

Identificando Casos de Folga Social em Repositórios de Código Aberto

João Américo Martins Caiafa de Andrade¹

¹Bacharelado em Engenharia de Software
Instituto de Ciências Exatas e Informática
Pontifícia Universidade Católica de Minas Gerais (PUC Minas)
Edifício Fernanda, Rua Claudio Manoel, 1.162, Funcionários
30.140-100 — Belo Horizonte — MG — Brasil

joao.andrade@sga.pucminas.br

Abstract. *In recent years, the need for large-scale software projects has meant that projects are generally developed in groups. However, unique challenges arise for team leaders to evaluate the work of each group member fairly and objectively. For generally, individuals who find themselves in team environments when they lack motivation and responsibility engage in social loafing behavior. Thus, this paper aims to propose and evaluate a method to identify cases of social loafing based on the contribution of members in the source code of open source projects. The results obtained demonstrate that the proposed method can be used as a tool to assist project managers in identifying possible cases of social loafing in teams.*

Keywords: *social loafing, GitHub, contribution, repository mining.*

Resumo. *Nos últimos anos, a necessidade de projetos de software em grande escala fez com que os projetos sejam geralmente desenvolvidos em grupos. No entanto, surgem desafios únicos para os líderes de equipe avaliarem o trabalho de cada membro do grupo com justiça e objetividade. Pois geralmente indivíduos que se encontram em ambientes de equipe quando não têm motivação e responsabilidade exercem comportamento de folga social. Dessa forma, esse artigo pretende propor e avaliar método para identificar casos de folga social a partir da contribuição dos membros no código-fonte de projetos em código aberto. Os resultados obtidos demonstram que o método proposto pode ser utilizado como uma ferramenta para auxiliar os gestores de projetos a identificar possíveis casos de folga social nas equipes.*

Palavras-chave: *folga social, GitHub, contribuição, mineração de repositórios.*

Bacharelado em Engenharia de Software - PUC Minas
Trabalho de Conclusão de Curso (TCC)

Orientador de conteúdo (TCC I): Cleiton Tavares - cleitontavares@pucminas.br

Orientadora de conteúdo (TCC I): Simone de Assis - simone@pucminas.br

Orientador acadêmico (TCC I): Laerte Xavier - laertexavier@pucminas.br

Orientador do TCC II: Johnatan Alves de Oliveira - johnatan.oliveira@pucminas.br

Belo Horizonte, 14 de maio de 2023.

1. Introdução

Nos últimos anos, a necessidade de projetos de *software* em grande escala fez com que os projetos sejam geralmente desenvolvidos em grupos. Sendo a capacidade de trabalho em equipe muito importante para Engenheiros de Software [Nguyen et al. 2020]. Normalmente, em projetos o código-fonte é disponibilizado em plataformas de hospedagem de código, onde cada desenvolvedor pode contribuir com suas atualizações do *software* [Fontão et al. 2020]. No entanto, juntamente com as vantagens de uma colaboração em equipe, surgem desafios únicos para os líderes de equipe avaliarem o trabalho de cada membro do grupo com justiça e objetividade. Pois geralmente indivíduos que se encontram em ambientes de equipe quando não têm motivação e responsabilidade exercem comportamento de folga social, fenômeno que se caracteriza onde os indivíduos exercem menos esforço em direção a uma meta de equipe do que quando trabalhassem sozinhos [Buffardi 2020].

Quando não se possui uma visão do desempenho dos desenvolvedores durante o andamento do projeto, torna-se um desafio para o gerenciamento de grandes equipes. Logo, para organização do projeto é ideal ter a visão do impacto que cada atualização de um desenvolvedor afeta em todas as funcionalidades do sistema. Além de ter a visão de que forma cada membro da equipe contribuiu para o projeto, rastreando seu histórico de atualizações, e como foi dividido entre o grupo as demandas [Nguyen et al. 2020]. Sendo essas visões necessárias para o gerenciamento de grandes equipes, que se tornaram cada vez mais comuns em *softwares* de grande escala. Por isso este trabalho buscou solucionar o problema: **a ausência de métodos para identificar a ocorrência de folga social a partir da análise do código-fonte em repositórios de projetos de código aberto.**

No passado, a contribuição de cada membro do grupo só podia ser julgada pela avaliação mútua dos membros ou pelo julgamento subjetivo do líder de equipe [Nguyen et al. 2020]. Atualmente, em projetos de *software* como as contribuições do desenvolvedor são geralmente armazenadas em repositórios como GitHub, elas podem fornecer medições efetivas para entender o desempenho do desenvolvedor [Fontão et al. 2020]. A partir das contribuições, é possível identificar casos de folga social. Como resultado da folga social, algumas equipes de engenharia de software incluem membros que não contribuem com sua parte do trabalho. Nesses casos, seria injusto que os outros membros da equipe sofressem, principalmente se tivessem tentando engajar e motivar os outros [Buffardi 2020]. Por isso, em muitas equipes acontece de ter uma falta de sintonia entre os membros, onde alguns desenvolvedores se esforçam mais para compensar a falta de comprometimento dos demais [Mihelič and Culiberg 2018].

Baseado no que foi discutido, este trabalho tem por objetivo **propor e avaliar método para identificar casos de folga social a partir da contribuição dos membros no código-fonte de projetos em código aberto.** Propõem-se os seguintes objetivos específicos: 1) analisar o impacto da contribuição ao código-fonte, a partir da relação de número de erros encontrados, correções de erros anteriores e inserção de novas linhas; 2) avaliar como o comportamento do desenvolvedor muda durante a vida do projeto, a partir da análise do histórico de atualizações; 3) analisar se é possível identificar casos de folga social a partir da análise do histórico de atualizações. Deseja-se responder 2 questões de pesquisa com esse trabalho: i) Como identificar casos de folga social em repositórios? ii) Como a quantidade de integrantes afeta a quantidade de casos de folga social?

A partir da mineração dos repositórios e análise dos códigos fontes para se obter as métricas de contribuição de cada autor. É possível mensurar como foi a participação de cada desenvolvedor no projeto e assim identificar casos de folga social na equipe de desenvolvimento. Além disso, agrupando os resultados para identificar possíveis padrões nas taxas de folga social pelo tamanho dos grupos.

Este trabalho está organizado da seguinte forma: Seção 2 apresenta a fundamentação teórica. Seção 3 aborda trabalhos relacionados. Seção 4 descreve a metodologia proposta no estudo. Seção 5 apresenta os resultados obtidos. Seção 6 detalha as ameaças à validade e suas mitigações. Seção 7 mostra as conclusões do estudo e propõem trabalhos futuros.

2. Fundamentação Teórica

Esta seção contém os principais conceitos abordados neste estudo. O primeiro referente a contribuição de cada membro em repositórios. No segundo se define o que é folga social e como surgiu o conceito.

2.1. Contribuição em Repositórios

Repositório é um tipo de armazenamento de projeto que utiliza controles de versão, o que possibilita manter todo o código e o histórico de edições em um só lugar para facilitar a colaboração. Os repositórios não só permitem padrões de colaboração flexíveis, mas também conectam toda a história de edição de arquivos de um determinado projeto. O controle da versão do sistema Git registra quem fez a alteração, quando, e com base em que estado dos dados atuais. Estes metadados estão disponíveis publicamente e podem ser consultados por meio de uma Interface de Programação de Aplicações (API). Os serviços de alojamento de Repositório são, portanto, uma valiosa fonte de informação para investigar características específicas de projetos de desenvolvimento de *softwares* [Bonvoisin et al. 2018].

As atualizações de um projeto são realizadas por meio de *commits*, que retêm conhecimento sobre o que mudou desde uma atualização anterior [Hokka et al. 2021]. As plataformas mantêm um registro de contribuições (ou *commits*), incluindo suas mensagens de *commits* (uma descrição da revisão), hora, autor, e quantas linhas mudaram (incluindo ambas as inserções e exclusões) [Buffardi 2020]. Com isso, é possível identificar em um repositório quais as contribuições foram realizadas por um usuário específico e aplicar métricas para análise de suas atualizações.

2.2. Folga Social

A folga social, é um fenômeno observável quando o esforço entregue por uma pessoa, exibem uma diminuição considerável no nível individual de esforço quando atuam em grupos em comparação com quando atuam sozinhos [Latané et al. 1979].

No início da década de 1880, um professor de engenharia agrícola Max Ringelmann projetou uma tarefa de puxar corda para seus alunos e mediu seu desempenho. O que ele observou foi um pior desempenho quando o número de alunos puxando a corda aumentou. Em outros termos, ao trabalhar com seus colegas, alunos eram mais propensos a relaxar, o que décadas mais tarde foi conhecido como o efeito Ringelmann. Somente

no final da década de 1970 é que esse estudo pioneiro foi replicado em uma série de experimentos demonstrando que os alunos que se apresentaram sozinhos bateram palmas e gritaram mais intensamente do que aqueles que fizeram o mesmo como parte de um grupo. A presença de outras pessoas os levaram a reduzir seu esforço em aplaudir. Foi então que Latané e colegas cunharam o termo folga social [Mihelič and Culiberg 2018].

Estudo prévio descobriu que é na falta de motivação que a folga social surge. Foram identificados fatores críticos para folga social do indivíduo a partir de duas dimensões: fatores de nível de grupo (ou seja, constituição, dispersão e justiça) e características da tarefa (ou seja, visibilidade da tarefa, complexidade da tarefa). Folga social pode ser usada para explicar perdas de produtividade em grupos de trabalho. Vários estudos na área de Tecnologia da Informação (TI) também investigaram folga no ambiente baseado de TI. Piezon descobriu que nas redes sociais a folga é significativa em grupos de estudantes online. Pesquisas anteriores estudaram folga social com métodos quantitativos e 149 consideravam a folga social como um estado estático [Zhang et al. 2017].

3. Trabalhos Relacionados

Os trabalhos analisados nesta seção são relacionados à medição da contribuição de usuários e identificação de casos de folga social.

Buffardi (2020) procura mensurar a contribuição de cada indivíduo em 42 projetos de software, a partir da análise dos commits realizados e histórias de usuário completados no quadro de projeto Kanban. Para análise foi realizado a mineração dos repositórios e agrupado os commits pelos autores, dessa forma identificando individualmente a contribuição de cada um no projeto. Os resultados obtidos apresentaram uma forte correlação entre a avaliação em pares e interação dos indivíduos no repositório. O autor ressalta a dificuldade de mensurar a total contribuição de um indivíduo em um projeto, visto que algumas etapas de um projeto de software não constam no histórico de um repositório [Buffardi 2020].

Nguyen et al (2020), analisa os resultados obtidos a partir da aplicação do Pro-Edu4Group, sistema de avaliação de projetos em grupo. O software se baseia na integração do código do grupo, realizando teste a cada implantação e alertando os estudantes caso infrinja o estilo de código adotado. Por meio do feedback fornecido pelo sistema, os alunos podem corrigir rapidamente partes dos projetos do grupo que não estão conforme o estilo de codificação e ajudar os alunos a manter a qualidade do código. Além disso, o sistema propõe métricas para avaliar a contribuição de cada indivíduo em um grupo. No final é feita uma relação das métricas obtidas com a nota que cada estudante recebeu (A, B, C ou D). Os resultados obtidos foram que estudantes com nota A e B contribuem em maior parte do código e o que está correlacionado por receberem uma maior nota. Os mesmos alunos também são responsáveis pela maior parte dos erros implantados no código, além de sua correção antes da entrega final [Nguyen et al. 2020].

Leung (1998) cita a importância de que profissionais da computação tenha habilidades para cooperar em projetos com muitas pessoas. Como em grupos alguns se esforçam muito, enquanto outros quase não participam. Dessa forma não sendo certo avaliar todos os integrantes do grupo da mesma forma. Por isso o autor propõe estimar a contribuição individual de cada integrante. Foi solicitado que cada estudante do grupo avaliasse seus companheiros de grupo a partir da contribuição que cada um teve. A pes-

quiza leva em consideração que os estudantes podem ser honestos com a avaliação dos outros, mas ser tendencioso com a sua nota. Ao mesmo tempo que o estudante possa ser desonesto e atribuir notas incorretas para os outros [Leung 1998].

Fronza and Wang (2021) afirmam que a folga social se trata de um problema comum em equipes de desenvolvimento de software. Em equipes ágeis de desenvolvimento de *software* são definidas regras para prevenir o comportamento de folga social. Os autores se propõem a estudar como são especificadas estas regras. Para isso realizaram primeiro um estudo de caso com 14 equipes de estudantes de desenvolvimento de software. Após isso uma análise dos conjuntos de regras consideradas mais eficazes por equipes profissionais de desenvolvimento ágil. Os resultados mostram que equipes de desenvolvimento de software ágil especificam as expectativas de toda a equipe sobre quando ter reuniões e como se comportar nelas, como lidar com bloqueios de trabalho, como comunicar e colaborar com colegas de equipe e que meios de comunicação devem ser utilizados. Estas regras destinam-se a prevenir comportamentos de folga social [Fronza and Wang 2021].

Robert (2020) apontam a falta de controle social e a incapacidade de observar ou de confiar que outros estão a cumprir os seus compromissos são frequentemente citadas como causas principais de folga social em equipes virtuais onde existem os desafios da dispersão geográfica e uma dependência das comunicações eletrônicas. O estudo visa analisar como o controle social e a confiança entre os membros da equipe impactam nos casos de folga social. Para isso foram utilizados questionários em 39 equipes com objetivo de analisar o controle, confiança individual, confiança afetiva e esforço individual. Os resultados demonstram que o controle da equipe e confiança dos membros tem um efeito positivo na redução de casos de folga social [Robert 2020].

4. Materiais e Métodos

O presente trabalho é classificado como pesquisa quantitativa. Esta característica se deve ao fato de se buscar casos de folga social através da coleta e análise de dados obtidos pela mineração de repositórios. A seguir, é explicado o método proposto e são apresentadas as etapas realizadas neste trabalho:

4.1. Método

O objetivo desse trabalho é propor um método para identificar casos de folga social, para isso é necessário identificar as contribuições que cada integrante fez em seu repositório.

A contribuição foi medida a partir de análise estática dos códigos fonte utilizando a ferramenta Radon¹ exclusiva para linguagem Python. A partir das métricas obtidas, se realizou uma comparação a partir da ordem de publicação de cada código, obtendo-se um histórico de como a qualidade do código evolui durante o projeto e como cada contribuição interferiu nos resultados de cada métrica obtida. Desse forma é possível comparar o desempenho entre cada desenvolvedor e verificar se algum deles possui um desempenho abaixo da média geral de seu grupo. Caso o desenvolvedor possua os menores valores em todas as métricas e não exista outro autor com valores iguais, o método classifica ele como um caso de folga social no grupo.

¹<https://radon.readthedocs.io/en/latest/>

4.2. Etapas

Conforme os procedimentos identificados no Seção 4.1, pode-se definir as seguintes etapas a serem seguidas neste trabalho:

1. **Criação de Dataset:** A primeira etapa se dá com desenvolvimento de um *script* para mineração de dados do GitHub. Sendo escolhido inicialmente os 1.000 repositórios de código aberto, de linguagem Python, com mais estrelas no GitHub. A partir do resultado da primeira consulta foi obtido o número de colaboradores para cada repositório, sendo escolhido apenas os que possuem mais que 2 colaboradores, com objetivo de filtrar os projetos individuais e utilizar apenas os projetos feitos por um grupo de pessoas. Com a segunda análise feita, obteve-se para cada repositório a lista de *commits* realizados na *branch* principal, que foi utilizada para obter o código-fonte necessário para as próximas etapas.
2. **Análise estática dos repositórios:** Utilizando a biblioteca Radon e os resultados da primeira etapa, realizou-se uma análise estática com objetivo de obter do código de cada *commit* as métricas de contribuição como: número de linhas modificadas, erros corrigidos e erros adicionados. As métricas foram escolhidas, a partir de como os autores concordam que a métrica de contribuição serve para avaliar a contribuição individual dos programadores. No entanto, salientam que esta métrica deve ser utilizada em conjunto com outras informações como correção de erros, sugerindo que estas métricas podem complementar a avaliação subjetiva da contribuição dos programadores. Através da leitura dos valores históricos das métricas, é possível detectar variações consideráveis, que podem indicar que os promotores lidam com problemas pessoais ou profissionais durante o projeto [Lima et al. 2015].
3. **Tratamento dos dados encontrados e aplicação de método:** A partir da análise dos códigos foi realizado uma comparação de cada *commit* com seu anterior e agrupar os resultados pelo autor de cada *commit*, com objetivo de medir a contribuição que cada colaborador realizou no projeto. Dessa forma obteu-se as métricas que ajudaram a identificar se a contribuição interferiu na qualidade do código, ao mesmo tempo, descartar *commits* que não representaram nenhuma mudança expressiva no código. Foram desconsiderados, como possíveis casos de folga social, os autores que não possuíam durante todo o projeto: nenhuma linha modificada, número de erros adicionados ou corrigidos, igual a zero e apenas um *commit* realizado. O objetivo é remover autores que não possuem métricas suficientes para demonstrar algum tipo de atuação no projeto.
4. **Análise dos resultados:** Realizado a análise dos resultados e comparação entre os repositórios para identificar os possíveis casos de folga social.

5. Resultados e Analises

Nesta seção, são apresentados os resultados obtidos e as análises realizadas neste estudo. A seção 5.1 responde a primeira questão de pesquisa. A seção 5.2 utilizando os resultados obtidos na seção anterior responde a segunda questão de pesquisa.

Os resultados são baseados nas métricas obtidas a partir da análise de 76 repositórios. Para cada repositório foram analisados um total de 1.000 *commits*. A quantidade de repositórios foi definida a partir da data limite para execução, para atender o tempo limitado para execução.

5.1. Identificar folga social em repositórios

Para analisar como identificar casos de folga social em repositórios, é necessário medir o nível de contribuição que cada autor teve de participação em um repositório. Foram analisados as métricas obtidas para cada *commit* de um repositório e agrupadas pelo nome do autor. Utilizando como base as métricas de número de linhas modificadas e número de submissões, utilizadas por Nguyen et al (2020). Assim como a quantidade de erros adicionados e erros corrigidos, utilizado por Lima et al. (2015).

A Tabela 1 representa um exemplo das métricas obtidas de um repositório. Identificamos como caso de folga social nos casos em que o mesmo autor possui os menores valores de linhas modificadas e submissões. Para a quantidade de erros a pessoa tem que de possuir o número de erros corrigidos menor ou igual ao número de erros adicionados, o objetivo é detectar casos falso positivos, em que o autor trabalhou no projeto com objetivo único de corrigir erros, o que poderia diminuir seu número de submissões, mas não a contribuição durante o projeto. Como exemplo o autor A possui o menor valor para linhas modificadas e número de *commit*, além de o número de erros corrigidos ser igual ao de erros adicionados. Este exemplo é um caso que o algoritmo classificou como caso de folga social.

A partir da análise de 76 repositórios foram identificados 32 possíveis casos de folga social, representando 42,10% de casos positivos. Os projetos identificados com folga social podem ter durante seu desenvolvimento apresentar, excesso de trabalho por parte de alguns integrantes do grupo e o tempo de entrega pode ter sido comprometido. Visto que esses repositórios não apresentaram uma divisão da carga de trabalho equilibrada para todos os integrantes.

Autor	Linhas modificadas	Erros corrigidos(%)	Erros adicionados(%)	Número de commits
A	1	0	0	4
B	221	10.235	3.41	9
C	794	32.73	86.59	46
D	843	6.82	17.05	81

Tabela 1. Exemplos das métricas obtidas para um repositório

5.2. Quantidade de autores por repositório

Para analisar como a quantidade de integrantes afeta a quantidade de casos de folga social, foi agrupado o número de repositórios com e sem folga social pelo número de pessoas que cada projeto possui. Podemos perceber a partir da Figura 1 um padrão em que na maioria dos grupos o número de casos de repositórios com folga social é menor. Mas é possível ver uma alta na quantidade de casos de folga social em projetos com 21 até 30 pessoas e de 31 a 40 integrantes, representando respectivamente 85,71% e 60% dos repositórios com essa faixa. Esses resultados superaram a média dos outros grupos que se encontra em 31,27%. Além disso, é possível perceber que a base utilizada apresenta uma alta concentração de projetos com até 10 integrantes, representando 47,36% de todos os repositórios analisados.

Foi identificado que em grupos com muitos desenvolvedores não é possível identificar casos apenas pela análise do código-fonte, pois o método classifica como folga social apenas quando só um integrante do grupo apresenta métricas abaixo do restante da equipe. Por isso para ser possível identificar casos de folga social em grupos com muitos integrantes é necessário complementar com análises mais diretas como, por exemplo, entrevistas e questionários com os desenvolvedores.

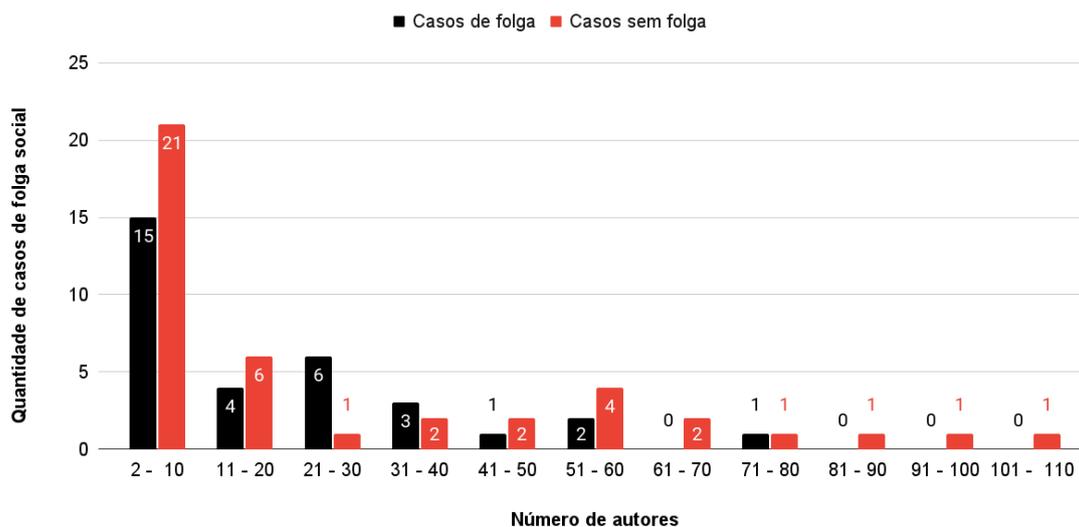


Figura 1. Quantidade de casos de folga social por número de autores

6. Ameaças à Validade

Nesta seção, são apresentadas as ameaças à validade deste estudo, assim como as estratégias adotadas para mitigá-las.

Para a ameaça à validade de construção, a quantidade de repositórios analisados pode não se estatisticamente representativa. Visto que não há o conhecimento do número total projetos executados no mundo real. Para mitigar essa ameaça foram minerados os repositórios mais populares do GitHub e com número de desenvolvedores diversos, com objetivo de avaliar os projetos mais representativos.

Quanto a ameaça a validade interna, é possível ocorrer problemas de conexão com API, erros de execução no *script* detalhado na Seção 4.1, assim como quedas de energia que podem interromper a máquina utilizada para mineração dos dados. Para lidar com essa ameaça o *script* armazena os resultados em arquivos temporários, para que caso seja interrompido o progresso não, seja perdido por completo. Também foi utilizado três máquinas para execução dos processos, com cada sistema executando uma parte da lista de repositórios. Dessa forma garantindo uma maior eficiência na execução dos processos caso houvesse alguma interrupção.

Para a ameaça à validade externa, assim como diversos estudos da Engenharia de Software, não é possível generalizar os resultados encontrados. Por isso foi utilizado um conjunto de dados vasto que consiste em 76 repositórios, 7.742 desenvolvedores e 72.254 *commits* analisados.

7. Conclusão e trabalhos futuros

Este estudo propôs avaliar um método para identificar casos de folga social a partir da contribuição dos membros em projetos de código aberto. A partir da mineração dos repositórios e análise do código de cada *commit*, foram obtidas métricas de contribuição para cada autor. Foram identificados que 42,10% dos repositórios analisados apresentaram indícios de folga social. Tendo uma alta expressiva de 85,71% para os grupos que possuem de 21 a 30 membros na equipe. Demonstrando que o método proposto pode ser utilizado como auxílio para identificação dos casos de folga social em projetos com grandes equipes.

Para trabalhos futuros pode-se executar o estudo em repositórios com código privado ou desenvolvido utilizando outras linguagens além da Python, com objetivo de identificar se o tipo de repositório afeta do quantidade de casos de folga social. Além disso, para se obter uma análise mais detalhada é possível complementar o estudo a partir da entrevista dos membros das equipes, com objetivo de que eles avaliem os outros integrantes. Assim, é possível obter uma visão melhor do desempenho de cada desenvolvedor durante o projeto.

Pacote de Replicação

O pacote de replicação deste trabalho encontra-se disponível em: [link do repositório GitHub do trabalho](#).

Referências

- [Bonvoisin et al. 2018] Bonvoisin, J., Buchert, T., Preidel, M., and Stark, R. G. (2018). How participative is open source hardware? insights from online repository mining. *Design Science*, 4:e19.
- [Buffardi 2020] Buffardi, K. (2020). Assessing individual contributions to software engineering projects with git logs and user stories. SIGCSE '20, page 650–656, New York, NY, USA. Association for Computing Machinery.
- [Fontão et al. 2020] Fontão, A., Cafeo, B. B. P., Bonifácio, B., dos Santos, R. P., and Dias-Neto, A. C. (2020). *On Developer Relations Team's Reasons for Using Repositories*, page 187–188. Association for Computing Machinery, New York, NY, USA.
- [Fronza and Wang 2021] Fronza, I. and Wang, X. (2021). Social loafing prevention in agile software development teams using team expectations agreements. *IET Software*, 15(3):214–229.
- [Hokka et al. 2021] Hokka, H., Dobslaw, F., and Bengtsson, J. (2021). Linking developer experience to coding style in open-source repositories. In *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 516–520.
- [Latané et al. 1979] Latané, B., Williams, K., and Harkins, S. (1979). Many hands make light the work: The causes and consequences of social loafing. *Journal of Personality and Social Psychology*, 37(6):822–832.
- [Leung 1998] Leung, Y.-W. (1998). Least-square-error estimate of individual contribution in group project. *IEEE Transactions on Education*, 41(4):282–285.

- [Lima et al. 2015] Lima, J., Treude, C., Filho, F. F., and Kulesza, U. (2015). Assessing developer contribution with repository mining-based metrics. In *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 536–540.
- [Mihelič and Culiberg 2018] Mihelič, K. K. and Culiberg, B. (2018). Reaping the fruits of another’s labor: The role of moral meaningfulness, mindfulness, and motivation in social loafing. *Journal of Business Ethics*, 160(3):713–727.
- [Nguyen et al. 2020] Nguyen, B.-A., Ho, K.-Y., and Chen, H.-M. (2020). Measure students’ contribution in web programming projects by exploring source code repository. In *2020 International Computer Symposium (ICS)*, pages 473–478.
- [Robert 2020] Robert, L. P. (2020). Behavior–output control theory, trust and social loafing in virtual teams. *Multimodal Technologies and Interaction*, 4(3).
- [Zhang et al. 2017] Zhang, X., Jiang, S., and Cheng, Y. (2017). Inferring the student social loafing state in collaborative learning with a hidden markov model. In *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion*. ACM Press.