

Validação e análise de um inventário de conceitos sobre programação introdutória

Ana Caroline R. Braz, Leandro
S. G. Carvalho, Elaine H. T.
Oliveira, David B. F. Oliveira
Universidade Federal do Amazonas
(UFAM)
Manaus – AM, Brasil
{ana.braz,galvao,elaine,david}@icomp.ufam.edu.br

Filipe Dwan Pereira
Universidade Federal de Roraima
(UFRR)
Boa Vista – RR, Brasil
filipe.dwan@ufrr.br

Roberto A. Bittencourt, Bianca
L. Santana
Universidade Estadual de Feira de
Santana (UEFS)
Feira de Santana – BA, Brasil
roberto@uefs.br
biancasantana.ls@gmail.com

Um inventário de conceitos (IC) é um conjunto de questões, de resposta discursiva ou de múltipla escolha, com o intuito de identificar as concepções errôneas (em inglês, *misconceptions*) dos alunos sobre determinado tópico ou assunto [4]. Desde que foram introduzidos na ciência no campo da Física [10], diversas outras áreas passaram a tentar replicar ICs para conhecer a evolução dos alunos em diversos tópicos, principalmente naqueles considerados fundamentais e complexos de serem entendidos em disciplinas ou cursos. Isso se deve principalmente ao fato de que ICs validados não são aplicáveis somente a um curso ou a uma instituição, mas possíveis de serem replicados independente do currículo, do curso, da instituição e dos métodos pedagógicos utilizados [4, 18, 24].

Apesar de serem amplamente conhecidos em diversos campos do conhecimento, os ICs em Computação ainda são poucos [3, 4, 11, 17, 18, 24]. Além disso, as metodologias já desenvolvidas foram validadas e/ou analisadas em contextos pouco diversos, o que dificulta sua replicação. Mais especificamente, no tópico de programação introdutória, não há instrumento validado para o português do Brasil e de uso aberto, no melhor do nosso conhecimento.

Note que a disciplina de introdução à computação (conhecidas na literatura como CS1) pode ser complexa para os alunos de graduação, já que regularmente ela demanda alta capacidade cognitiva de abstração dos problemas propostos [5, 6, 8, 13, 15, 19, 20, 22]. Isso vale tanto na área da computação, como em outros cursos de graduação das ciências exatas e engenharias em que CS1 é uma disciplina obrigatória [2, 7, 9, 21]. Destaca-se que os alunos de graduação de áreas que não tem computação como atividade fim, conhecidos como *non-CS-majors*, podem apresentar mais de um terço de reprovação em CS1 [1, 9, 21].

Com isso em vista, alguns ICs foram desenvolvidos na literatura para CS1 tendo como principal destaque o trabalho desenvolvido por [24], que criou o primeiro IC independente de linguagem de programação e adaptável a qualquer currículo pedagógico, passando pela definição dos conceitos-chave de CS1 e pela especificação do teste das questões. Apesar de seu trabalho ser notório, as questões desse IC não são abertas para reprodução, o que fez com que [18] desenvolvesse um trabalho de replicação dessas questões obtendo

resultados que sustentam sua validação. Embora o trabalho de [18] legitime bons resultados, como uma correlação positiva em relação ao IC de [24] e mesmo nível de dificuldade [18], os próprios autores destacam alguns pontos para avanço: o nível de dificuldade das questões, a fragilidade em validar o questionário, dentre outros.

Assim, percebe-se que é fundamental desenvolver ICs para as disciplinas introdutórias de programação, pois é notória a crescente dificuldade de abordagens educacionais para CS1 e avaliação dos alunos nessa disciplina [13, 17]. De fato, muitos estudantes finalizam a disciplina com pouco entendimento concreto sobre esses conceitos, o que os leva a ter as concepções errôneas [3, 14, 16, 22, 23]. Então, para além de conhecer tais concepções errôneas e aplicar metodologias e intervenções para que estas possam ser sanadas, os ICs também são de extrema necessidade para comparar abordagens de ensino de CS1 e medir o conhecimento dos professores [17, 18], sendo estas também razões válidas como motivação para este trabalho.

No cenário da Universidade Federal do Amazonas (UFAM), a disciplina CS1 é ofertada para 17 cursos presenciais de Ciências Exatas e Engenharias do campus em Manaus [1]. Essa disciplina está focada na solução de problemas em uma linguagem de programação de alto nível. Segundo dados do sistema acadêmico utilizado na instituição, de 2017 a 2019 foram ofertadas 18 turmas anualmente, envolvendo cerca de 600 matrículas em 2017 e em 2018, e 685 matrículas em 2019. A taxa de reprovação em 2017 foi de 54,3%; em 2018, foi de 51,2%; e, em 2019, aumentou para 60,6%, mostrando a grande dificuldade dos alunos nessa disciplina.

O objetivo principal da pesquisa é validar a tradução, para o português brasileiro, do inventário de conceitos de programação proposto por [18], utilizando a mesma metodologia, a fim de garantir a isomorfia dos instrumentos. Para tanto, primeiro houve a tradução do material para o português brasileiro. Logo após, efetuou-se a aplicação do questionário entre os alunos já aprovados na disciplina de CS1 no nível de graduação, por meio da técnica de *Think-aloud* [12]. Em seguida, foi realizada uma nova revisão, a fim de corrigir erros e mal-entendidos para assim aplicar em uma turma teste no CodeBench¹, uma plataforma de juiz online.

No momento, o questionário está sendo novamente revisado e adaptado para aplicação pré e pós-teste final nas turmas de CS1. Os resultados serão analisados por meio da Teoria de Resposta ao Item e Alpha de Cronbach, que são recomendadas pela literatura educacional para este fim [17].

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'21, Abril 26–30, 2021, Jataí, Goiás, Brasil (On-line)

© 2021 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

¹codebench.icomp.ufam.edu.br/

AGRADECIMENTOS

Os autores agradecem ao apoio e financiamento prestado pela Universidade Federal do Amazonas – UFAM por meio do Edital 081/2019 – PROPESP/UFAM, do Programa Institucional de Bolsas de Iniciação Científica (PIBIC). Além disso, esta pesquisa, realizada no âmbito do Projeto Samsung-UFAM de Ensino e Pesquisa (SUPER), de acordo com o artigo 48 do Decreto no 6.008/2006 (SUFRAMA), foi parcialmente financiada pela Samsung Electronics da Amazônia Ltda., nos termos de Lei Federal no 8.387/1991, mediante contrato 001/2020, firmado com a Universidade Federal do Amazonas e a FAEPI, Brasil. Contamos também com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e do Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (Processo 308513/2020-7).

REFERÊNCIAS

- [1] Arthur Alves, Leandro Silva Galvão de Carvalho, Elaine Oliveira, and David Fernandes. 2019. Análise comportamental em juízes online para predição do desempenho final de alunos em disciplinas de computação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 30, 1906.
- [2] Ada Araujo, Daniel Lopes Zordan Filho, Elaine Harada Teixeira de Oliveira, Leandro Silva Galvão de Carvalho, Filipe Dwan Pereira, and David Braga Fernandes de Oliveira. 2021. Mapeamento e análise empírica de misconceptions comuns em avaliações de introdução à programação. In *Anais do Simpósio Brasileiro de Educação em Computação*. SBC, 123–131.
- [3] Ricardo Caceffo, Pablo Frank-Bolton, Renan Souza, and Rodolfo Azevedo. 2019. Identifying and Validating Java Misconceptions Toward a CS1 Concept Inventory. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (Aberdeen, Scotland Uk) (ITICSE '19)*. Association for Computing Machinery, New York, NY, USA, 23–29. <https://doi.org/10.1145/3304221.3319771>
- [4] Ricardo Caceffo, Steve Wolfman, Kellogg S Booth, and Rodolfo Azevedo. 2016. Developing a computer science concept inventory for introductory programming. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 364–369.
- [5] Hermino Barbosa de Freitas Júnior, Filipe Dwan Pereira, Elaine Harada Teixeira de Oliveira, David Braga Fernandes de Oliveira, and Leandro Silva Galvão de Carvalho. 2020. Recomendação Automática de Problemas em Juizes Online Usando Processamento de Linguagem Natural e Análise Dirigida aos Dados. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*. SBC, 1152–1161.
- [6] Marcos Avner Pimenta de Lima Lima, Leandro Silva Galvão de Carvalho, Elaine Harada Teixeira de Oliveira, David Braga Fernandes de Oliveira, and Filipe Dwan Pereira. 2021. Uso de atributos de código para classificação da facilidade de questões de codificação. In *Anais do Simpósio Brasileiro de Educação em Computação*. SBC, 113–122.
- [7] Ingrid Lima dos Santos, David Braga Fernandes Oliveira, Leandro Silva Galvão de Carvalho, Filipe Dwan Pereira, and Elaine Harada Teixeira de Oliveira. 2020. Tempos de Transição em Estados de Corretude e Erro como Indicadores de Desempenho em Juizes Online. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*. SBC, 1283–1292.
- [8] Samuel Fonseca, Elaine Oliveira, Filipe Pereira, David Fernandes, and Leandro Silva Galvão de Carvalho. 2019. Adaptação de um método preditivo para inferir o desempenho de alunos de programação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 30, 1651.
- [9] Samuel C Fonseca, Filipe Dwan Pereira, Elaine HT Oliveira, David BF Oliveira, Leandro SG Carvalho, and Alexandra I Cristea. 2020. Automatic Subject-based Contextualisation of Programming Assignment Lists. EDM.
- [10] David Hestenes, Malcolm Wells, and Gregg Swackhamer. 1992. Force concept inventory. *The physics teacher* 30, 3 (1992), 141–158.
- [11] Wanda M Kunkle and Robert B Allen. 2016. The impact of different teaching approaches and languages on student learning of introductory programming concepts. *ACM Transactions on Computing Education (TOCE)* 16, 1 (2016), 1–26.
- [12] Clayton Lewis. 1982. *Using the "thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY.
- [13] Marcos Lima, Leandro Silva Galvão de Carvalho, Elaine Harada Teixeira de Oliveira, David Braga Fernandes Oliveira, and Filipe Dwan Pereira. 2020. Classificação de dificuldade de questões de programação com base em métricas de código. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*. SBC, 1323–1332.
- [14] Raymond Lister, Elizabeth S Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik Moström, Kate Sanders, Otto Seppälä, et al. 2004. A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin* 36, 4 (2004), 119–150.
- [15] Andrew Luxton-Reilly, Ibrahim Albluwi, Brett A Becker, Michail Giannakos, Amruth N Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 2018. Introductory programming: a systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*. 55–106.
- [16] Michael McCracken, Vicki Almstrum, Danny Diaz, Mark Guzdial, Dianne Hagan, Yifat Ben-David Kolikant, Cary Laxer, Lynda Thomas, Ian Utting, and Tadeusz Wilusz. 2001. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In *Working group reports from ITICSE on Innovation and technology in computer science education*. 125–180.
- [17] Andreas Mühlhling, Alexander Ruf, and Peter Hubwieser. 2015. Design and first results of a psychometric test for measuring basic programming abilities. In *Proceedings of the workshop in primary and secondary computing education*. 2–10.
- [18] Miranda C Parker, Mark Guzdial, and Shelly Engleman. 2016. Replication, validation, and use of a language independent CS1 knowledge assessment. In *Proceedings of the 2016 ACM conference on international computing education research*. 93–101.
- [19] Filipe Dwan Pereira, Linnick Maciel de Souza, Elaine Harada Teixeira de Oliveira, David Braga Fernandes de Oliveira, and Leandro Silva Galvão de Carvalho. 2020. Predição de desempenho em ambientes computacionais para turmas de programação: um Mapeamento Sistemático da Literatura. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*. SBC, 1673–1682.
- [20] Filipe Dwan Pereira, Samuel C Fonseca, Elaine HT Oliveira, David BF Oliveira, Alexandra I Cristea, and Leandro SG Carvalho. 2020. Deep learning for early performance prediction of introductory programming students: a comparative and explanatory study. *Brazilian journal of computers in education*. 28 (2020), 723–749.
- [21] Filipe D Pereira, Elaine HT Oliveira, David BF Oliveira, Alexandra I Cristea, Leandro SG Carvalho, Samuel C Fonseca, Armando Toda, and Seiji Isotani. 2020. Using learning analytics in the Amazonas: understanding students' behaviour in introductory programming. *British Journal of Educational Technology* (2020).
- [22] A. V. Robins. 2019. Novice programmers and introductory programming. In *The Cambridge Handbook of Computing Education Research*. Cambridge University Press, Cambridge, Chapter 12, 327–376.
- [23] Juha Sorva. 2018. Misconceptions and the Beginner Programmer. *Computer Science Education: Perspectives on Teaching and Learning in School* (2018), 171.
- [24] Allison Elliott Tew, W Michael McCracken, and Mark Guzdial. 2005. Impact of alternative introductory courses on programming concept understanding. In *Proceedings of the first international workshop on Computing education research*. 25–35.