auxílio à construção da solução de issues?

Em relação aos desafios, sete códigos foram identificados representando dificuldades enfrentadas pelos participantes durante o estudo.

O principal desafio observado foi a relação entre a issue proposta e as recomendações, ocorrendo tanto em casos de alta quanto de baixa similaridade. Alguns participantes relataram que os WE apresentavam uma ou mais issues distintas da issue a ser resolvida. O participante P4 afirmou: *"realmente é bem diferente do que ele me mostrou, até porque ele tava mostrando mais lógica do que UI"*. Além disso, participantes relataram Issues com PR que apresentavam soluções desalinhadas da issue proposta, desviando significativamente da solução aceita. Conforme relatado pelo P9 ao resolver uma issue sobre mudanças funcionais em botões: embora a issue proposta e a recomendação abordassem funcionalidades similares, eram não relacionadas na implementação e contexto. Relatos similares surgiram de outros participantes, indicando que, apesar dos WE apresentarem soluções com tópicos similares, em múltiplos casos introduziam Issues e PRs aparentemente relevantes mas funcionalmente distintos, criando uma falsa sensação de proximidade e levando participantes a interpretar erroneamente o problema e/ou mirar em soluções incorretas. Isso foi evidenciado pelo P14, que abandonou a solução correta devido à similaridade percebida com WE; e pelos P5 e P12, que inicialmente notaram similaridades (padrões de código e reuso de classes, respectivamente) mas concluíram não haver alinhamento real após revisão da solução.

Outros desafios observados relacionam-se ao nível de complexidade das recomendações. Alguns participantes relataram que as recomendações careciam de detalhamento suficiente ou eram excessivamente genéricas, forçando-os a examinar PRs de outras issues resolvidas ou buscar fontes externas para formular soluções eficientes. Por exemplo, o P11

declarou antes da revisão da solução: *eu não percebi (nada que possa ajudar na solução da issue), no caso, vendo assim pelo menos. Eu acredito que eu precisaria buscar entender mais ainda.*”, acrescentando que usar as informações do WE exigiria consultar outras fontes. Da mesma forma, o participante P6 não reconheceu similaridade útil entre as soluções das issues recomendadas e proposta.

Por outro lado, algumas recomendações mostraram-se excessivamente complexas, criando falsas impressões sobre a dificuldade da solução. Isso levou a investimentos desproporcionais de tempo e conseqüentemente impediu resolução efetiva. O P8 observou: *“é, a solução é mais fácil que eu esperava (...) A recomendação deu uma assustada, porque mostrou umas coisas muito complexas e no final era facinho, mas eu também não entendo muito do JabRef e do Java”*.

Finalmente, surgiram desafios relativos a arquivos e diretórios recomendados. Alguns WE apresentaram PRs modificando arquivos irrelevantes ao problema alvo, causando confusão entre diretórios. O P16 exemplificou isso: *“O diretório já vai pra um lado diferente meio no início aqui, e a maneira como foi resolvido é bem diferente, até. Ok que os problemas eram diferentes também, né... nesse aqui é JabRef/GUI, o outro ele era JabRef/Bibtex já, ele não passou por esse GUI, mas eu acho que ta meio perto”*.

4.3 RQ3: Quais métricas do EyeTracker evidenciam os benefícios e desafios identificados na construção da solução?

Os resultados desta seção foram divididos de acordo com as métricas apresentadas na Seção 3.6. Os dados foram analisados de acordo com duas páginas de interesse no GitHub presentes nos Pull Requests de cada issue: Conversation (que contém a descrição do Pull Request, imagens, comentários e trechos de código citados pelos usuários) e Files Changed (que contém o código relacionado ao Pull Request). A análise também foi realizada sob duas perspectivas: grau de similaridade (alta ou baixa) e de auxílio na elaboração da solução (ajudou ou não ajudou). Considerando que o teste de Shapiro-Wilk indicou que os dados não são normalmente distribuídos (valor-p < 0,05), aplicou-se o teste não paramétrico de Mann-Whitney. É importante salientar que houveram problemas na captura dos dados do rastreamento ocular do participante P1 para as duas tarefas e do participante P9 para a segunda tarefa devido a problemas no plugin iTrace-Chrome durante o estudo. Sendo assim, os dados destes dois participantes não foram utilizados nesta análise.

4.3.1 Fixation

A Figura 1 apresenta um histograma que apresenta a distribuição dos fixations de acordo com o intervalo de tempo. Foram identificados 274 fixations no total, sendo a grande maioria (218) contabilizada uma ou duas vezes. Dentre os 56 fixations com 3 ou mais contabilizações,

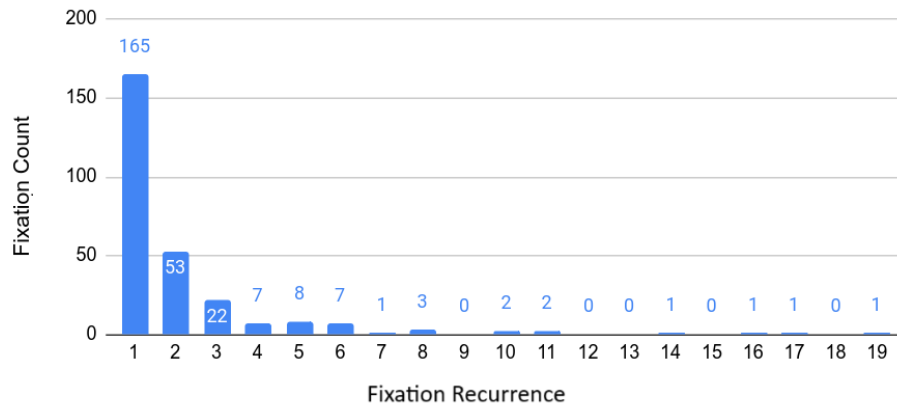


Figura 1 – Distribuição de Fixations.

identificou-se uma maior ocorrência de fixations em issues com menor quantidade de arquivos alterados e linhas de código modificadas. Durante a condução do estudo, observou-se que os participantes tenderam a olhar com menos atenção PRs que apresentavam grandes alterações no código, descendo mais rapidamente pela página. É importante ressaltar que em estudos com rastreadores oculares, a rolagem de páginas é um desafio, especialmente em relação à Fixation, já que a captura da visão de um usuário é feita por meio da resolução do monitor em que o estudo é conduzido, utilizando coordenadas X e Y (Larigaldie; Dreneva; Orquin, 2024). Porém, este não é o caso nesse estudo, já que, por meio do plugin iTrace, é possível capturar diretamente a área de interesse que foi focada na página, independentemente das coordenadas.

4.3.2 Fixation in Conversation

Os participantes tiveram uma maior quantidade de pontos de foco (Fixation Count) e por mais tempo (Fixation Duration) em recomendações que foram consideradas úteis para direcionar na solução do problema (Figura 2). Recomendações consideradas úteis tiveram em média 3.72 pontos de fixation e 20.97 segundos de duração, enquanto recomendações não úteis tiveram 2.23 pontos de fixation e 12.49 segundos de duração. A diferença entre os grupos é significativa para as duas métricas, com tamanho de efeito médio ($W = 940.500$, $p = 0.007$, $r = .357$; $W = 915.000$, $p = 0.017$, $r = .320$), indicando que os participantes tendem a dedicar mais atenção visual ao conteúdo descritivo da issue em recomendações consideradas mais úteis para solucionar a issue proposta, reforçando a percepção de relevância e utilidade do Worked Example.

Resultados similares foram encontrados ao analisar os dados de acordo com o grau de similaridade da recomendação com a tarefa a ser resolvida. A quantidade de pontos de fixation e duração foi maior nas recomendações com alta similaridade (Figura 3). Os participantes tiveram, em média, 3.16 pontos de fixação com duração de 21.69 segundos em recomendações de alta similaridade e 2.63 pontos de fixação com duração de 10.9 segundos em recomendações de baixa similaridade. Contudo, a diferença entre os grupos para as duas métricas não

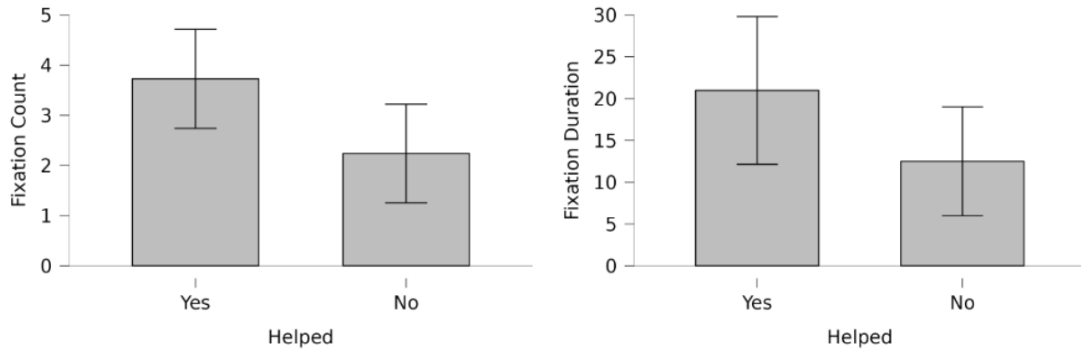


Figura 2 – Boxplot do número de Fixations e duração dos Fixations nas Conversations de acordo com a utilidade do WE.

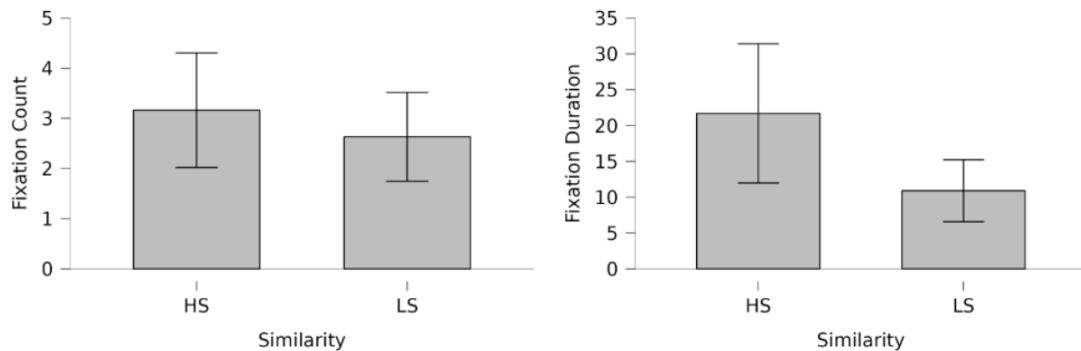


Figura 3 – Boxplot do número de Fixations e duração dos Fixations nas Conversations de acordo com a similaridade do WE.

foi significativa ($W = 746.000$, $p = 0.647$; $W = 801.500$, $p = 0.293$), portanto, a alta similaridade das recomendações não garante engajamento visual consistente aos elementos descritivos da issue recomendada.

4.3.3 Fixation in PR Code

A análise dos dados referentes ao rastreamento ocular dos PRs indicou uma maior quantidade de fixations e maior duração nas recomendações consideradas úteis (Figura 4). Os participantes tiveram, em média, 6 fixations e duração de 13.34 segundos ao analisar os PRs das recomendações úteis e 4.16 fixations e duração de 12.05 segundos ao analisar os PRs das recomendações não úteis. Contudo, a análise não identificou diferenças significativas entre os grupos para as duas métricas ($W = 820.500$, $p = 0.172$; $W = 783.000$, $p = 0.337$), indicando que a utilidade percebida não garante engajamento visual consistente aos elementos de código da Pull Request da issue recomendada.

Em relação ao grau de similaridade das recomendações, identificou-se resultados semelhantes, com um número maior de fixations e tempo de duração em recomendações de alta similaridade (Figura 5). Em média, os participantes tiveram 7.18 fixations e 19.74 segundos em recomendações de alta similaridade, enquanto para issues de baixa similaridade houveram 2.81 fixations e 5.68 segundos. A análise estatística indicou diferença significativa entre os grupos para as duas métricas, com tamanho do efeito pequeno e médio respectivamente ($W = 897.500$,

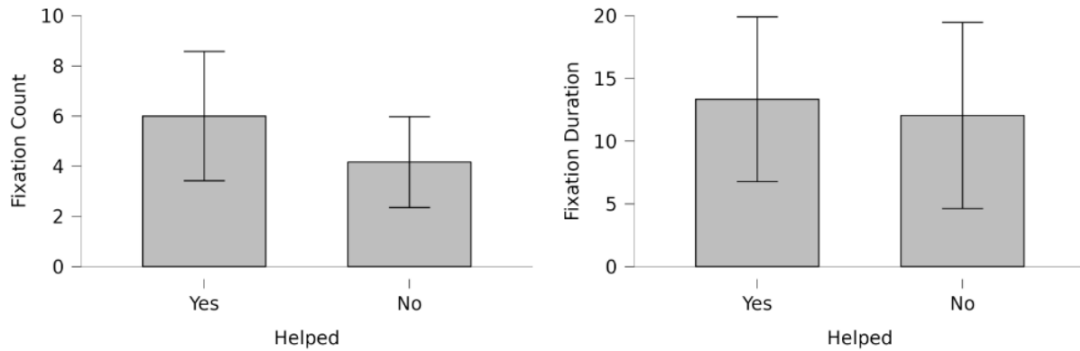


Figura 4 – Boxplot do número de Fixations e Fixation Duration no código do PR de acordo com a utilidade do WE

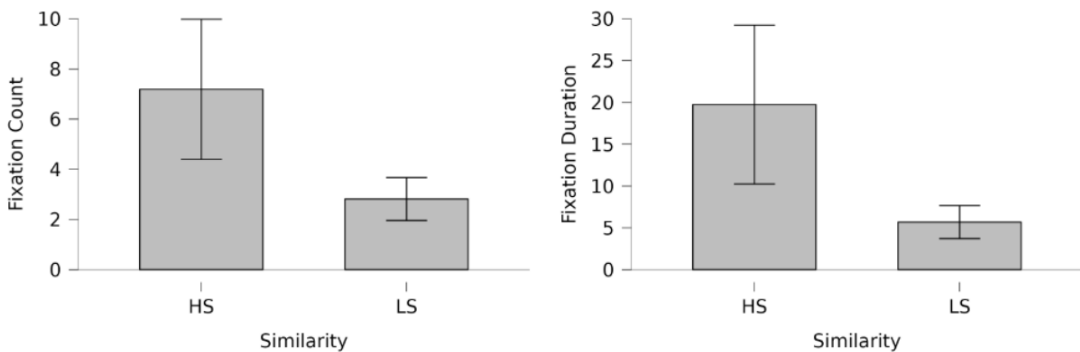


Figura 5 – Boxplot do número de Fixations e Fixation Duration no código do PR de acordo com a similaridade do WE

$p = 0.038$, $r = 0.277$; $W = 921.000$, $p = 0.021$, $r = 0.310$), indicando que os participantes tendem a dedicar mais atenção visual aos elementos de código do Pull Request em recomendações de maior similaridade à issue proposta, reforçando a percepção de relevância e utilidade do Worked Example.

4.3.4 Average Fixation Duration (AFD)

Nesta análise, a média de duração dos Fixations para cada AOI foi ordenada do maior para o menor. Para identificar os elementos mais relevantes, buscou-se AOIs que foram identificadas em pelo menos 50% dos participantes associados à recomendação. Considere, por exemplo, que o PR 9921 recomendado pelo bot foi associado a 10 participantes. Somente AOIs associados a essa recomendação que foram identificadas em pelo menos 5 participantes foram consideradas relevantes. Dentre os Top 10 elementos identificados com as maiores AFDs estão três Checklists (41.7s, 14.3s e 9.1s) que indicam as alterações realizadas no PR, uma imagem (17.9s) que apresenta um exemplo visual de que o PR resolve o problema, três títulos da PR (13.6s, 6.3s e 4.6s), uma linha de código (9s), um nome de arquivo da PR (7.4s) e uma data do comentário presente na PR (4.7s).

4.3.5 Re-review frequency

Ao todo, foram identificados 135 elementos que foram revisitados pelo menos mais uma vez pelos participantes. Contudo, houveram somente 7 elementos que foram revisitados por mais de um participante. O elemento mais revisitado foi a linha de código 27 do arquivo 'EditExternalFileTypeViewModel.java' do PR 9921 sugerido pelo bot para resolver as issues 7 e 9, cujo conteúdo consiste na chamada de um método para definir o valor de um atributo ('defaultApplicationSelectedProperty.setValue(true);'). Cinco participantes revisitaram a linha de código 27 em um total de 12 vezes para pensar em uma solução para a issue 7. Quatro participantes revisitaram essa mesma linha também em um total de 12 vezes para pensar em uma solução para a issue 9.

As issues 7 e 9 são tarefas relacionadas a problemas de interface que consistem, respectivamente, na remoção da proteção de um elemento selecionado e edição de preferências preenchidas pelo valor padrão. O arquivo que contém a linha 27 observada faz parte de uma estrutura de diretórios relacionados às preferências da interface do JabRef (gui/preferences/), além de possuir em seu próprio nome os termos "default" e "selectedProperty", o que pode ter despertado a atenção dos participantes. Considerando essas características, os participantes podem ter vislumbrado a possibilidade dessa chamada de método ser útil na resolução das issues propostas. Tais insights são corroborados pelas citações de participantes como o P6, por exemplo, que após ser apresentado à solução para a issue 9 mencionou: "eu acho que bateu um pouco com o que eu tinha pensado... usar o setValue, procurar algo no banco, que ele usou o mesmo do cara lá, o getDatabase, e ele procurou alguém que já existia e setou o valor. Essa daqui eu senti que ajudou mais do que na outra". O participante P2 também afirma, após a visualização da solução para a issue 7, que "esse daqui (o participante passa o mouse sobre o método setValue) ajuda se você quiser adicionar aquela outra partezinha, aquela outra opção, que a issue tava pedindo".

Outro exemplo está na linha 527 do arquivo ArgumentProcessor.java do PR 9117 sugerido pelo bot para resolver a issue 15. Essa linha consiste na instanciação da classe AtomicFileWriter em uma estrutura try/catch. Três participantes revisitaram essa mesma linha em um total de 4 vezes. A issue 15 consiste em uma tarefa para alterar a nomenclatura para o salvamento de arquivos '.bib'. O nome da classe instanciada 'AtomicFileWriter' pode ter chamado a atenção dos participantes, já que o processo de salvamento envolve a escrita de arquivos. O participante P3, por exemplo, mencionou, após visualizar a solução para a issue 15 que "entre a resposta e a recomendada, acredito que eles estão no mesmo contexto, porque a recomendada está com problema de salvar, gerando um arquivo vazio. Esse daqui, ele tá mudando o tipo de arquivo pra salvar, a extensão do arquivo. Então, a issue recomendada não resolve, porém ela está no mesmo contexto". O participante P7, por sua vez, destaca que "provavelmente essa parte do file writer aqui, que ele colocou, ajudaria".

Considerando os resultados descritos e a métrica de Re-review Frequency (Cheng *et al.*, 2022), os dados indicam a tendência dos participantes de visitar seletivamente elementos cuja

Tabela 2 – Percepção dos participantes sobre cada recomendação

#	Issue	Similarity	15			12			18			14			9			7								
			79.3	78.23	70.97	69.07	68.07	67.83	69.76	67.98	56.26	34.07	33.88	33.46	30.77	28.21	25.77	28	23.14	21.84						
JAVA Exp.	OSS Contrib.		9117	9067	9067	9590	9604	9686	8994	8909	8993	9320	9461	9343	9836	9806	9711	9921	9716	9836						
P1	None	No	DNH	DNH	DNH																DNH					
P2	Medium	No	DNH																		DNH	DNH	H			
P3	None	Yes	H																		H	DNH	DNH			
P4	High	Yes	DNH	DNH	DNH																		DNH			
P5	High	Yes	DNH	H	H										DNH											
P6	High	Yes	DNH												H	NV	NV									
P7	Low	Yes	H												DNH	H	DNH									
P8	Medium	Yes	H	H	H										H											
P9	Medium	Yes				H	H	DNH															DNH			
P10	Low	Yes				DNH															H	NV	NV			
P11	Low	Yes				DNH															H	H	H			
P12	Low	Yes				H	H	H															DNH			
P13	Low	Yes	H	H	H																		DNH			
P14	Medium	Yes	DNH																				DNH	DNH	DNH	
P15	None	Yes							H	H	DNH	DNH														
P16	None	No							H				DNH	DNH	DNH											
P17	Low	No							H				H	DNH	H											
P18	High	Yes							H	DNH	H	DNH														
P19	Low	No							DNH	H	DNH													DNH	DNH	DNH
P20	None	No							DNH																	
H			4	3	3	2	2	1	4	2	1	1	0	1	2	1	0	3	1	2						
DNH			6	2	2	2	0	1	2	1	2	3	2	1	4	1	2	7	3	2						

nomenclatura ou funcionalidade apresentam correspondência semântica com a issue proposta. A correlação entre a terminologia dos elementos revisitados (ex.: 'setValue', 'AtomicFileWriter') e os requisitos das issues indica que a relevância semântica é um fator importante na re-revisão de AOI, evidenciando a relação entre a percepção de utilidade e o alinhamento conceitual entre o elemento revisitado e a issue proposta.

4.4 RQ4: Qual a percepção das pessoas na construção da solução de acordo com o grau de similaridade do Worked Example?

No geral, os resultados apontam para uma percepção positiva dos participantes sobre os Worked Examples. Dentre os 20 participantes do estudo, 15 apontaram pelo menos um caso em que os Worked Examples ajudaram a pensar na solução da Issue. A Tabela 2 apresenta a percepção dos participantes em relação a cada recomendação (H = Ajudou, DNH = Não ajudou, NV = Não visualizou) sugerida de acordo com a tarefa a ser resolvida pelo participante e ordenadas pelo nível de similaridade. A tabela também apresenta o perfil dos participantes em relação à experiência com JAVA e contribuições em SL. As duas últimas linhas apresentam a contagem por percepção para cada recomendação. Recomendações em que pelo menos metade dos participantes consideraram que auxiliou no direcionamento para uma possível solução foram sinalizadas na cor verde. Recomendações em que menos da metade dos participantes consideraram que não auxiliou foram sinalizadas em vermelho.

Worked Examples com maior similaridade à issue proposta foram mais eficazes em guiar os participantes em direção à solução, particularmente quando três recomendações foram apresentadas. Embora a diferença entre recomendações de alta similaridade úteis e não úteis tenha sido pequena (úteis: 22 classificações; não úteis: 18 classificações), recomendações de alta similaridade foram classificadas como úteis com maior frequência que recomendações de baixa similaridade (apenas 11 participantes consideraram recomendações de baixa simi-

laridade úteis). Em casos com três recomendações, exemplos de alta similaridade receberam 18 classificações úteis contra 10 para exemplos de baixa similaridade. Para casos com única recomendação, exemplos de alta similaridade receberam 4 classificações úteis versus 1 para exemplos de baixa similaridade.

Entre os 10 participantes que receberam três recomendações, 80% consideraram pelo menos uma recomendação útil (1 participante: considerou 1 recomendação útil; 4 participantes: consideraram 2 recomendações úteis; 3 participantes: consideraram todas as recomendações úteis), e 20% não consideraram nenhuma recomendação útil. Por outro lado, a apresentação de apenas 1 Worked Example, mesmo com alta similaridade, não foi tão eficaz, ajudando 4 dos 10 participantes a formular a solução para a issue.

Worked Examples com menor similaridade foram menos úteis aos participantes. Entre os 10 participantes que receberam três recomendações, 70% consideraram pelo menos uma recomendação útil (5 participantes: consideraram 1 recomendação útil; 1 participante: considerou duas recomendações úteis; 1 participante: considerou todas as recomendações úteis), e 30% não consideraram nenhuma recomendação útil. Por outro lado, a recomendação de apenas um único Worked Example com baixa similaridade à issue foi percebida como útil por apenas um participante.

Para entender melhor o efeito dos WE, analisamos os resultados de acordo com a experiência dos participantes em Java (Figura 7). Os resultados sugerem que participantes sem experiência em Java foram os menos beneficiados pelos WE, com apenas 5 recomendações consideradas úteis. A falta de conhecimento sobre a linguagem de programação pode ter afetado sua compreensão da issue, reduzindo a utilidade dos WE. O participante P13, por exemplo, afirmou: "não consegui identificar muita (utilidade), porque não sei muito sobre Java e não entendi o código que li". O participante P20 também relatou: "dado meu conhecimento limitado de Java, não acho que ajudaria, analisando a solução e a recomendação do Chatbot. Vejo que não funciona para mim, como um não-desenvolvedor Java".

Participantes altamente experientes também relataram maior número de recomendações que não ajudaram. Dada sua experiência, eram mais propensos a formular uma solução baseada em seu entendimento da issue. Quando apresentados a uma solução diferente, sentiram que o WE os levou por um caminho equivocado. O participante P4, por exemplo, afirmou antes de visualizar a solução: "Honestamente, olhando para isso, meu primeiro pensamento quando vi esta issue com este problema aqui, foi que ele simplesmente não conseguia ler o enum corretamente. Então comparando com a resposta que ele enviou aqui, não posso dizer se é tão similar." Após analisar a solução, o participante declarou: "Realmente é muito diferente do que me mostrou, principalmente porque mostrava mais lógica do que UI. [...] E de fato, revisando o que me mostra aqui é completamente diferente, aqui ele está lidando com lógica, adicionou um try aqui que é para tratamento de erros. Então posso afirmar que não é nem similar. É apenas um problema de exibição, não de lógica."

O grupo que mais se beneficiou dos WE foi o com baixa experiência em Java (ou seja, compreensão básica de Java), com 16 recomendações consideradas úteis. Esses participan-

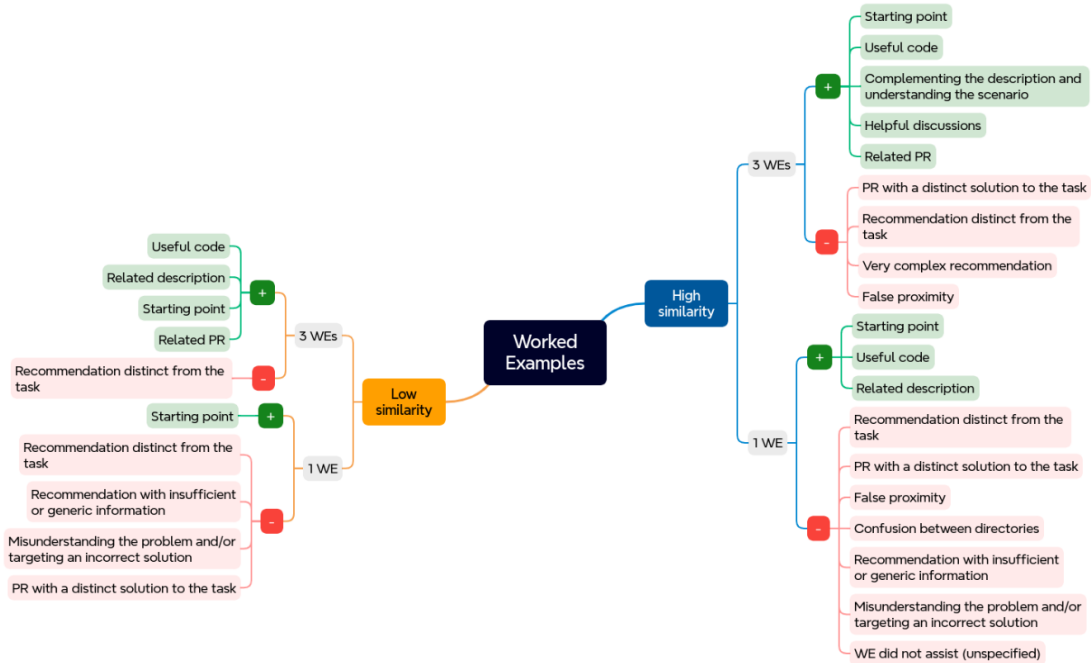


Figura 6 – Distribuição dos códigos gerados na análise qualitativa.

tes têm conhecimento necessário para entender o código, mas ainda precisam de ajuda para chegar a uma solução. Portanto, buscam extrair o máximo de elementos que possam ajudá-los a entender melhor o problema, como estrutura do projeto, padrões de relato de issues e PRs, funções utilizadas e estrutura de arquivos. O participante P19, por exemplo, afirmou: "Sim, direcionou mais ou menos lá para arquivos que parecem estar relacionados à interface do usuário para gerenciamento de grupos, os problemas eram meio diferentes das outras coisas, tipo, o problema que estamos tentando resolver aqui parece ser que a informação que está sendo enviada para edição é diferente do que aparece na tela, e aqui é o que acontece nos bastidores no JabRef para lidar com grupos. Mas parece que direcionou mais ou menos lá, tipo, se eu desse uma olhada nos arquivos ali, se eu tivesse conhecimento do JabRef, tivesse usado, poderia tentar olhar aqui 'oh, estes são arquivos da aba de grupos', então parece que é mais ou menos para esses arquivos aqui, e ele parece ter acertado. Acho que ajudaria, sim." O participante P11 também relatou: "Me ajudou a ter uma base de como repassar as alterações feitas, neste caso, as issues que sugeri para mim, que é um padrão que eu já teria como base para outras issues. Com as recomendações saberia onde tenho que fazer as alterações, conforme onde foi fechado, neste caso, conforme a solução."

A Figura 6 mostra a distribuição dos códigos gerados na análise qualitativa de acordo com o grau de similaridade do WE e o número de WE fornecidos. No geral, fornecer apenas um WE resultou em vários aspectos negativos relatados pelos participantes. WE com baixa e alta similaridade compartilham os mesmos códigos, como "Recomendação distinta da tarefa", "PR com solução distinta da tarefa", "Recomendação com informação insuficiente ou genérica" e "Interpretação equivocada do problema e/ou direcionamento para solução incorreta". No caso de WE com alta similaridade, há também códigos referentes a "Falsa proximidade" e "Confusão entre diretórios".

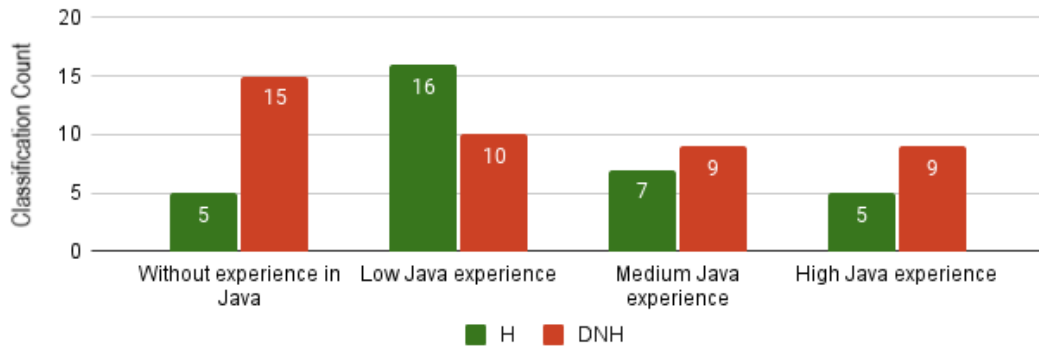


Figura 7 – Distribuição de Worked Examples que Ajudaram (H) e Não Ajudaram (DNH) de acordo com a experiência do participante em Java.

Quando três WE são fornecidos, há uma percepção mais positiva entre os participantes, com maior incidência de códigos positivos. Códigos compartilhados entre WE de baixa e alta similaridade são: "Código útil", "Ponto de partida" e "PR relacionado". WE com baixa similaridade também foram associados ao código "Descrição relacionada". WE com alta similaridade tiveram os códigos "Complementa a descrição e compreensão do cenário" e "Discussões úteis", indicando maior alinhamento entre os WE e a tarefa a ser resolvida. Por outro lado, tanto WE com alta quanto baixa similaridade tiveram casos onde a recomendação foi distinta da tarefa. WE com alta similaridade também receberam os códigos "PR com solução distinta da tarefa", "Falsa proximidade" e "Recomendação muito complexa". Este resultado indica a necessidade de incluir outros dados além do título e descrição da Issue para melhorar a recomendação de WE mais relacionados à tarefa a ser resolvida.

4.5 RQ5: Qual a percepção das pessoas na construção da solução de acordo com o grau de similaridade do Worked Example?

No geral, os participantes tiveram uma percepção positiva da abordagem. Os resultados do TAM (Figura 8) indicam que os participantes consideraram-na relativamente útil, bastante fácil de usar e com intenção de utilizá-la se a mesma estivesse disponível. Contudo, é importante destacar que a tecnologia apresentou as menores notas na dimensão "Utilidade Percebida" (PU). Como os participantes foram atribuídos tanto à Worked Examples com alta similaridade quanto com baixa similaridade, com somente uma ou com três recomendações, a maior parte dos participantes (17) teve pelo menos um caso em que o Worked Example não ajudou na elaboração da solução, o que pode ter afetado a percepção quanto à utilidade da abordagem. A segmentação das avaliações entre os três grupos (Ajudou, Ajudou parcialmente e Não ajudou) reforça o impacto das WE consideradas não úteis na dimensão PU (Figura 8). Em relação à Facilidade de Uso Percebida (PEOU), a menor nota referente ao item PEOU4 ("Eu acho fácil fazer com que o bot faça o que eu quero") pode ser explicada pela interação limitada do

participante com o bot, que consiste simplesmente em visualizar as recomendações e acessá-las.



Figura 8 – Resultado do questionário TAM: (a) por item avaliado e (b) por dimensão avaliada.

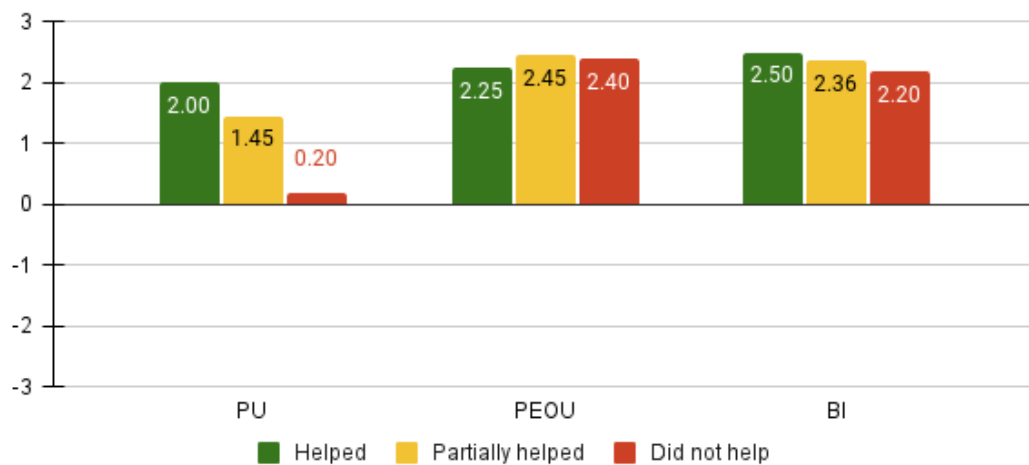


Figura 9 – Resultado do questionário TAM de acordo com a percepção dos participantes sobre a utilidade do WE.

5 DISCUSSÕES

Os resultados indicam o potencial do uso de Worked Examples para auxiliar desenvolvedores novatos na resolução de issues em aberto. No geral, os participantes perceberam a utilidade das WE no auxílio à resolução de issues em aberto.

Em relação à **RQ1** (“*Quais os benefícios do uso de worked examples no auxílio à construção da solução de issues?*”), a análise qualitativa revelou que os WE de alta similaridade com três recomendações forneceram um ponto de partida para direcionar os participantes no desenvolvimento da solução, como por exemplo, a localização de arquivos e pastas potencialmente relevantes para a resolução do problema. As recomendações possibilitaram também compreender melhor o cenário do problema, seja por meio de discussões relevantes entre os membros da comunidade, por meio do fornecimento de issues relacionados ao contexto ou pela análise da estrutura de arquivos presente nos PRs. Os participantes relataram ainda a identificação de trechos de código úteis, que possibilitaram a identificação de métodos, extensões e funções relacionadas, bem como PR relacionados com a tarefa a ser resolvida. Tais resultados estão em acordo com o trabalho anterior de Tan *et al.* (2023), os quais identificaram que bugs similares possibilitam contextualizar melhor sobre o bug e sua solução, reforçando a importância e a relevância do uso de WE na resolução de issues.

Em relação à **RQ2** (“*Quais os desafios do uso de worked examples no auxílio à construção da solução de issues?*”), o principal desafio refere-se à relação entre a recomendação e a tarefa. Apesar de issues com maior similaridade terem sido considerados mais úteis, os participantes relataram que algumas recomendações não possuem muita relação com a tarefa ou possuem soluções distintas que não resolvem diretamente o problema. Em alguns casos, os participantes relataram, antes de visualizar a solução, terem percebido uma relação entre a tarefa e a recomendação. Porém, após visualizar a solução, perceberam que se tratavam de coisas diferentes. Embora a abordagem de recomendação baseada na similaridade textual do título e do conteúdo da issue tenha trazido WE relevantes e úteis, é importante que outros dados, tais como arquivos modificados, diretórios, bem como nomes de métodos, funções e classes sejam também considerados, de forma a retornar recomendações mais relacionadas à tarefa.

Outro desafio está na complexidade das recomendações. Um participante relatou ter se assustado com a complexidade da recomendação em comparação com a solução efetiva da tarefa, que era bastante simples. Em linha com o desafio anterior, é necessário refinar a abordagem proposta para direcionar melhor os desenvolvedores na solução. Durante o estudo, notou-se que os participantes tendiam a fixar menos a atenção em PRs muito extensos, descendo pela página mais rapidamente. Tais resultados revelam a necessidade de se filtrar os códigos úteis em potencial, de forma a fornecer conteúdo mais direcionado ao desenvolvedor, principalmente em PRs extensos com modificações em diversos arquivos e inclusão/exclusão de muitas linhas de código. Assim, desenvolvedores podem focar a sua atenção em trechos de código mais relevantes.

Em relação à **RQ3** (“*Quais métricas do EyeTracker evidenciam os benefícios e desafios identificados na construção da solução?*”), a análise dos fixations revelou uma grande quantidade de elementos que foram focados somente uma vez durante o estudo. Diferentemente de outros trabalhos que envolvem pequenos programas ou trechos de código (Cheng *et al.*, 2022; Jbara; Feitelson, 2015), este estudo envolveu a análise de PRs completos de uma ou mais recomendações que resultam em uma quantidade maior de arquivos e linhas de código a serem analisadas, o que pode explicar a grande quantidade de elementos focados pelos participantes.

A maior quantidade e duração dos fixations nas discussões presentes nos WE que auxiliaram os participantes e em WE que possuem maior similaridade com a tarefa indicam que essas recomendações forneceram informações em potencial que chamaram a atenção dos participantes para pensar na solução da tarefa, fazendo-os analisá-las com mais atenção. Resultados similares foram identificados na análise dos fixations sobre o código das PRs. A maior quantidade e duração dos fixations indica que houveram mais elementos em potencial no código que os participantes analisaram com mais atenção para tentar compreender a sua função e verificar se os mesmos podem ou não auxiliar na resolução da tarefa.

Por outro lado, a maior quantidade de elementos observados e a duração das observações implica também em uma maior dispersão da atenção e um maior esforço cognitivo dos participantes, indicando uma dificuldade em localizar rapidamente elementos que podem ser úteis na resolução do problema. Tais resultados estão de acordo com o trabalho de Morrison, Margulieux e Guzdial (2015), que identificou uma maior carga cognitiva quando os problemas divergem de contexto dos WE. Isso indica que os participantes acabam levando mais tempo tentando identificar informações úteis ou compreender o código presente na recomendação. Os resultados do EyeTracker revelaram, por exemplo, que os participantes buscam por elementos que possuem relação com a tarefa a ser resolvida analisando o nome da classe, método ou função. Assim, destacar esses elementos em potencial no código da recomendação pode possibilitar uma maior eficiência das recomendações no auxílio à solução da tarefa e reduzir a carga cognitiva.

Em relação à **RQ4** (“*Qual a percepção das pessoas na construção da solução de acordo com o grau de similaridade do Worked Example?*”), os resultados indicam uma percepção mais positiva sobre WE com maior similaridade com a tarefa, principalmente quando são fornecidas três recomendações. Embora a maior quantidade de recomendações resulte em um maior tempo para analisá-las, os resultados indicam a importância de se fornecer mais opções para que o desenvolvedor explore potenciais soluções. Estudos envolvendo a busca por exemplos de código em ferramentas de busca como o Google indicam que nem sempre as melhores soluções estão entre os primeiros retornados pelo motor de busca (Chatterjee; Juvekar; Sen, 2009; Hora, 2021). Assim, é essencial que sejam recomendados mais de um WE para os desenvolvedores, principalmente os com maior grau de similaridade. Embora a recomendação de WE por meio da análise de similaridade do título em conjunto com a descrição da issue pela distância de cosseno tenha retornado issues relevantes, os participantes ainda relataram PRs com solução distinta da tarefa mesmo em WE com alta similaridade. Assim, é importante que sejam incluí-

dos outros dados na análise de similaridade, de forma a recomendar WE mais direcionados à tarefa, como por exemplo, a identificação de arquivos modificados, funções, métodos e classes que possuam relação com a tarefa a ser trabalhada.

Em relação à **RQ5** (“*Qual a percepção sobre a utilidade, facilidade de uso e intenção de uso dos participantes em relação a uma ferramenta de sugestão de Worked Examples?*”) os resultados do TAM evidenciam uma percepção positiva dos participantes em relação à proposta, os quais consideraram fácil de usar e que pretendem utilizá-la quando estivesse disponível. A menor avaliação na Utilidade Percebida tem relação direta com o design balanceado do estudo, em que os participantes foram apresentados à issues com alta e baixa similaridade, estes últimos mais frequentemente considerados como não úteis no auxílio à resolução das tarefas. Ao segmentar a análise entre participantes que consideraram que o WE ajudou, ajudou parcialmente e não ajudou, é possível observar um aumento na avaliação da dimensão PU, indicando que o fornecimento somente de WE com maior similaridade potencialmente resultará em avaliações mais positivas quanto à utilidade da abordagem proposta.

6 AMEAÇAS À VALIDADE

Os participantes deste estudo são todos de uma mesma instituição de ensino, o que pode não representar com precisão a população de desenvolvedores novatos. Para mitigar essa limitação, foram recrutados participantes com perfis variados (diferentes períodos do curso, diferentes níveis de conhecimento em Java e contribuições em projetos de SL).

Os resultados do estudo podem ser afetados pelo perfil dos participantes. Para minimizar esse risco, os participantes foram balanceados e distribuídos entre as condições de acordo com o nível de conhecimento em Java e contribuições em projetos de SL.

A seleção de issues a serem analisadas no estudo também pode afetar os resultados do estudo. Para minimizar esse risco, foram selecionadas as 3 issues que possuíam WE com maior similaridade e 3 issues que possuíam WE com menor similaridade, de forma a obter uma maior variedade de issues a serem analisadas. O processo de codificação da análise qualitativa é sujeito à subjetividade do pesquisador. Para minimizar esse risco, a codificação foi realizada por um dos autores do artigo e revisada por outros dois pesquisadores. As discordâncias das análises e dos resultados foram discutidas de forma exaustiva até que houvesse um consenso.

A condução do estudo em somente um repositório limita a generalização dos resultados. Estudos em outros repositórios são necessários para validar os achados deste trabalho.

7 CONCLUSÃO

Este artigo apresentou um estudo para investigar o potencial do uso de Worked Examples (WEs) no auxílio a desenvolvedores novatos na resolução de issues abertas em projetos de software livre. A pesquisa foi conduzida com 20 alunos do curso de bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná, os quais analisaram issues fechadas do repositório do JabRef selecionadas para os testes, distribuídas em grupos com exemplos de alta e baixa similaridade recomendadas por um bot desenvolvido pelos autores.

Os resultados indicam que WEs com maior similaridade à tarefa a ser resolvida auxiliaram mais os participantes em direção à solução, principalmente quando fornecidos três recomendações, revelando o potencial do uso da distância de cosseno para identificar issues em relevantes. A análise qualitativa das respostas dos participantes revelou que os WEs ajudaram a identificar diretórios, arquivos e trechos de código relevantes, servindo de ponto de partida para o desenvolvimento da solução. Além disso, a visualização de issues relacionadas e as discussões presentes nessas recomendações possibilitaram uma maior compreensão sobre o cenário da tarefa a ser resolvida.

Os resultados também apontaram alguns desafios na utilização de WEs. O principal desafio está na identificação de issues relevantes. Embora no geral WEs com maior similaridade tenham sido considerados mais úteis, houveram situações em que WEs com alta similaridade apresentaram soluções distintas da tarefa, mesmo com descrições que remetem a um problema similar, dando uma sensação de falsa proximidade. Nesse sentido, é importante que pesquisas futuras envolvendo recomendações de issues considerem não somente a similaridade textual (título e corpo da issue), mas também outros elementos como o nome dos arquivos, diretórios, funções, métodos e classes.

Outro desafio está na redução da carga cognitiva e na identificação de elementos relevantes nos WEs. Os resultados do rastreamento ocular revelaram uma maior quantidade e duração de fixations, indicando que os participantes tiveram dificuldade em localizar informações relevantes nas recomendações. Assim, é importante que futuras abordagens identifiquem e destaquem trechos de código, descrições e discussões relevantes, de forma a direcionar melhor os desenvolvedores e reduzir a carga cognitiva.

Os resultados deste trabalho são um ponto de partida para a investigação do uso de WEs em projetos de Software Livre e abrem diversas perspectivas futuras. Pesquisas podem ser conduzidas em outros repositórios, de forma a validar os achados deste estudo, bem como envolver recomendações de problemas similares em repositórios de projetos distintos. Estudos também podem ser conduzidos para identificar padrões de comportamento no consumo de WEs de forma a desenvolver estratégias que facilitem a identificação de elementos relevantes.

REFERÊNCIAS

- ALDERLIESTEN, J. W. D.; ZAIDMAN, A. An initial exploration of the "good first issue" label for newcomer developers. *In: 2021 IEEE/ACM 13TH INTERNATIONAL WORKSHOP ON COOPERATIVE AND HUMAN ASPECTS OF SOFTWARE ENGINEERING (CHASE)*. 2021. **Anais [...]** [S.l.: s.n.], 2021. p. 117–118.
- BHASIN, T.; MURRAY, A.; STOREY, M.-A. Student experiences with github and stack overflow: An exploratory study. *In: 2021 IEEE/ACM 13TH INTERNATIONAL WORKSHOP ON COOPERATIVE AND HUMAN ASPECTS OF SOFTWARE ENGINEERING (CHASE)*. 2021. **Anais [...]** [S.l.: s.n.], 2021. p. 81–90.
- BOOTH, J. L. *et al.* Using example problems to improve student learning in algebra: Differentiating between correct and incorrect examples. **Learning and Instruction**, v. 25, p. 24–34, 2013.
- CHATTERJEE, S.; JUVEKAR, S.; SEN, K. Sniff: A search engine for java using free-form queries. *In: FUNDAMENTAL APPROACHES TO SOFTWARE ENGINEERING*. 2009. **Anais [...]** [S.l.: Springer Berlin Heidelberg, 2009. p. 385–400.
- CHEN, O.; KALYUGA, S.; SWELLER, J. The worked example effect, the generation effect, and element interactivity. **Journal of Educational Psychology**, v. 107, n. 3, p. 689–704, 2015.
- CHEN, O. *et al.* The effect of worked examples on learning solution steps and knowledge transfer. **Educational Psychology**, v. 43, n. 8, p. 914–928, 2023.
- CHEN, O.; RETNOWATI, E.; KALYUGA, S. Effects of worked examples on step performance in solving complex problems. **Educational Psychology**, v. 39, n. 2, p. 188–202, 2019.
- CHENG, S. *et al.* Gaze-based annotations for reading comprehension. *In: PROCEEDINGS OF THE 33RD ANNUAL ACM CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS*. 2015. **Anais [...]** [S.l.: s.n.], 2015. p. 1569–1572.
- CHENG, S. *et al.* Collaborative eye tracking based code review through real-time shared gaze visualization. **Frontiers of Computer Science**, v. 16, n. 3, p. 163704, 2022.
- DAVIS, F. D.; BAGOZZI, R. P.; WARSHAW, P. R. Technology acceptance model. **J Manag Sci**, v. 35, n. 8, p. 982–1003, 1989.
- ERICSON, B. J.; MARGULIEUX, L. E.; RICK, J. Solving parsons problems versus fixing and writing code. *In: PROCEEDINGS OF THE 17TH KOLI CALLING INTERNATIONAL CONFERENCE ON COMPUTING EDUCATION RESEARCH*. 2017, New York, NY, USA. **Anais [...]** New York, NY, USA: Association for Computing Machinery, 2017. p. 20–29. Disponível em: <https://doi.org/10.1145/3141880.3141895>.
- FITZSIMMONS, C. J. *et al.* Can feedback, correct, and incorrect worked examples improve numerical magnitude estimation precision? **The Journal of Experimental Education**, v. 91, n. 1, p. 20–45, 2023.
- GARCES, S. *et al.* Engaging students in active exploration of programming worked examples. **Education and Information Technologies**, v. 28, n. 3, p. 2869–2886, 2022.
- GAWEDA, A. M. *et al.* Typing exercises as interactive worked examples for deliberate practice in cs courses. *In: PROCEEDINGS OF THE TWENTY-SECOND AUSTRALASIAN COMPUTING EDUCATION CONFERENCE*. 2020. **Anais [...]** [S.l.: s.n.], 2020. p. 105–113.

- GUARNERA, D. T. *et al.* itrace: Eye tracking infrastructure for development environments. *In: PROCEEDINGS OF THE 2018 ACM SYMPOSIUM ON EYE TRACKING RESEARCH & APPLICATIONS*. 2018. **Anais [...]** [S.l.: s.n.], 2018. p. 105:1–105:3.
- HORA, A. Googling for software development: What developers search for and what they find. *In: 2021 IEEE/ACM 18TH INTERNATIONAL CONFERENCE ON MINING SOFTWARE REPOSITORIES (MSR)*. 2021. **Anais [...]** [S.l.: s.n.], 2021. p. 317–328.
- HÜSING, S. *et al.* Using worked examples for engaging in epistemic programming projects. *In: PROCEEDINGS OF THE 55TH ACM TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION*. 2024. **Anais [...]** [S.l.: s.n.], 2024. p. 443–449.
- JBARA, A.; FEITELSON, D. G. How programmers read regular code: A controlled experiment using eye tracking. *In: 2015 IEEE 23RD INTERNATIONAL CONFERENCE ON PROGRAM COMPREHENSION*. 2015. **Anais [...]** [S.l.: s.n.], 2015. p. 244–254.
- LARIGALDIE, N.; DRENEVA, A.; ORQUIN, J. L. eyescrollr: A software method for reproducible mapping of eye-tracking data from scrollable web pages. **Behavior Research Methods**, v. 56, n. 4, p. 3380–3395, 2024.
- MORRISON, B. B.; MARGULIEUX, L. E.; GUZDIAL, M. Subgoals, context, and worked examples in learning computing problem solving. *In: PROCEEDINGS OF THE ELEVENTH ANNUAL INTERNATIONAL CONFERENCE ON INTERNATIONAL COMPUTING EDUCATION RESEARCH*. 2015. **Anais [...]** [S.l.: Association for Computing Machinery, 2015. p. 21–29.
- SANTOS, F. *et al.* How to choose a task? mismatches in perspectives of newcomers and existing contributors. *In: PROCEEDINGS OF THE 16TH ACM/IEEE INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING AND MEASUREMENT*. 2022. **Anais [...]** [S.l.: s.n.], 2022. p. 114–124.
- SHARAFI, Z. *et al.* A practical guide on conducting eye tracking studies in software engineering. **Empirical Software Engineering**, v. 25, p. 3128–3174, 2020.
- STEINMACHER, I. *et al.* Being a mentor in open source projects. **Journal of Internet Services and Applications**, v. 12, n. 1, p. 7, 2021.
- STEINMACHER, I. *et al.* Overcoming open source project entry barriers with a portal for newcomers. *In: 2016 IEEE/ACM 38TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING*. 2016. **Anais [...]** [S.l.: s.n.], 2016. p. 273–284.
- TAN, X. *et al.* Is it enough to recommend tasks to newcomers? understanding mentoring on good first issues. *In: PROCEEDINGS OF THE 45TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING*. 2023. **Anais [...]** [S.l.: s.n.], 2023. p. 653–664.
- TAN, X.; ZHOU, M.; SUN, Z. A first look at good first issues on github. *In: PROCEEDINGS OF THE 28TH ACM JOINT MEETING ON EUROPEAN SOFTWARE ENGINEERING CONFERENCE AND SYMPOSIUM ON THE FOUNDATIONS OF SOFTWARE ENGINEERING*. 2020. **Anais [...]** [S.l.: s.n.], 2020. p. 398–409.
- ZHI, R. *et al.* Exploring the impact of worked examples in a novice programming environment. *In: PROCEEDINGS OF THE 50TH ACM TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION*. 2019. **Anais [...]** [S.l.: s.n.], 2019. p. 98–104.