

# Em Direção a uma Ferramenta de Recomendação de Código para Apoiar o Aprendizado de Programação

Guilherme de Ávila de Moura, Williamson Silva  
{guilhermemoura.aluno, williamsonsilva}@unipampa.edu.br  
Universidade Federal do Pampa, Campus Alegrete, Rio Grande do Sul, Brasil

Gabriel Machado Lunardi  
gabriel.lunardi@ufsm.br  
Universidade Federal de Santa Maria, Campus Cachoeira do Sul, Rio Grande do Sul, Brasil

Aprender programação de computadores é uma atividade desafiadora, especialmente para quem está iniciando na área. Os principais desafios enfrentados pelos estudantes novatos são abstração e compreensão da lógica de programação [8], ou seja, sistematizar problemas em algoritmos e materializá-los em código-fonte [14, 19]. Como resultado, alguns possíveis problemas a curto e longo prazo podem emergir [10, 11]: (i) a curto prazo, o tempo gasto para realização das atividades e fixação de novos conceitos de programação, que ocorre principalmente quando o estudante ainda está na fase inicial de aprendizado; (ii) a longo prazo, em que o estudante, já atuando no mercado de trabalho, apresenta menor produtividade em traduzir as necessidades do cliente em uma solução algorítmica, impactando no atraso do projeto e até mesmo na entrega para o cliente. Uma maneira para tentar minimizar esse problema pode ser por meio de um apoio ferramental, em que o estudante verifica a semântica e a completude do seu raciocínio em relação a um determinado problema. Além disso, tal apoio pode auxiliar o estudante na descoberta de diferentes formas sobre como solucionar um problema, tais como a escrita de uma estrutura de controle condicional na forma *if-else* ou, alternativamente, na forma de um teste ternário.

Os sistemas de recomendação baseados em código-fonte (*Source Code based Recommendation System - SCoReS*) fornecem suporte para algumas tarefas, por exemplo: como usar uma determinada API (*Application Programming Interface*) ou uma estrutura; dicas sobre elementos que estão omissos ou ambíguos no código; sugestões sobre como reutilizar ou corrigir um código existente; ou, ajudar programadores novatos a aprender um novo projeto, paradigma de programação, ou estilo [7]. Com isso, um SCoReS é relevante para a melhoria do desenvolvimento de software, principalmente do código-fonte, por este ser o artefato mais atualizado e, portanto, a fonte de dados mais confiável [13, 20]. É importante destacar que o código-fonte fornece uma fonte rica e estruturada de informações, com base nas quais os sistemas de recomendação podem ser construídos para fornecer recomendações úteis a desenvolvedores de software ou iniciantes em programação.

Com isso em mente, foi realizada uma pesquisa em artigos científicos, publicados em conferências e periódicos, e na loja de extensões do editor de código *VSCode*, tendo em vista sua popularidade [15], a fim de verificar ferramentas que pudessem ser utilizadas como auxílio aos estudantes. Dentre as identificadas, a que mais se aproximou

da ideia de apoiar os estudantes durante a escrita de códigos foi a SoCeR (*A New Source Code Recommendation Technique for Code Reuse*) [16], uma ferramenta que recomenda código-fonte por meio da interação em linguagem natural com o usuário. Entretanto, a ferramenta não está disponível para uso e não existem detalhes sobre seu desenvolvimento. Ainda, descrevem outras ferramentas similares, como Koder [2] e Krugle [1], que também foram aqui testadas e suas funcionalidades se mostraram insuficientes para atender às necessidades de nossa metodologia [16]. Já as extensões do editor *VSCode* testadas foram: Clara [4], Blackbox [5], Captain Stack [6], Tabnine [3] e GitHub Copilot [9]. Após uma minuciosa avaliação de todas as ferramentas apresentadas, foi evidenciado que nenhuma delas atendia plenamente às necessidades da metodologia proposta. As únicas ferramentas que se mostraram adequadas para o propósito não estavam disponíveis para uso (SoCeR) ou eram de caráter pago (GitHub Copilot). Diante desse cenário, torna-se evidente a necessidade de proposição, desenvolvimento e avaliação experimental de uma ferramenta de código aberto que seja capaz de atender às especificações requeridas e listadas anteriormente.

Portanto, este trabalho tem por objetivo desenvolver um SCoReS, para apoiar o ensino de programação, com acesso público e facilitado a todos. Para tanto, será necessário conduzir algumas etapas. (i) Identificar, construir e validar uma base de conhecimento útil em programação para desenvolvedores de software iniciantes que servirá como entrada para o sistema de recomendação. A partir dela, vislumbra-se a recomendação de código a partir da própria interação do usuário, conforme este projeta seus códigos durante o aprendizado de programação. Em outras palavras, o sistema SCoReS aprenderá como a pessoa pensa para resolver os problemas e sugerirá melhorias e correções, aumentando continuamente sua base de dados. Além disso, será necessário (ii) definir qual(is) modelo(s)/algoritmo(s) de recomendação, baseados em Inteligência Artificial, serão empregados. À primeira vista está o emprego do modelo baseado em conteúdo que emprega técnicas de mineração de textos [17], seguida de modelos pré-treinados de linguagens, tais como *codeBERT* [12]. Também espera-se (iii) desenvolver uma interface para acesso aos resultados da recomendação por meio de uma extensão para o editor *VSCode*. A escolha do editor está ligada à popularidade da ferramenta entre programadores iniciantes e experientes. Por fim, espera-se (iv) avaliar a assertividade das recomendações geradas [18] para o apoio do aprendizado de programação, assim como a experiência de aprendizagem dos estudantes (*learning experience*) com a ferramenta proposta. Ressalta-se que esta proposta está será pautada em uma metodologia baseada em experimentação, que utiliza estudos experimentais para avaliar e evoluir tecnologias de software [21].

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

*EduComp'23, Abril 24-29, 2023, Recife, Pernambuco, Brasil (On-line)*

© 2023 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

## 1 AGRADECIMENTOS

Os autores agradecem a CAPES - 001 e UNIPAMPA. Os autores também agradecem pelo apoio financeiro da FAPERGS (Projeto ARD/ARC – processo 22/2551-0000606-0) e CNPq (Projeto Universal nº 405973/2021-7).

## REFERÊNCIAS

- [1] [n.d.]. Krugle. Disponível em: <https://www.krugle.com>. Acesso em 28 de janeiro de 2023.
- [2] 2005. Koder. Disponível em: <https://www.koders.com>. Acesso em 28 de janeiro de 2023.
- [3] 2018. Tabnine AI Autocomplete. Disponível em: <https://marketplace.visualstudio.com/items?itemName=TabNine.tabnine-vscode>. Acesso em 28 de janeiro de 2023.
- [4] 2021. Clara Copilot. Disponível em: <https://marketplace.visualstudio.com/items?itemName=BadBoy17G.clara-copilot>. Acesso em 28 de janeiro de 2023.
- [5] 2022. Blackbox. Disponível em: <https://marketplace.visualstudio.com/items?itemName=Blackboxapp.blackbox>. Acesso em 28 de janeiro de 2023.
- [6] 2022. Captain Stack. Disponível em: <https://marketplace.visualstudio.com/items?itemName=captainstack.captain-stack>. Acesso em 28 de janeiro de 2023.
- [7] Muhammad Azeem Akbar, Arif Ali Khan, and Zhiqiu Huang. 2022. Multicriteria decision making taxonomy of code recommendation system challenges: A fuzzy-AHP analysis. *Information Technology and Management*, 1–17.
- [8] Ivanilse Calderon, Williamson Silva, and Eduardo Feitosa. 2021. Um Mapeamento Sistemático da Literatura sobre o uso de Metodologias Ativas durante o Ensino de Programação no Brasil. In *Anais do XXXII Simpósio Brasileiro de Informática na Educação*. SBC, 1152–1161.
- [9] Marcos Ronaldo Melo Cavalheiro, Gerson Battisti, Barbara Gündel, and Isabel Koltermann Battisti. 2022. GITHUB COPILOT E O FUTURO DA PROGRAMAÇÃO. *Salão do Conhecimento* 8, 8.
- [10] Bárbara Gabriela Sousa da Silva, Laysa Mabel de Oliveira Fontes, and Brígido Conrado de Brito Freitas. 2022. Autoavaliação na aprendizagem de programação introdutória: uma revisão sistemática da literatura Self-assessment in introductory programming learning: a systematic. *Brazilian Journal of Development* 8, 5, 32616–32643.
- [11] Faruk Demir. 2022. The effect of different usage of the educational programming language in programming education on the programming anxiety and achievement. *Education and Information Technologies* 27, 3, 4171–4194.
- [12] Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiao Cheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. 2020. CodeBERT: A Pre-Trained Model for Programming and Natural Languages.
- [13] Marko Gasparic and Andrea Janes. 2016. What recommendation systems for software engineering recommend: A systematic literature review. *Journal of Systems and Software* 113, 101–113.
- [14] Anabela Gomes, Joana Henriques, and Antonio Jose Mendes. 2008. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias* 1, 1, 93–103.
- [15] Akira Hanashiro. 2021. VS Code - O que é e por que você deve usar? Disponível em: <https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>. Acesso em 28 de janeiro de 2023.
- [16] Md Mazharul Islam and Razib Iqbal. 2020. SoCeR: A New Source Code Recommendation Technique for Code Reuse. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 1552–1557.
- [17] Hyeyoung Ko, Suyeon Lee, Yoonseo Park, and Anna Choi. 2022. A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. *Electronics* 11, 1.
- [18] Guilherme Medeiros Machado, Vinicius Maran, Gabriel Machado Lunardi, Leandro Krug Wives, and José Palazzo Moreira de Oliveira. 2021. AwARE: A Framework for Adaptive Recommendation of Educational Resources. *Computing* 103, 4, 675–705.
- [19] MACILON PEIXOTO et al. 2022. UMA REVISÃO BIBLIOGRÁFICA ACERCA DO USO DE FERRAMENTAS DIDÁTICAS NO ENSINO DE LÓGICA DE PROGRAMAÇÃO.
- [20] Martin P. Robillard, Walid Maalej, Robert J. Walker, and Thomas Zimmermann. 2014. *Recommendation Systems in Software Engineering*. Springer Publishing Company, Incorporated.
- [21] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.