

Participação computacional na aprendizagem de uma linguagem de programação no ensino superior: um estudo multicaseos

Computational participation in programming language learning in higher education: a multi-case study

Gabryella Rodrigues¹, Ana Francisca Monteiro², António José Osório³

Centro de Investigação em Educação (CIEE), Instituto de Educação
Universidade do Minho (UMinho)

4710-057 Braga, Portugal

¹gabryella.rocha@gmail.com, ²anafmonteiro@gmail.com, ³ajosorio@ie.uminho.pt

Resumo — Aprender a programar envolve uma variedade de atividades cognitivas e representações mentais. Face à complexidade do problema e à intenção de apreender as particularidades e singularidades da experiência do estudante na aprendizagem da programação em contexto formal e não formal, esta pesquisa adota uma abordagem qualitativa, a fim de caracterizar a complexa inter-relação da participação computacional no processo de aprendizagem da programação, sem a isolar do contexto natural em que se desenvolve, procurando atingir sua compreensão a partir da observação de múltiplos casos.

Palavras Chave – programação introdutória; programadores iniciantes; aprendizado da programação; participação computacional; estudo de caso.

Abstract — Learning to program involves a variety of cognitive activities and mental representations. Given the problem's complexity and the intention to apprehend the particularity and peculiarity of student's experience in learning programming informal and non-formal contexts, this research adopts a qualitative approach to characterize the complex interrelationship between computational participation and the learning process of programming, without isolating it from the natural context in which it develops, seeking to achieve its understanding from the observation of multiple cases.

Keywords – introductory programming; novice programming; learning programming; computational participation; case study.

I. INTRODUÇÃO

Adquirir e desenvolver aptidão na codificação de um *software* envolve variadas atividades cognitivas e representações mentais [1]. Este campo é alvo de diversas pesquisas [2], que se focam em diferentes conceitos, práticas e perspectivas [3, 4, 5]. Apesar do aumento da discussão e dos avanços acerca de

métodos e ferramentas usadas no processo de ensino e aprendizado da programação, a literatura relata [6,7] uma predominância da combinação do ensino tradicional com práticas de codificação individuais em laboratório, nomeadamente na unidade curricular (UC) de programação introdutória, que se detêm em explorar conceitos e habilidades técnicas [8].

Mover a concepção individualista da programação para uma concepção designada de participação computacional [9] que inclua um maior foco em dimensões sociais e criativas, com o objetivo de reformular o conceito do pensamento computacional de Jeannette Wing [10], a fim de ressignificar o aprendizado da programação. Kafai & Burke [9] descrevem três dimensões (cognitiva, pessoal e política) para a reformulação do conceito inicial de Wing [10] a partir de diferentes perspectivas de aprendizagem [11].

A primeira propõe o desenvolvimento de habilidades e competências técnicas ligadas à programação, enfatizando o aprendizado da sintaxe e semântica de uma linguagem de programação [11]. Essa perspectiva cognitiva é a que prevalece na última recomendação curricular proposta pela ACM/IEEE-CS aos cursos de computação [7], entretanto fornece pouca atenção à forma como o aprendizado é inserido em contextos sociais e culturais [12].

A segunda dimensão enfatiza como os alunos podem desenvolver habilidades técnicas através da construção em equipe de *software* partilhável. Esta dimensão vê o pensamento computacional como um veículo para a expressão pessoal e a participação em comunidades *online*, na qual aprender é tornar-se membro de uma comunidade que constrói e compartilha metas e valores [12]; por sua vez, a dimensão política discute um enquadramento crítico do pensamento computacional, buscando contextualizar as práticas cognitivas e situando-as no contexto do aluno, a fim de que esses sejam ensinados a usar a computação para o bem social.

Diante desse cenário, esta pesquisa irá investigar através do estudo de casos específicos, como as dimensões da participação computacional poderão influenciar nos processos de aprendizagem da programação introdutória, promovendo um conjunto de habilidades essenciais.

Para tanto, foram estabelecidos os seguintes objetivos específicos: Obj_esp_01 - identificar a UC de programação introdutória; Obj_esp_02 - verificar as habilidades exigidas aos alunos no aprendizado da programação; Obj_esp_03 - identificar as metodologias usadas no ensino-aprendizado da UC; Obj_esp_04 - observar nove casos de aprendizado da programação introdutória em contexto formal e não formal; Obj_esp_05 - verificar como os alunos operacionalizam a construção, compreensão e correção de um algoritmo; Obj_esp_06 - analisar os dados coletados nas entrevistas, questionários e análise de documentos, em função da aplicação da taxonomia revista de Bloom e Obj_esp_07 - esboçar um modelo que explique como a participação computacional influencia na qualidade do processo de aprendizado da programação de alunos em diferentes contextos.

Pretende-se, assim, encontrar e aprofundar conhecimento que possa ser mobilizado para apoiar os estudantes a adquirirem habilidades e competências que lhes permitam percursos bem sucedidos e coerentes com as suas expectativas pessoais, profissionais e sociais, independentes do contexto em que estão inseridos.

II. DESENHO DO ESTUDO DE CASO

Face à complexidade do problema e à intenção de apreender a experiência do estudante acerca da programação, esta pesquisa adota uma abordagem

qualitativa, de paradigma fenomenológico-interpretativo, através do estudo de “casos”, com o propósito de atingir o entendimento das particularidades do problema descrito.

À vista disso esclarece-se que o interesse por estudar o fenômeno da aprendizagem de programação é resultado tanto das discussões que a literatura reflete, quanto da percepção que a primeira autora possui acerca do problema, evidenciado pela sua experiência profissional.

O projeto consiste num estudo de caso múltiplo exploratório baseado nas diretrizes de Robert Yin [13]. Serão estudados, sem controle ou intervenção, casos de aprendizado de programação na UC de programação introdutória, tema atual e com relevante discussão na literatura.

O processo de aprendizado da programação será observado em seis unidades curriculares (UC) de programação introdutória, ofertadas em contexto formal, e três cursos de programação ofertados em contexto não formal, em países de Língua Portuguesa, designadamente Angola, Brasil, Moçambique e Portugal. Nesses países, serão estudados três cursos de programação desenvolvidos em contexto não formal, em outras palavras, cursos que se configuram como de curta duração, ofertados como acréscimo ou complemento à educação formal. No campo da computação esses cursos são identificados através da terminologia *bootcamp* e são caracterizados como um treinamento intensivo e específico que une prática e teoria através de dinâmicas imersivas e produtivas [14].

Dos cursos formais, ofertados por instituições de educação superior, três serão observados no Brasil e um em Angola, um em Moçambique e um em Portugal; quanto aos cursos não formais de curta duração, um será observado um em África (Angola ou Moçambique), um no Brasil e um em Portugal.

A definição da quantidade de cursos formais a serem observados foi estabelecida de forma proporcional à população de cada país. Além da expressão em língua portuguesa, o critério usado para a escolha dos países e instituições para recolha dos dados, deve-se à facilidade na abertura do campo de investigação. Existindo outros países lusófonos, não poderão ser incluídos por limitações de tempo disponível para a realização da pesquisa.

Para além desses critérios, nesse conjunto de países selecionados há uma comunidade alargada de profissionais e estudantes de computação que fazem uso de uma linguagem de programação e do português como primeira língua e que podem ofertar uma quantidade de dados que possibilitem a transferência dos conhecimentos adquiridos para outros contextos.

III. PREPARAÇÃO PARA COLETA DE DADOS

Antes de iniciar a fase de coleta de dados foi realizada uma revisão sistemática da literatura em junho de 2021, com a análise de 33 artigos que exploram o ensino e a aprendizagem de programação na UC de programação introdutória, a fim de coletar sistematicamente dados recentes disponíveis em bases de dados científicas em resposta aos objetivos específicos Obj_esp_01, Obj_esp_02 e Obj_esp_03 (secção I).

Foi revelado uma nomenclatura padrão que nomeia a UC que aborda os conceitos fundamentais da programação em cursos de computação, os dados revelaram que a maioria dos estudos mencionam o nome “*introductory program course*”. Quanto ao quesito abordagem, as pesquisas analisadas apresentaram uma certa diversidade, umas focaram-se nos conceitos fundamentais do desenvolvimento de software associados a uma determinada linguagem de programação, outras deixaram de lado a ênfase em programação e forneceram uma introdução mais ampla aos conceitos, com ênfase na algoritmia.

Quanto à identificação das habilidades envolvidas na programação, notou-se uma preocupação em maior proporção ao estímulo das habilidades técnicas. Por outro lado, as habilidades como pensamento crítico, comunicação, criatividade, persistência, participação voluntária, perseverança, confiança e trabalho em equipe emergiram em alguns dos artigos analisados. Esses dados levam a crer que o conhecimento e as habilidades, sejam elas técnicas ou sociais, são claramente identificados, transferidos e alcançados através de práticas específicas. Contudo, as pesquisas ainda não identificam essa tríade como parte de um modelo em potencial.

Por último, em resposta ao objetivo específico Obj_esp_03, identificou-se que 54% dos estudos fizeram referência ao uso de métodos inovadores no

processo de ensino e aprendizagem de uma linguagem de programação, entretanto a maioria descreveu relatos de um único experimento.

Com esse estudo foi possível cobrir trabalhos publicados recentemente, obter uma visão geral do processo de ensino-aprendizado na UC de programação introdutória e identificar se as pesquisas desenvolvidas pelas universidades a nível mundial, estão em sintonia com as propostas feitas pelo grupo ACM/IEEE acerca das recomendações curriculares para os cursos de computação.

IV. COLETA E ANÁLISE

O grupo de indivíduos que serão estudados nesta pesquisa será formado por quem tem um discurso sobre aprendizagem da programação: os alunos. A princípio, a definição dos participantes será de forma intencional, seguindo alguns critérios.

Para estarem aptos a serem selecionados, esses alunos deverão estar cursando a UC de programação introdutória ou participando de um *bootcamp* de programação. A faixa etária e o sexo são características irrelevantes, mas deverão estar frequentando o curso ou a UC pela primeira vez. Os estudantes serão convidados a participar da pesquisa por *e-mail*. Os três primeiros alunos de cada caso a preencherem o formulário de aceite, estarão inseridos oficialmente na investigação. Os demais respondentes serão cadastrados em uma lista de espera e serão contactados caso haja alguma desistência de participantes durante o processo de coleta de dados.

Como exigência de toda pesquisa que envolva seres humanos, e antes do início da fase de coleta de dados, será necessário que o projeto seja autorizado pelo conselho de ética da universidade do Minho, além de se fazer necessário o preenchimento dos Termos de Consentimento Informado, Livre e Esclarecido, Termos de Áudio e Termos de Imagem pelos alunos participantes ou responsáveis.

E a fim de obter informações suficientes e pertinentes, será usada a técnica da triangulação [13] como estratégia metodológica para atribuir validade e fidedignidade a este estudo de caso, pois assegurará o cruzamento das informações colhidas através dos vários instrumentos: questionário fechado, entrevista aberta, recolha de verbalizações e análise documental.

O questionário com perguntas fechadas, será usado para perceber quem são os alunos envolvidos na pesquisa, propiciando a identificação das suas características, comportamentos e expectativas, nomeadamente ao aprendizado da programação.

Com o objetivo de evidenciar as dimensões da participação computacional no processo de aprendizado de uma linguagem de programação e detalhes sobre construção, compreensão e correção de um algoritmo, prevê-se a realização de dois encontros com cada participante.

O primeiro encontro propiciará a oportunidade de conhecer o participante com profundidade através da realização de uma entrevista aberta, que será usada para validar os dados coletados no questionário e identificar particularidades do curso/UC que frequentam, dos métodos de ensino e das habilidades exploradas. No segundo encontro, os alunos deverão construir, ler e corrigir um código, para que se possa observar como esses indivíduos operacionalizam o design de um algoritmo, que estratégias utilizam para resolver problemas e como lidam com os erros (fracassos). A técnica de coleta de dados será a de recolha de verbalizações, uma estratégia usada em pesquisas qualitativas para aceder e analisar os processos mentais ativados na resolução de problemas [15].

O ambiente onde os dados serão coletados será através de videoconferência, que proporcionará o devido registro de áudio e vídeo.

Essa fase será concluída com a análise documental do plano de estudos dos cursos/UC. Logo, identificar quais são as habilidades prévias necessárias e as competências que deverão ser adquiridas ao final da UC/curso, bem como verificar a metodologia usada pelos professores.

Os dados colhidos serão processados usando o software de análise de dados *NVivo*, sob a concepção do método de análise de conteúdo. Por fim, será esboçado um modelo que explique quais estratégias de aprendizado são mais eficientes para o desenvolvimento da participação computacional. Espera-se que o entendimento do contexto, das ferramentas, das linguagens, dos paradigmas e das práticas de aprendizado usadas pelos alunos no desenvolvimento de códigos, como eles o fazem e por quê, possa ajudar-nos a perceber novas formas de pensar a programação.

AGRADECIMENTOS

Este trabalho é apoiado pelo CIEd - Centro de Investigação em Educação, Instituto de Educação, Universidade do Minho e financiado pela agência portuguesa FCT - Fundação para a Ciência e a Tecnologia, bolsa nº2021.07850.BD.

REFERÊNCIAS BIBLIOGRÁFICA

- [1] Fedorenko, E., Ivanova, A., Dhamala, R., & Bers, M. U. (2019). The Language of Programming: A Cognitive Perspective. *Trends in Cognitive Sciences*, 23(7), 525–528. <https://doi.org/10.1016/j.tics.2019.04.010>.
- [2] Becker, B. A., & Quille, K. (2019). 50 Years of CS1 at SIGCSE: A Review of the Evolution of Introductory Programming Education Research. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 338–344. <https://doi.org/10.1145/3287324.3287432>.
- [3] Figueiredo, J., & García-Peñalvo, F. (2021). Teaching and Learning Strategies for Introductory Programming in University Courses. *Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21)*, 746–751. <https://doi.org/10.1145/3486011.3486540>.
- [4] Gomes, A., Marcelino, M. J., Correia, F., & Mendes, A. J. (2020). Study methods in introductory programming courses. *2020 IEEE Global Engineering Education Conference (EDUCON)*, 898–904. <https://doi.org/10.1109/EDUCON45650.2020.9125228>.
- [5] Sobral, S. R. (2021). Strategies on Teaching Introducing to Programming in Higher Education. Em Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, & A. M. Ramalho Correia (Orgs.), *Trends and Applications in Information Systems and Technologies* (p. 133–150). Springer International Publishing.
- [6] Brown, D. W., Ghafoor, S. K., & Canfield, S. (2018). Instruction of Introductory Programming Course Using Multiple Contexts. *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 147–152. <https://doi.org/10.1145/3197091.3197105>.
- [7] Force, C. T. (2020). *Computing Curricula 2020: Paradigms for Global Computing Education*. Association for Computing Machinery.
- [8] Burbekova, S. (2021). Soft Skills as the Most In-Demand Skills of Future IT Specialists. *2021 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, 1–5. <https://doi.org/10.1109/SIST50301.2021.9465935>.
- [9] Kafai, Y. B., & Burke, Q. (2017). Computational Participation: Teaching Kids to Create and Connect Through Code. In P. J. Rich & C. B. Hodges (Orgs.), *Emerging Research, Practice, and Policy on Computational Thinking* (p. 393–405). Springer International Publishing. https://doi.org/10.1007/978-3-319-52691-1_24.
- [10] Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- [11] Kafai, Y. B., Proctor, C., & Lui, D. A. (2019). Framing Computational Thinking for Computational Literacies in K-12 Education. Em *Proceedings of the Weizenbaum Conference 2019 “Challenges of Digital Inequality—Digital Education, Digital Work, Digital Life”* (p. 6). <https://doi.org/10.34669/wi.cp/2.21>.
- [12] Kafai, Y. B., & Proctor, C. (2022). A Reevaluation of Computational Thinking in K–12 Education: Moving Toward Computational Literacies. *Educational Researcher*, 51(2), 146–151. <https://doi.org/10.3102/0013189X2111057904>.
- [13] Yin, R. K. (2015). *Estudo de Caso-: Planejamento e métodos*. Bookman editora.
- [14] Thayer, K., & Ko, A. J. (2017). Barriers faced by coding Bootcamp students. *Proceedings of the 2017 ACM Conference on International Computing Education Research*, 245–253.
- [15] Newell, A., Simon, H. A., & others. (1972). *Human problem solving* (Vol. 104). Prentice-hall Englewood Cliffs, NJ.