Digital Educational Resources in Programming Education: A Systematic Review of Resource Types and Teaching Challenges

Ana Paula Silva Paulina Bezerra

Graduate Program in Computer Science, State University of Rio Grande do Norte (UERN) and Federal Rural University of the Semi-Arid Region (UFERSA), Brazil

Gabryella Rocha Rodrigues

Federal Institute of Pará (IFPA), Brazil

Ceres Germanna Braga Morais

Department of Informatics, State University of Rio Grande do Norte (UERN), Brazil
Graduate Program in Computer Science, State University of Rio Grande do Norte (UERN) and Federal Rural University of the Semi-Arid Region (UFERSA), Brazil

Abstract

Teaching programming requires the use of different digital educational resources to support the understanding of abstract concepts and the development of problem-solving skills. However, these resources are often dispersed across multiple platforms, which makes their selection, evaluation, and reuse by teachers more difficult. This article presents a Systematic Literature Review on the use of digital educational resources in programming education, focusing on the types of resources employed, as well as their sources and reported difficulties. The review was conducted based on a structured protocol, including the definition of search strings, the selection of databases, and the application of inclusion and exclusion criteria. After the quality assessment was applied, 27 primary studies composed the final analysis set. The results reveal a predominance of practice-oriented resources, such as programming exercises and interactive environments, along with the use of multimedia materials and reusable content. Despite this diversity, significant challenges persist, particularly those related to resource selection, pedagogical integration, material quality, and teachers' workload. These findings highlight the need for more structured approaches to support teachers in the effective use and organization of digital educational resources in programming education.

2012 ACM Subject Classification Applied computing → Education; Information systems → Information retrieval; Computing methodologies → Knowledge representation and reasoning

Keywords and phrases Digital Educational Resources, Programming Education, Resource Types, Teaching Challenges, Resource Selection

Digital Object Identifier 10.4230/OASIcs.ICPEC.2026.12

Supplementary Material *Text*: `https://drive.google.com/open?id=1WMa233ZR-LRSCwsrNX7SMtLi9cgrP3-1&usp=drive_fs` [27]

Funding The authors acknowledge the financial support provided by the Graduate Program of the State University of Rio Grande do Norte (UERN) and Federal University of the Semi-Arid Region (UFERSA) for this submission.

1 Introduction

Programming education plays a central role in the training of professionals in Computing, following the growing demands of the labor market and society for digital competencies and problem-solving skills. To support this process, different Digital Educational Resources (DERs) have been used in educational contexts, including interactive platforms, practical exercises, videos, programming environments, simulators, automatic graders [25], educational games, and multimedia materials [2, 7].

However, the use of these resources in programming courses involves specific pedagogical challenges, especially due to the abstract nature of computational concepts and the need to integrate theory and practice [21]. International curricular guidelines, such as those proposed by the Association for Computing Machinery (ACM), highlight that Computing education should be guided by fundamental competencies, such as computational thinking, problem solving, and software development [4]. However, since these guidelines do not specify particular tools, they transfer to teachers the responsibility for selecting and adapting materials.

More recently, the advancement of Artificial Intelligence (AI)-based technologies, such as intelligent tutoring systems, automatic content recommendation, and generative AI tools [31], has expanded the possibilities for personalization and automated support. Despite this rich ecosystem, most materials are dispersed across multiple platforms and lack pedagogical standardization [29, 12]. This fragmentation makes it difficult to locate and reuse resources, increasing teachers' workload due to manual curation processes.

Although there are studies focused on the development of specific tools for programming education, there is a predominance of research centered on isolated technologies, such as interactive environments, automatic assessment systems, practice platforms, or AI-based applications [1, 31]. In contrast, investigations that analyze, in an integrated way, the different types of digital educational resources used in programming education, as well as the challenges faced by teachers in selecting, organizing, and integrating these materials, remain limited.

In this context, this paper presents a Systematic Literature Review (SLR) with the objective of investigating how programming teachers select, use, and produce digital educational resources, as well as mapping the types of resources employed in programming education, considering their formats and sources. In addition, it seeks to identify the main difficulties faced by teachers in selecting, adapting, and using these resources in their educational contexts.

This study contributes by synthesizing and organizing evidence from the literature, offering a structured view of the use of digital educational resources in programming education and the challenges associated with their use.

2 Related Work

2.1 Interactive Environments and Practice-Oriented Resources

Recent studies have explored interactive environments and practice-oriented resources as strategies to support programming education. Hidayat et al. [14] investigated *live coding* platforms associated with gamification to increase engagement in Object-Oriented Programming courses. Similarly, Buffardi and Wang [7] analyzed the integration of videos directly into programming practice interfaces, seeking to reduce the cognitive load caused by switching between instructional materials and coding environments.

Other studies have explored tools focused on the construction of interactive code narratives and the integration between theory and practice. Al-Gahmi et al. [2] highlighted the use of *Jupyter Notebooks* to support content modularization and practical learning. Overall, these studies emphasize resources that promote immediate feedback, active learning, and experimentation during the code development process.

2.2 Artificial Intelligence-Based Technologies in Programming Education

Recent literature also points to a growing trend in the use of Generative Artificial Intelligence (AIGC) technologies in programming education. Abolnejadian et al. [1] investigated the use of ChatGPT as a personalized tutor through *prompt engineering* strategies, highlighting its potential to provide individualized support and automated feedback to students. Similarly, Xie and Chen [31] discussed the application of generative AI in the teaching of the C programming language, using the *Outcome-Based Education* (OBE) model to prioritize the development of computational thinking rather than focusing exclusively on language syntax.

These studies highlight the potential of AI to support learning personalization, content recommendation, and automated assistance during programming practice. However, most research focuses primarily on the effectiveness of these technologies and their impact on student performance, paying less attention to aspects related to the organization, curation, and pedagogical integration of digital educational resources [11, 12].

2.3 Repositories, Open Educational Resources, and Resource Organization

Regarding the authorship and sharing of Open Educational Resources (OER), studies such as Chen and Nunes [8] explore active learning through the creation of videos by students themselves (*Stubents*). Staubitz et al. [29] and Fleenor and Carroll [12], in turn, focus on interoperability and the creation of repositories of automatically graded exercises, aiming to support scalability in large classes.

2.4 Research Gap

Despite these contributions, most studies focus on isolated tools or methodologies. Table 1 summarizes the main approaches and highlights the positioning of this study in relation to the literature.

Although the studies cited contribute to the understanding of specific scenarios, the fragmentation of the literature makes practical adoption by teachers more difficult. Much of the research focuses only on the effectiveness of tools, leaving aside the teacher's perspective in the selection process. Thus, gaps remain regarding:

- The organized classification of the different types of digital educational resources (DERs) available for programming education;
- The sources and repositories that organize these resources;
- The real difficulties faced by teachers in selecting, adapting, and using these materials in their classes.

Thus, this research seeks to address this gap through a Systematic Literature Review (SLR). The study not only maps the types of resources, but also investigates the challenges related to their organization, offering an integrated view to support teachers in decision-making.

Table 1 Summary of related studies.

Study	Type of Resource	Focus / Aspect Investigated
Hidayat et al. [14]	Live Coding Platform	Gamification and motivation in OOP teaching.
Al-Gahmi et al. [2]	Jupyter Notebooks	Code narratives and reduction of cognitive load.
Buffardi and Wang [7]	Integrated Videos	Immediate practice and visual support in coding.
Abolnejadian et al. [1]	Generative AI (ChatGPT)	Adaptive learning through a personalized AI tutor.
Xie and Chen [31]	Generative AI (AIGC)	Competency-based curriculum reform (OBE).
Staubitz et al. [29]	Exercise Repository	Interoperability and automatic grading of assignments.
Chen and Nunes [8]	Videos (Stubents)	Student authorship and peer-based active learning.
Oliveira et al. [11]	OER Platform	Collective intelligence for resource curation.
This study	**Multiple DERs**	**Integrated analysis of resource types, sources, and teaching challenges.**

3 Methodology

This study was conducted as a Systematic Literature Review (SLR), following the guidelines proposed by Kitchenham and Charters [19]. The review protocol is available for consultation[1]. The objective was to identify and analyze the main data sources and materials used in programming education, the types of educational resources employed, and the difficulties reported in teaching practice. In addition, the review sought to understand how these materials are selected, organized, and used.

3.1 Research Questions

The review was guided by the following research questions:

- RQ1: How are digital educational resources used in programming education made available and organized?
- RQ2: What types of digital educational resources, considering their formats and media, are used in programming education?
- RQ3: What difficulties do teachers face in selecting, using, and producing digital educational resources for programming education?

To answer these questions, different dimensions related to the use of digital educational resources in programming education were analyzed, including resource sources, forms of organization, pedagogical practices, types of materials, and teaching challenges.

[1] https://docs.google.com/document/d/1Ty0Q-h4xdHNUPdFtUM8eBleB7EaiR7ZsWKpJvtV8mgg/edit?tab=t.0

3.2 Search Strategy

The search was conducted in IEEE Xplore, ACM Digital Library, ScienceDirect, and Web of Science.

The search string was defined by combining terms related to programming education, digital educational resources, and the teaching context, as follows:

```
(teacher OR educator)
AND ("educational resources" OR "learning materials")
AND (repository OR platform)
AND ("teaching programming" OR programming OR "programming education")
```

3.3 Inclusion and Exclusion Criteria

Table 2 presents the criteria adopted for selecting the studies.

Table 2 Inclusion and exclusion criteria adopted in the SLR.

Inclusion Criteria (IC)	Exclusion Criteria (EC)
Available in full text.	Not available in full text or duplicated studies.
Published in English or Portuguese.	Published in languages other than English or Portuguese.
Peer-reviewed studies.	Grey literature.
Published between 2016 and February 2026.	Outside the defined time frame.
Address the use, selection, production, or organization of educational materials and resources for programming education.	Do not address programming education or focus exclusively on student learning without discussing materials or teaching practices.
Describe or analyze sources, platforms, repositories, or digital environments for programming education materials.	Address only programming tools without an educational purpose.
Target audience composed of teachers, instructors, or educators involved in programming education.	Studies focused exclusively on other audiences, without implications for teaching practice.

The inclusion and exclusion criteria were established to ensure the selection of relevant, current studies aligned with the scope of the research. These criteria considered aspects related to publication quality, the educational context addressed, and adherence to the topic of digital educational resources in programming education, with a focus on teaching practice.

3.4 Study Selection Process

The selection process was carried out in four successive stages to ensure methodological rigor and accuracy in narrowing down the results. The detailed selection flow, from initial identification to the final sample, is illustrated in the PRISMA diagram shown in Figure 1.

1. **Initial Screening and Duplicates:** The automated search returned 434 studies. After identifying and removing 3 duplicates, 431 articles remained for analysis.
2. **Title and Abstract Screening:** The 431 studies were screened by title and abstract based on the inclusion and exclusion criteria (IC and EC). At this stage, 186 articles were pre-selected for the eligibility phase.
3. **Eligibility and Admissibility Assessment:** The 186 articles were submitted to a technical analysis focused on their adherence to the research questions (RQs). At this stage, the introduction, methodology, and conclusion sections were read to verify whether

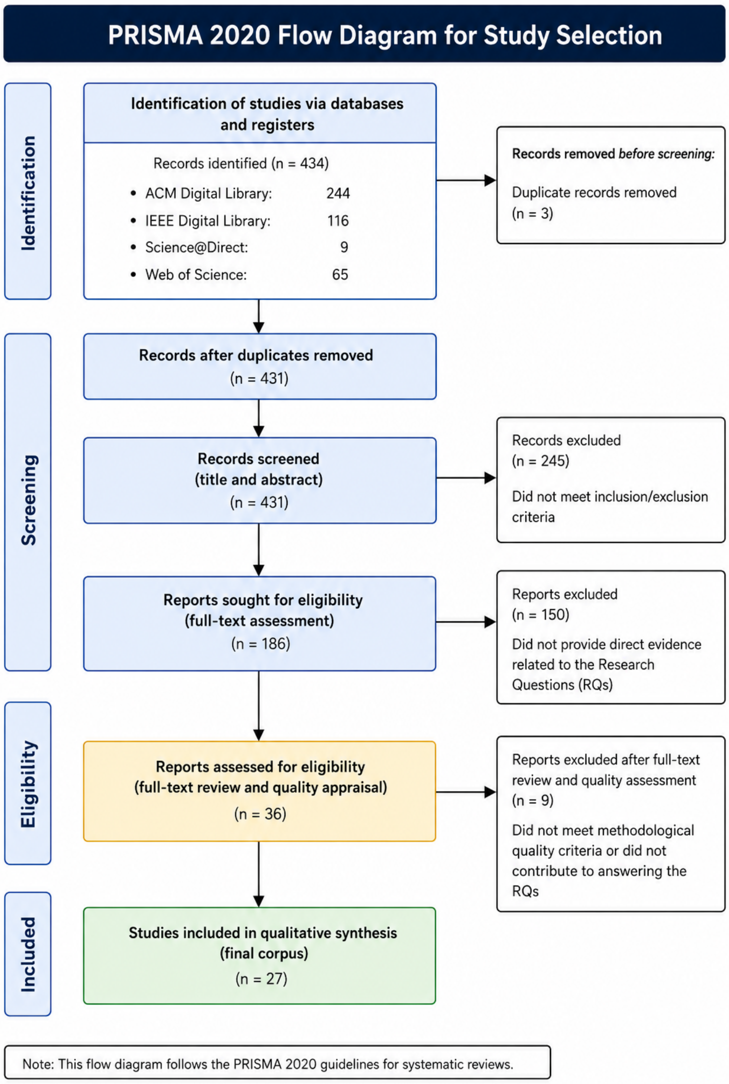

Figure 1 PRISMA flow diagram of the study selection process.

each study provided data on resource types or teaching challenges. This filter led to the exclusion of 150 articles that did not provide direct evidence for the RQs, leaving 36 studies for in-depth analysis.

4. **Full Critical Reading and Synthesis:** The 36 candidate studies were submitted to a full critical reading. After this detailed analysis and the application of the Quality Assessment (QA) protocol, 9 articles were excluded for not reaching the cut-off score. As a result, 27 primary studies composed the final corpus of this Systematic Literature Review.

3.5 Quality Assessment

To ensure that the final synthesis included only studies with methodological rigor and thematic relevance, the 36 articles that reached the full-reading stage were evaluated using a checklist composed of seven criteria.

Each criterion was scored objectively: 1.0 (Yes/Fully meets the criterion), 0.5 (Partially), or 0.0 (No/Not reported).

- QA1 – Does the study clearly present its research objectives?
- QA2 – Does the study describe the educational context in which programming education takes place, such as course type, educational level, or digital environment?
- QA3 – Does the study describe the types of educational materials or resources used for programming education?
- QA4 – Does the study report the sources, platforms, or repositories used to obtain or make the materials available?
- QA5 – Does the study address aspects related to teacher training or difficulties faced by teachers in programming education?
- QA6 – Does the study clearly describe the methodology used, such as case study, survey, experiment, or document analysis?
- QA7 – Does the study present relevant contributions to teaching practice or to the use of digital tools in programming education?

The final score was obtained by summing the points, with a maximum score of 7.0. A cut-off score of 4.0 was established. This final stage resulted in the exclusion of 9 articles that, although related to the topic, did not reach the required level of detail or methodological rigor, consolidating 27 primary studies for the final analysis.

For these studies, data extraction focused on information about types of educational resources, teaching practices, platforms used, and reported difficulties. The data were analyzed through qualitative synthesis and descriptive statistics, allowing the results to be organized according to the research questions defined in this review.

4 Results

This section presents the results obtained from the analysis of the 27 primary studies selected in the review, considering the time frame from 2016 to 2026 adopted in the search strategy. Although the defined period covers ten years, no eligible studies published in 2016 or 2026 were identified after applying the inclusion and exclusion criteria.

Regarding publication sources, there was a predominance of studies indexed in the ACM Digital Library, which accounted for 55.6% of the selected publications, followed by IEEE Xplore, with 37.0%. Web of Science represented 7.4% of the analyzed studies, indicating a higher concentration of research in the areas of Computing, Educational Technologies, and Digital Learning Environments. As for temporal distribution, publication growth was observed over the years, especially from 2021 onward. The year 2025 recorded the highest volume, with **18.5%** of the sample. The years 2022 and 2024 each accounted for **14.8%**, while 2018, 2021, and 2023 each represented **11.1%**. The years 2017 and 2020 each accounted for **7.4%**, and 2019 had the lowest frequency, with **3.7%** of the selected articles.

4.1 Sources, Organization, and Good Practices in the Use of Digital Educational Resources (RQ1)

The results show that the use of Digital Educational Resources (DERs) in programming education involves different resource sources, technological platforms, and pedagogical strategies. Figure 2 synthesizes these elements through a conceptual model that shows how different resource sources, such as LMSs, MOOCs, programming environments, digital repositories, and artificial intelligence-based tools, are made available through platforms and educational environments that enable their access and use.

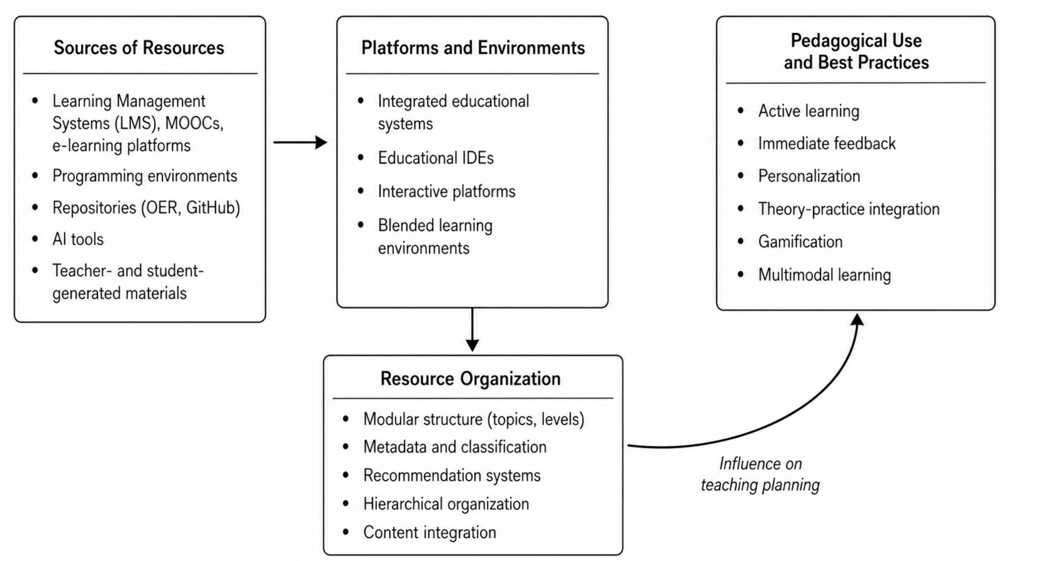

Figure 2 Conceptual model of the digital educational resource ecosystem in programming education, highlighting the relationship between resource sources, educational platforms, organization mechanisms, and their impact on pedagogical use.

The analyzed studies show that these resources are frequently used in integrated platforms, such as educational IDEs [22, 28], interactive environments, and automatic feedback systems [30, 20], supporting practices related to active learning, immediate feedback, teaching personalization, and integration between theory and practice.

Regarding resource sources, there is a predominance of digital teaching platforms, such as Learning Management Systems (LMSs) [1, 17, 30], e-learning environments [18, 3, 8], and MOOCs [6, 26, 13], as well as programming environments, digital repositories, such as OER and GitHub [12, 24, 20], and, more recently, artificial intelligence-based tools. The production of materials by teachers and students also stands out, indicating practices of adaptation and authorship within the educational context.

However, the results indicate that the pedagogical use of these resources depends directly on how materials are organized in these environments. Aspects such as modular structure, hierarchical organization, use of metadata, and recommendation systems [11, 29, 28] were identified as important elements for facilitating search, retrieval, reuse, and pedagogical integration of materials.

As shown in Figure 2, although platforms enable access to resources, the organization of materials has a direct influence on pedagogical practices and on the planning of teaching activities. This result suggests that the challenges faced by teachers are not only related to resource availability, but mainly to how these materials are structured and integrated into educational contexts.

4.2 Types of Digital Educational Resources (RQ2)

The diversity of digital educational resources identified in the selected primary studies ($N = 27$) reveals a scenario strongly oriented toward practice and interactivity in programming education. Since the articles could be classified into multiple categories simultaneously, the accumulated percentages reflect the coexistence of different media and formats in the educational ecosystems analyzed. Figure 3 presents the categorized distribution of the main types of resources.

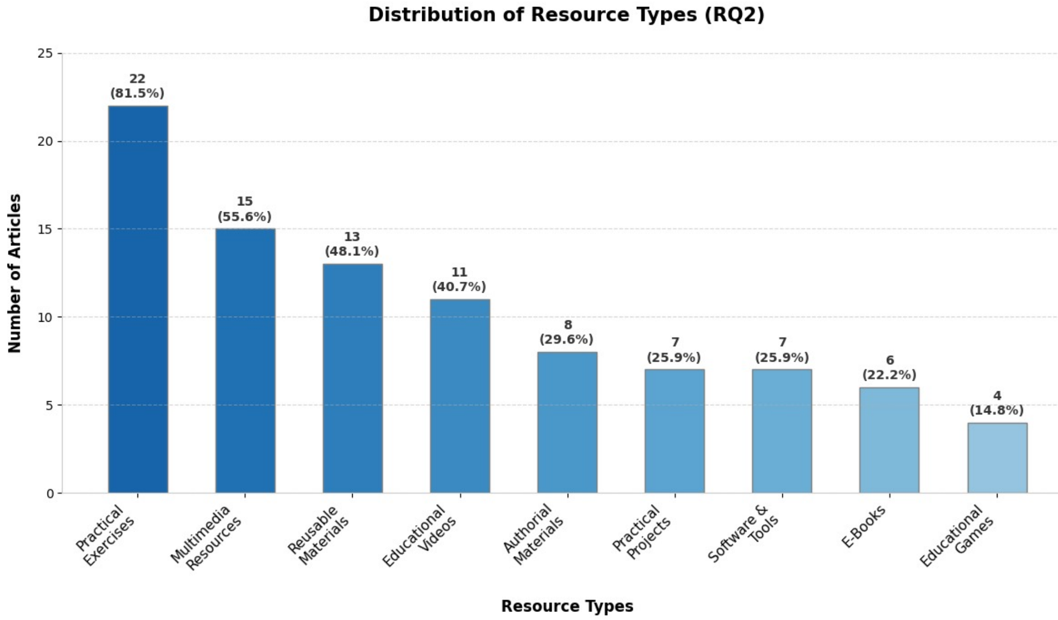

Figure 3 Distribution of resource types identified in the primary studies (N = 27, multiple categories per article permitted).

Among the most recurrent resources, practical exercises stand out, appearing in **81.5%** of the studies, highlighting the centrality of approaches based on problem solving and code implementation [12, 15]. This result reinforces the predominance of pedagogical strategies aligned with active learning, in which students take a more participatory role in the knowledge construction process.

Multimedia resources, which include structured texts, PDFs, images, and audio materials, appear in **55.6%** of the studies. These resources play an important complementary role, acting as theoretical support that precedes or accompanies programming activities [11, 23]. Similarly, reusable materials, such as structured courses, shared source code, and open repositories, were mapped in **48.1%** of the cases, indicating a strong tendency in the academic community to adapt and reuse pre-existing artifacts, especially those hosted on collaborative platforms such as GitHub [6, 28].

Educational videos are present in **40.7%** of the studies and are widely used to support hybrid models, synchronous moments of *live coding*, and asynchronous teaching strategies [8, 26]. Author-created materials, which include artifacts directly produced by teachers, such as *slides* and quizzes, appear in **29.6%** of the studies. This reflects teachers' efforts to personalize and align expository content and reading checks with the specific objectives of their classes [16].

Less frequently, but with similar pedagogical relevance, practical projects and the use of software and simulations were both identified in **25.9%** of the articles [9, 10]. These formats usually require greater student autonomy and are applied in contexts focused on real-world scenarios or the development of complex systems [30]. Finally, *e-books*, with **22.2%**, and educational games, with **14.8%**, complete the distribution, the latter being restricted to specific dynamics of gamification and the introduction of abstract concepts for beginners [2, 18, 14].

Overall, the data point to the consolidation of learning ecosystems based on media hybridization. Although this diversity of formats enriches the student experience, it also transfers to teachers the critical challenge of articulating and integrating these different resources into a coherent and manageable curriculum design.

4.3 Challenges in the Use of Digital Educational Resources (RQ3)

The difficulties reported in the literature show that the use of digital educational resources in programming education involves challenges that go beyond access to technologies, encompassing pedagogical, organizational, and technical aspects. Figure 4 presents the distribution of the main categories of difficulties identified.

Among the most recurrent challenges, those related to teacher training and traditional methods, as well as technological and tool limitations, stand out equally, both appearing in **37.0%** of the studies [2, 8, 20]. This result indicates, on the one hand, the need for continuous training to support the integration of technologies and practice-oriented approaches and, on the other hand, the obstacles imposed by complex platforms.

Next, difficulties associated with the quality and updating of materials were mapped in **33.3%** of the studies [16, 17]. This points to the rapid obsolescence of technological content and to the need for careful evaluation of the sources used. Additionally, the alignment between theory and practice and the adequacy of resources to students' level were reported in **29.6%** of the cases [5, 16], suggesting that resource effectiveness depends critically on contextualization and on the ability to manage the cognitive complexity of abstract concepts.

With the same frequency of **25.9%**, three interconnected bottlenecks emerge: resource organization and standardization, workload and time required for support, and student engagement and learning. This group of results reveals a challenging cycle for teachers: the possible lack of standardized repositories generates fragmentation [29], which substantially increases the time spent structuring classes and providing individual feedback [12], directly affecting the ability to mitigate low student engagement in the face of traditional methodologies.

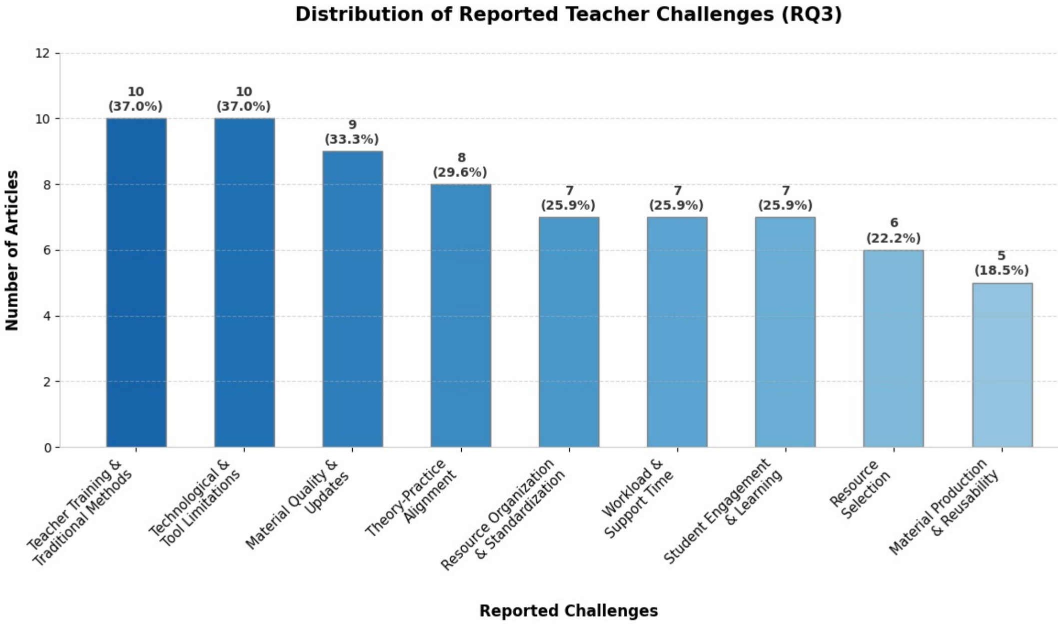

Figure 4 Distribution of difficulties reported in the analyzed studies.

Less frequent, but still limiting, are challenges related to resource selection, identified in **22.2%** of the studies, which expose bureaucracy and difficulties in searching government portals or open repositories [11]. Finally, problems related to the production and reuse of materials appear in **18.5%** of the studies, revealing barriers such as lack of modularity, which hinder the efficient sharing of artifacts among different authors and institutional contexts [28].

Overall, the results indicate that the main obstacles involve the organization and integration of resources into the teaching process.

5 Discussion

The analysis of the studies reveals a scenario marked less by the scarcity of digital educational resources and more by the complexity involved in their pedagogical use. Although there is a wide diversity of materials and platforms, the central challenge lies in selecting, organizing, and integrating these resources into teaching contexts.

This result is consistent with the literature on Digital Educational Resources (DERs), which indicates that their effectiveness depends not only on availability, but also on adequate description, organization, and pedagogical alignment. As discussed in the related work, the use of metadata is essential to support the retrieval and reuse of educational materials [11, 29, 28]. However, the findings of this review indicate that, in practice, significant limitations remain, especially those related to lack of standardization and difficulty in locating suitable resources.

Regarding the sources and forms of material availability (RQ1), the results show that resources are distributed across multiple platforms, such as virtual learning environments, open repositories, development environments, and artificial intelligence-based tools. Although this diversity expands pedagogical possibilities, it also contributes to the fragmentation of the teaching process, requiring teachers to act as integrators of different technologies. This scenario reinforces the need to consolidate structured mechanisms that support the organization and integration of these resources.

In relation to the types of resources used (RQ2), the predominance of practical and interactive activities confirms the emphasis, present in Computing education guidelines, on practice-based learning and problem solving. These guidelines advocate the integration of theory and practice as an essential element for the development of programming competencies. However, the results show that this integration remains a challenge in real teaching contexts, especially when resources are used in an isolated or disconnected manner.

The difficulties identified (RQ3) help explain this scenario. Issues related to teacher training, material quality, and the integration between theory and practice show that the use of digital resources is not only a technological issue, but also a pedagogical and organizational one. The recurrent need for continuous teacher training indicates a gap between the demands of programming education and the support available for teachers.

In addition, aspects such as teachers' workload and tool limitations reinforce that the adoption of digital resources can increase the complexity of teaching when adequate support is not available. This factor is directly related to resource fragmentation and the absence of organization standards, which makes reuse and adaptation more difficult.

In this context, the results of this review reinforce the need for more structured approaches to support the use of digital educational resources in programming education. Initiatives based on ontologies, structured repositories, and recommendation systems appear promising because they can contribute to improving the organization, retrieval, and pedagogical alignment of resources [11, 29, 28].

Thus, more than increasing the number of available materials, it is necessary to invest in strategies that enable their effective integration and use in the teaching and learning process.

6 Study Limitations

Although this review followed a structured protocol for searching, selecting, and analyzing studies, some limitations should be considered. The search was conducted only in specific databases and considered studies published between 2016 and February 2026, written in English or Portuguese. This may have restricted the scope of the results and introduced linguistic and geographical limitations.

In addition, the search strategy prioritized terms related to the pedagogical context of programming education, which may have excluded part of the technical literature on specialized tools used as digital educational resources. The categorization of resources, pedagogical practices, and reported difficulties also involved qualitative interpretation of the analyzed studies, which may introduce some subjectivity into the classification process.

To reduce possible biases, inclusion and exclusion criteria, quality assessment criteria, and data categorization procedures were defined in advance. Despite these limitations, the results provide a relevant overview of the use of digital educational resources in programming education and the challenges associated with teaching practice.

7 Conclusion

This systematic review showed that the main challenge in the use of digital educational resources in programming education is not centered on their availability, but on how these resources are organized and integrated into pedagogical practices.

Although the literature reveals a significant diversity of materials and approaches, their effective use still depends on factors such as teacher training, pedagogical alignment, and the ability to articulate different types of resources. In this context, teachers play a central role not only as users, but also as mediators and integrators of educational technologies.

The findings also indicate that the fragmentation of resources across multiple platforms, combined with the absence of standards for organization and description, makes retrieval and reuse more difficult, limiting their educational potential. This scenario reinforces the need for solutions that go beyond simply making materials available and that include mechanisms to support their structuring and pedagogical use.

In this sense, this systematic review is not limited to synthesizing the literature, but also provides a conceptual and empirical basis for the development of a repository of digital educational resources for programming education. The results obtained guide the definition of requirements for this repository, especially regarding the organization of materials, the use of metadata, and the need for recommendation mechanisms and support for teaching practice.

As its main contribution, this study highlights a gap between the availability of digital resources and their effective pedagogical application, indicating the need for approaches based on semantic organization. The use of ontologies, in this context, emerges as a promising strategy to structure, classify, and retrieve resources more efficiently.

As future work, the development and validation of an ontology-based repository is proposed, with the aim of supporting teachers in the selection and integration of digital educational resources and contributing to the reduction of complexity in programming education.

Thus, more than increasing the number of available resources, it is essential to advance in how these resources are organized and used, in order to enhance their impact on the teaching and learning process.

References

1. Mohammad Abolnejadian, Sharareh Alipour, and Kamyar Taeb. Leveraging chatgpt for adaptive learning through personalized prompt-based instruction: A cs1 education case study. In *Extended abstracts of the CHI conference on human factors in computing systems*, pages 1–8, 2024. doi:10.1145/3613905.3637148.
2. Abdulmalek Al-Gahmi, Yong Zhang, and Hugo Valle. Jupyter in the classroom: An experience report. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*, pages 425–431, 2022. doi:10.1145/3478431.3499379.
3. Rahadyan Fannani Arif, Harits Ar Rosyid, and Utomo Pujianto. Design and implementation of interactive coding with gamification for web programming subject for vocational high school students. In *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, volume 6, pages 177–182. IEEE, 2019.
4. Association for Computing Machinery. Computing curricula 2020: Paradigms for global computing education, 2020. Accessed on: Apr. 30, 2026. URL: https://www.acm.org/education/curricula-recommendations.
5. Lorena A Barba. Engineers code: reusable open learning modules for engineering computations. *Computing in Science & Engineering*, 22(4):26–35, 2020. doi:10.1109/MCSE.2020.2976002.
6. Anastasiia Birillo, Mariia Tigina, Zarina Kurbatova, Anna Potriasaeva, Ilya Vlasov, Valerii Ovchinnikov, and Igor Gerasimov. Bridging education and development: Ides as interactive learning platforms. In *Proceedings of the 1st ACM/IEEE Workshop on Integrated Development Environments*, pages 53–58, 2024. doi:10.1145/3643796.3648454.
7. Kevin Buffardi and Richert Wang. Integrating videos with programming practice. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 1*, pages 241–247, 2022. doi:10.1145/3502718.3524778.
8. Yige Chen and Bernardo Pereira Nunes. Students: Videos created by and for students, active learning resources in large and diverse computer science classrooms. In *Proceedings of the 2024 Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2024)*, pages 653–659, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3649217.3653569.

9 Hugo Costelha and Carlos Neves. Technical database on robotics-based educational platforms for k-12 students. In *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 167–172. IEEE, 2018. doi:10.1109/ICARSC.2018.8374178.

10 Josivan Pereira Da Silva, Paulo Henrique Gonçalves Pimentel, Luciano Gonçalves Pimentel, and Ismar Frango Silveira. Pixel python rpg: repurposing an entertainment game to an open educational resource for computer programming fundamentals. In *2021 XVI Latin American Conference on Learning Technologies (LACLO)*, pages 326–333. IEEE, 2021.

11 Marcela Ribeiro de Oliveira, Israel Barreto Sant'Anna, Guilherme Scariot Ramos, Luis Carlos Erpen De Bona, Marcos Alexandre Castilho, Marcos Didonet Del Fabro, and Eduardo Todt. Open educational resources platform based on collective intelligence. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*, pages 346–353. IEEE, 2018. doi:10.1109/CIC.2018.00053.

12 Hillary Fleenor and Hyrum D Carroll. Creating an oer collection of automatically scored practice exercises for computer science 1. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 1280–1280, 2020.

13 Elena Harris and Richert Wang. Codewit. us: A platform for diverse perspectives in coding. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*, pages 780–786, 2022.

14 Wahyu Nur Hidayat, Wiji Dwi Prasetyo, Ayu Najmatus Zahiroh, Esa Pramudheva Maydi Syahri, Mochammad Mu'iz Afdloly, and Rahajeng Kartika Sari. Development of gamification and live coding-based programming learning platform to foster learning motivation of vocational students. In *2023 International Conference on Electrical and Information Technology (IEIT)*, pages 299–304. IEEE, 2023.

15 Muntasir Hoq, Atharva Patil, Kamil Akhuseyinoglu, Peter Brusilovsky, and Bita Akram. An automated approach to recommending relevant worked examples for programming problems. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, pages 527–533, 2025. doi:10.1145/3641554.3701951.

16 Wenhui Hou and Xiaohan Yu. Exploring the blended teaching mode of python programming language course based on obe concepts. In *Proceedings of the 2nd Guangdong-Hong Kong-Macao Greater Bay Area Education Digitalization and Computer Science International Conference*, pages 434–439, 2025.

17 Xiaoyan Jian, Feilong Li, and Zhaohong Liu. A study on the application of aigc context in the development of digital resources for online courses. In *Proceedings of the 2024 International Conference on Artificial Intelligence and Teacher Education*, pages 136–141, 2024.

18 Diamantis Kintsakis and Maria Rangoussi. An early introduction to stem education: teaching computer programming principles to 5 th graders through an e-learning platform: a game-based approach. In *2017 IEEE Global Engineering Education Conference (EDUCON)*, pages 17–23. IEEE, 2017. doi:10.1109/EDUCON.2017.7942816.

19 Barbara Kitchenham and Stuart Charters. *Guidelines for performing systematic literature reviews in software engineering.* Citeseer, 2007.

20 Andrew Meads, Yu-Cheng Tu, and Gillian Dobbie. Moving a bootcamp-style computer science programme online: An experience report. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, pages 785–791, 2023. doi:10.1145/3545945.3569814.

21 Ceres Germanna Braga Morais. *Ensino e aprendizagem de programação: estudo de caso no Ensino Superior.* Tese de doutoramento em ciências da educação, especialidade em tecnologia educativa, Universidade do Minho, Braga, Portugal, 2021.

22 Nektarios Moumoutzis, George Boukeas, Vassilis Vassilakis, Nikos Pappas, Chara Xanthaki, Ioannis Maragkoudakis, Antonios Deligiannakis, and Stavros Christodoulakis. Design, implementation and evaluation of a computer science teacher training programme for learning and teaching of python inside and outside school: Establishing and supporting code clubs to learn computer programming by self-contained examples. In *Interactive Mobile Communication, Technologies and Learning*, pages 575–586. Springer, 2017. doi:10.1007/978-3-319-75175-7_56.

23 İbrahim Halil Özdemır, Tarık Kişla, and Serdar Stefano Tito. Mobile programming with kuika: A course design. In *2023 32nd Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE)*, pages 1–6. IEEE, 2023.

24 James Paterson, Joshua Adams, Brian Hainey, and Sajid Nazir. A repository of resources and exemplars for the cloud curriculum. In *Proceedings of the 5th Conference on Computing Education Practice*, pages 9–12, 2021. `doi:10.1145/3437914.3437971`.

25 Filipe Portela. A Generative Artificial Intelligence Tool to Correct Programming Exercises. In Ricardo Queirós, Mário Pinto, Filipe Portela, and Alberto Simões, editors, *6th International Computer Programming Education Conference (ICPEC 2025)*, volume 133 of *Open Access Series in Informatics (OASIcs)*, pages 7:1–7:16, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/OASIcs.ICPEC.2025.7`.

26 Xunchao Qiu, Chunyue Zhang, and Libo Zhao. Preliminary study on the online teaching mode of" technical fundamentals of web front-end" based on spoc and wechat. In *Proceedings of the 7th International Conference on Distance Education and Learning*, pages 133–138, 2022. `doi:10.1145/3543321.3543343`.

27 Ana Paula Silva. SLR Protocol. Text (visited on 2026-07-13). URL: `https://drive.google.com/open?id=1WMa233ZR-LRSCwsrNX7SMtLi9cgrP3-1&usp=drive_fs`, `doi:10.4230/artifacts.27126`.

28 Sebastian Speiser. Composition of open educational resources through dynamic linking of modular components. In *Proceedings of the 2024 the 16th International Conference on Education Technology and Computers*, pages 255–260, 2024. `doi:10.1145/3702163.3702423`.

29 Thomas Staubitz, Ralf Teusner, and Christoph Meinel. Towards a repository for open auto-gradable programming exercises. In *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 66–73. IEEE, 2017. `doi:10.1109/TALE.2017.8252306`.

30 Yan Watequlis Syaifudin, Siti Rohani, Nobuo Funabiki, and Pramana Yoga Saputra. Blending android programming learning assistance system into online android programming course. In *2021 9th International Conference on Information and Education Technology (ICIET)*, pages 26–33. IEEE, 2021.

31 Nan Xie and Weimin Chen. Research on teaching reform of aigc empowering c language programming course. In *Proceedings of the 2024 2nd International Conference on Information Education and Artificial Intelligence*, pages 176–181, 2024.