

Dificuldades e desafios do processo de aprendizagem de algoritmos e programação no ensino superior: uma revisão sistemática de literatura

Difficulties and challenges in the learning process of algorithms and programming in higher education: a systematic literature review

Dificultades y desafíos en el proceso de aprendizaje de algoritmos y programación en la educación superior: una revisión sistemática de la literatura

Recebido: 14/10/2020 | Revisado: 22/10/2020 | Aceito: 26/10/2020 | Publicado: 27/10/2020

Ceres Germanna Braga Morais

ORCID: <https://orcid.org/0000-0001-6270-2508>

Universidade do Estado do Rio Grande do Norte, Brasil

E-mail: ceresmoraes@uern.br

Francisco Milton Mendes Neto

ORCID: <https://orcid.org/0000-0003-3462-3364>

Universidade Federal Rural do Semi-Árido, Brasil

E-mail: miltonmendes@ufersa.edu.br

António José Meneses Osório

ORCID: <https://orcid.org/0000-0002-9931-1962>

Universidade do Minho, Portugal

E-mail: ajosorio@ie.uminho.pt

Resumo

Aprender a programar geralmente é um desafio para os estudantes: cada aluno possui suas dificuldades individuais e os professores precisam identificá-las para dar o suporte necessário a seus alunos. Neste sentido, este artigo apresenta parte dos resultados obtidos em uma Revisão Sistemática de Literatura sobre o processo de aprendizagem de algoritmos e programação. O principal objetivo do trabalho aqui apresentado é apontar as principais dificuldades e desafios vividos pelos alunos do nível superior em disciplinas de Algoritmos e programação, bem como o que estas dificuldades podem acarretar aos alunos ao longo do seu percurso acadêmico. A escolha pelo método da pesquisa se deu pelo fato de que a Revisão Sistemática de Literatura apresenta um tipo de estudo cuja metodologia é bem definida, sendo capaz de avaliar a pesquisa de maneira criteriosa, confiável e objetiva. Para este estudo, a

revisão foi realizada em quatro bases de dados: Web of Science, Scopus, IEEE Xplore e RCAAP. Para tanto, foi definida uma questão de revisão, as palavras-chave, a equação necessária ao estudo, bem como o protocolo de revisão. Inicialmente foram encontrados 350 artigos, dos quais 169 artigos foram pré-selecionados. Após a aplicação dos critérios de inclusão e exclusão, 33 artigos foram selecionados para a análise e desenvolvimento deste trabalho. Com a análise, pudemos verificar e categorizar as dificuldades e desafios enfrentados pelos estudantes durante o processo de aprendizagem de algoritmos e programação. Com os resultados apresentados, este artigo pode contribuir com uma visão geral capaz de servir de ponto de partida para outras pesquisas, bem como para auxiliar docentes e estudantes na percepção destas dificuldades.

Palavras-chave: Aprendizagem de programação; Revisão sistemática de literatura; Algoritmos; Desafios.

Abstract

Learning to program is often a challenge for students: each student has their individual difficulties and teachers need to identify them in order to provide the necessary support to their students. In this sense, this article presents part of the results obtained in a Systematic Literature Review on the process of learning algorithms and programming. The main objective of the work presented here is to point out the main difficulties and challenges experienced by higher education students in Algorithms and Programming disciplines, as well as what these difficulties can bring to students along their academic path. The choice of the research method was because the Systematic Literature Review presents a type of study whose methodology is well defined, being able to evaluate the research in a careful, reliable and objective way. For this study, the review was conducted in four databases: Web of Science, Scopus, IEEE Xplore and RCAAP. For that, a review question was defined, the keywords, the necessary equation for the study, as well as the review protocol. Initially, 350 articles were found, of which 169 articles were pre-selected. After applying the inclusion and exclusion criteria, 33 articles were selected for the analysis and development of this work. With the analysis, we were able to verify and categorize the difficulties and challenges faced by students during the process of learning algorithms and programming. With the results presented, this article can contribute with an overview capable of serving as a starting point for other research, as well as to assist teachers and students in the perception of these difficulties.

Keywords: Programming learning; Systematic literature review; Algorithms; Challenges.

Resumen

Aprender a programar es a menudo un desafío para los estudiantes: cada estudiante tiene sus dificultades individuales y los maestros deben identificarlas para brindar el apoyo necesario a sus estudiantes. En este sentido, este artículo presenta parte de los resultados obtenidos en una Revisión Sistemática de la Literatura sobre el proceso de aprendizaje de algoritmos y programación. El principal objetivo del trabajo aquí presentado es señalar las principales dificultades y desafíos que experimentan los estudiantes de educación superior en disciplinas de Algoritmos y Programación, así como lo que estas dificultades pueden traer a los estudiantes a lo largo de su carrera académica. La elección del método de investigación se debió a que la Revisión Sistemática de Literatura presenta un tipo de estudio cuya metodología está bien definida, pudiendo evaluar la investigación de manera cuidadosa, confiable y objetiva. Para este estudio, la revisión se realizó en cuatro bases de datos: Web of Science, Scopus, IEEE Xplore y RCAAP. Para ello, se definió una pregunta de revisión, las palabras clave, la ecuación necesaria para el estudio, así como el protocolo de revisión. Inicialmente, se encontraron 350 artículos, de los cuales se preseleccionaron 169 artículos. Luego de aplicar los criterios de inclusión y exclusión, se seleccionaron 33 artículos para el análisis y desarrollo de este trabajo. Con el análisis pudimos verificar y categorizar las dificultades y desafíos que enfrentan los estudiantes durante el proceso de aprendizaje de algoritmos y programación. Con los resultados presentados, este artículo puede aportar un panorama que sirva de punto de partida para otras investigaciones, así como para ayudar a profesores y estudiantes en la percepción de estas dificultades.

Palabras clave: Aprendizaje de programación; Revisión sistemática de la literatura; Algoritmos; Desafíos.

1. Introdução

O ensino de algoritmos e programação está inserido nas diretrizes curriculares dos cursos de Computação e áreas correlatas (Rum & Ismail, 2017). Embora estas disciplinas sejam obrigatórias nestes cursos, aprender uma linguagem de programação é uma tarefa difícil e desafiadora para alguns estudantes (Saito, Washizaki & Fukuzawa, 2016). Isso por que cada aluno possui suas dificuldades individuais, seu ritmo de aprendizagem, seus interesses e motivações, de forma que os professores precisam identificar as características de seus alunos e as dificuldades por eles enfrentadas para que consigam dar o suporte devido (Kawagush et al, 2019), de maneira a garantir um melhor desempenho de cada um e da turma como um

todo. Vários são os estudos, desenvolvidos nos últimos anos, a fim de verificar as dificuldades enfrentadas pelos alunos e o que estas dificuldades acarretam durante seu percurso acadêmico.

Para sedimentar propostas educacionais voltadas para a aprendizagem de algoritmos e programação, é importante mapear esforços e resultados de produções científicas que estão sendo desenvolvidas. Para tanto foi realizada uma Revisão Sistemática de Literatura (RSL), de trabalhos desenvolvidos entre os anos 2000 e 2020, representando o Século XXI, com o objetivo de analisar artigos que tragam metodologias, estratégias e ferramentas para a aprendizagem de algoritmos e programação para o ensino superior. Este artigo apresenta parte dos resultados obtidos na RSL tendo como foco apresentar os resultados da Questão de Pesquisa sobre os desafios e dificuldades enfrentadas pelos estudantes no processo de aprendizagem de algoritmos e programação, assim como o que estas dificuldades podem acarretar no percurso acadêmico destes alunos.

Para melhor apresentação da pesquisa, este documento está organizado da seguinte maneira: a Seção 2 apresenta o método de pesquisa utilizado; a Seção 3 apresenta os resultados e discussões em torno da questão das duas questões de pesquisa; e a Seção 4 apresenta as considerações finais do trabalho.

2. Metodologia

O presente trabalho apresenta parte dos resultados de uma RSL que teve como objetivo investigar trabalhos científicos que apresentem metodologias, estratégias e ferramentas utilizadas na aprendizagem de algoritmos e programação, dentre os anos de 2000 e 2020. Esse período de tempo foi escolhido para poder representar pesquisas desenvolvidas ao longo do século XXI, para verificar a evolução das abordagens adotadas, e também contemplar estudos mais recentes. Para isso, optou-se por desenvolver uma RSL que, de acordo com Faria (2014):

“Caracteriza-se por empregar uma metodologia de pesquisa com rigor científico e de grande transparência, cujo objetivo visa minimizar o enviesamento da literatura, na medida em que é feita uma recolha exaustiva dos textos publicados sobre o tema em questão. Assim, é basilar o objetivo de estruturar todos os procedimentos de forma a garantir a qualidade das fontes, logo pela definição de uma equação de pesquisa, de critérios de inclusão e exclusão e de todas as normas que julguem convenientes para o caso. A credibilidade da pesquisa será, desta forma, proporcional ao grau de como se estabelecem as regras, pela possibilidade de replicação do processo” (Faria, 2014, p. 21).

A metodologia utilizada neste trabalho foi baseada nas diretrizes propostas por Kitchenham *et al* (2009), que definem a RSL como uma pesquisa em profundidade de um fenômeno de interesse que produz resultados específicos e detalhados por meio da análise de conteúdo e qualidade do material pesquisado. Tendo em vista o objetivo principal desta revisão, a saber “Como é realizado o processo de aprendizagem de algoritmos e programação no nível superior de ensino”, foi elaborada uma Questão de Pesquisa (QP), a qual guia todo o desenvolvimento do trabalho.

Ao longo que a pesquisa foi sendo desenvolvida, surgiram outras quatro questões que serviram de apoio para responder questionamentos que foram levantados ao longo das análises dos dados obtidos, e que se relacionam entre si, às quais chamaremos de Questão de Apoio (QA), das quais duas fazem parte do escopo deste artigo. A Tabela 1 apresenta a QP e as duas QAs que compõem este trabalho.

Tabela 1. Questão de pesquisa e Questões de Apoio.

QP	Quais abordagens (metodologias, estratégias, ferramentas) estão sendo propostas para a aprendizagem de algoritmos e programação no nível superior de ensino?
QA1	Quais as principais dificuldades enfrentadas pelos alunos durante a aprendizagem de programação?
QA2	O que as dificuldades enfrentadas pelos alunos podem desencadear durante seu percurso acadêmico?

Fonte: Elaborado pelos autores.

As questões apresentadas na Tabela 1 deixam claro que, além de verificarmos as metodologias, estratégias e ferramentas que foram propostas e utilizadas para a aprendizagem de algoritmos e programação, ao longo do Século XXI, ao desenvolver a RSL também pudemos verificar e analisar as principais dificuldades enfrentadas pelos alunos, durante o processo de aprendizagem de algoritmos e programação; e o que estas dificuldades desencadeiam ao aluno durante seu percurso acadêmico.

Para chegarmos aos dados da nossa RSL, desenvolvemos uma *string* de busca, a qual foi adaptada de acordo com cada base escolhida. No entanto, de forma geral, o objetivo da *string* era buscar trabalhos científicos que abordassem a temática de aprendizagem de programação, ensino de programação, e também as metodologias, ferramentas e estratégias de ensino e de aprendizagem. Para a realização das buscas, configuramos os mecanismos de forma que estas ocorressem no título dos artigos. Para efeito de início da pesquisa, a *string* para a QP é:

((“Programming learning”) OR (“Programming teaching”)

AND

((“method”) OR (“technique”) OR (“strategy”) OR (“tool”) OR (“methodology”)OR
 (“practice”)OR (“technology”)))

Para o desenvolvimento da RSL realizou-se pesquisas em quatro bases de dados distintas, a saber: Web of Science, Scopus, IEEE Xplore e Rcaap. Os artigos encontrados na busca em cada uma das quatro bases escolhidas foram então analisados segundo os critérios de inclusão (CI) e critérios de exclusão (CE) definidos no protocolo dessa RSL, os quais estão apresentados nas Tabelas 2 e 3, respectivamente.

Tabela 2. Critérios de Inclusão.

Critério de Inclusão	Descrição
CI1	Estudos que apresentem revisão por pares (<i>peer review</i>)
CI2	Trabalhos nas línguas portuguesa ou inglesa
CI3	Artigos com resumo
CI4	Todas as áreas de pesquisa
CI5	Artigos, documentos de conferências, atas.

Fonte: Elaborado pelos autores.

Tabela 3. Critérios de Exclusão.

Critério de Exclusão	Descrição
CE1	Artigos que estejam fora do intervalo de tempo dos anos 2000 a 2020
CE2	Artigos que não apresentem pesquisa voltada para a aprendizagem de algoritmos e programação no nível superior
CE3	Artigos cuja pesquisa seja voltada para o ensino e/ou aprendizagem de alguma linguagem de programação específica
CE4	Artigos que estejam duplicados (manter o mais recente)
CE5	Trabalhos sem resultado ou sem metodologia

Fonte: Elaborado pelos autores.

A primeira etapa de seleção dos artigos baseou-se nos títulos e palavras-chaves, de forma que foram removidos os estudos que não indicavam ligação com o objeto de estudo. A segunda etapa de análise envolveu a seleção dos artigos com base na leitura de seu resumo. A

terceira etapa da análise envolveu a seleção dos artigos com base na leitura completa dos mesmos. Por fim, para o tratamento dos dados, foram definidos os seguintes Critérios de Qualidade (CQ), para os quais foram determinados pesos, e que devem ser aplicados a cada artigo, após a realização da terceira etapa de análise. A Tabela 4 apresenta os Critérios de Qualidade desenvolvidos para essa RSL.

Tabela 4. Critérios de Qualidade.

Critério de Qualidade	Descrição	Não	Parcialmente	Sim
CQ1	O trabalho apresenta uma metodologia/técnica/ferramenta de ensino de programação?	0	0,5	1
CQ2	A metodologia/ferramenta/técnica foi testada e validada?	0	0,5	1
CQ3	O trabalho apresenta fatores que podem influenciar no aprendizado de algoritmos?	0	0,5	1
CQ4	O trabalho apresenta as competências/conhecimentos/aptidões que um aluno deve ter para aprender algoritmos?	0	0,5	1
CQ5	O trabalho apresenta as dificuldades em aprender algoritmos?	0	0,5	1

Fonte: Elaborado pelos autores.

Para efeitos desta pesquisa, com a aplicação dos CQ ao artigo pode ir de 0 (não contempla nenhum dos critérios) a 5 (contempla todos os critérios). Foram considerados os trabalhos que obtiveram pelo menos 3 pontos, de forma que pelo menos um dos critérios de qualidade seja contemplado em sua totalidade.

Antes de apresentar os resultados da pesquisa, é importante destacar o uso de critérios de validade metodológica. É recomendado tanto por Petersen *et al* (2008) quanto por Kitchenham *et al* (2009) que existam pelo menos dois avaliadores para aplicar os critérios de seleção, de forma a garantir uma maior confiabilidade do processo. Apesar do adequado ser a avaliação de cada estudo realizada por pelo menos dois avaliadores e, em caso de divergência, acionar um terceiro avaliador, não é atípico encontrar este tipo de estudo sendo realizado por apenas um pesquisador por conta da limitação de recursos. Isso, entretanto, diminui a credibilidade do resultado. Desta forma, dois pesquisadores participaram desta RSL como avaliadores, de forma que ambos replicaram todo o processo de busca nas quatro bases, de acordo com as *strings* definidas e os resultados obtidos foram condizentes com os que já haviam sido encontrados, validando, portanto, as *strings*, o âmbito da pesquisa e os critérios de inclusão e exclusão definidos no protocolo, trazendo credibilidade à RSL aqui apresentada.

3. Resultados da Pesquisa

A pesquisa e a verificação dos resultados foram realizadas durante os dias 21 e 22 de fevereiro de 2020. Nas quatro bases escolhidas para a realização das pesquisas, a delimitação temporal foi estabelecida a partir do ano 2000 até o presente, contemplando, assim, pesquisas desenvolvidas no Século XXI. A partir da temática “*programming learning*” e “*programming teaching*”, foram efetuadas algumas pesquisas de cunho exploratório, para se verificar as principais ocorrências, e para que pudesse aprofundar melhor na temática “o processo de aprendizagem de algoritmos e programação no nível superior de ensino” a ser pesquisada, para então formular a equação de busca.

Dessa forma, ao pesquisar nas bases sobre o tema, incluímos na equação os termos “*method*”, “*technique*”, “*strategy*”, “*tool*”, “*methodology*”, “*practice*” e “*technology*”, para posteriormente formar a *string* de busca. Sabemos que cada base tem suas particularidades para formar as equações de busca e, por este motivo, a *string*.

A pesquisa na base de dados Web of Science ocorreu no dia 21 de fevereiro de 2020. Para a realização do processo de busca, optou-se pela opção de “busca avançada” disponibilizada pela base. Com a pesquisa inicial realizada na base de dados Web of Science, o resultado bruto foi de 165 ocorrências.

A pesquisa na base de dados Scopus também ocorreu no dia 21 de fevereiro de 2020. Para a realização do processo de busca, optou-se pela opção de “busca avançada” disponibilizada pela base. Com essa pesquisa inicial, o resultado bruto foi de 55 ocorrências.

A pesquisa realizada na base de dados IEEE Xplore ocorreu também no dia 21 de fevereiro de 2020. Para a realização do processo de busca, optou-se pela opção de “busca avançada” disponibilizada pela base. Com essa pesquisa inicial, o resultado bruto foi de 79 ocorrências. Por fim, realizou-se a pesquisa na base de dados RCAAP no dia 22 de fevereiro de 2020. Com essa pesquisa inicial, o resultado bruto foi de 51 ocorrências.

Logo, como resultado de buscas com a *string* descrita anteriormente, e com os critérios de inclusão e exclusão configurados nas bases digitais, foram encontrados um total de 350 trabalhos, sendo 165 na Web of Science, 55 na Scopus, 79 na IEEE Xplore e 51 na base RCAAP.

A partir desses dados retornados, partiu-se para a primeira etapa de análise e seleção dos artigos para a RSL. Inicialmente, por meio da leitura dos títulos e palavras-chave dos dados brutos retornados na pesquisa, foram pré-selecionados 223 artigos científicos, sendo 99 da Web of Science, 34 da Scopus, 49 da IEEE Xplore e 41 da RCAAP. A pré-seleção se deu

em relação aos critérios de inclusão e exclusão definidos no protocolo. Foram excluídos trabalhos que, de acordo com o título e as palavras-chave, não apresentassem a temática de aprendizagem de algoritmos e programação no nível superior de ensino, que se tratassem de trabalhos voltados para uma linguagem de programação específica, ou ainda que não contemplassem o âmbito da pesquisa.

Após essa pré-seleção, foram exportados das bases de dados Web of Sciece, Scopus e IEEE Xplore os arquivos *bibtex* contendo os metadados dos artigos retornados por cada base. Em relação aos artigos da base RCAAP, como não conseguimos o arquivo de dados *bibtex*, exportamos os metadados para uma ferramenta de planilha eletrônica para posterior análise dos dados.

Assim, foram gerados três arquivos *bibtex*, totalizando 177 artigos, os quais foram importados para a ferramenta de gestão bibliográfica Zotero. Esse processo foi necessário para que se desenvolvesse, primeiramente, a exclusão de artigos duplicados e, posteriormente, a leitura dos resumos de todos os artigos selecionados. Depois verificamos se havia também artigos duplicados entre os existentes no Zotero, e entre os artigos da RCAAP catalogados na planilha eletrônica.

Após este processo, foram encontrados um total de 49 artigos duplicados, sendo mantidos 62 artigos da base Web of Science, 23 da base Scopus, 48 da IEEE Xplore e 36 da RCAAP, totalizando 169 artigos para a segunda etapa de análise e seleção. Para este estudo, os artigos indexados na IEEE que estavam duplicados na base de dados Web of Science ou Scopus estão contabilizados como sendo da base IEEE Xplore; não houve ocorrência de duplicidade entre a IEEE Xplore e a RCAAP. A Tabela 5 apresenta os resultados das buscas e primeira etapa da análise, separando-os de acordo com a base consultada.

Tabela 5. Resultado das buscas.

Fonte	Artigos encontrados	Artigos pré-selecionados	Total após exclusão de artigos duplicados
Web of Science	165	99	62
Scopus	55	34	23
IEEE Xplore	79	49	48
RCAAP	51	41	36
Total	350	223	169

Fonte: Elaborado pelos autores.

Para o desenvolvimento da segunda etapa de análise e seleção dos artigos, foram realizadas as leituras dos seus resumos, de forma a compreender o estudo apresentado, bem como reaplicar os critérios de inclusão e exclusão definidos no protocolo desta RSL. Com isso, 84 artigos foram incluídos na análise da RSL, sendo 31 da Web of Science, 15 da Scopus, 32 da IEEE Xplore e 6 da RCAAP.

Uma vez que a terceira etapa se trata da leitura dos artigos completos, partimos para a realização do *download* dos artigos selecionados para essa fase da RSL. No entanto, nem todos os artigos encontram-se disponíveis na *web*. Entramos em contato com alguns autores e obtivemos sucesso em alguns casos, no entanto, dos 84 artigos selecionados para esta etapa, 9 estudos foram retirados por não conseguirmos o acesso do texto completo, totalizando, portanto, em 75 artigos para a leitura completa. Destes 75 artigos, 24 são da Web of Science, 13 da Scopus, 32 da IEEE Xplore e 6 da RCAAP.

Após a leitura completa dos artigos, e novamente a aplicação dos critérios de inclusão e de exclusão, foram escolhidos 38 artigos para compor esta RSL, sendo 14 da Web of Science, 3 da Scopus, 18 da IEEE Xplore e 3 da RCAAP.

Tendo realizado as três etapas de análise e seleção dos artigos, foi desenvolvida a análise de acordo com os critérios de qualidade definidos e apresentados anteriormente. Com isso, foram selecionados 33 artigos para o estudo, sendo 13 da Web of Science, 2 da Scopus, 15 da IEEE Xplore e 3 da RCAAP. A Tabela 6 apresenta os resultados das segunda e terceira etapas de análise e seleção da revisão.

Tabela 6. Resultado das análises da segunda e terceira etapa.

Fonte	Escolha após leitura dos resumos	Total de artigos completos disponíveis	Escolha após leitura completa	Escolha após Critérios de Qualidade
Web of Science	31	24	14	13
Scopus	15	13	3	2
IEEE Xplore	32	32	18	15
Rcaap	6	6	3	3
Total	84	75	38	33

Fonte: Elaborado pelos autores.

Caracterização dos artigos selecionados para a RSL

Como apresentado anteriormente, foram escolhidos para a RSL um total de 33 artigos científicos. Nesta Seção iremos apresentá-los de acordo com suas características, tais como ano de publicação, país de desenvolvimento da pesquisa e língua em que foi publicado.

Em relação ao ano de publicação, seis foram publicados entre os anos 2000 e 2010, e 27 foram publicados entre os anos 2011 e 2019. Até o final desta pesquisa, nenhum dos artigos publicados no ano 2020 havia sido selecionado para a RSL. O Gráfico 1 apresenta o número de publicações por ano dos artigos selecionados.

Gráfico 1. Artigos selecionados para a RSL por ano de publicação.

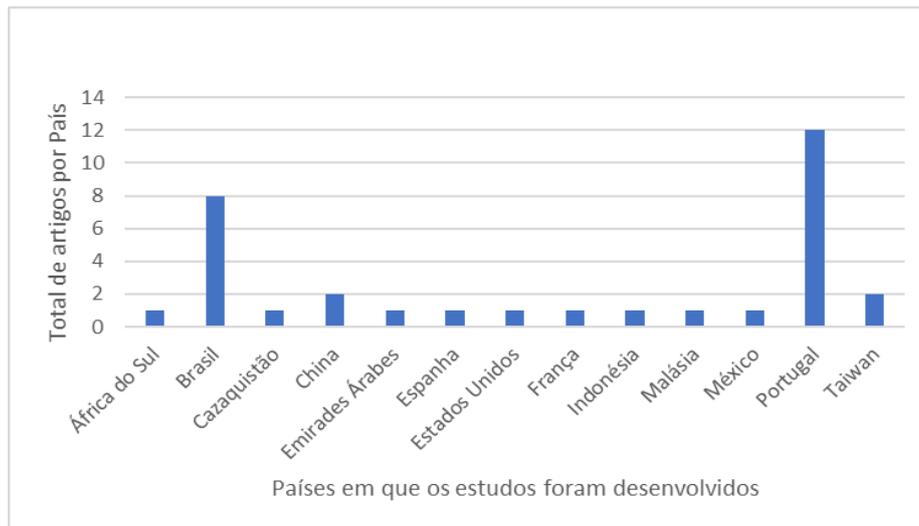


Fonte: Elaborado pelos autores.

De acordo com o Gráfico 1, podemos observar que na primeira década do Século XXI a temática desta RSL já era abordada, embora de forma incipiente. Percebemos que, ao passar dos anos, foram surgindo novos trabalhos na área, principalmente a partir do ano 2015. Vale salientar que, de acordo com os critérios de exclusão e inclusão adotados para este trabalho, aqui abordamos apenas os estudos voltados para a aprendizagem de programação no ensino superior, sem focar em nenhuma linguagem de programação, especificamente.

Em relação ao País de desenvolvimento da pesquisa, a maioria dos trabalhos selecionados para esta RSL foram realizados em Portugal, seguido do Brasil. O Gráfico 2 apresenta o número de artigos por País de publicação.

Gráfico 2. Artigos selecionados para a RSL por País de publicação.



Fonte: Elaborado pelos autores.

Em relação à Língua de escrita, para esta RSL foram considerados artigos que estivessem na Língua Inglesa ou na Língua Portuguesa. Dessa forma, dos 33 artigos selecionados, 26 estão escritos em Inglês e sete estão escritos na Língua Portuguesa. A Tabela 7 apresenta todos os artigos selecionados, estando organizados alfabeticamente de acordo com o seu título. Além do título, apresentamos ainda o ID de cada artigo, por meio do qual guiaremos a referência destes ao longo das próximas seções desta revisão, o ano de publicação, o país de publicação e a base de dados de onde o trabalho foi extraído.

Tabela 7. Lista de artigos selecionados para a RSL.

ID	Título do Artigo	Ano	Local	Base
A01	A Class Record and Reviewing System Designed to Promote Programming Learning	2011	Portugal	IEEE Xplore
A02	A conceptual framework to implement gamification on online courses of computer programming learning: implementation	2017	Espanha	Web of Science
A03	A Programming Learning System for Beginners— A Completion Strategy Approach	2000	Taiwan	IEEE Xplore
A04	A Study of Cooperative Computer Programming Learning Behavior and its Influence on Learning Performance	2012	Taiwan	Web of Science
A05	A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education	2019	Brasil	IEEE Xplore
A06	A Systematic Review of Literature on Methodologies, Practices, and Tools for Programming Teaching	2018	Brasil	IEEE Xplore

A07	A taxonomy of exercises to support individual learning paths in initial programming learning	2013	Portugal	Web of Science
A08	A teacher's view about introductory programming teaching and learning: difficulties, strategies and motivations	2015	Portugal	Scopus
A09	A Virtual Robot Solution to support Programming Learning An Open Source approach	2018	Portugal	IEEE Xplore
A10	A Web Framework to Improve Computer Programming Learning	2018	México	IEEE Xplore
A11	Applying Unidirectional Versus Reciprocal Teaching Strategies in Web-based Environment and Their Effects on Computer Programming Learning	2013	China	Scopus
A12	Attributes of Students Engagement in Fundamental Programming Learning	2015	África do Sul	IEEE Xplore
A13	Collaborations leads to success: a study of effects of using pair-programming teaching technique on student performance in a middle eastern society	2015	China	IEEE Xplore
A14	Comparison between Pseudocode Usage and Visual Programming with Scratch in programming learning	2018	Brasil	IEEE Xplore
A15	Computer Programming Learning How to Apply Gamification on Online Courses	2018	Portugal	RCAAP
A16	Difficulties of Programming Learning from the	2017	Brasil	IEEE Xplore
A17	Engineering Meaningful Computing Education Programming Learning Experience Model	2019	Malasia	IEEE Xplore
A18	Evaluating Success of a Programming Learning Tool	2009	Portugal	Web of Science
A19	Gamificação como Solução para os Problemas da Aprendizagem da Programação	2017	Portugal	Web of Science
A20	Identifying the Need to Intervene: Analysis and Representation of Interaction Patterns in Group Programming Learning	2011	Brasil	Web of Science
A21	Increasing student commitment in introductory programming learning	2012	Portugal	Web of Science
A22	Integrating New Technologies and Existing Tools to Promote Programming Learning	2010	Portugal	Web of Science
A23	Micro-atividades para a Aprendizagem de Programação	2016	Portugal	RCAAP
A24	Microworlds with Different Pedagogical Approaches in Introductory	2019	Portugal	Web of Science
A25	On the Development of a Programming Teaching Tool: The Effect of Teaching by Templates on the Learning Process	2006	Emirados Árabes	Web of Science
A26	Orchestration and adaptation of learning scenarios – Application to the case of programming learning / teaching	2017	França	IEEE Xplore
A27	Priscilla – Proposal of System Architecture for Programming	2018	Cazaquistão	IEEE

	Learning and Teaching Environment			Xplore
A28	Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático	2016	Brasil	RCAAP
A29	Studies and Proposals about Initial Programming Learning	2010	Portugal	IEEE Xplore
A30	The Analysis of a Case Study for Group Programming Learning	2008	Brasil	Web of Science
A31	The role of personalized education	2017	Indonésia	Web of Science
A32	The use of Pair Programming to Support Introductory Programming Teaching: A Qualitative Study	2018	Brasil	IEEE Xplore
A33	Utilising behavioural analytics in a blended programming learning environment	2019	Estados Unidos da América	Web of Science

Fonte: Elaborado pelos autores.

4. Análise e Discussão

A análise e discussão aqui apresentada diz respeito às duas Questões de Apoio desenvolvidas para esta pesquisa. Durante a análise, buscamos responder às questões, de acordo com as abordagens apresentadas em cada artigo selecionado para compor esta revisão. Para apresentar a análise, dividimos em duas partes, sendo a primeira parte para a QA1 e a segunda parte para a QA2.

4.1- Quais os principais desafios e dificuldades enfrentados pelos alunos durante a aprendizagem de programação?

A taxa de insucesso dos alunos nas disciplinas de programação nos anos iniciais dos cursos da área de Informática é bastante elevada. Os professores têm feito um grande esforço no sentido de melhorar os resultados baseando as aulas na resolução prática de exercícios em detrimento da exposição teórica. O decorrer das aulas tem por base a resolução de fichas de exercícios que os alunos vão fazendo com a ajuda do professor, que acaba muitas vezes por ter que explicar uma solução já pré-definida sem que os alunos tenham tempo de conseguir resolver por si próprios. Normalmente, os alunos percebem a solução mas não são capazes de produzi-la. O aluno deveria ter um tempo flexível e adequado à sua rotina de aprendizagem para a resolução dos exercícios, usufruindo simultaneamente das ajudas necessárias a um

iniciante de programação de forma a manter a motivação na procura de uma solução (Chuchulashvili, Goziashvili, Pereira & Lopes, 2016).

É sabido que, em todo o mundo, muitos estudantes apresentam dificuldades em aprender a programar (Santos, Gomes, & Mendes, 2011) e a comunidade científica vem desenvolvendo várias propostas para superá-las (Costa Júnior, 2019). De acordo com Souleiman (2018), essas dificuldades se devem ao fato de programação exigir várias habilidades: (i) capacidade de resolução de problemas; (ii) conhecimento de ferramentas de programação; e (iii) domínio da sintaxe e da semântica de uma linguagem de programação. Por outro lado, introdução a cursos de programação podem reunir um grande número de estudantes que têm níveis heterogêneos. Isso complica a tarefa do professor de fazer exercícios mais práticos e ajudar os alunos que têm dificuldades.

Desta forma, ao longo dos últimos 20 anos, vem crescendo o número de pesquisas e desenvolvimento de novas ferramentas que possam apoiar alunos no processo de aprendizagem de programação. No entanto, segundo Santos, Gomes & Mendes (2011), embora tenham surgido várias ferramentas, é difícil que haja apenas uma que englobe todas as necessidades do aluno e, portanto, acreditam que seja benéfico reunir várias ferramentas e abordagens de ensino para criar um ambiente de aprendizagem.

Existem vários problemas associados ao processo de aprendizagem de programação. Como apresentado por Gomes & Mendes (2015), em um estudo realizado com professores da área de informática, percebeu-se que a maioria dos professores apontam a deficiência na competência dos alunos, principalmente na solução de problemas, raciocínio lógico e base matemática. Além disso, o tamanho das turmas impede um monitoramento mais individualizado, bem como a realização de tarefas mais interessantes aos alunos. Outro fator apontado pelos professores entrevistados nesse mesmo estudo trata-se da forma como as aulas de programação são conduzidas, as quais são consideradas muitas vezes ineficientes por eles “devido à organização das aulas teóricas e práticas no laboratório. No entanto, os professores entendem que estas decisões não estão relacionadas com eles, e são inconvenientes para serem mudadas” (Gomes & Mendes, 2015, p. 2). Além destes problemas elencados, Shadiev *et al* (2013) citam que os estudantes nem sempre conseguem obter um *feedback* instantâneo ou comentários do professor e colegas da turma quando se faz necessário, devido ao tempo da aula, ao *design* instrucional e ao tamanho da turma, fazendo com que muitas vezes o aluno iniciante em programação seja forçado a aprender uma linguagem de programação sozinho, raramente interagindo com os colegas.

Diante das análises realizadas nos artigos presentes nesta RSL, a Tabela 8 apresenta a categorização das dificuldades enfrentadas pelos alunos durante a aprendizagem de algoritmos e programação.

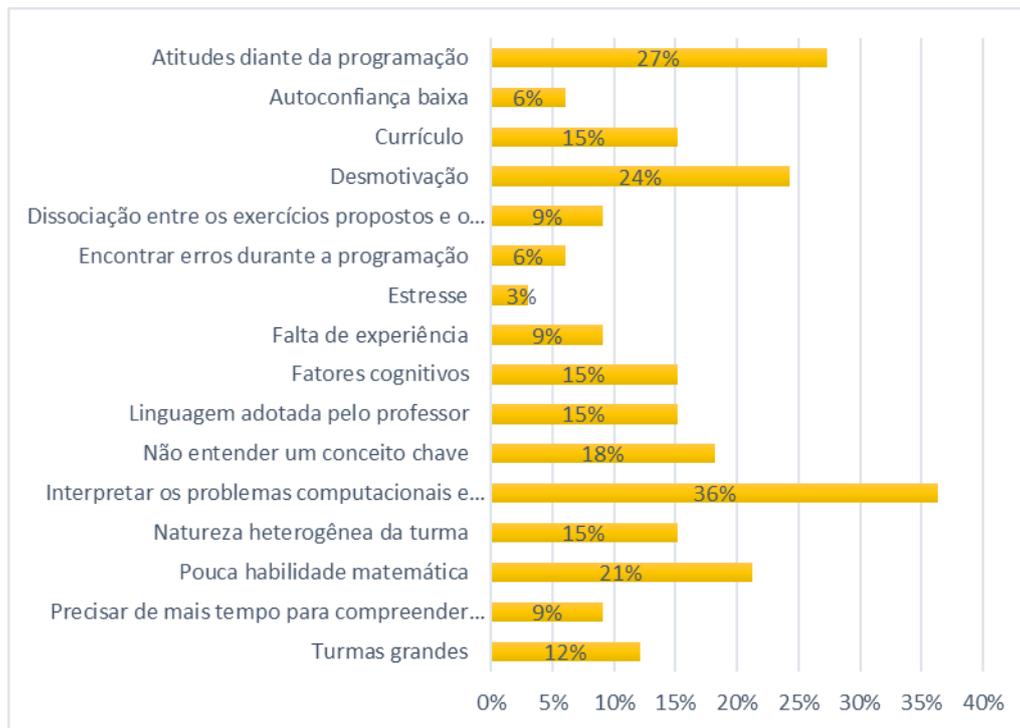
Tabela 8. Categorização das dificuldades enfrentadas pelos alunos durante a aprendizagem de algoritmos e programação.

Categorização dificuldades enfrentadas	Artigos
Atitudes diante da programação	A02, A08, A09, A12, A16, A19, A22, A29, A30
Autoconfiança baixa	A08, A12
Currículo	A02, A08, A21, A23, A29
Desmotivação	A02, A07, A08, A10, A19, A21, A23, A27
Dissociação entre os exercícios propostos e o mundo real	A14, A21, A25
Encontrar erros durante a programação	A19, A26
Estresse	A27
Falta de experiência	A12, A19, A29
Fatores cognitivos	A02, A09, A12, A29, A30
Linguagem adotada pelo professor	A02, A08, A09, A16, A19
Não entender um conceito chave	A01, A02, A09, A12, A14, A19
Interpretação dos problemas computacionais e como resolvê-los	A08, A09, A10, A12, A14, A16, A19, A21, A23, A26, A29, A30
Natureza heterogênea da turma	A07, A08, A21, A22, A26
Pouca habilidade matemática	A08, A11, A12, A14, A16, A19, A29, A30
Precisar de mais tempo para compreender um assunto	A01, A14, A23
Turmas grandes	A07, A21, A22, A26

Fonte: Elaborado pelos autores.

Conforme os dados apresentados na Tabela 8, foi desenvolvido o Gráfico 3 que representa de forma percentual as dificuldades enfrentadas pelos alunos no processo de aprendizagem de algoritmos e programação.

Gráfico 3. Percentual de artigos por categoria de dificuldades enfrentadas durante o processo de aprendizagem de algoritmos e programação.



Fonte: Elaborado pelos autores.

A Tabela 8 e o Gráfico 3 mostram que vários são os fatores que causam dificuldade no processo de aprendizagem de algoritmos e programação. O fator interpretação dos problemas computacionais e como resolvê-los é o de maior porcentagem entre os trabalhos analisados, estando presente em 36% dos artigos. De acordo com Gomes & Mendes (2015), vários autores associam as dificuldades da aprendizagem de programação à natureza abstrata da programação bem como à natureza da resolução de problemas. Os autores Aparicio & Costa (2018) referem como dificuldades de aprendizagem de algoritmos a recolha dos dados, a identificação do conceito problema, e como aplicar o algoritmo correto para novas situações. Logo, para muitos estudantes, utilizar uma linguagem de programação para resolver problemas é uma tarefa difícil, de forma que uma das maiores causas dos problemas de aprendizagem é a falta de habilidade em resolver problemas, de modo que alunos iniciantes geralmente não sabem como criar algoritmos (Mendes, Paquete, Cardoso & Gomes, 2012).

Rezende & Bispo Jr. (2018) apontam que muitas são as dificuldades vinculadas à aprendizagem de programação, tais como: falta de conhecimento prévio em matemática (apresentado em 21% dos artigos), dissociação entre os exercícios propostos e o mundo real (abordado em 9% dos trabalhos) e dificuldades na compreensão do assunto (18% dos artigos).

O segundo fator mais citado pelos artigos em relação às dificuldades na aprendizagem de algoritmos e programação trata-se das atitudes dos alunos diante da programação, presente em 27% dos artigos. Segundo Elteğani & Butgereit (2015, p. 104) “Quando os novatos aprendem programação, eles geralmente têm uma variedade de conhecimentos básicos e atitude em relação à programação. Uma das atitudes que afetam sua aprendizagem é seu comportamento em relação aos erros / erros”, logo as atitudes dos alunos diante da interpretação ou da ocorrência de erros em um programa pode afetar o seu processo de aprendizagem de algoritmos e programação. Como citado anteriormente, a motivação é um dos fatores que mais favorece a aprendizagem de algoritmos e programação. Ao contrário, a desmotivação dificulta esse processo, sendo este problema citado por 24% dos trabalhos analisados nesta RSL, de forma que “a desmotivação é comum em muitos alunos novatos em programação, que não são capazes de lidar com as dificuldades naturais associadas ao aprendizado de programação” (Gomes & Mendes, 2015, p. 1).

A falta ou pouca habilidade matemática é abordada em 21% dos artigos como dificuldade enfrentada pelos alunos. Em seu estudo, Gomes & Mendes (2010) perceberam que as dificuldades demonstradas pelos alunos durante a aprendizagem de programação poderiam ser superadas com o desenvolvimento de habilidades matemáticas, isto porque a necessidade desse conhecimento é fundamental para a aprendizagem de algoritmos e programação (Shadiev et al., 2013; Rezende & Bispo, 2018). Não compreender um conceito chave é citado como uma dificuldade por 18% dos trabalhos presentes nesta RSL, uma vez que o processo de aprendizagem pode ser interrompido se houver lacunas no conhecimento do aluno, portanto, se um aluno não entender um conceito ou estrutura chave, terá muitas dificuldades em todas as atividades de aprendizagem (Santos, Gomes, & Mendes, 2011). A natureza heterogênea da turma pode dificultar o processo de aprendizagem, uma vez que os alunos podem apresentar níveis diferentes de conhecimento, de motivação, de comprometimento e de ritmo de aprendizagem (Santos, Gomes, & Mendes, 2013), dificultando também o acompanhamento do professor e o progresso de alunos que estejam em níveis abaixo da turma. O mesmo fenômeno ocorre pelo fato de haver turmas grandes em unidades curriculares de algoritmos e programação (Santos, Gomes, & Mendes, 2013; Mendes, Paquete, Cardoso, & Gomes, 2012), fator este citado por 12% dos artigos.

Uma das preocupações dos docentes no processo de ensino de algoritmos e programação, citada em 15% dos artigos, está vinculada à escolha da linguagem de programação (Piteira, Costa & Aparício, 2017), de forma que alguns professores acreditam que algumas linguagens de programação, como Python ou Processing são mais apropriadas

para o ensino inicial que outras. Alguns professores escolhem a linguagem de acordo com o curso e o perfil do aluno, no entanto, de acordo com a pesquisa realizada por Gomes & Mendes (2015), a maioria dos professores iniciam com a linguagem C. Também citados em 15% dos trabalhos analisados estão o currículo e fatores cognitivos dos alunos, os quais são reportados na literatura como problemas para a aprendizagem de algoritmos e programação (Piteira, Costa & Aparício, 2017), de forma que o domínio cognitivo do aluno é considerado crucial para o processo de aprendizagem de programação (Aparicio & Costa, 2018). 9% dos artigos apontam como dificuldades no processo de aprendizagem a dissociação entre os exercícios propostos e o mundo real (Al-Imamy, Alizadeh & Nour, 2006), a falta de experiência em programação (Pereira, Costa & Aparicio, 2017) e o fato de o aluno precisar de mais tempo para compreender um conceito (Chuchulashvili, Goziashvili, Pereira & Lopes, 2016). Ainda encontramos como dificuldades a autoconfiança baixa (Elteгани & Butgereit, 2015), com 6% e o estresse (Skalka & Drlik, 2018), citado em 3%.

Embora espere-se que os alunos consigam superar estas e outras dificuldades existentes durante o percurso acadêmico, seja com o apoio docente, dos colegas ou por determinação própria, por exemplo, sabemos que estas podem desencadear vários fatores, tais como frustração, desistência da disciplina e até mesmo o abandono do curso. A seguir, são apresentados os fatores encontrados durante as análises desenvolvidas nesta RSL.

4.2- O que as dificuldades enfrentadas pelos alunos podem desencadear durante seu percurso acadêmico?

Disciplinas introdutórias de programação sofrem devido a altas taxas de reprovações e desistências. Esta é uma situação que afeta principalmente os novatos, pois estas disciplinas geralmente são ofertadas no início dos cursos. De acordo com Mendes, Paquete, Cardoso & Gomes (2012), muitos autores sugeriram possíveis causas para as dificuldades dos alunos, como a natureza da programação, o histórico dos alunos e as atitudes de estudo bem como as estratégias pedagógicas comumente usadas em disciplinas de algoritmos e programação.

Estas dificuldades podem causar desde reprovações nas disciplinas de algoritmos e programação até desistências do curso. De acordo com a análise realizada nesta RSL, categorizamos o que as dificuldades enfrentadas pelos estudantes podem desencadear ao longo do seu percurso acadêmico, como apresenta a Tabela 9.

Tabela 9. Categorização das causas das dificuldades em programar.

Categoria	Artigo
Abandono do curso ou da disciplina	A01, A07,A08, A10, A16, A19, A21, A26, A27, A32
Desmotivação	A07, A08, A27
Frustração	A07, A10, A27
Reprovação na disciplina	A01, A07, A10, A12, A16, A21, A22, A23, A26, A27, A29, A32

Fonte: Elaborado pelos autores.

A reprovação na disciplina é o desencadeamento mais citado nos artigos, ocorrendo em 36%, seguido do abandono do curso ou da disciplina de algoritmos ou programação, citado em 30%. Em todos os artigos, para ambas as ocorrências, os autores citam que a principal causa destas é devido às dificuldades enfrentadas pelos alunos durante o processo de aprendizagem de algoritmos e programação. A frustração e a desmotivação aparecem cada uma delas em 9% dos artigos analisados. De acordo com (Castro, Fuks, Spósito & Castro Júnior, 2008), a frustração e a desmotivação do aluno podem, ainda, causar a reprovação ou o abandono do curso.

5. Considerações Finais

Identificar as dificuldades e desafios enfrentados pelos alunos como as apresentadas neste artigo pode ajudar os professores de algoritmos e programação a orientar seus alunos durante o processo de aprendizagem. Por exemplo, o professor pode adaptar sua metodologia, de acordo com as experiências anteriores dos alunos, de forma a contextualizar o ensino e assim proporcionar melhor experiência na aprendizagem; pode propor estratégias de motivação aos alunos, como forma de estimulá-los a participarem mais ativamente das aulas, e dessa forma, poder ter mais êxito no percurso acadêmico. Outro aspecto a ser visto é, quando as turmas forem numerosas e heterogêneas, sempre que possível, dividir as turmas, para torná-las menores, permitindo que o professor perceba melhor as dificuldades enfrentadas por cada aluno, de forma individualizada e, dessa forma, poder intervir com mais propriedade, para ajudar os alunos no processo de aprendizagem. Acreditamos que a utilização da Revisão Sistemática de Literatura pôde contribuir de forma significativa para a

identificação de tais fatores e dos exemplos acima elencados, uma vez que com esta pudemos verificar e analisar trabalhos relevantes existentes na literatura.

Além disso, com a análise de conteúdo dos artigos, pudemos realizar a categorização das dificuldades, de acordo com os termos mais citados nos artigos, apresentando ainda uma discussão que nos ajuda a perceber o que as dificuldades acarretam aos alunos durante seu percurso acadêmico no curso. Como trabalho futuro, espera-se desenvolver uma metodologia que auxilie os alunos durante o processo e aprendizagem de algoritmos e programação, tendo como apoio os resultados obtidos nesta pesquisa.

Referências

Al-Imamy, S., Alizadeh, J., & A. Nour, M. (2006). On the Development of a Programming Teaching Tool: The Effect of Teaching by Templates on the Learning Process. *Journal of Information Technology Education: Research*, 5, 271-283.

Aparicio, J., & Costa, C. (2018). A virtual robot solution to support programming learning an open source approach. *Iberian Conference on Information Systems and Technologies, CISTI. 2018-June*, 1-6. IEEE Computer Society.

Borges, R., Oliveira, P., Lima, R., & De Lima, R. (1 de 5 de 2018). A systematic review of literature on methodologies, practices, and tools for programming teaching. *IEEE Latin America Transactions*, 16(5), 1468-1475.

Bosse, Y., & Gerosa, M. (1 de 11 de 2017). Difficulties of Programming Learning from the Point of View of Students and Instructors. *IEEE Latin America Transactions*, 15(11), 2191-2199.

Castro, T. H., Fuks, H., Spósito, M. A., & Castro Júnior, A. N. (2008). The Analysis of a Case Study for Group Programming Learning. *8th IEEE International Conference on Advanced Learning Technologies (ICALT)*.

Castro, T., Robertson, D., Fuks, H., & Castro, A. (2011). Identifying the need to intervene: Analysis and representation of interaction patterns in group programming learning. *Lecture*

Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6969 LNCS, 158-174.

Chang, K.-E., Chiao, B.-C., Chen, S.-W., & Hsiao, R.-S. (2000). *A Programming Learning System for Beginners-A Completion Strategy Approach*.

Chiu, M. M. (2008). Flowing toward correct contributions during groups' mathematics problem solving: a statistical discourse analysis. *Journal of the Learning Sciences*, 17(3), 415–463.

Chuchulashvili, M., Goziashvili, N., Pereira, M. J., & Lopes, R. P. (4 a 6 de Julho de 2016). Micro-atividades para a Aprendizagem de Programação. *Estilos De Aprendizagem: Educação, Tecnologias E Inovação. VII Congresso Mundial de Estilos de Aprendizagem*, 1503-1514.

Costa, C., & Aparicio, M. (2014). Evaluating success of a programming learning tool. *ACM International Conference Proceeding Series*. 73-78. Association for Computing Machinery.

Costa, J. (2019). Microworlds with different pedagogical approaches in introductory programming learning: Effects in programming knowledge and logical reasoning. *Informatica (Slovenia)*, 43(1), 145-148.

De Castro, T., Fuks, H., Spósito, M., & De Castro, A. (2008). The analysis of a case study for group programming learning. *Proceedings - The 8th IEEE International Conference on Advanced Learning Technologies, ICALT 2008*, (pp. 850-854).

De Oliveira, T., & Reboucas, A. (2018). The use of pair programming to support introductory programming teaching: A qualitative study. *Proceedings - 13th Latin American Conference on Learning Technologies, LACLO 2018*, 65-68. Institute of Electrical and Electronics Engineers Inc.

Elteğani, N., & Butgereit, L. (2015). Attributes of Students Engagement in Fundamental Programming Learning. *International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering*, 101-106.

Gomes, A., & Mendes, A. (2010). Studies and proposals about initial programming learning. *Proceedings - Frontiers in Education Conference, FIE*.

Gomes, A., & Mendes, A. (2015). A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. *Proceedings - Frontiers in Education Conference, FIE. 2015-February*. Institute of Electrical and Electronics Engineers Inc.

Ho, S., Chean, S., Chai, I., & Tan, C. (2019). Engineering Meaningful Computing Education: Programming Learning Experience Model. *Proceedings of the 2019 IEEE IEEM*, 925-929.

Hwang, W., Shadiey, R., Wang, C., & Huang, Z. (5 de 2012). A pilot study of cooperative programming learning behavior and its relationship with students' learning performance. *Computers and Education*, 58(4), 1267-1281.

Institute of Electrical and Electronics Engineers. Macao Section, IEEE Technology and Engineering Management Society. Singapore Chapter, IEEE Technology and Engineering Management Society. Hong Kong Chapter, & Institute of Electrical and Electronics Engineers. (s.d.). *IEEE IEEM2019. 2019 IEEE International Conference on Industrial Engineering & Engineering Management : 15-18 Dec, Macau*.

Junus, F. (2017). The Role Of Personalized Education Tools In Computer Programming Learning. 92-98.

Medeiros, R., Ramalho, G., & Falcao, T. (1 de 5 de 2019). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions on Education*, 62(2), 77-90.

Mendes, A., Paquete, L., Cardoso, A., & Gomes, A. (2012). Increasing student commitment in introductory programming learning. *Proceedings - Frontiers in Education Conference, FIE*.

Micael Souza, D., Helena da Silva Batista, M., & Francine Barbosa, E. (s.d.). Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático

Problems and Weaknesses in the Teaching and Learning of Programming: A Mapping Review.

Nawahdah, M., Taji, D., & Inoue, T. (2016). Collaboration leads to success: A study of the effects of using pair-programming teaching technique on student performance in a Middle Eastern society. *Proceedings of 2015 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2015*. 16-22. Institute of Electrical and Electronics Engineers Inc.

Oliveira, T. A., & Rebouças, A. D. (2018). The use of Pair Programming to Support Introductory Programming Teaching: A Qualitative Study. *2018 XIII Latin American Conference on Learning Technologies (LACLO)*, 65-68.

Ortíz, O. O., Murillo, J. A., & Hernández, E. M. (2018). A Web Framework to Improve Computer Programming Learning. *2018 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC 2018), Ixtapa, Mexico*.

Paredes, Y., Huang, P., & Hsiao, I. (3 de 7 de 2019). Utilising behavioural analytics in a blended programming learning environment. *New Review of Hypermedia and Multimedia*, 25(3), 89-111.

Pereira, R., Costa, C., & Aparicio, J. (2017). Gamificação como Solução para os Problemas da Aprendizagem da Programação. *Iberian Conference on Information Systems and Technologies, CISTI*. IEEE Computer Society.

Piteira, M., Costa, C., & Aparício, M. (2017). A Conceptual Framework To Implement Gamification On Online Courses Of Computer Programming Learning: Implementation. *ICERI2017 Proceedings. 1*, 7022-7031. IATED.

Piteira, M., Costa, C., & Aparicio, M. (7 de 4 de 2018). Computer Programming Learning: How to Apply Gamification on Online Courses? *Journal of Information Systems Engineering & Management*, 3(2).

Rezende, C., & Bispo, E. (2018). Comparison between pseudocode usage and visual programming with scratch in programming teaching. *Proceedings - 13th Latin American Conference on Learning Technologies, LACLO 2018* (pp. 492-498). Institute of Electrical and Electronics Engineers Inc.

Saeed, R., Mokhtar, R., Jāmi‘at al-Nīlayn, IEEE Sudan Subsection, & Institute of Electrical and Electronics Engineers. (2015). *2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE) : 7th-9th September 2015, Khartoum, Sudan.*

Santos, Á., Gomes, A., & Mendes, A. (6 de 2010). Integrating new technologies and existing tools to promote programming learning. *Algorithms*, 3(2), 183-196.

Santos, Á., Gomes, A., & Mendes, A. (2011). A class record and reviewing system designed to promote programming learning. *Proceedings - Frontiers in Education Conference, FIE.*

Santos, Á., Gomes, A., & Mendes, A. (2013). A taxonomy of exercises to support individual learning paths in initial programming learning. *Proceedings - Frontiers in Education Conference, FIE*, 87-93.

Shadiev, R., Hwang, W., Yeh, S., Yang, S., Wang, J., Han, L., Liu, C. (2013). Applying unidirectional versus reciprocal teaching strategies in web-based environment and their effects on computer programming learning. *Proceedings - 2013 IEEE 13th International Conference on Advanced Learning Technologies, ICALT 2013*, 5-9.

Skalka, J., & Drlik, M. (2018). Priscilla-Proposal of System Architecture for Programming Learning and Teaching Environment. *IEEE 12th International Conference on Application of Information and Communication Technologies, AICT 2018 - Proceedings*. Institute of Electrical and Electronics Engineers Inc.

Souleiman, A. (2018). Orchestration and adaptation of learning scenarios - Application to the case of programming learning / teaching. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA. 2017-October*, 7-11. IEEE Computer Society.

Porcentagem de contribuição de cada autor no manuscrito

Ceres Germanna Braga Moraes – 60%

Francisco Milton Mendes Neto – 20%

Antônio José Meneses Osório – 20%