UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII - GOVERNADOR ANTÔNIO MARIZ
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ALINA NÓBREGA DE BARROS GOMES

OS OBSTÁCULOS NO APRENDIZADO DE ALGORITMOS DE PROGRAMAÇÃO E SUAS INFLUÊNCIAS NA EVASÃO EM CURSOS DE COMPUTAÇÃO: UMA REVISÃO DE LITERATURA

PATOS - PB
2023

ALINA NÓBREGA DE BARROS GOMES

OS OBSTÁCULOS NO APRENDIZADO DE ALGORITMOS DE PROGRAMAÇÃO E SUAS INFLUÊNCIAS NA EVASÃO EM CURSOS DE COMPUTAÇÃO: UMA REVISÃO DE LITERATURA

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba, Campus VII, em cumprimento à exigência para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração: Compreensão de Código

Orientador(a): Ma. Keila Lucas dos Santos

PATOS - PB

2023

G633o Gomes, Alina Nobrega de Barros.

Os obstáculos no aprendizado de algoritmos de programação e suas influências na evasão em cursos de computação [manuscrito] : uma revisão de literatura / Alina Nobrega de Barros Gomes. - 2023.

40 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2023.

"Orientação : Profa. Ma. Keila Lucas dos Santos, Coordenação do Curso de Computação - CCEA. "

1. Linguagem de Programação. 2. Compreensão de código. 3. Aprendizagem em Algoritmos. 4. Evasão na computação. I. Título

21. ed. CDD 005.1

Elaborada por Edson M. A. Monteiro - CRB - CRB 15/743 BSC7/UEPB

ALINA NOBREGA DE BARROS GOMES

OS OBSTÁCULOS NO APRENDIZADO DE ALGORITMOS DE PROGRAMAÇÃO E SUAS INFLUÊNCIAS NA EVASÃO EM CURSOS DE COMPUTAÇÃO: UMA REVISÃO DE LITERATURA

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 28/06/2023.

BANCA EXAMINADORA

Keila Lucas dos Santos

Prof. Ma. Keila Lucas dos Santos
(Orientadora)

Pablo Roberto Fernandes de Oliveira

Prof. Me. Pablo Roberto Fernandes de Oliveira
(Examinador)

Angélica Félix Medeiros

Prof. Ma. Angélica Félix Medeiros
(Examinadora)

Aos meus pais, por sempre acreditarem em mim e nunca medirem esforços para que hoje eu chegasse aqui.

AGRADECIMENTOS

Neste momento de imensa gratidão, antes de tudo, agradeço a Deus por estar sempre presente em minha vida e por sua orientação divina para enfrentar todos os desafios acadêmicos, me concedendo sabedoria, clareza de pensamento e discernimento. Também sou grata a Deus por colocar em meu caminho pessoas incríveis que contribuíram para o meu crescimento acadêmico e pessoal.

Agradeço aos meus pais Albério e Walkyria, pelo amor incondicional e constante incentivo aos estudos. Obrigada por serem exemplo, abrigo e proteção, por acreditarem em mim e por todas as oportunidades que me proporcionaram para tornar esse sonho realidade. Eu amo vocês!

À minha orientadora, Keila, pela sua orientação, apoio e cordialidade. Obrigada por ser paciente, por acreditar na minha capacidade e por sua experiência e conhecimento, você foi fundamental para o desenvolvimento deste trabalho.

À minha família e amigos, por estarem comigo dia após dia, me incentivando e dando forças para superar os desafios. Obrigada por todos os sorrisos e abraços que tornaram meus dias mais leves e felizes.

Às amizades feitas no decorrer da minha graduação, pelas batalhas enfrentadas juntos, pelos conselhos e incentivos, e pelos momentos incríveis. Muito obrigada, vocês tornaram essa jornada mais leve e repleta de lembranças felizes.

RESUMO

O cenário de evasão observado em cursos da Computação transmite uma séria preocupação sobre a aprendizagem de alunos iniciantes no ensino superior dessa área. A dinâmica e o aprendizado em disciplinas da Programação são aspectos que demandam uma análise sobre os fatores que podem contribuir para o abandono dos cursos. Esta pesquisa tem como objetivo analisar as principais dificuldades encontradas por estudantes iniciantes na área da Computação, a partir de uma Revisão Sistemática de Literatura, identificando soluções práticas que visam reduzir os problemas em torno da aprendizagem da Programação, além da problemática da evasão nas disciplinas de Algoritmos. Os objetivos principais da pesquisa foram definidos para identificar as principais obras na área de programação, concentrando-se nos fatores que causam compreensão falha em Algoritmos, e apresentar noções acerca da evasão, ressaltando-se o impacto da dificuldade no aprendizado de programação no Ensino Superior. Seguindo a metodologia *Design Science Research,* os resultados da pesquisa indicaram que algumas ferramentas e estratégias de ensino como o uso da programação em blocos e a aplicação de cursos introdutórios preliminares, auxiliam e fomentam o estudo dos Algoritmos, facilitando a compreensão sobre os aspectos da Programação. O trabalho tem como principal contribuição divulgar as principais técnicas e metodologias que auxiliam na aprendizagem da Programação e pode contribuir para minimizar as taxas de evasão escolar nas disciplinas relacionadas aos cursos de Computação.

Palavras-Chave: Linguagem de Programação; Compreensão de Código; Evasão na Computação; Aprendizagem em Algoritmos.

ABSTRACT

The dropout scenario observed in Computer Science courses raises serious concerns about the learning of beginner students in higher education in this field. The dynamics and learning in Programming subjects are aspects that require an analysis of the factors that may contribute to course abandonment. This research aims to analyze the main difficulties encountered by beginner students in the field of Computer Science through a Systematic Literature Review, identifying practical solutions aimed at reducing problems related to the learning of Programming, as well as the issue of dropout in Algorithm subjects. The main objectives of the research were defined to identify the main works in the programming field, focusing on the factors that cause poor understanding of Algorithms, and to provide insights into dropout, highlighting the impact of difficulty in programming learning in higher education. Following the Design Science Research methodology, the research results indicated that some teaching tools and strategies such as the use of block programming and the implementation of preliminary introductory courses assist and promote the study of Algorithms, facilitating the understanding of Programming aspects. The main contribution of this work is to disseminate the main techniques and methodologies that aid in the learning of Programming and may contribute to minimizing dropout rates in subjects related to Computer Science courses.

Keywords: Programming Language; Code Comprehension; Dropout in Computer Science; Learning Algorithms.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1 INTRODUÇÃO

1.1 Problema

A disciplina de Algoritmos é a base para a maior parte das disciplinas nos cursos da área de Computação. Contudo, a evasão nas disciplinas iniciais é um fato recorrentemente estudado por vários autores (GIRAFFA, MORAES, 2016; DE JESUS, RODRIGUEZ, COSTA, 2021). A situação de estudantes que iniciam, mas não terminam seus cursos pode ser classificada como desperdício social, acadêmico e econômico, pois tudo que foi investido não traz o devido retorno (SILVA FILHO *et al.*, 2007).

Segundo Hoed (2016), alunos evadidos indicaram a baixa qualidade das aulas e a dificuldade em disciplinas que requerem conhecimentos matemáticos como algumas das causas da desistência do curso de Computação. Giraffa e Moraes (2016) apontam que as disciplinas causadoras dessas desistências são associadas ao ensino de Cálculo, Programação e Algoritmos. Gomes, Henriques e Mendes (2008) também ressaltam que uma das razões para esse alto índice de desistência é a dificuldade que alunos iniciantes encontram para desenvolver habilidades necessárias às disciplinas de Algoritmos.

Krzyzanowski *et al*. (2015) destacam que as disciplinas introdutórias, tais como Algoritmos e Programação, demandam um bom nível de compreensão em alguns conteúdos da área de Exatas e que a deficiência nesses conteúdos é um fator determinante para as taxas de reprovação e de evasão na disciplina de Algoritmos e no curso de Computação.

Zacarias e Mello (2019, *apud* BELCHIOR *et al.,*2016) alegam que em disciplinas de Algoritmos o aluno precisa apresentar uma capacidade alta de abstração para que possa compreender por completo o problema. A falta dessa habilidade cria uma série de obstáculos, como a dificuldade para entender a lógica computacional, desmotivação e desânimo, gerando-se fatores determinantes para a não continuidade no curso.

Dentro dessa perspectiva busca-se identificar quais são as dificuldades específicas enfrentadas pelos alunos de Ciência da Computação em relação ao

aprendizado de algoritmos e qual o impacto dessas dificuldades no índice de evasão eminente desses alunos no respectivo curso.

1.2 Proposta de Solução

Diante da problemática apresentada, e da busca por respostas relacionadas à evasão e a aprendizagem de algoritmos, foi realizada uma Revisão Sistemática da Literatura (RSL), buscando trabalhos desenvolvidos no período de 2014 a 2023, sendo esses os trabalhos referentes aos últimos 10 anos de estudo, com objetivos de pesquisa associados às dificuldades encontradas pelos alunos na aprendizagem de uma linguagem de programação e à compreensão de algoritmos de programação.

Na pretensão de compreender as dificuldades predominantes na compreensão de código, referente ao aprendizado de Algoritmos por alunos iniciantes e relacioná-las possivelmente como a evasão em disciplinas referentes a Programação e posteriormente em cursos de Computação. Os dados obtidos nesta pesquisa podem auxiliar professores nas tomadas de decisões relacionadas ao desenvolvimento de estratégias para motivar a permanência dos alunos na área e no desenvolvimento ou aprimoramento de ferramentas voltadas para a aprendizagem dos estudantes de Computação.

1.3 Objetivos

1.3.1 Objetivo Geral

Este trabalho tem como objetivo investigar os principais fatores que impedem a progressão de alunos iniciantes no estudo de algoritmos computacionais, visando identificar o impacto desses obstáculos em relação à evasão nos cursos da área de Computação.

1.3.2 Objetivos Específicos

- Identificar no estado da arte, a partir da revisão de literatura, pesquisas relacionadas à aprendizagem de linguagens de programação e compreensão de código;
- Evidenciar quais fatores causam a má compreensão de Algoritmos, visando auxiliar no direcionamento de outros métodos pedagógicos para aprimorar o ensino da Programação;
- Apresentar dados coletados em literaturas acerca da evasão em cursos de Ciência da Computação;
- Compreender o impacto que as dificuldades na aprendizagem de programação causam na alta evasão de estudantes na área da Computação.

1.4 Justificativa

A dificuldade de compreensão em conteúdos matemáticos por parte dos alunos, incidente desde a formação básica, é um fator impactante nas altas taxas de reprovação e de desistência nas disciplinas integrantes do curso de Computação, principalmente as que são relacionadas à programação (GOMES *et al*., 2015). A presente pesquisa busca identificar as principais dificuldades associadas à aprendizagem de algoritmos, objetivando o reconhecimento de estratégias para reduzir a taxa de evasão dos cursos na área de Computação e também contribuir para o desenvolvimento de ferramentas que aprimorem o processo de ensino/aprendizado de alunos iniciantes.

1.5 Organização do Trabalho

Este trabalho dispõe de 6 capítulos, incluindo o capítulo de Introdução, na qual foram apresentadas a contextualização, a problemática, a proposta de solução, os objetivos geral e específicos, a justificativa e a forma de organização do trabalho. No capítulo 2 é apresentado todo o Referencial Teórico, incluindo o conceito de

linguagem de programação e de algoritmos. O referido Capítulo discorre também sobre a aprendizagem efetiva em programação, as dificuldades enfrentadas e aborda a evasão em cursos de Computação. O Capítulo 3 é composto pela apresentação dos trabalhos relacionados aos objetivos da pesquisa. No Capítulo 4 apresenta-se o processo metodológico da pesquisa, com a descrição do desenvolvimento da Revisão Sistemática da Literatura e a da análise dos trabalhos. No capítulo 5 apresenta-se os resultados obtidos no estudo, concentrando nas respostas para as questões de pesquisa que relatam as principais dificuldades apresentadas pelos alunos no aprendizado de uma linguagem de programação e a relação dessas dificuldades na evasão em cursos de Computação. No capítulo 6 são apresentadas as Considerações Finais do trabalho, as limitações da pesquisa, as contribuições e sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo apresenta-se a revisão bibliográfica do trabalho, discutindo-se os conceitos que fundamentam o tema escolhido para a pesquisa. O capítulo está organizado da seguinte forma: Na Seção 2.1 explica-se o conceito de linguagem e algoritmos de programação; na Seção 2.2 são expostas as estratégias obtidas por meio da RSL que podem levar a uma aprendizagem efetiva em programação; a Seção 2.3 associa algumas dificuldades à aprendizagem; e por fim, na Seção 2.4 aborda-se a evasão em cursos da área de Computação, em especial a evasão existente em disciplinas de algoritmos de programação.

2.1 Linguagem de Programação

A programação é uma parte muito importante para a Ciência da Programação. Atualmente existem diversas linguagens de programação, assim como existem diversos tipos de computadores, mas apesar disso, todas elas têm um objetivo principal semelhante que é tornar possível a comunicação entre os humanos e as máquinas. Por meio da programação os humanos são capazes de informar aos computadores uma sequência de passos que devem ser seguidos para solucionar problemas específicos.

> Uma linguagem de programação é um método padronizado que usamos para expressar as instruções de um programa a um computador programável. Ela segue um conjunto de regras sintáticas e semânticas para definir um programa de computador. Regras sintáticas dizem respeito à forma de escrita e regras semânticas ao conteúdo (GOTARDO, 2015, p. 17).

Pode-se dizer que as linguagens de programação são os idiomas que os computadores são capazes de compreender, é o meio utilizado para se comunicar com máquinas programáveis, e por intermédio dessas linguagens o computador é conduzido a seguir uma sequência de instruções, denominadas de Algoritmos. Gotardo (2015) também enfatiza que algoritmos são a base para o uso das linguagens de programação e para a construção de programas. Um computador só

é capaz de executar com maestria um determinado comando se um passo a passo de instruções for bem definido.

Um algoritmo pode ser definido como uma sequência finita de passos criados para a resolução de um determinado problema (FERRARI; CECHINEL, 2008), ou seja, é uma sequência de ações que, se seguidas corretamente, levam ao resultado de um problema. No livro “Introdução a algoritmos e programação” escrito por Ferrari e Cechinel (2008), são evidenciadas algumas regras para o desenvolvimento de um algoritmo eficaz, sendo elas:

- Definir ações simples e sem ambiguidades;
- Organizar as ações de forma ordenada;
- Ter uma sequência finita de passos.

Os algoritmos podem ser classificados em tipos, sendo alguns desses tipos: Pseudocódigo, fluxograma e descrição narrativa. Segue abaixo um exemplo de cada um dos algoritmos supracitados:

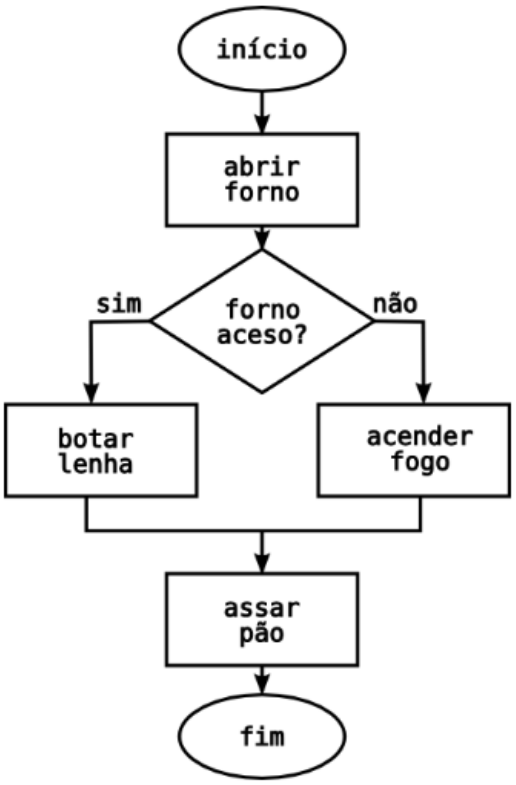
Figura 1- Exemplo de um algoritmo de pseudocódigo

Algoritmo 1 Exemplo de Pseudocódigo.

```
leia (x, y) {Esta linha é um comentário}
se x > y então
    escreva ("x é maior")
senão
    se y > x então
        escreva ("y é maior")
    senão
        escreva ("x e y são iguais")
    fim-se
fim-se
```

Fonte: Ferrari; Cechinel (2008)

Figura 2 - Exemplo de um algoritmo em forma de fluxograma



Fonte: Ferrari; Cechinel (2008)

Figura 3- Exemplo de um algoritmo de descrição narrativa

Algoritmo 1 Troca de pneu do carro.

1: desligar o carro
2: pegar as ferramentas (chave e macaco)
3: pegar o estepe
4: suspender o carro com o macaco
5: desenroscar os 4 parafusos do pneu furado
6: colocar o estepe
7: enroscar os 4 parafusos
8: baixar o carro com o macaco
9: guardar as ferramentas

Fonte: Ferrari; Cechinel (2008)

A Figura 1 refere-se a uma representação algorítmica chamada de Pseudocódigo, que é o modelo de algoritmos que mais se aproxima da linguagem de programação. Essa representação faz uso de palavras-chave na linguagem

natural do programador similares aos termos encontrados nas linguagens de programação, porém não necessita do mesmo rigor sintático que uma linguagem de programação requer. Essa figura descreve o passo a passo para a realização de uma comparação entre duas variáveis x e y, com o objetivo de retornar como a maior delas, ou ambas caso possuam o mesmo valor.

Na Figura 2 é apresentado um algoritmo no formato de Fluxograma, cuja representação é feita em formato de gráfico, onde cada ação a ser seguida é retratada por uma figura geométrica específica, a depender do objetivo de cada uma. Na imagem, por exemplo, as ações ou situações são representadas por retângulos e as tomadas de decisão por losangos. Esse tipo de representação não se aproxima da linguagem de programação, servindo mais como orientação para a resolução do problema em si, mostrando possíveis caminhos que podem ser seguidos. O algoritmo mostrado na Figura 2 demonstra quais ações são necessárias para assar um pão em forno a lenha.

Por fim, na Figura 3, é retratado um algoritmo de descrição narrativa, na linguagem do programador, simplesmente descrevendo linha a linha os passos necessários até que o problema seja resolvido. Nesse caso, o algoritmo descreve com clareza todas as ações que precisam ser executadas para uma troca de pneu.

2.2 Aprendizagem Efetiva em Programação

Através da Revisão Sistemática da Literatura, pôde-se verificar uma diversidade de estudos voltados para o ensino e a aprendizagem de programação, nos quais vários deles apresentam alternativas que auxiliam neste processo. Também foi possível identificar quais são os principais fatores necessários para uma aprendizagem efetiva e alguns dos meios utilizados nos estudos para obtenção de um retorno associando a compreensão, ou seja, quais técnicas foram usadas para avaliar a aprendizagem dos alunos em programação.

Whalley e Kasto (2014) afirmam que educadores de Ciência da Computação vêm aplicando taxonomias educacionais como a Taxonomia de Bloom, a Taxonomia de Bloom Revisada e a Taxonomia SOLO, buscando um ensino de programação mais eficiente para iniciantes. Izu, Weerasinghe e Pope (2016) apoiam o uso da

taxonomia SOLO para avaliar a capacidade dos alunos de programar e identificar relações de nível superior. O uso dessas taxonomias facilitou a realização de diversas coletas de dados ajudando professores a definir expectativas em relação ao progresso de alunos iniciantes na codificação.

Em seu trabalho Falckembach e Araujo (2006) salientaram que o uso da programação orientada a objetos como primeira disciplina para o ensino de programação, com o propósito de resolver as dificuldades encontradas no estudo inicial da lógica de programação, não gerou os resultados esperados. A introdução à lógica de programação se fez mais eficaz com a programação estruturada, sendo mais fácil para a compreensão e determinação das ações que constituem um algoritmo. Por essa razão, acredita-se que os procedimentos básicos devem ser trazidos de forma gradual, de modo que o aluno seja levado a desenvolver de modo mais progressivo a habilidade de resolver problemas via computador.

Dümmel *et al.* (2018) realizou um estudo para a criação e análise da influência de um curso introdutório de programação, no qual afirmaram que o primeiro passo é compreender as ideias básicas por trás de um algoritmo, como eles funcionam e como podem ser executados em uma máquina teórica, para só então ser possível entender com mais clareza a ideia de pensamento computacional e aprender as diferenças entre dados e instruções. Dessa forma, os alunos serão levados a enxergar os problemas introduzindo Algoritmos de forma mais eficiente, podendo dividir os problemas em problemas menores, mais fáceis de entender e resolver, metodologia similar a programação em Blocos.

Foi possível observar também através da revisão bibliográfica a aprendizagem efetiva ser uma tarefa bastante complexa, e por isso alguns grupos de pesquisa têm desenvolvido softwares a fim de auxiliar estudantes e professores nesse processo de ensino-aprendizagem. Grande parte dos trabalhos coletados por meio da RSL estão relacionados ao objetivo de avaliar a usabilidade de alguma ferramenta específica na aprendizagem da programação.

2.3 Dificuldades na Aprendizagem

A revisão bibliográfica permitiu a identificação de fatores importantes para uma aprendizagem efetiva, assim como também foi possível obter conhecimento em relação às muitas dificuldades enfrentadas nesse processo. Falckembach e Araujo (2006) afirmam que a dificuldade na resolução de problemas computacionais enfrentada pelos alunos está no reconhecimento dos procedimentos fundamentais para se chegar a uma solução. Por isso, é necessária a busca por meios eficientes para trabalhar os processos cognitivos, sobretudo a abstração e a formalização.

Dümmel *et al.* (2018) ressaltam que muitas vezes os alunos têm dificuldade em transferir o conhecimento adquirido anteriormente para problemas novos e desconhecidos devido a falta de experiência com as ferramentas para a construção da linguagem, ou seja, não entendem como as ferramentas funcionam para implementar sua solução algorítmica. Os autores afirmam que a linguagem Python, apesar de ser uma linguagem cujo paradigma é Orientado a Objeto, possui pouca sobrecarga sintática e é ideal para cursos introdutórios, auxiliando no desenvolvimento da compreensão acerca de estruturas de controle, seleção e iteração; introdução a listas e funções. Essa linguagem é bastante próxima à encontrada nos pseudocódigos, o que permite aos alunos se concentrarem mais na compreensão e resolução dos problemas.

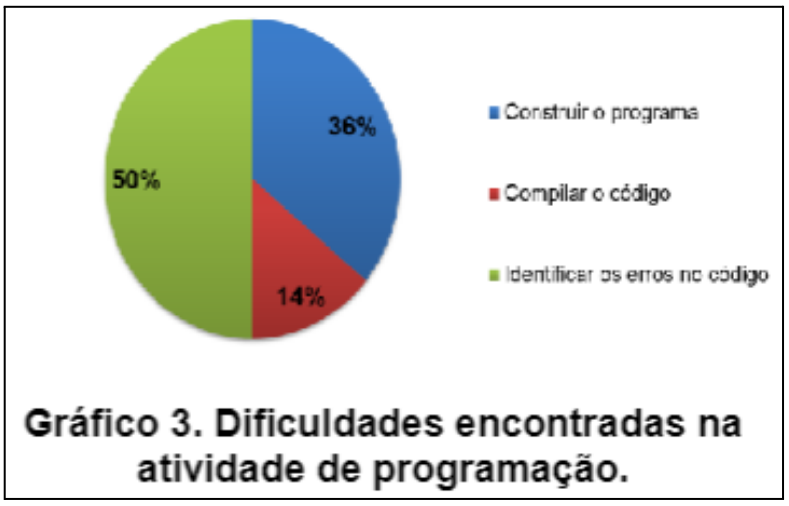
Com base nos estudos de Obaidellah *et al.* (2018), muitos pesquisadores associam a falta de habilidade ou incapacidade de compreensão de conceitos para a resolução de problemas como um dos principais problemas enfrentados por estudantes novatos no aprendizado de linguagens de programação. Whalley e Kasto (2014) estimaram a dificuldade de tarefas de escrita de código por novatos através de um estudo usando métricas de software. Com isso, verificou-se que a complexidade ciclomática (complexidade de um determinado módulo), a métrica de expressão regular (quantidade de comandos e operadores presentes), e a legibilidade estão fortemente relacionadas à dificuldade de compreensão do código.

Izu, Weerasinghe e Pope (2016), fazendo uso da Taxonomia SOLO, realizaram um estudo avaliando as habilidades básicas de programadores iniciantes, e por meio desse estudo foi possível identificar a presença de muitos loops infinitos

ou vazios, sendo esses os erros mais comuns cometidos pelo grupo. Os autores reforçaram a necessidade de atividades de suporte para melhorar a compreensão tanto na semântica dos loops quanto na manipulação de vetores.

Gomes *et al.* (2015) realizaram análise sobre os principais erros cometidos por estudantes de programação, apontando os principais temas que devem ser prioridade no processo de ensino-aprendizagem de programação. Foram eles: Depuração (encontrar e solucionar erros), Compilação (erros sintáticos) e desenvolver um programa para realizar executar uma tarefa (erros semânticos), reforçando a falta de uma boa prática de interpretação e criação de algoritmos.

Figura 4 - Dificuldade encontrada na atividade de programação



Fonte: Gomes *et al.* (2015)

A Figura 4 foi extraída de um estudo feito sobre erros em programação, direcionada para o reconhecimento das principais dificuldades enfrentadas por programadores iniciantes. Após a realização do estudo, ficou evidente que 50% das dificuldades estavam relacionadas à identificação de erros no próprio código. A segunda maior dificuldade observada, com 36%, foi a de construir o programa, ou seja, de usar corretamente uma linguagem de programação. Por fim, com 14%, erros ao compilar o código. Nesse mesmo estudo eles associam a dificuldade de identificar erros no próprio código com a dificuldade de entender as mensagens de erro geradas pela máquina devido ao idioma utilizado pelo compilador.

2.4 Evasão nas Disciplinas de Programação

De acordo com o Dicio (Dicionário Online de Português), a palavra evasão é descrita como ato de abandono, desistência de algo ou alguém em um lugar ou determinada situação. O tipo de evasão discutida nesta pesquisa refere-se à evasão escolar, especificamente a evasão em disciplinas iniciais no curso de Ciência da Computação. Para discutir acerca desse sério problema que é o abandono de disciplinas de programação e posteriormente desistência no curso, ressalta-se a afirmação de Freire *et al.* (2019) sobre a análise de que as disciplinas introdutórias nos cursos de Ciência da Computação apresentam elevados índices de reprovação e evasão, em decorrência de dificuldades referentes à metodologia.

Os alunos apresentam grandes dificuldades em disciplinas ligadas a programação ao longo do primeiro ano de graduação, as quais muitas vezes são associadas a falta de conhecimento prévio, dificuldades na resolução de problemas, raciocínio lógico, abstração, entre outros fatores envolvidos (DE JESUS, RODRIGUEZ, COSTA, 2021).

De Jesus, Rodriguez e Costa (2021) realizaram um estudo coletando dados relevantes para tentar prever o risco de evasão no curso de Licenciatura em Computação da Universidade do Estado do Amazonas, e concluíram que as disciplina de Introdução a Programação de Computadores e Algoritmos estão entre as disciplinas que mais reprovam os alunos no primeiro ano do curso, apesar de serem disciplinas de formação básica e que dão suporte para a compreensão de outras disciplinas no decorrer dos cursos de Ciências da Computação.

Na Figura 5 são mostradas as disciplinas contempladas no primeiro ano de curso e o número de aprovados e reprovados em cada disciplina. Foi possível verificar que as disciplinas de Introdução a Programação de Computadores, Cálculo I, Álgebra Linear I, Matemática Discreta e Programação de Computadores e Algoritmos foram as disciplinas que apresentaram o número de reprovados acima do número de aprovados.

Figura 5 - Disciplinas introdutórias e o número de aprovados e reprovados

	Disciplina	Aprovados	Reprovados
0	Filosofia da Educacao	109	29
1	Introducao a Programacao de Computadores	85	101
2	Introducao a Computacao	95	60
3	Calculo I	66	113
4	Legislacao e Organizacao da Educacao Brasileira	110	22
5	Algebra Linear I	40	80
6	Matematica Discreta	37	125
7	Programacao de Computadores e Algoritmos	54	65
8	Probabilidade e Estatistica	44	37
9	Psicologia da Educacao	92	10
10	Portugues Instrumental	83	20

Fonte: De Jesus; Rodriguez; Costa (2021)

Figura 6 - Disciplinas de programação

	Disciplina	Aprovado	Rep Nota	Trancado	Rep Freq
0	Introducao a Programacao de Computadores	85	80	2	21
2	Programacao de Computadores e Algoritmos	54	52	2	13
3	Algoritmos e Estrutura de Dados I	35	32	2	10
1	Projeto de Programas	40	16	0	9
4	Algoritmos e Estrutura de Dados II	20	6	3	4

Fonte: De Jesus; Rodriguez; Costa (2021)

Já na Figura 6, o estudo foi feito incluindo somente disciplinas que incluíssem conceitos e práticas de programação, onde foram analisados os números de aprovados, trancamentos e reprovações por nota ou por falta. As disciplinas de Introdução a Programação de Computadores, Programação de Computadores e Algoritmos e Algoritmos e Estrutura de Dados I foram as disciplinas que apresentaram número de reprovações e trancamentos superior ao número de aprovados.

Em geral, neste capítulo foram mostrados conceitos que fundamentam as linguagens de programação assim como 3 modelos de algoritmos bastante usados no decorrer dos cursos de Ciências da Computação. Em seguida, foram verificados estudos voltados para o ensino e aprendizagem de programação, e foram evidenciados alguns fatores necessários para uma aprendizagem efetiva em programação, como a necessidade de se compreender as ideias básicas de um algoritmo e ser capaz de transcrevê-las para uma máquina, ou seja, desenvolver a habilidade de resolver problemas via computadores.

Logo após, foram discutidas as principais dificuldades enfrentadas no aprendizado por programadores iniciantes e foi possível perceber que muitos autores abordaram a falta de habilidades ou incapacidades de compreensão e resolução de problemas, em outras palavras, eles não têm o que é necessário para implementar uma solução algorítmica para um determinado problema, ressaltando a importância de buscar meios que auxiliem nesse problema. Por fim, foi escolhido um trabalho que realizou uma análise de evasão no curso de Computação da Universidade do Estado do Amazonas, onde foram observados altos índices de reprovação e evasão em disciplinas introdutórias. Entre as principais disciplinas destacadas nessa categoria estavam incluídas:

- Introdução a Programação de Computadores;
- Programação de Computadores e Algoritmos;
- Algoritmos e Estrutura de Dados I

As três disciplinas elencadas ganharam destaque pois são citadas na pesquisa com taxas de reprovação superiores às de aprovação e estão totalmente relacionadas com o objetivo deste trabalho.

No capítulo seguinte serão discutidos os trabalhos relacionados, os quais foram obtidos por meio de uma Revisão Sistemática da Literatura, que serviram como referencial técnico e fonte para obtenção dos resultados da pesquisa.

3 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados os trabalhos relacionados aos objetivos da pesquisa para dar embasamento científico e acadêmico necessários para a discussão do tema. Os trabalhos citados a seguir foram identificados mediante uma Revisão Sistemática da Literatura (RSL) com o propósito de explorar diversos trabalhos relacionados ao ensino/aprendizado de novatos em programação, buscando descobrir as dificuldades enfrentadas e associá-las com a evasão em cursos de Ciência da Computação. Todo o processo de desenvolvimento da RSL está descrito no Capítulo 4, juntamente com a metodologia de trabalho.

Dümmel *et al.* (2018) abordam sobre as dificuldades que programadores iniciantes, identifica-se que as falhas na resolução de problemas e na compreensão de algoritmos estão associadas, diversas vezes, na limitação dos alunos conseguirem associar semelhanças com problemas passados e não aplicar os algoritmos necessários.

No trabalho de Izu *et al.* (2019) foi realizada uma pesquisa sobre as dificuldades específicas enfrentadas por programadores novatos e destacaram a importância de tratar esses obstáculos por meio de estratégias de ensino e avaliação adequadas que equilibrem a teoria e a prática, pois a compreensão de programas envolve uma combinação de conhecimento conceitual e habilidades práticas.

Tshukudu e Cutts (2020) avaliaram o processo de transferência de alunos novatos de uma linguagem de programação para outra, e afirmaram que a transferência de conhecimentos prévios pode levar a concepções equivocadas, porém, os conhecimentos não precisam necessariamente vir de outra linguagem, onde ele cita como exemplo a diferença entre o conceito de variável na programação e na matemática.

Considerando alternativas de baixo custo que podem auxiliar professores a aprimorar a metodologia de ensino de habilidades de compreensão de código, ressalta-se o trabalho de Donaldson e Cutts (2018), que analisaram três exercícios de aprendizagem/avaliação distintos usando Modelos de Blocos, os quais eram baseados em anotações de código, permitindo também serem aplicados em papel e

lápis, reduzindo a complexidade, pois divide as atividades em componentes mais gerenciáveis para o aluno e permite flexibilidade de acordo com as necessidades dos professores, ou seja, é uma alternativa acessível, voltada também para a visão dos estudantes, diferentemente da maioria dos outros trabalhos analisados, que concentraram seus estudos na visão pedagógica.

Gomes *et al.* (2015) descrevem em seu artigo “Um estudo sobre erros de programação: Reconhecendo as dificuldades de programadores iniciantes” um trabalho realizado com alunos iniciantes em programação, no qual foi feita uma análise nos códigos coletados, onde 71% das compilações não foram bem sucedidas. Ao fim da análise dos dados, declararam que a maior parte dos erros cometidos pelos alunos foram erros de sintaxe, como a falta de tokens ao final de estruturas (parênteses ou chaves), a não finalização de comandos com ponto e vírgula, erros na declaração de variáveis ou a falta delas ou a falta de argumentos em funções.

No estudo realizado por Hoed (2016), foi feita uma comparação entre a evasão em cursos de Computação classificados pelo INEP com a média nacional considerando as 8 grandes áreas do conhecimento. A comparação entre instituições públicas e privadas foi feita separadamente como mostrado nas Figuras 7 e 8, e foi possível notar que a média da evasão em cursos de Computação em ambas ficou acima da média de evasão nacional no período de 2010 a 2014.

Figura 7 - Evasão na Computação em relação à média nacional (Instituições Privadas)

Ano	Média Nacional(%)	Computação (%)
2010	15,86	19,42
2011	17,46	20,74
2012	16,82	20,16
2013	16,81	21,10
2014	17,16	21,89

Fonte: Hoed (2016)

Figura 8 - Evasão na Computação em relação à média nacional (Instituições Públicas)

Ano	Média Nacional (%)	Computação (%)
2010	12,16	14,23
2011	11,46	13,87
2012	12,00	16,18
2013	12,45	15,12
2014	12,35	15,88

Fonte: Hoed (2016)

Segundo Giraffa e Moraes (2016), estudos realizados por diversos pesquisadores da área de Educação e Computação e do Ministério da Educação brasileiro (MEC) apontam que muitos estudantes desistem dos cursos logo no primeiro ano da graduação, e alegam que as disciplinas responsáveis pelas desistências são as associadas ao ensino de Cálculo e de Programação, incluindo a disciplina de Algoritmos. Após a realização do estudo, eles evidenciaram que os principais motivos do abandono ou cancelamento da disciplina de Algoritmos de Programação estão associados à falta de tempo para se dedicarem e à habilidade de compreensão dos exercícios. Foi citado também que a condução didática dos professores em sala de aula influencia a permanência dos alunos na disciplina.

Considerando a relevância dos estudos citados, em suas percepções e abordagens sobre o aprendizado de algoritmos e evasão no curso de Computação, também é importante ressaltar que segundo Falckembach e Araujo (2006) é bastante necessário trabalhar mais processos cognitivos necessários à construção de algoritmos, visando desenvolver a agilidade dos alunos na solução de problemas.

Os trabalhos selecionados neste capítulo têm em comum o interesse pela busca de informações e alternativas que possam influenciar positivamente na educação de conteúdos ligados ao aprendizado introdutório na área da Computação, a fim de trazer melhorias capazes de descomplicar a compreensão da programação de computadores e consequentemente reduzir as elevadas taxas de evasão que os cursos da área apresentaram.

4 METODOLOGIA

4.1 Aspectos Metodológicos

O principal desafio dessa pesquisa é analisar os fatores mais notáveis que dificultam o aprendizado na disciplina de algoritmos para alunos iniciantes no curso de Ciência da Computação e se há relação dessa adversidade com a acentuada evasão que os cursos superiores desta área apresentam. A investigação é feita por meio de uma Revisão Sistemática da Literatura (RSL), detalhada no Tópico 4.2 deste Capítulo.

A metodologia aplicada na pesquisa foi a *Design Science Research*, paradigma de pesquisa pragmático proposto por Hevner *et al.* (2004), na qual a resolução de problemas no mundo real é feita por intermédio da criação e análise de artefatos inovadores. Para isso, a modalidade de pesquisa implementada foi do tipo bibliográfica, com o propósito de conhecer melhor as dificuldades apresentadas nas disciplinas de algoritmos e analisar a evasão.

De acordo com Dresch *et al.* (2015), a *Design Science Research* é fundamental para a resolução de problemas específicos e relevantes, que ainda não apresentem respostas suficientes para ser solucionado. Os autores enfatizam também que não necessariamente se busca uma solução perfeita, mas uma solução que possibilite que outros pesquisadores consigam fazer uso desse conhecimento gerado.

O paradigma *Design Science Research* permite que seja definida uma Questão Geral de Pesquisa (QGP) e que esta pode ser decomposta em Questões Secundárias de Pesquisa (QSP).

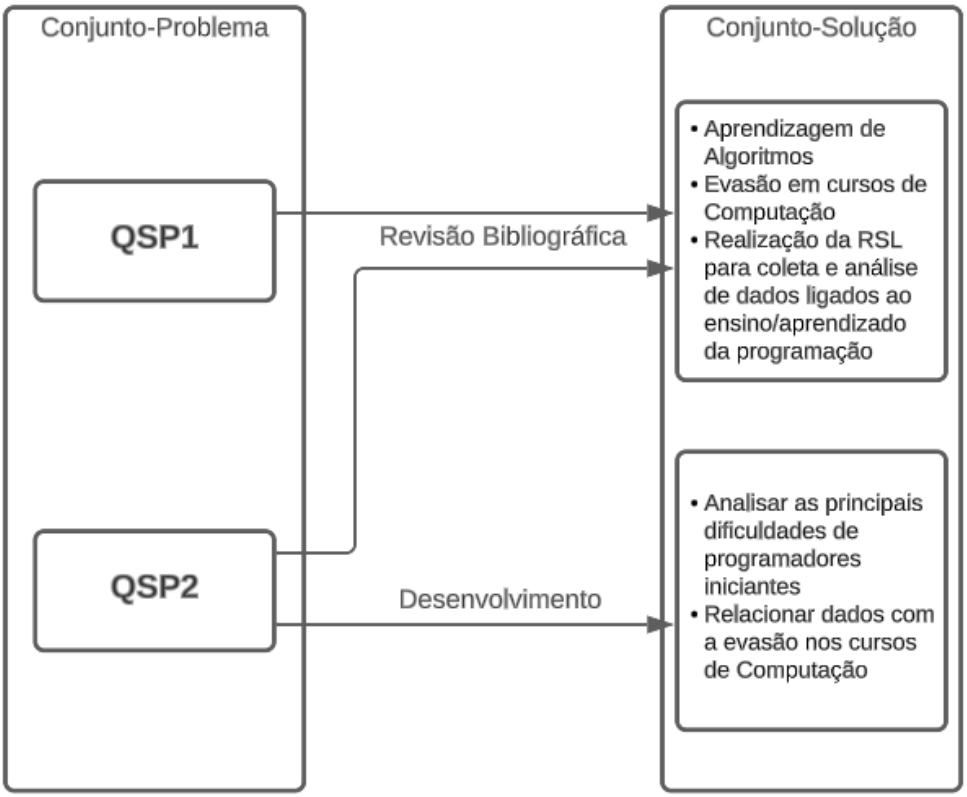
A QGP que orienta esta pesquisa é definida a seguir:

QGP - Quais são as principais dificuldades no aprendizado de algoritmos enfrentadas por alunos em cursos de Computação que geram impacto nos índices de evasão registrados no curso?

QSP1: Quais os obstáculos mais comuns que os recém ingressantes no curso de Ciência da Computação enfrentam em disciplinas de programação?

QSP2: Qual a relação entre a limitação na aprendizagem em programação e evasão registrada em turmas iniciantes do curso de Computação?

Figura 9 - Metodologia da pesquisa conforme o *Design Science*



Fonte: Autoria própria (2023)

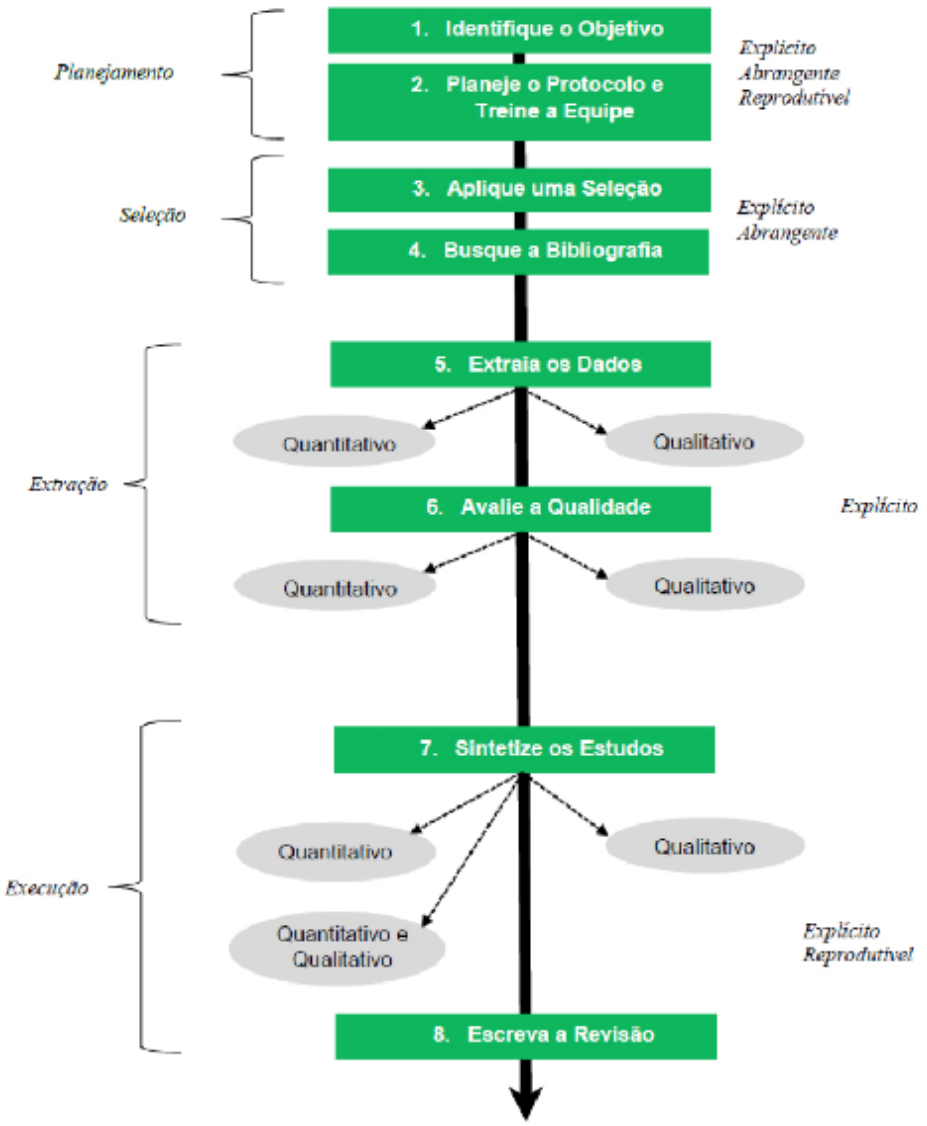
Na Figura 9 está ilustrado o processo de desenvolvimento da pesquisa, no qual as QSPs mencionadas pertencem ao Conjunto-problema partindo em busca de uma resposta no Conjunto-solução por meio das técnicas:

- **Revisão bibliográfica** - Técnica aplicada para estruturar o embasamento teórico dos assuntos abordados no trabalho. Também será realizada uma Revisão Sistemática da Literatura, para a coleta de dados que possuam as informações necessárias à pesquisa, que neste caso é a respeito do ensino e aprendizagem de programação por alunos iniciantes.
- **Desenvolvimento** - Ocorrerá a análise dos dados coletados nas literaturas selecionadas pela RSL e tentativa de relacioná-los com a evasão nos cursos de Computação.

4.2 Revisão Sistemática da Literatura

Brizola e Fantin (2016 *apud* Morandi e Camargo, 2015, p. 141), destacam que a Revisão Sistemática da Literatura (RSL) é uma etapa de extrema importância para a condução de pesquisas científicas, em especial pesquisas realizadas sob o paradigma da *Design Science*. A RSL segue etapas bem definidas que devem ser seguidas com rigor, sendo elas: Identificar o objetivo, selecionar fontes de busca da temática, elaborar estratégias para definir o rumo da pesquisa, avaliar os estudos a serem utilizados na RSL, sintetizar os resultados, e por fim concluir o estudo.

Figura 10 - Guia sistemático para o desenvolvimento de revisões de literatura



Fonte: Okoli; Duarte apud: Duarte; Mattar (2019)

A Figura 10 mostra sistematicamente o processo de desenvolvimento de uma RSL, subdividindo as etapas de planejamento, seleção das bases de estudos, extração dos dados e a execução.

No caso do estudo presente, a RSL foi aplicada para explorar os trabalhos relacionados aos objetivos da pesquisa. A princípio, a busca retornou 153 publicações abrangendo os últimos 10 anos, incluindo o ano atual (2014 a 2023).

Foram examinadas três bases de publicações científicas: *Science Direct*, ACM *Digital Library* e Periódicos CAPES. As duas primeiras bases foram escolhidas para a busca por fazerem parte dos principais repositórios de publicações internacionais, e a base Periódicos CAPES por ser a principal fonte de trabalhos nacionais em Língua Portuguesa.

Na Tabela 1 a seguir estão detalhadas as *strings* de busca usadas em cada Base, e logo em seguida o total de trabalhos retornados nas consultas realizadas. A busca de trabalhos por meio da revisão sistemática foi realizada em abril de 2023.

TABELA 1 - Strings de busca para RSL

Base	**String**	**Total**
Science Direct	(“programming language” OR “linguagem de programação”) AND (“learning” OR “aprendizagem”) AND (“code comprehension” OR “compreensão de código”)	39
ACM DL	(“programming language” OR “linguagem de programação”) AND (“learning” OR “aprendizagem”) AND (“code comprehension” OR “compreensão de código”)	108
Periódicos CAPES	(“programming language” OR “linguagem de programação”) AND (“learning” OR “aprendizagem”) AND (“code comprehension” OR “compreensão de código”) AND (“difficulty” OR “dificuldade”)	6

Fonte: Autoria própria (2023)

Após o retorno desses trabalhos, foi necessário realizar um refinamento com o intuito de filtrar os mais relevantes para a pesquisa. O refinamento foi feito a partir dos seguintes Critérios de Exclusão (CE):

- CE1 - Trabalhos que não têm como foco o ensino/aprendizado de linguagens de programação.
- CE2 - Trabalhos que não abordam a compreensão de código por grupos com pouco ou nenhum contato com a programação.
- CE3 - Trabalhos que sejam um resumo ou uma Revisão Sistemática de Literatura.
- CE4 - Trabalhos escritos em idioma diferente do Inglês ou Português.
- CE5 - Trabalhos idênticos ou que tenham versão atualizada (o mais recente deverá ser mantido caso seja relevante a pesquisa).
- CE6 - Trabalhos cuja visualização ou download não estejam disponíveis no momento da pesquisa.

A Tabela 2 apresenta o refinamento da RSL através da aplicação dos Critérios de Exclusão mencionados anteriormente. Finalmente, foram selecionados 39 trabalhos para a avaliação completa, sendo eles 7 publicações coletadas na Base *Science Direct*, 32 na ACM *Digital Library* e 0 no repositório da CAPES.

TABELA 2 - Aplicação dos Critérios de Exclusão

Base	Total	CE1	CE2	CE3	CE4	CE5	CE6	Incluídos
Science Direct	39	8	16	7	0	0	1	7
ACM DL	108	10	42	9	0	3	12	32
Periódicos CAPES	6	2	2	1	0	1	0	0

Fonte: Autoria própria (2023)

Após a revisão completa dos 39 trabalhos incluídos na RSL, com o objetivo de identificar as publicações que tinham seu foco voltado diretamente para o estudo

das dificuldades apresentadas por estudantes novatos na aprendizagem de uma linguagem de programação, sete se destacaram no contexto de análise de componentes fundamentais à compreensão de programas para programadores iniciantes e buscaram evidenciar quais as dificuldades predominantes que esse grupo apresenta ao aprender uma nova linguagem de programação. Dentre os outros 32 trabalhos, a maior parte tinha como objetivo principal analisar uma única estratégia ou aplicar uma ferramenta que auxiliasse na compreensão ou no aprendizado de uma linguagem.

Grande parte dos trabalhos revisados abordaram o estudo de linguagens de programação introdutórias de maneira superficial, direcionando sua pesquisa para o estudo de movimentos oculares ou para a programação colaborativa, voltando-se para a visibilidade do código, portanto, estes não foram utilizados para este estudo.

Ao fim da revisão, foi possível verificar diversas técnicas que podem auxiliar no ensino/aprendizado das linguagens e contribuir para uma melhor compreensão de código. A revisão realizada a partir da RSL permitiu que as questões de pesquisa, definidas no Tópico 4.1, fossem respondidas e a discussão produzida será apresentada no Capítulo de Resultados.

5 RESULTADOS

Neste capítulo, serão apresentados os resultados obtidos a partir da síntese dos dados extraídos dos trabalhos selecionados por meio da Revisão Sistemática da Literatura, e nesta seção serão discutidas as seguintes Questões de Pesquisa:

QGP - Quais são as principais dificuldades no aprendizado de algoritmos enfrentadas por alunos em cursos de Computação que geram impacto nos índices de evasão registrados no curso?

QSP1: Quais os obstáculos mais comuns que os recém ingressantes no curso de Ciência da Computação enfrentam em disciplinas de programação?

QSP2: Qual a relação entre a limitação na aprendizagem em programação e evasão registrada em turmas iniciantes do curso de Computação?

Os resultados obtidos nessa pesquisa são de extrema relevância para a compreensão dos obstáculos enfrentados na aprendizagem da programação e para o avanço na busca de métodos que auxiliem na docência. Através da análise meticulosa dos trabalhos selecionados, foi possível identificar respostas significativas para as questões de pesquisa. A seguir serão explorados os resultados coletados nas literaturas complementando o que já foi apresentado nos capítulos anteriores, buscando relacioná-los com a teoria que fundamenta essa investigação.

QSP1: Quais os obstáculos mais comuns que os recém ingressantes no curso de Ciência da Computação enfrentam em disciplinas de programação?

Portnoff (2018) afirma que há um alto número de programadores novatos que falham em curso introdutórios de programação, e considera esse problema como “um dos grandes sete desafios da educação em Ciência da Computação”. O autor culpa os educadores da área por subestimarem a importância da aplicação de novos métodos para o ensino de linguagens de programação.

Donaldson e Cutts (2018 *apud* Du Boulay, 1986) identificaram que a falta de habilidade na compreensão de código, a sintaxe e a semântica de uma linguagem estão entre as principais dificuldades enfrentadas por iniciantes na programação, alegando que são fatores consideráveis para resultados de aprendizagem

insatisfatórios. Tshukudu e Cutts (2020) analisaram a transferência do conhecimento conceitual/semântico ao aprender novas linguagens e ressaltaram que o processo de abstração de dados, como na programação Orientada a Objetos, é particularmente desafiador para alunos com conhecimento introdutório.

Contudo, Dümmel *et al.* (2018) declara que a maior parte das dificuldades enfrentadas pelos alunos está relacionada à resolução de problemas, e que estes apresentam dificuldades em transferir o conhecimento adquirido anteriormente para problemas novos e desconhecidos. Os autores alegam ainda que a falta de experiência com as ferramentas para as construções da linguagem prejudicam ainda mais o desempenho dos estudantes.

A realização de um curso preliminar foi vista por Dümmel *et al.* (2018) como uma boa alternativa, pois além de introduzir conceitos básicos aos alunos, serviu também como filtro para estudantes que decidiram não continuar por não se identificarem com a área da Computação. Neste estudo, o trabalho em dupla auxiliou os alunos a compartilharem suas dificuldades com o outro e também a ajudar nas dificuldades da sua dupla, reduzindo o desequilíbrio entre alunos mais habilidosos e menos habilidosos.

Izu *et al.* (2019) ressaltam a importância da abstração na compreensão de código, destacando a segmentação do código e a identificação de componentes chaves como auxílios para uma melhor compreensão. Ainda segundo os autores, as dificuldades podem ser subdivididas em quatro áreas: Representações conceituais sobre o dispositivo de computador (estruturas de controle que interrompem a linearidade do texto); variáveis; estruturas e representações de dados; métodos de programação (estratégias adequadas para resolver os problemas)

As habilidades em iteração e manipulação de vetores são essenciais para cursos introdutórios de programação. Visto isso, Izu, Weerasinghe e Pope (2016) realizaram a aplicação de um questionário para realizar avaliação no domínio das habilidades básicas em uma turma de programadores iniciantes, no qual identificaram diversos erros de sintaxe e lógica, além da falta de conhecimento da semântica de loops, havendo muitos loops infinitos ou vazios. Izu, Weerasinghe e Pope (2016) ainda sugerem que uma boa alternativa para a detecção de erros é o rastreamento do código em um papel.

Apesar de nenhum trabalho incluído na revisão ser direcionado para a abordagem de gamificação, notou-se que essa estratégia vem ganhando espaço no ensino da programação em escolas primárias. Diversos estudos também analisam as influências da introdução de diagramas UML para a compreensão de código (GRAVINO, SCANNIELLO, TORTORA, 2014; YANG, LEE, CHANG, 2018).

QSP2: Qual a relação entre a limitação na aprendizagem em programação e evasão registrada em turmas iniciantes do curso de Computação?

Para Giraffa e Moraes (2016) a permanência ou não de um aluno em uma disciplina ou até mesmo em um curso não pode ser ligada apenas a um único fator. Porém, a falta de tempo para realizar atividades propostas por professores e a dificuldade presente desde o ingresso dos alunos na compreensão dos conteúdos relacionados à Algoritmos e Programação foram apontados como fatores determinantes para a evasão dos mesmos nos cursos de Computação.

Além disso, as taxas de evasão nesses cursos estavam acima da média nacional levando em consideração as outras áreas do conhecimento entre os anos 2010-2014, conforme apresentado por Hoed (2016). É importante ressaltar que a análise da evasão não foi feita em um período de tempo mais recente por falta de estudos relacionados que abrangesse todo o Brasil.

Após a análise dos estudos coletados por meio da RSL, foram discutidos os principais obstáculos encontrados no ensino-aprendizado da programação, algumas formas e métodos usados para a realização dessas análises e foi debatida a relação entre a limitação na aprendizagem de disciplinas de programação com a evasão registrada nos cursos de Computação. Observou-se um aumento no interesse em projetos visando o ensino de programação para crianças e adolescentes, visível nos trabalhos de Donaldson e Cutts (2018); Dümmel, Westfechtel e Ehmann (2018); Martin, Hughes e Richards (2017) e Portnoff (2018).

Os dados de evasão em cursos superiores da área de Computação que sustentaram essa pesquisa foram extraídos da análise feita por Hoed (2016), que comparou a evasão nos cursos de Computação com a média nacional das outras áreas do conhecimento em instituições públicas e privadas separadamente. Em

ambos os casos a evasão nos cursos de Computação mostrou-se acima da média de evasão nacional, contudo, a evasão em Instituições privadas apresentou resultados superiores aos das Instituições públicas, mostrando que esse problema vai além da desigualdade na educação fornecida a diferentes classes sociais.

Direcionar estudos para o ensino da programação posteriormente ao ingresso em cursos superiores pode a longo prazo acabar reduzindo significativamente os problemas encontrados na aprendizagem de linguagens de programação no decorrer dos cursos de Ciência da Computação. Também foi possível observar uma busca expressiva por ferramentas e abordagens que auxiliem na obtenção de resultados e que visem trazer melhorias para a prática pedagógica nessa área.

6 CONCLUSÃO

Com a motivação de compreender as dificuldades na compreensão de código, referente ao aprendizado de Algoritmos, e o impacto na evasão em cursos de Computação, foi desenvolvida uma Revisão Sistemática da Literatura, voltada para a análise de trabalhos dos últimos 10 anos (2014 a 2023), direcionados ao ensino-aprendizado da Programação. Os trabalhos foram extraídos das bases *Science Direct*, ACM *Digital Library* e Periódicos CAPES, resultando em 153 publicações, das quais, sete trabalhos foram selecionados como norteadores para responder às questões de pesquisa.

Na investigação dos principais fatores que dificultam a progressão de alunos iniciantes no estudo da programação e de algoritmos computacionais, foram identificadas que as principais dificuldades estão relacionadas à sintaxe e semântica das linguagens, a falta de conhecimento das ferramentas utilizadas e a falta de habilidade para compreensão de código para a resolução de problemas.

Foi possível perceber que esses obstáculos, apesar de não se ter um confirmação de que são a causa principal, estão sim relacionados com o alto índice de evasão que os cursos da área de Computação apresentam, portanto é um tema que necessita ter maior visibilidade e busca de soluções por parte dos pesquisadores. De acordo com o que foi apresentado, ficou visível a necessidade de ações por parte dos professores, como realizar atividades simples que possam ser realizadas em sala de aula para auxiliar no desenvolvimento da habilidade de compreensão de código e procurar maneiras de estimular os estudantes a se engajarem mais no aprendizado das linguagens de programação. Dar foco na resolução do problema inicialmente por Algoritmos, sem focar nas linguagens em si pode ser também uma alternativa para melhorar a resolução de problemas.

Notou-se que a maior parte dos trabalhos analisados na RSL estão voltados para a prática pedagógica, ou seja, direcionados aos professores, e que o interesse em projetos visando o ensino de programação para crianças e adolescentes tem ganhado bastante notoriedade. Ainda a partir da análise bibliográfica, foram identificadas técnicas e ferramentas que podem ser utilizadas para auxiliar no ensino

da programação, sendo as principais citadas: programação em blocos, cursos introdutórios preliminares, desenvolvimento de ambientes de aprendizagem, correção de código com erros, gamificação e introdução de diagramas UML.

Foram observadas no decorrer da pesquisa possíveis limitações e ameaças que podem ter interferido nos resultados dessa Revisão Sistemática da Literatura. Uma limitação a ser destacada refere-se ao fato da pesquisa ter sido realizada em um período muito restrito de tempo e isso afetou o desenvolvimento do trabalho, tanto na busca quanto na avaliação de um conjunto maior de publicações científicas. Outro ponto em destaque é que, mediante a RSL, poucos trabalhos nacionais foram retornados na busca, e todos eles foram removidos da análise devido a aplicação dos critérios de exclusão da revisão.

É esperado que essa pesquisa sirva de apoio para futuros estudos direcionados à criação ou avaliação de ferramentas que contribuam para o processo de ensino/aprendizagem da programação, ressaltando-se que é de suma importância conhecer as principais dificuldades que os grupos de alunos novatos enfrentam na área da Computação. Além disso, o desenvolvimento de estratégias voltadas para a gamificação mostrou-se um caminho possível a ser explorado para aprimorar o ensino da programação. Espera-se também que a revisão bibliográfica realizada colabore com a descoberta de alternativas dedicadas à diminuição das taxas de evasão nos cursos da área da Computação, declarando-se a pretensão de desenvolver, em trabalhos futuros, um estudo de caso para analisar mais profundamente possíveis soluções para a evasão de alunos em Ciência da Computação.

7 REFERÊNCIAS

BRIZOLA, J.; FANTIN, N.. Revisão da literatura e revisão sistemática da literatura. **Revista de Educação do Vale do Arinos-RELVA**, v. 3, n. 2, 2016 *apud* MORANDI e CAMARGO (2015).

DE JESUS, H. O.; RODRIGUEZ, L. C.; COSTA, A. O. Jr. Predição de Evasão Escolar na Licenciatura em Computação. **Revista Brasileira de Informática na Educação**, v. 29, p. 255-272, 2021.

DONALDSON, P.; CUTTS, Q. Flexible low-cost activities to develop novice code comprehension skills in schools. **In Proceedings of the 13th Workshop in Primary and Secondary Computing Education.** Association for Computing Machinery, New York, NY, USA, Article 19, 1-4, 2018.

DRESH, A.; LACERDA, D. P.; ANTUNES, J. **Design Science Research**: Método de Pesquisa para Avanço da Ciência e Tecnologia. Edição 1. Bookman. Editora Grupo A. 2015.

DÜMMEL, N.; WESTFECHTEL, B.; EHMANN, M. Effects of a preliminary programming course on students' performance. **In Proceedings of the 3th European Conference of Software Engineering Education.** Association for Computing Machinery, New York, NY, USA, 77-86. 2018.

Evasão in.: Dicio, Dicionário Online de Português. Porto: 7Graus, 2023. Disponível em: https://www.dicio.com.br/evasao/ Acesso em: 14/06/2023.

FALCKEMBACH, G. A. M.; ARAUJO, F. V. Aprendizagem de Algoritmos: Dificuldades na resolução de problemas. **Anais SULCOMP**, Congresso Sul Brasileiro de Computação, v. 2, 2006.

FERRARI, F.; CECHINEL, C. Introdução a algoritmos e programação. **Bagé: Universidade Federal do Pampa**, 2008.

FREIRE, L.; COUTINHO, J.; LIMA, V.; LIMA, N. Uma proposta de encontros de tutoria baseada em Metodologias Ativas para disciplinas de Programação Introdutória. In: **Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação**. p.298, 2019.

GIRAFFA, M. M.; MORAES, M. da C. Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. **Congresso CLABES**, 3 nov. 2016.

GOMES, A., HENRIQUES, J., MENDES, A. J. (2008). "Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores". **Educação, Formação & Tecnologias**-ISSN 1646-933X, vol.1(1), pp. 93-103. Disponível em http://eft.educom.pt

GOMES, M.; BECKER, L.; GESTARO, L.; AMARAL, E.; TAROUCO, L. M. R. Um estudo sobre erros em Programação: reconhecendo as dificuldades de programadores iniciantes. **Anais dos Workshop do IV CBIE,** I Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação, 2015.

GOTARDO, R. A. **Linguagem de Programação I**. Edição 1. Editora Estácio. 2015.

GRAVINO, C.; SCANNIELLO, G.; TORTORA, G. Source-code comprehension tasks supported by UML design models: Results from a controlled experiment and a differentiated replication. **Journal of Visual Languages & Computing**, v. 28, p. 23 - 38. 2015.

HEVNER, A. R.; MARCH, S. T.; PARK, J.; RAM, S. **Design Science in Information Systems Research**. MIS Quarterly, vol. 28, n. 1, pp. 75-105. 2004.

HOED, R. M. **Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de computação**. 2016. Dissertação (Mestrado Profissional em Computação Aplicada) — Universidade de Brasília, Brasília, 2016.

IZU, C.; SCHULTE, C.; AGGARWAL, A.; CUTTS, Q.; DURAN, R.; GUTICA, M.; HEINEMANN, B.; KRAEMER, E.; LONATI, V.; MIROLO, C.; WEEDA, R. Fostering Program Comprehension in Novice Programmers - Learning Activities and Learning Trajectories. **In Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education**. Association for Computing Machinery, New York, NY, USA, 27–52. 2019.

IZU, C.; WEERASINGHE, A.; POPE, C. A Study of Code Design Skills in Novice Programmers using the SOLO taxonomy. **In Proceedings of the 2016 ACM Conference on International Computing Education Research**. Association for Computing Machinery, New York, NY, USA, pages 251–259. 2016

KRZYZANOWSKI, L.; BELETI, C. Jr; SANTIAGO, R. Jr; TOSTES, R. A. Ensino de programação - Um estudo preliminar nos cursos de licenciatura em Computação no Brasil. **Anais dos Workshop do CBIE**, 2019.

OBAIDELLAH, U.; MOHAMMED, A. H.; CHENG, P. "A Survey on the Usage of Eye-Tracking in Computer Programming." **ACM Computing Surveys**, volume 51, pages 1-58. 2018.

OKOLI,C., DUARTE, T. por: DUARTE, D. W. A., & Mattar, J. **Guia Para Realizar uma Revisão Sistemática de Literatura**. *EaD Em Foco*, v. 9, n. 1. (2019). https://doi.org/10.18264/eadf.v9i1.748

PORTNOFF, S. R. **The introductory computer programming course is first and foremost a language course**. ACM Inroads v. 9, issue 2, p. 34–52. 2018.

SILVA FILHO, R. L. L.; MOTEJUNAS, P. R.; HIPÓLITO, O.; LOBO, M. B. C. M. (2007). A evasão no ensino superior brasileiro. **Cadernos de Pesquisa**, 37(132), 641-659.

TSHUKUDU, E.; CUTTS, Q. Understanding Conceptual Transfer for Students Learning New Programming Languages. **In Proceedings of the 2020 ACM Conference on International Computing Education Research**. Association for Computing Machinery, New York, NY, USA, 227-237. 2020.

WHALLEY, J.; KASTO, N. How difficult are novice code writing tasks? a software metrics approach. **In Proceedings of the Sixteenth Australasian Computing Education Conference** - Volume 148. Australian Computer Society, Inc., AUS, pages 105–112. 2014.

YANG, J; LEE, Y; CHANG, K. H. Evaluations of JaguarCode: A web-based object-oriented programming environment with static and dynamic visualization. **Journal of Systems and Software**, v. 145, pages 147 -163. 2018.

ZACARIAS, R. O.; MELLO, D. R. B. METODOLOGIAS DE ENSINO DE LÓGICA DE PROGRAMAÇÃO E ALGORITMOS EM CURSOS DE GRADUAÇÃO. **Revista Interdisciplinar Pensamento Científico**, v. 5, n. 2, 29 dez. 2019 *apud* BELCHIOR *et al.* (2016).