

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GILMAR GOMES DO NASCIMENTO

O USO DE LOGS NO ENSINO DE PROGRAMAÇÃO

CURITIBA

2025

GILMAR GOMES DO NASCIMENTO

O USO DE LOGS NO ENSINO DE PROGRAMAÇÃO

USING LOGS TO SUPPORT PROGRAMMING EDUCATION

Proposta de Dissertação apresentado como requisito para obtenção do título de Mestre em Ciência da Computação do Mestre em Computação Aplicada da Universidade Tecnológica Federal do Paraná.

Orientador(a): Prof. Dr. Laudelino Cordeiro Bastos

Coorientador(a): Profa. Dra. Maria Claudia Figueireido Pereira Emer

CURITIBA

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.

RESUMO

No desenvolvimento de software, é comum utilizar métricas para avaliar a qualidade do código e o tempo despendido em sua produção. No ambiente profissional, essa análise ocorre principalmente por meio de *pull requests*, enquanto no contexto acadêmico, a avaliação costuma ser feita por meio de envios pontuais (como e-mails ou sistemas de gestão de aprendizagem - LMS). Embora os LMS registrem *logs* de atividades e eventos, há uma lacuna na coleta estruturada de dados sobre o processo de aprendizagem em programação, especialmente em ambientes como laboratórios de informática. Este trabalho propõe a criação de um *dataset* com registros (logs) do processo de desenvolvimento de código, capturando o rastro da aprendizagem em tempo real. Diferentemente de abordagens convencionais, que dependem de Sistemas de Gestão de Aprendizagem, a coleta será realizada por meio de um *plugin* integrado a um editor de código amplamente utilizado, voltado tanto para programação quanto para documentação. Espera-se fornecer uma base de dados sobre o ensino-aprendizagem de programação que possa auxiliar na identificação de dificuldades dos estudantes, na personalização do ensino e no aprimoramento das práticas pedagógicas.

Palavras-chave: ensino de programação; log; telemetria; logging.

ABSTRACT

In software development, it is common to use metrics to assess the quality of code and the time spent producing it. In Education is not default, however, this project will help to obtain new informations to recognize standard and metrics. This work proposes the creation of a *dataset* with records (logs) of the code development process, capturing the learning trail in real time. Unlike conventional approaches, which rely on Learning Management Systems, the collection will be carried out using a *plugin* integrated with a widely used code editor, aimed at both programming and documentation. The goal of this paper is to provide a detailed database for future analyses, contributing to the study of programming teaching and learning.

Keywords: programming education; log; telemetry; logging.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1	FUNDAMENTAÇÃO TEÓRICA	8
LinkbordercolorLinkcolor1.1	Desafios no Ensino de Programação e a Lacuna Instrucional sobre Logging.....	8
LinkbordercolorLinkcolor1.2	Logging: Conceitos e Aplicações na Engenharia de Software	8
LinkbordercolorLinkcolor1.3	A Análise de Logs como Ferramenta Educacional	9
LinkbordercolorLinkcolor1.4	Metodologia para Análise de Logs Educacionais	9
1.4.1	Coleta e Pré-processamento	9
1.4.2	Geração de Indicadores Educacionais	9
1.4.3	Análise de Padrões e Autoria.....	10
LinkbordercolorLinkcolor1.5	A Extensão Logado: Proposta de Implementação	10
	REFERÊNCIAS.....	11

1 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica necessária para o entendimento desta dissertação. Inicialmente, contextualiza os desafios do ensino de programação e introduz o conceito de *logging* como uma prática crucial da Engenharia de Software. Em seguida, explora o uso de *logs* no contexto educacional, detalha a taxonomia e os tipos de informações registradas, e apresenta a metodologia proposta para a análise desses dados. Por fim, é descrita a extensão *Logado*, que constitui a contribuição prática deste trabalho.

1.1 Desafios no Ensino de Programação e a Lacuna Instrucional sobre Logging

O ensino de programação transcende a mera transmissão de conhecimentos técnicos, abrangendo aspectos como boas práticas de desenvolvimento, integridade acadêmica e a compreensão do ciclo de vida do software. No entanto, uma análise da literatura pedagógica fundamental revela uma lacuna significativa.

Para mapear o estado da arte, consultou-se os conteúdos de livros-texto amplamente adotados globalmente (como os de Sommerville e Pressman) e a principal referência nacional, “Engenharia de Software Moderna” de Marco Tulio Valente. Constatou-se que, embora essas obras abordem tópicos como testes e qualidade de software, não dedicam seções ou discussões substantivas ao *logging* como uma prática de engenharia a ser ensinada, com suas próprias estratégias e melhores práticas (Gu *et al.*, 2022).

Esta omissão é notável dada a criticalidade dos *logs* na indústria e contrasta com o potencial educacional da análise dos rastros digitais que os alunos naturalmente geram durante a codificação. No Brasil, o tema é ainda mais incipiente; uma busca no portal da SBC com os termos `log OR logging AND "ensino de programação"` retorna poucos ou nenhum resultado, indicando um nicho de pesquisa promissor.

1.2 Logging: Conceitos e Aplicações na Engenharia de Software

Para estabelecer uma base terminológica unificada, adota-se a taxonomia de (Gu *et al.*, 2022), que define os conceitos fundamentais:

- | | | |
|-----------------|---------------|--------------------|
| • Log statement | • Log message | • Log |
| • Log level | • Log content | • Log location |
| • Log placement | • Logging | • Logging practice |

Segundo (Rice; Borgman, 1983), o monitoramento de sistemas pode ser entendido como um registro automático de transações. A importância dos *logs* é vasta, sendo utilizados em tarefas como detecção de anomalias, depuração, diagnóstico de desempenho e modelagem de comportamento de sistemas (Fu *et al.*, 2014). Estudos como o de (Yang *et al.*, 2021) confir-

mam que os desenvolvedores os utilizam primariamente para análise de problemas, buscando informações como a propagação de erros, *timestamps* e o *dataflow* entre componentes.

1.3 A Análise de Logs como Ferramenta Educacional

Aplicar a análise de *logs* no ensino de programação significa transpor essa prática da manutenção de sistemas para a análise do processo de aprendizagem. Os algoritmos de ensino não produzem apenas resultados; geram *logs* e rastros digitais detalhados que, quando analisados, podem fornecer insights sobre a curva de aprendizagem.

Como sugerido por (Silva *et al.*, 2022), variáveis preditoras podem ser extraídas do código, tais como: número de linhas lógicas, uso de estruturas de repetição, quantidade de identificadores, funções utilizadas, entre outras. Esta abordagem está alinhada com a *Learning Analytics*, cuja meta é melhorar a qualidade do ensino e da aprendizagem através da análise de dados sobre o comportamento dos estudantes (Huang *et al.*, 2020). Os *logs* de programação funcionam, assim, como um registro de aprendizagem (*learning record*) digital, revelando a experiência pessoal e as reflexões do estudante (Du; Wagner, 2005).

1.4 Metodologia para Análise de Logs Educacionais

A análise dos dados coletados será realizada em três etapas principais:

1.4.1 Coleta e Pré-processamento

Os dados brutos são gerados a partir da interação dos estudantes com o ambiente de programação, capturando eventos como edições de código, compilações e erros. Esta fase envolve:

- **Registro de Logs:** Captura de *timestamp*, trechos de código editados e eventos de desenvolvimento.
- **Filtragem e Anonimização:** Substituição de dados sensíveis por identificadores únicos para garantir a privacidade.

1.4.2 Geração de Indicadores Educacionais

Os *logs* brutos são transformados em métricas educacionais significativas (e.g., tempo para resolver um erro, padrões de tentativa-e-erro), criando um *dataset* para pesquisa. Este *dataset* permitirá estudos sobre a eficácia de metodologias de ensino e a correlação entre padrões de codificação e desempenho.

1.4.3 Análise de Padrões e Autoria

Com base no perfil de usuário (*user profile*) construído—que inclui frequência de *commits* e uso de estruturas específicas—é possível identificar discrepâncias na autoria de códigos, auxiliando na detecção de plágio e fornecendo uma base para avaliação personalizada.

1.5 A Extensão Logado: Proposta de Implementação

A motivação para o desenvolvimento da extensão *Logado* reside na evolução dos ambientes de programação. Argumenta-se que é mais eficaz integrar a funcionalidade de *logging* em uma IDE já consagrada e amplamente utilizada, evitando a curva de aprendizado associada a novas ferramentas. Dessa forma, a extensão será executada diretamente na IDE do estudante, capturando de forma integrada os eventos de desenvolvimento detalhados na seção 1.4.

REFERÊNCIAS

- ANNAMAA, A. Thonny, a python ide for learning programming. *In: PROCEEDINGS OF THE 2015 ACM CONFERENCE ON INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION*. 2015. **Anais [...]** [S.l.: s.n.], 2015. p. 343–343.
- ARABYARMOHAMADY, S.; MORADI, H.; ASADPOUR, M. A coding style-based plagiarism detection. *In: IEEE. PROCEEDINGS OF 2012 INTERNATIONAL CONFERENCE ON INTERACTIVE MOBILE AND COMPUTER AIDED LEARNING (IMCL)*. 2012. **Anais [...]** [S.l.], 2012. p. 180–186.
- CLEMENTE, R. G. Uma arquitetura para processamento de eventos de log em tempo real. **Agosto de**, „, 2008.
- DU, H. S.; WAGNER, C. Learning with weblogs: An empirical investigation. *In: IEEE. PROCEEDINGS OF THE 38TH ANNUAL HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES*. 2005. **Anais [...]** [S.l.], 2005. p. 7b–7b.
- EDUCAÇÃO, C. e. T. d. A. Instituto Federal de. **Projeto Pedagógico do Curso Técnico de Nível Médio em Informática, na Forma Subsequente**. Boca do Acre: , 2020. Disponível em: <https://cbda.ifam.edu.br/cursos/subsequentes/info/ppc-info-subsequente.pdf>.
- FOUNDATION, L. **The Linux Foundation**. [S.l.]: , . <https://trainingportal.linuxfoundation.org/learn/course/getting-started-with-opentelemetry-lfs148/why-do-we-need-opentelemetry/how-we-got-here?page=5>. Accessed: 2024-12-3.
- FU, Q. *et al.* Where do developers log? an empirical study on logging practices in industry. *In: COMPANION PROCEEDINGS OF THE 36TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING*. 2014, Hyderabad India. **Anais [...]** Hyderabad India: ACM, 2014. p. 24–33. ISBN 978-1-4503-2768-8. Disponível em: <https://dl.acm.org/doi/10.1145/2591062.2591175>.
- GATEV, R. Observability: Logs, metrics, and traces. *In: Introducing Distributed Application Runtime (Dapr): Simplifying Microservices Applications Development Through Proven and Reusable Patterns and Practices*. Berkeley, CA: Apress, 2021. p. 233–252. ISBN 978-1-4842-6998-5. Disponível em: https://doi.org/10.1007/978-1-4842-6998-5_12.
- GU, S. *et al.* Logging practices in software engineering: A systematic mapping study. **IEEE Transactions on Software Engineering**, IEEE v. 49, n. 2, p. 902–923, 2022.
- GUPTA, A.; JINDAL, M.; GOYAL, A. Identification of student programming patterns through clickstream data. *In: IEEE. 2024 IEEE INTERNATIONAL CONFERENCE ON COMPUTING, POWER AND COMMUNICATION TECHNOLOGIES (IC2PCT)*. 5., 2024. **Anais [...]** [S.l.], 2024. p. 1153–1158.
- HEVNER, A. R. *et al.* Design science in information systems research. **MIS quarterly**, JSTOR,, p. 75–105, 2004.
- HUANG, A. Y. *et al.* Predicting students' academic performance by using educational big data and learning analytics: evaluation of classification methods and learning logs. **Interactive Learning Environments**, Taylor & Francis v. 28, n. 2, p. 206–230, 2020.
- KENNEDY, A. R. **Towards a data-driven analysis of programming tutorials' telemetry to improve the educational experience in introductory programming courses**. 2015. Tese (Doutorado) 2015.

KITCHENHAM, B. Procedures for performing systematic reviews. **Keele, UK, Keele University**, v. 33, n. 2004, p. 1–26, 2004.

LIMA, N. A. *et al.* Avaliação da aprendizagem nos institutos federais em goiás durante a pandemia: uma investigação documental, . Insitituto Federal de Educação, Ciência e Tecnologia de Goiás,, 2022.

LIU, C. *et al.* User behavior discovery from low-level software execution log. **IEEEJ Transactions on Electrical and Electronic Engineering**, Wiley Online Library v. 13, n. 11, p. 1624–1632, 2018.

MANGAROSKA, K. *et al.* Exploring students' cognitive and affective states during problem solving through multimodal data: Lessons learned from a programming activity. **Computer Assisted Learning**, v. 38, n. 1, p. 40–59, fev. 2022. ISSN 0266-4909, 1365-2729. Disponível em: <https://onlinelibrary.wiley.com/doi/10.1111/jcal.12590>.

NGUYEN, B.-A.; HA, T.-V. T.; CHEN, H.-M. Analyzing git log in an code-quality aware automated programming assessment system: A case study. **tvujs**, ,, set. 2023. ISSN 2815-6099, 2815-6072. Disponível em: <https://journal.tvu.edu.vn/index.php/journal/article/view/2430>.

PAGE, M. J. *et al.* The prisma 2020 statement: an updated guideline for reporting systematic reviews. **Systematic reviews**, Springer v. 10, n. 1, p. 1–11, 2021.

PEFFERS, K. *et al.* A design science research methodology for information systems research. **Journal of management information systems**, Taylor & Francis v. 24, n. 3, p. 45–77, 2007.

PEREIRA, F. D. *et al.* Using learning analytics in the amazonas: understanding students' behaviour in introductory programming. **British journal of educational technology**, Wiley Online Library v. 51, n. 4, p. 955–972, 2020.

QIAN, Y.; LEHMAN, J. Students' misconceptions and other difficulties in introductory programming: A literature review. **ACM Transactions on Computing Education (TOCE)**, ACM New York, NY, USA v. 18, n. 1, p. 1–24, 2017.

RAABE, A. L. A.; SILVA, J. d. Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. *In*: SN. XIII WORKSHOP DE EDUCAÇÃO EM COMPUTAÇÃO (WEI'2005). SÃO LEOPOLDO, RS, BRASIL. 3 n. 5., 2005. **Anais [...]** [S.l.], 2005.

RICE, R. E.; BORGMAN, C. L. The use of computer-monitored data in information science and communication research. **Journal of the American Society for Information Science**, Wiley Online Library v. 34, n. 4, p. 247–256, 1983.

RODRIGUES, F. C. R.; GAVA, R. Universidades federais no estado de minas gerais: Um estudo comparativo. **REAd | Porto Alegre – Edição 83**, SciELO Brasil,, 2016.

SILVA, E. S. *et al.* Previsão de indicadores de dificuldade de questões de programação a partir de métricas do código de solução. *In*: SBC. ANAIS DO XXXIII SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. 2022. **Anais [...]** [S.l.], 2022. p. 859–870.

UMEZAWA, K. *et al.* Analysis of logic errors utilizing a large amount of file history during programming learning. *In*: IEEE. 2020 IEEE INTERNATIONAL CONFERENCE ON TEACHING, ASSESSMENT, AND LEARNING FOR ENGINEERING (TALE). 2020. **Anais [...]** [S.l.], 2020. p. 630–634.

YANG, N. *et al.* An interview study of how developers use execution logs in embedded software engineering. *In*: IEEE. 2021 IEEE/ACM 43RD INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: SOFTWARE ENGINEERING IN PRACTICE (ICSE-SEIP). 2021. **Anais [...]** [S.l.], 2021. p. 61–70.