PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO
EDUCAÇÃO
EM FOCO UEMG

Pesquisas experimentais no desenvolvimento do pensamento computacional: um mapeamento sistemático de literatura no ensino de conceitos de computação

Carlos BELETI[1]

Marta SFORNI[2]

Resumo

O Pensamento Computacional, caracterizado como uma forma de pensamento amparado nos fundamentos da Ciência da Computação, tem sido amplamente investigado com a finalidade de se conhecer meios para promovê-lo. Com o objetivo de identificar como tem sido as pesquisas experimentais sobre o desenvolvimento do Pensamento Computacional no ensino de conceitos de Computação, realizamos um mapeamento sistemático de literatura contemplando publicações em línguas portuguesa, inglesa e espanhola. Constatamos que não há um padrão no formato da realização das investigações experimentais, mas estas tendem a ser de curta duração e, apesar das diversas subáreas da Ciência da Computação, a programação tem sido predominantemente utilizada no desenvolvimento do Pensamento Computacional. Concluímos ser necessário a realização de pesquisas experimentais com maior tempo de duração sobre o desenvolvimento desse tipo de pensamento, bem como de investigações acerca dos fundamentos didáticos de uma ação educativa que possa promovê-lo.

Palavras-chave: Educação e Tecnologia. Mapeamento Sistemático de Literatura. Pensamento Computacional.

[1] Master's (2013) in Computer Science, and PhD (ongoing) in Education from the State University of Maringá (UEM). Adjunct Professor at the Federal University of Paraná - Jandaia do Sul Campus. He has experience in the area of Computer Science Education and has researched the development of Computational Thinking based on Historical-Cultural Theory, and its interface between Education and Computing. ORCID: https://orcid.org/0000-0002-0158-8673

E-mail: beleti.junior@gmail.com

[2] PhD in Education from the University of São Paulo (USP) (2003) and Postdoctoral Researcher at the State University of Campinas (Unicamp) (2014). Professor in the Graduate Program in Education at UEM, focusing primarily on the following topics: conceptual learning, didactics, Historical-Cultural Theory, and Activity Theory. ORCID: https://orcid.org/0000-0002-9614-2075

E-mail: martasforni@uol.com.br

PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO
EDUCAÇÃO
EM FOCO | UEMG

Experimental research in the development of computational thinking: a systematic mapping of literature in the teaching of computing concepts

Carlos BELETI

Marta SFORNI

Abstract

Computational Thinking, as a form of thinking supported by the foundations of Computer Science, has been widely investigated with a view to finding ways to promote it. In order to identify how experimental research on the development of Computational Thinking in the teaching of Computing concepts has been developed, we conducted a systematic mapping review of publications in Portuguese, English, and Spanish. We found that there is no pattern in the format of how experimental investigations are conducted, but they tend to be of short duration and, despite the various sub-fields within Computer Science, programming has been predominantly used in Computational Thinking development. We concluded that it is necessary to conduct experimental research with a longer duration on the development of this type of thinking, as well as investigations into the didactic foundations of an educational action that can promote it.

Keywords: Computational Thinking. Education and Technology. Systematic Mapping of Literature.

EDUCAÇÃO EM FOCO | UEMG
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO

Investigación experimental en el desarrollo del pensamiento computacional: un mapeo sistemático de la literatura en la enseñanza de conceptos de computación

Carlos BELETI

Marta SFORNI

Resumen

El Pensamiento Computacional, caracterizado como una forma de pensamiento apoyada en los fundamentos de la Ciencia de la Computación, ha sido ampliamente investigado con el objetivo de encontrar formas de promoverlo. Para identificar cómo ha sido la investigación experimental sobre el desarrollo del Pensamiento Computacional en la enseñanza de conceptos de Computación, realizamos un mapeo sistemático de la literatura de publicaciones en portugués, inglés y español. Encontramos que no hay un estándar en el formato de la realización de las investigaciones experimentales, sino que tienden a ser de corta duración y, a pesar de los diversos subcampos dentro de la Ciencia de la Computación, la programación se ha utilizado predominantemente en el desarrollo del Pensamiento Computacional. Concluimos que es necesario realizar investigaciones experimentales de mayor duración sobre el desarrollo de este tipo de pensamiento, así como investigaciones sobre los fundamentos didácticos de la acción educativa que pueden promoverlo.

Palabras clave: Educación y Tecnología. Mapeo Sistemático de la Literatura. Pensamiento Computacional.

Introduction

The integration of computing and technology into people's daily lives has become commonplace and has intensified in recent years. In various sectors, technological artifacts are increasingly embedded in processes and services. In the educational field, there has been a rise in discussions regarding the inclusion of computational concepts. One concept that has stimulated academic research and investigations in this area is Computational Thinking (CT), a form of thinking that uses concepts from Computer Science to aid in the analysis and solution of problems in various contexts. However, despite the numerous studies on the topic, there is still no consensus in the literature regarding its definition and structure, nor on how to promote its development in students.

In the National Common Curricular Base (BNCC), computational thinking encompasses the abilities to analyze and solve problems through computational concepts, developing skills and capabilities related to various areas (BRAZIL, 2017). The BNCC establishes that computing and technology concepts, as well as their interrelation with CT, should be addressed in teaching in a transversal manner, permeating other areas of knowledge. Recently, CT was included by the Organization for Economic Cooperation and Development (OECD) as a capability to be assessed through the Program for International Student Assessment (PISA) (OECD, 2019).

The Brazilian Computer Society (SBC) has sought to guide and direct computing education in the country, contributing to discussions in the field by proposing guidelines, curricular references, and formative itineraries for teaching computing in Basic Education (EB) (SBC, 2019; SBC, 2018a; RAABE et al., 2017). Regarding CT, SBC emphasizes its importance for integrating computational and technological concepts at all stages of EB, defining it as "[...] the ability to systematize problem-solving activities, represent and analyze solutions through algorithms" and that "[...] requires mastery of abstract objects necessary to describe both information and the processes that manipulate it" (SBC, 2018a, p. 3).

The recent emphasis on the topic of computational thinking is linked to its (re)emergence[3], with Jeannette Wing, in the mid-2000s. According to the author, "[...] computational thinking involves problem-solving, system design, and understanding human behavior based on the fundamental

[3] The term Computational Thinking was first mentioned by Seymour Papert (1980), referring to the possibility of "integrating computational thinking into everyday life" in the sense of involving computational systems in people's activities.

concepts of computer science" (WING, 2006, p. 33, our translation).

Following Wing's definition, numerous studies have investigated the topic (BARR; STEPHENSON, 2011; CSTA; ISTE, 2011; ROYAL SOCIETY, 2012; BRENNAN; RESNICK, 2012; SEITER; FOREMAN, 2013; SELBY; WOOLLARD, 2013; SHUTE; SUN; ASBELL-CLARKE, 2017; HSU; CHANG; HUNG, 2018; PALTS; PEDASTE, 2020), aiming to contribute to research in the field and make Computational Thinking accessible to everyone at all educational levels and stages. Upon reviewing the current literature, we find that the development of Computational Thinking has been extensively researched, generally in studies that conduct experiments or interventions connecting Computational Thinking to programming-related activities. The reviews or mappings found in the literature (GROVER; PEA, 2013; KALELIOGLU; GULBAHAR; KUKUL, 2016; ZHANG; NOURI, 2019; TASLIBEYAZ; KURSUN; KARAMAN, 2020; TIKVA; TAMBOURIS, 2021), although also categorizing data for analysis similarly to this mapping, did not specifically focus on experimental research with students in computing education and the development of computational thinking. They aimed to cover other perspectives, as well as programming environments and tools, simulation and robotics kits; evaluative metrics on the development of Computational Thinking; introduction of Computer Science in Basic Education; definitions and concepts of Computational Thinking; promotion of Computational Thinking skills through Scratch; and the development of conceptual models for Computational Thinking through programming.

Given that experimental research involves direct contact with the concrete reality of Basic Education and can indicate pedagogical pathways for teaching computing at this level of education, we conducted a systematic literature mapping (SLM) with the aim of analyzing how Computational Thinking has been addressed in the teaching of computing concepts in experimental research.

In this article, we describe the methodological procedures and the documentary *corpus* of the SLM, the extraction and analysis of information, present the discussions generated and the results achieved by the investigation, and conclude with the final considerations.

Development

Methodological procedures and documentary *corpus*

According to Kitchenham and Charters (2007), much of the research begins with some form of literature review to analyze gaps and research possibilities in a given area. For the authors, a Systematic Literature Mapping (SLM) is a more elaborate literature review with rigorous methodological precision. From this perspective, this work presents the results of an SLM that is part of a doctoral research project in Education. The study was based on the methodological procedures outlined by Kitchenham and Charters (2007), consisting of the following stages:

1) Research Question and Strategy
2) Selection of Studies
3) Extraction and Analysis of Information
4) Discussion.

We will address the first two stages in this section, with the remaining two stages covered in the subsequent sections.

Research Question and Strategy

Based on published work on the theme of Computational Thinking (CT) in scientific journals, particularly in the fields of Computing and Education, we questioned how CT has been developed in the teaching of computational concepts through experimental research. This inquiry is supported by recent literature reviews in the area (TIKVA; TAMBOURIS, 2021; TASLIBEYAZ; KURSUN; KARAMAN, 2020; ZHANG; NOURI, 2019).

Thus, we defined the following questions:

- Q1: How has Computational Thinking (CT) been addressed in the teaching of computational concepts in experimental research?
- Q2: Which computational concepts have been investigated? At what educational levels and stages are they being applied?

As a research strategy, we defined the database to search for the papers, as well as the composition of the search string. The chosen database was the CAPES[4], Periodicals Portal, which hosts over 200 relevant academic databases, such as ACM, ERIC, IEEE, Wiley, Taylor & Francis, Scopus, SpringerLink, Web of Science, Science Direct, among others.

The portal was accessed through the "my space" option, where users can register and store research information in a personal session. Using this option, we conducted several searches with

[4] Disponível em: https://www.periodicos.capes.gov.br/. Acesso em: 30/12/2020.

keywords such as "computational thinking," "teaching," "learning," "basic education," and "computer science education" in Portuguese, English, and Spanish each word translated into these three languages aiming to create a single search string. However, we found that combining keywords in three languages into one string was unfeasible due to the unique specifics of each language. Additionally, we noticed that the keyword "computer science education" was limiting the number of records returned, so we removed it from the composition. After some searches with the remaining keywords, we proposed the following search string:

- ("pensamento computacional") AND (ensino OR aprend*) AND ((educação) OR ("ensino básico") OR ("educação básica") OR ("ensino fundamental") OR ("ensino médio") OR ("ensino superior")));
- ("computational thinking") AND (teach* OR learn*) AND ((education) OR ("elementary school") OR ("secondary school") OR ("high school") OR ("university education") OR ("higher education")));
- ("pensamiento computacional") AND (enseñ* OR aprend*) AND (educación) OR ("educación primaria") OR ("educación secundaria") OR ("formación profesional") OR ("bachillerato") OR ("enseñanza superior").

On December 30, 2020, the search strings returned 2175 works, being: 27 in Portuguese, 2101 in English and 47 in Spanish. All records were exported by the platform and stored locally for later use.

Selection of papers

To assist in the organization and documentation of the MSL, we used the online tool Parsif.al, developed specifically for this purpose. In addition to systematizing the planning and execution of the mapping, the tool allows for simultaneous sharing of the entire process with other users. Moreover, it enables the planning of all stages of the systematic mapping, as well as the definition of selection criteria – inclusion and exclusion – for each paper. The inclusion criteria were defined as follows:

- Publications, in the form of full journal articles, that present Computational Thinking (CT) in the teaching of computational concepts through experimental approaches.

Regarding the exclusion criteria, they were as follows:

- Publications that were not in Portuguese, English, or Spanish;

- Incomplete publications (without title, abstract, irrelevant or incomplete information);
- Duplicate publications (only one was considered);
- Did not qualify as experimental research or addressed a different research topic.

Once the criteria were defined, the stored records were imported into the tool, which organized them to facilitate the investigation of the works.

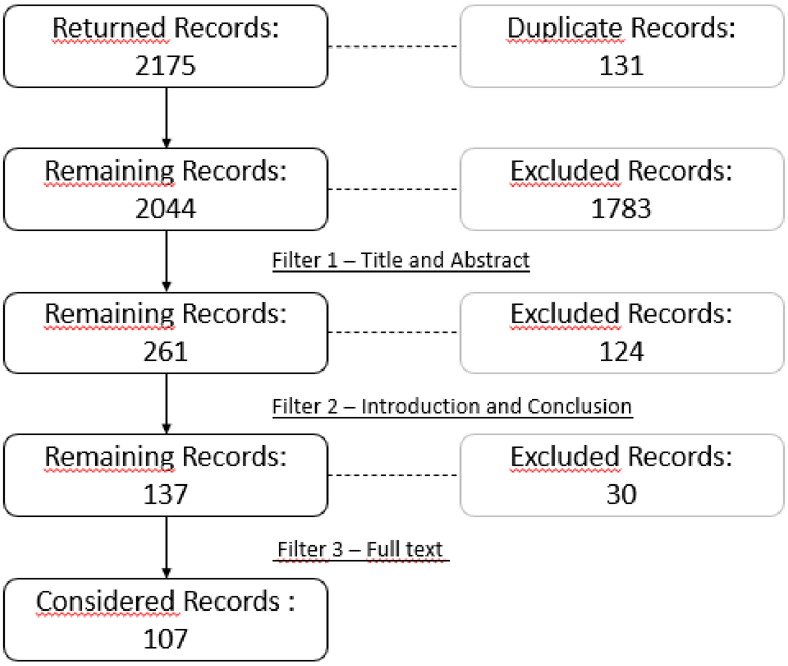
To guide the investigation, the following selection filters were proposed:

- First Filter: Reading the title, abstract, and keywords, evaluating them according to the publication selection criteria.
- Second Filter: Reading the introduction and conclusion, evaluating them according to the publication selection criteria.
- Third Filter: Reading the full publication, evaluating it according to the publication selection criteria.

Before starting the reading of the works, the software itself identified 131 duplicate records, which were immediately removed. The remaining 2044 records were analyzed according to the first filter and the selection criteria, resulting in 261 works. With this quantity, we proceeded with the analysis according to the second filter, from which we obtained 137 works. After a complete reading (third filter), we ended up with 107 works. Figure 1 illustrates the process of analyzing the records to compose the documental *corpus*.

EDUCAÇÃO EM FOCO | UEMG
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO

Figure 1 - Process of Analysis and Composition of the Documental *Corpus*



Source: Prepared by the authors.

Extraction and Analysis of Information

At this stage, we analyzed the selected papers by classifying them into categories to organize the information in a way that contributes to the central investigation of the systematic mapping. The following categories were defined:

- Publication Year: Year the paper was published;
- Country of Origin: Country where the research was conducted;
- Repository and Journal: Repositories and scientific journals where the papers can be found;
- Target Audience Level and Stage: Early Childhood Education, Elementary School – early and late years –, High School, Higher Education/Technical – Teacher Training;
- Teaching and Learning Methodologies: Approaches used in the experiments;
- Experiment Control: Division into control and experimental groups;
- Thematic Area Covered: Subarea or concept of computing addressed;

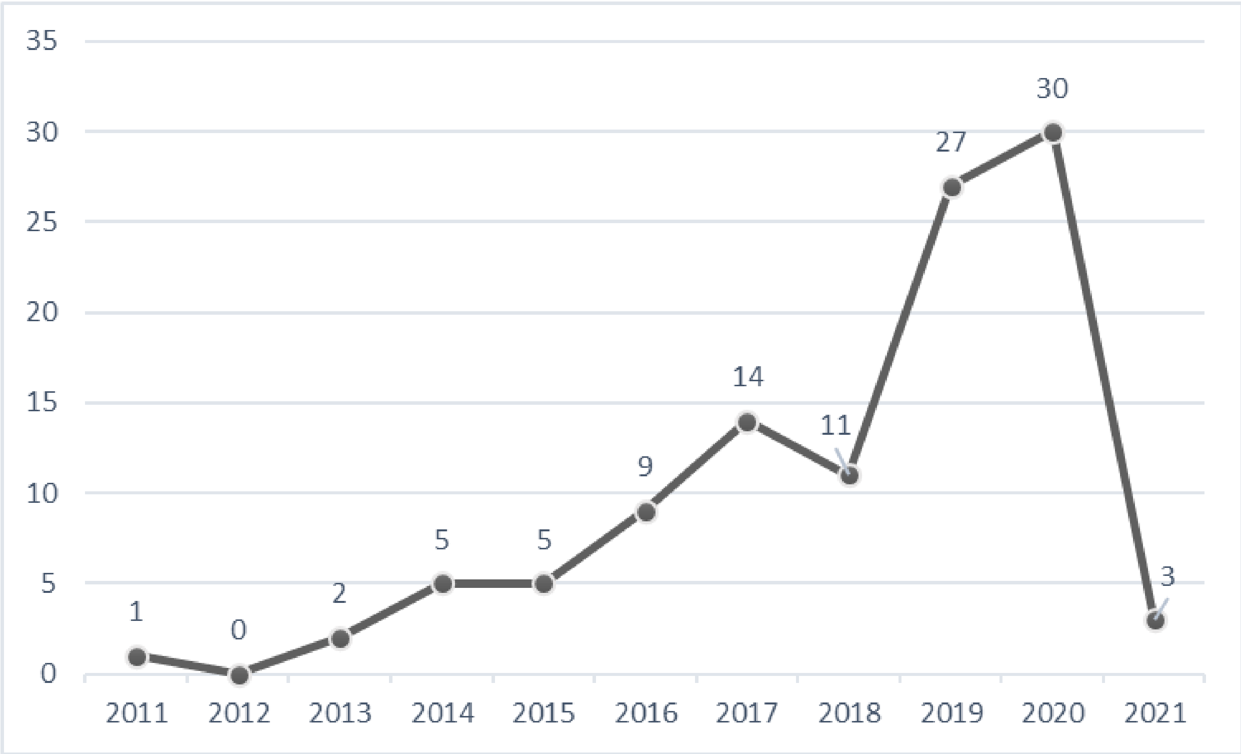
- Method of Promoting PC Development: Artifact used in the approach (software, hardware, unplugged);
- Duration of Intervention: Time spent conducting the experiment (less than 3 hours, 3 to 10 hours, 11 to 20 hours, 21 to 30 hours, more than 30 hours);
- Number of Participants: Number of people involved in the experiment;
- Data Analysis Methodology: Qualitative, quantitative, or mixed;
- Data Collection and Analysis Instrument: Type of instrument used for analysis and data collection;
- Statistical Procedures: Statistical tools used in data analysis;
- Theoretical Foundation: Theoretical background mentioned in the paper.

To proceed with the analysis of the papers, the selected records were exported from Parsif.al into a spreadsheet. For each paper, we reviewed and recorded the information for each category, thus forming the document *corpus*[5] or this research.

Regarding the year of publication, the highest number of selected papers were published in 2020 and 2019, with 30 and 27 papers, respectively. As previously mentioned, the search date on the periodical portal was December 30, 2020, so papers published after this date were not considered. Figure 2 illustrates the classification by year of publication.

[5] The list of all the works in the document *corpus* is available at: https://docs.google.com/spreadsheets/d/1ozMin4cWbEkVjB6pLKANkhwWEDg9UFMCooMvQBcKFrg/.

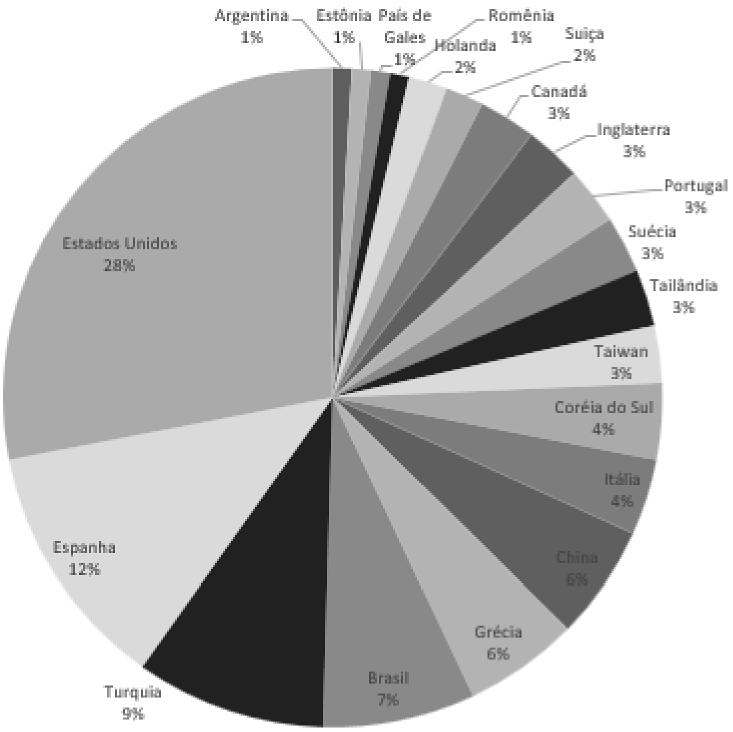
Figure 2 - Selected Articles by Year of Publication



Source: Prepared by the authors.

Regarding the country of publication, the majority of the papers originated from the United States, followed by Spain, Turkey, and Brazil. Similar to other systematic reviews (ZHANG; NOURI, 2019; HSU; CHANG; HUNG, 2018), the United States also appears with a large number of published papers. Figure 3 illustrates the distribution of papers by country.

Figure 3 - Distribution by Country



Source: Prepared by the authors.

Despite the majority of the papers being from the United States and consequently in English, some papers from Brazil were also written in English. This factor has two interesting aspects for analysis. On one hand, it broadens the reach of the research globally, disseminating its results to other countries especially since many internationally relevant journals only accept papers in English. On the other hand, it limits the sharing of information within the country among people who do not speak English, restricting its circulation, for example, in the context of basic education (EB).

Regarding the repository and journal where the papers were hosted, although the search base was the CAPES Journal Portal, most of the papers were in journals indexed in other databases or repositories. The largest quantities of databases and journals are illustrated in Table 1. Elsevier, Springer, ACM, ERIC, and IEEE were the indexing databases that appeared most frequently. The journals "Computer and Education" and "ACM Transactions on Computing Education" had the highest number of published papers, with ten and nine articles, respectively. Other databases and journals can be viewed in the complete listing of the documentary corpus.

Tabela 1 - Repositories or Indexing Databases and Most Frequent Journals

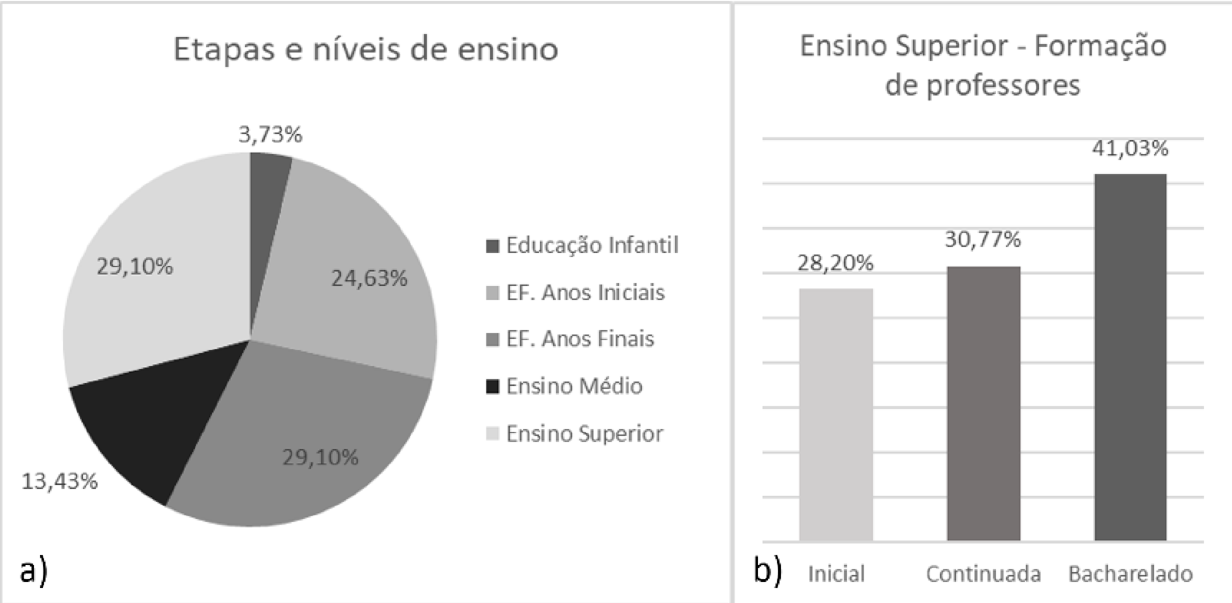
BASE	JOURNAL	QUANTITY
ACM	ACM Transactions on Computing Education	9
ELSEVIER	Computers and Education	10
	Computers in human behavior	6
ERIC	Informatics in Education	5
	International Journal of Computer Science Education in Schools	3
	Journal of Information Technology Education	2
	Participatory Educational Research (PER)	2
IEEE	IEEE Access	3
MDPI	Education sciences	2
REDIB	Revista Observatório	2
SAGE	Journal of educational computing research	3
SPRINGER	Journal of science education and technology	5
	Education and Information Technologies	2
	Educational Technology Research and Development	2
	Technology, Knowledge and Learning	2
	TechTrends	2
WILEY	Computer Applications in Engineering Education	2

Source: Prepared by the authors.

Regarding the level and stage of education in which the studies were conducted, we observed a certain balance between Elementary Education (early and late years) and Higher Education, with fewer studies in Secondary Education and even fewer in Early Childhood Education. We also observed that a single study could address experiments conducted at more than one educational level. Figure 4 – a) details the percentages related to the levels and stages of education.

Specifically regarding Higher Education, we identified a total of 39 studies, which we divided between undergraduate programs and teacher training programs (initial and continuing), as we understand that these are distinct courses with different objectives and therefore require separate analysis. Figure 4 – b) particularly highlights these quantities.

Figure 4 – a) Levels and Stages of Education; b) Teacher Training



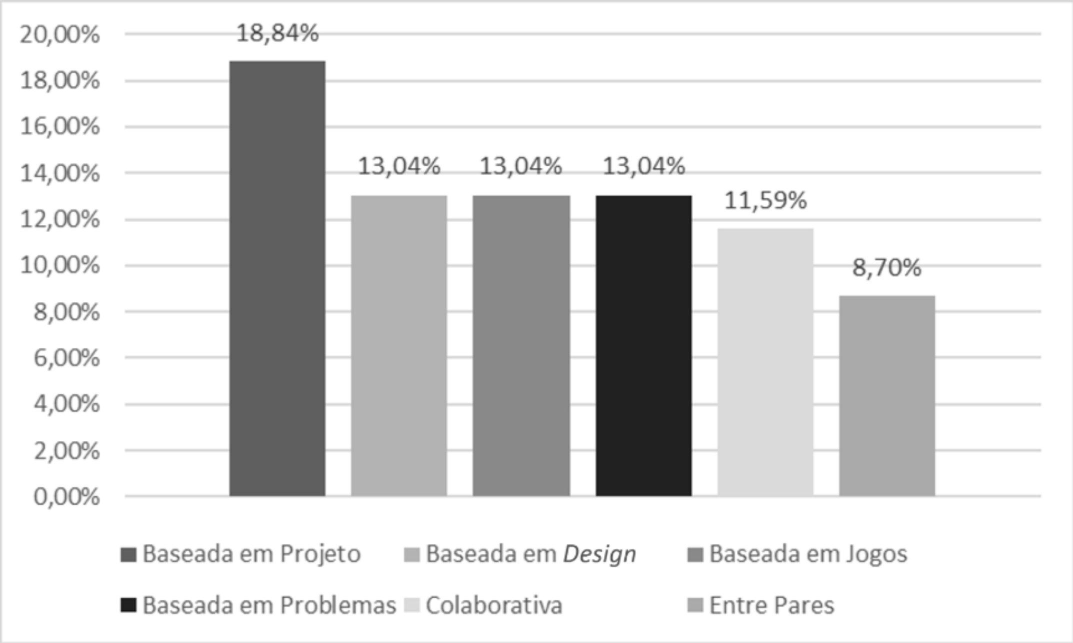
Source: Prepared by the authors.

To analyze the experimental methods used in the research, we examined each study to determine if there was any definition of control over the experiment, such as the establishment of control and experimental groups, with or without random assignment of subjects (experimental or quasi-experimental study). In the studies investigated, the distribution of participants into control and experimental groups occurred in 22 of them (20,56%).

Regarding the teaching and learning methodologies used by the authors of each study, we identified six main methodologies: Project-Based, Design-Based, Game-Based, Problem-Based, Collaborative, and Peer-Based. These methodologies accounted for 78.26% of the 69 approaches declared in the studies. Figure 5 illustrates the percentage of the mentioned approaches.

EDUCAÇÃO EM FOCO UEMG
PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO

Figure 5 - Teaching and Learning Methodologies



Source: Prepared by the authors.

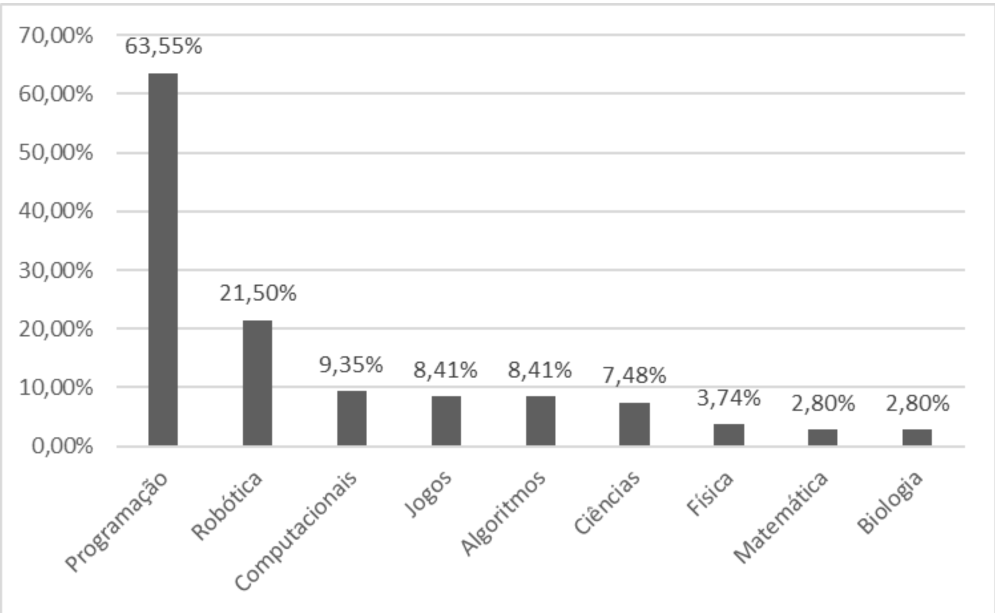
It is important to note that the definitions of the methodologies used were recorded based on the authors' self-description, which sometimes referred to them as teaching approaches, methodologies, or learning methodologies. Additionally, some studies reported using one or more methodological approaches. Other approaches, not mentioned here, appeared in only one study each.

Regarding the thematic areas of the studies, programming was present in a significant portion of the studies (85.98%), either as a means (instrument) or an end in teaching practices, development, or promotion of computational thinking skills. Even when not considered as a disciplinary content, programming appeared as a study subject, for example, in studies focusing on games or robotics as primary elements. Specifically, programming as a tool to develop or promote computational thinking skills appeared in 63.55% of the studies. Robotics also had a notable presence, reaching 21.50%. The intersection of studies explicitly mentioning both robotics and programming in the development of computational thinking skills amounted to 10.28%. It is emphasized that a division between programming and robotics is not feasible, as robotics cannot be developed without programming; for instance, a robot cannot be operated without programming.

Furthermore, we found research focused on the development of computational thinking through concepts related to the field of Computing, such as data structures, agile methods, logic, Turing machines, artificial intelligence concepts, simulators, and software, collectively reaching 9.35%. Concepts like algorithms and games in the development of computational thinking were also present in some studies, each accounting for 8.41%. We identified studies addressing computational thinking and programming in other fields, such as Science, Mathematics, Physics, Biology, Arts, Journalism, and Financial Education. Although the focus of the mapping was on teaching computing concepts, these studies were selected due to their direct correlation with programming.

Figure 6 presents an overview of the main thematic areas related to the teaching of computing and computational thinking present in the studies.

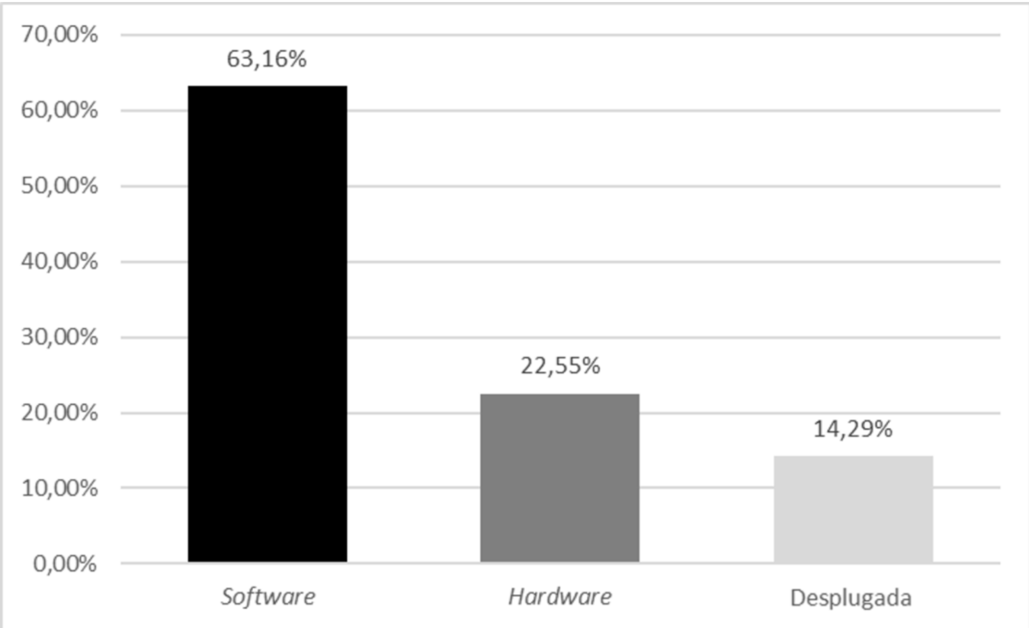
Figure 6 - Thematic Areas Covered



Source: Prepared by the authors.

About the way PC has been addressed in studies, ie, which materials, tools or learning objects have been used in support of the development of the PC, we divide the item into three possibilities, namely: Software - environment, platform or application; Hardware - electronic equipment (computer, tablet, smartphone, programmable board, robotics), and; Unplugged - computing disconnected or disconnected, without the use of hardware or software. Again, we emphasize that every hardware device depends on its software to get up and running, and thus most of the work that made use of hardware was also made use of software. Figure 7 shows the percentages of each subcategory.

Figure 7 - Materials and tools in the approach



Source: Prepared by the authors.

Within each subcategory, we highlight the artifacts that have been used. For example, in "Software", which presents the largest quantity of artifacts in use as a tool in 112 works, we divided again the subcategory into: visual programming languages (blocks) (86), game usage (14), written programming language (9), use of simulations (3), in addition to some works that did not specify the artifact used.

PWe can see that most of the works that utilized software relied on block-based programming platforms. Among these, Scratch[6] appeared in 41.86% (36) of the works, which aligns with findings from other research in this field (ZHANG; NOURI, 2019; HSU; CHANG; HUNG, 2018). Lego's[7] block-based programming software was also present in 13.95% of the works. Although Scratch allows for the creation of games and animations, works that involved playing a game rather than programming it were categorized under "Games." The "Written" category referred to conventional programming languages such as Python and C/C++, which do not use block-based syntax. The "Simulation" category referred to the use of simulators, typically online, where teaching is mediated by reproducing or demonstrating phenomena related to a particular theme.

In the "Hardware" subcategory, the use of robots and robotics kits, such as Lego kits, was predominant, followed by electronic prototyping boards (e.g., Arduino). Robots were more commonly used compared to prototyping boards. Of the 32 works involving robots, Lego kits accounted for 15 (46,88%).

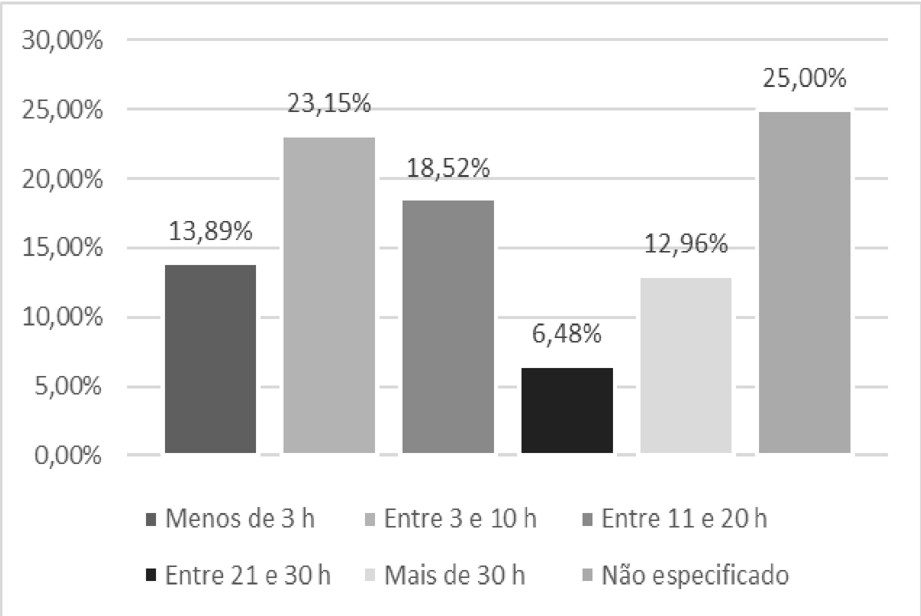
The "Unplugged" subcategory totaled 20 instances across the works, with the artifacts being well distributed. In 6 of these, we categorized some initiatives as "Projects" offering unplugged or disconnected activities, including: Computer Science Unplugged (2) (BELL; WITTEN; FELLOWS, 2011), Bebras Challenge (1) (BEBRAS, n.d.), Kesfet Project (1), Barefoot Computing Project[8] (1), Barefoot Computing Project[9] (1) e Codekinderen Project[10] (1). The use of flowcharts and non-electronic games followed, each appearing 4 times. Additionally, Code.org[11] activities, which, despite being an online platform, offer practices that can be performed disconnected, were also noted. We also found 3 works mentioning unplugged activities without specifying the origin, developed and applied by the authors themselves.

Regarding the duration of the intervention, whether developed in the form of a workshop, course, seminar, or one-time intervention, we divided this category based on the total hours spent: i) intervention with less than 3 hours; ii) intervention between 3 and 10 hours; iii) intervention between 11 and 20 hours; iv) intervention between 21 and 30 hours; v) intervention with more than 30 hours; vi) unspecified duration. As shown in Figure 8, we can see that many works did not specify the

[6] Available at: https://scratch.mit.edu/. Accessed on: May 20, 2021.
[7] Available at: https://www.lego.com/pt-br/themes/mindstorms. Accessed on: May 20, 2021.
[8] Available at: https://www.kesfetprojesi.org. Accessed on: June 10, 2021.
[9] Available at: https://www.barefootcomputing.org. Accessed on: June 10, 2021.
[10] Available at: http://www.codekinderen.nl/leerling/unplugged/index.html. Accessed on: June 10, 2021.
[11] Available at: https://code.org. Accessed on: May 20, 2021.

duration (although some mentioned the total time in days or months, they did not state the total hours). Fewer studies reported interventions lasting between 3 and 10 hours, followed by those between 11 and 20 hours, and then those with less than 3 hours. Interventions with a duration exceeding 30 hours and those between 21 and 30 hours reached 12.96% and 6.48%, respectively.

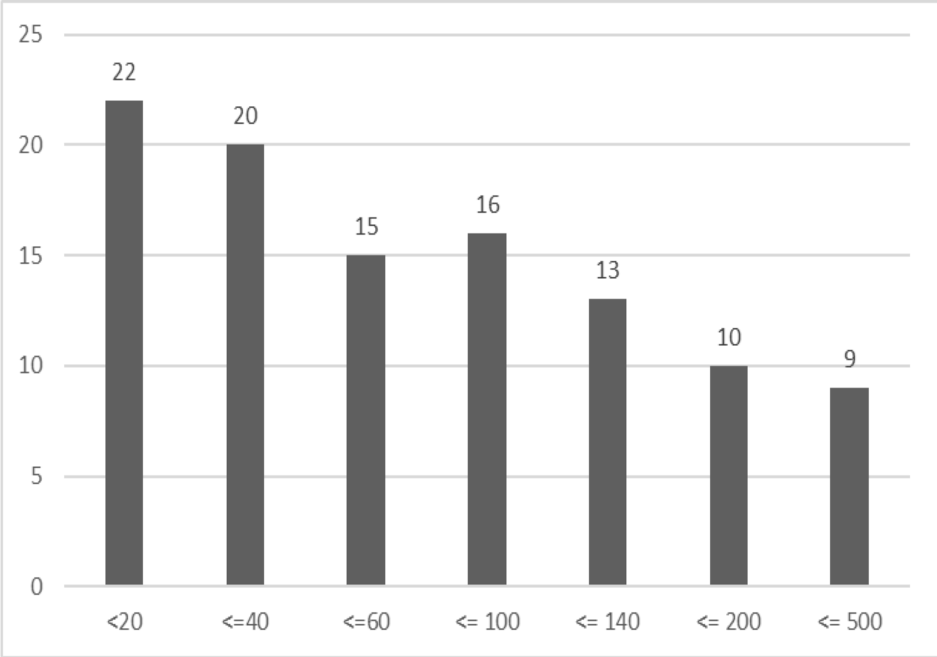
Figure 8 - Duration of the intervention



Source: Prepared by the authors.

The number of participants in the interventions varied widely, ranging from two to 1340 (in just one study). Only one research study did not specify the number of participants. Figure 9 illustrates the number of participants.

Figure 9 - Number of Participants in the Interventions

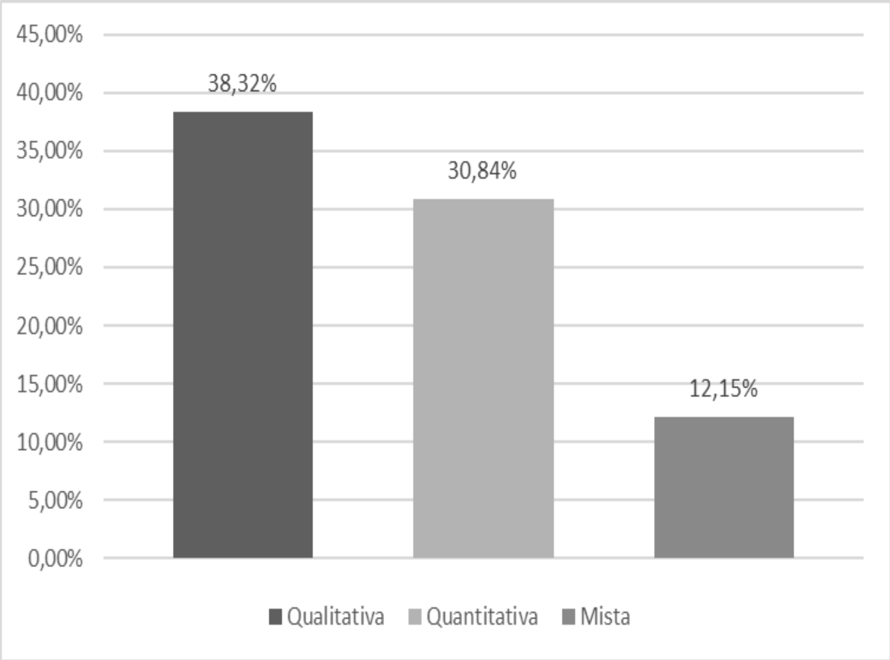


Source: Prepared by the authors.

We can observe that the largest number of participants fell within the range of up to 100, compared to other variations in participant numbers. The interventions generally involved students from regular classes in educational institutions (across various levels) or in teacher training courses.

In terms of data analysis methodologies used in the studies, we found, based on the authors' self-descriptions, that qualitative, quantitative, and mixed methodologies were employed. In total, 87 studies indicated the use of one or more of these methodologies. The percentages for this category are presented in Figure 10.

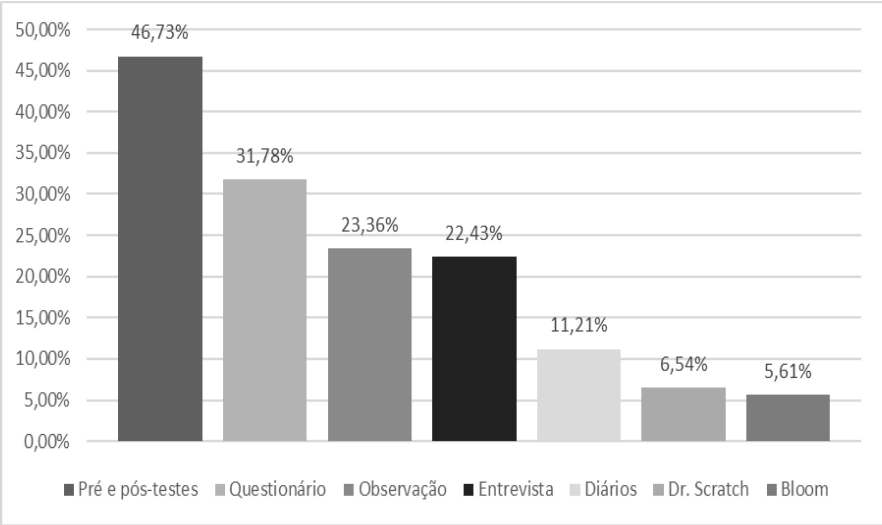
Figure 10 - Data Analysis Methodologies



Source: Prepared by the authors.

Pre- and Post-Tests were the primary instruments for quantitative assessment, appearing in nearly half of the analyzed studies (46.73%). These tests compared performance improvements in activities (learning), self-assessment of performance, attitude, motivation, and self-efficacy, all related to computational thinking (CT). Notably, three of these tests were frequently mentioned: the Computational Thinking Test (Román-González, Pérez-González, & Jiménez-Fernández, 2017), the Bebras Task (Bebras, n.d.), and the Computational Thinking Levels Scale (Korkmaz, Çakır, & Özden, 2017); these were referenced in 20 of the studies that used pre- and post-tests. The Dr. Scratch tool, which automatically evaluates projects from the Scratch platform, was also used in some studies. In qualitative methods, observations (video recordings), questionnaires, interviews, and diaries (notes) were present. The Taxonomy of Educational Objectives (Bloom's Taxonomy) was also investigated in some studies. Figure 11 shows the most commonly used instruments.

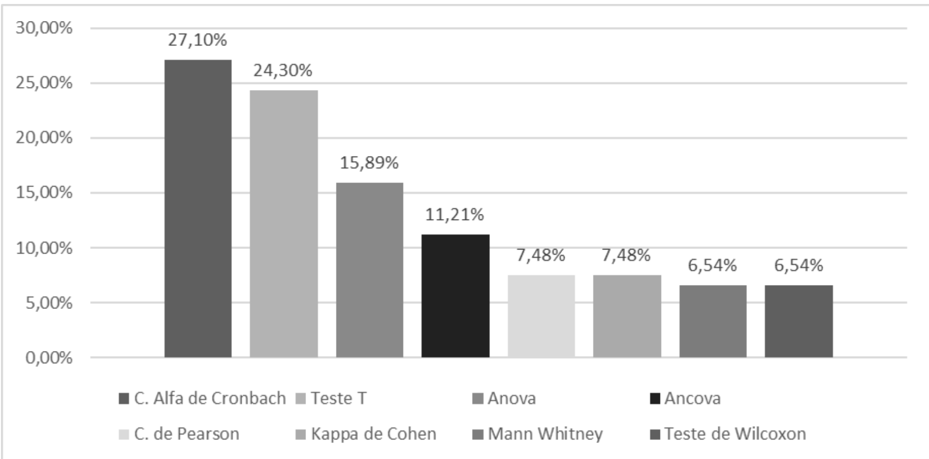
Figure 11 - Data Analysis and Collection Instruments



Source: Prepared by the authors.

In addition to these instruments, both parametric and non-parametric statistical procedures were used in the analysis of results across various studies. The main statistical methods are presented in Figure 12.

Figure 12 - Statistical procedures



Source: Prepared by the authors.

In the analysis of the works that comprised the documentary *corpus* of this research, we observed theoretical foundations in the works of prominent authors on the topic of Computational Thinking (Seymour Papert and Jeannette Wing), as well as references related to theories of teaching, learning, and development in the field of Psychology, such as Jean Piaget and Lev Vygotsky. The motivation for these checks was the assumption that works on Computational Thinking might not be investigating the relationship between Computational Thinking and teaching, learning, and human development. We found that 93.46% of the research referenced some work by Wing. Papert, who some researchers (Nouri et al., 2019; Shute, Sun, & Asbell-Clarke, 2017) consider to be the first to address the concept of Computational Thinking, was mentioned in 60.75% of the analyzed research. Regarding psychological assumptions, only 23.36% referenced authors from this field, with most briefly describing the assumptions of Piaget's and Vygotsky's theories (DÍAZ-LAUZURICA; MORENO-SALINAS, 2019; WU *et al.* 2019).

Discussion and Results

Based on the objectives of this work and aiming to identify similarities between the analyzed categories, we organized the information from the studies into groups to guide the discussion. These groups are:

- Programming as a Fundamental Concept for Developing Computational Thinking (CT);
- Other Computing Concepts Used for Developing CT;
- How Experimental Interventions Have Been Conducted.

The following subsections provide detailed information on each of the groups.

Programming as a Fundamental Concept for the Development of Computational Thinking

This category is linked to the first part of the research question (Q2) of this study: "What concepts of Computing have been investigated?" We hypothesized that programming is the principal Computing concept used in the development of Computational Thinking (CT) in the literature reviewed.

Recent systematic reviews support this hypothesis, showing that programming is often used as a fundamental tool for developing CT skills (TASLIBEYAZ; KURSUN; KARAMAN, 2020; HSU; CHANG; HUNG, 2018). Programming is considered an attractive means to promote CT and is used as a key element in developing conceptual models for education (TIKVA; TAMBOURIS, 2021).

Additionally, programming has been identified as a focus in higher education (LYON; MAGANA, 2020) and the use of block-based programming environments/tools (ZHANG; NOURI, 2019) has been highlighted (ZHANG; NOURI, 2019).

Thus, we investigated how the 68 studies that mentioned using programming in the development of computational thinking (CT) conducted their experiments. Regarding the target audience of the studies grouped in this category, 2.94% were preschool children, 20.59% were early primary school students, and 27.94% were late primary school students. Additionally, 11.76% targeted high school students, 17.65% were in higher education, excluding teacher training programs, which accounted for 19.12%. Despite the majority of the studies (63.24%) targeting primary education students, the inclusion of CT in the Base National Common Curriculum (BNCC) is minimal. Considering that this educational document treats CT as something to be developed, generally related to Mathematics, linked to flowcharts and algorithms, but without specific guidance on how to develop it in primary education (BRASIL, 2017), these investigations could contribute to a closer integration of CT with pedagogical practices at this educational stage.

When analyzing the total number of studies categorized by target audience, higher education and teacher training programs represent the highest quantity, accounting for 64.10% of the 39 studies that used programming for this level. This indicates that programming has been widely used in higher education for the development of computational thinking (CT). Specifically, among these studies, 75% were in bachelor's programs unrelated to teacher training, with 12 out of 16 studies utilizing programming for CT development. These numbers reflect the reality of higher education curricula in Brazil, where the presence of technology or computing concepts in course disciplines often implies programming. This raises the question of whether there is a real need for bachelor’s students to simply learn programming techniques. Shouldn't there be a broader integration of computing and technology concepts, such as the development of a form of thinking based on computer science concepts, like CT?

Regarding the duration of the intervention, most (30.18%) did not specify the workload or the duration of the activities, with 24.53% describing a duration between 11 and 20 hours, and 20.75% conducting their activities within a range of 3 to 10 hours. This indicates that the activities do not have a considerable amount of time allocated, given that this is a relatively new field of knowledge that has not been extensively explored in basic education (BE). Despite its presence in educational curricula in other countries (ROYAL SOCIETY, 2012; HUBWIESER et al., 2015; HEINTZ;

MANNILA; FÄRNQVIST, 2016; CSTA, 2017), in Brazil, computing is not a curricular component of BE. This raises questions about the necessary time for developing computational thinking (CT), even through programming, considering that an introductory programming course at the higher education level, for example, typically has an average of 70 hours of instruction. With this, we ask ourselves: In experiments with reduced time duration, is it possible to learn programming? And regarding CT, is it feasible to develop it in a short period?

As for the tools used in the experiments, we found that 78.57% employed some type of software (environment, platform, or application), 12.50% used hardware, and 8.93% utilized unplugged or disconnected computing as a tool. Among the studies that used software as a tool, more than half (61.36%) employed Scratch. Therefore, Scratch is the most commonly used platform for developing CT when programming is involved, regardless of the target age group. The justification for using this platform, as mentioned in the studies, is its free availability and its approach to teaching programming through blocks rather than written language, which facilitates learning. The tool also includes an artifact that automatically evaluates projects, called Dr. Scratch.

With this context established, and recognizing that programming has been the most used foundation for developing CT, we must revisit a factor briefly mentioned: the teaching of programming as a curricular component in higher education. Undergraduate programs, typically in the exact sciences, include at least one programming course in their curricula, and despite not discussing how it is approached, there are high rates of failure in these courses (SIMON et al., 2019; AURELIANO; TEDESCO; GIRAFFA, 2016). This leads us to reflect on the difficulty level of understanding concepts related to programming. If a higher education student struggles with these concepts, can BE students understand them in short-duration interventions or experiments? Furthermore, is it possible to develop CT through programming learning in such a short time?

Other Computing Concepts Used for the Development of Computational Thinking

This category was established to analyze studies that use other computing concepts or themes for promoting the development of computational thinking (CT), excluding programming. As previously noted, programming, specifically targeted to develop or promote CT skills, was present in 63.55% of the studies, reaching 85.98% when considered not only as content but also as a means to teach robotics or games, for example, since both a game and a robot need to be programmed.

Therefore, we consider here the 14.02% (15) of studies that did not use programming as a concept in their experimental studies on CT development. Despite not mentioning programming, these studies involved themes related to the concept, such as studying algorithms through unplugged or disconnected computing, without using electronic devices. Some of these studies used flowcharts, puzzles, and even a specific type of robot; processes of requirements analysis and evaluation in the software development cycle; work with gameplay in games; use of simulators, illustrations, and animations; project tasks like Kesfet and the Bebras challenge.

The analysis of the studies in this category leads us back to the question (Q2) of this research: what computing concepts have been investigated? That is, while understanding the importance of programming in the field of computing, why is it the most commonly used concept for developing CT? Additionally, what is the importance of other concepts or subfields of computing?

To provide information that could help answer these questions, we sought to identify the fundamental subfields or disciplines of computing by analyzing documents that underpin the creation and maintenance of computing courses. We found references from the Brazilian Computer Society (SBC) for undergraduate computing programs (ZORZO et al., 2017), guided by the National Curriculum Guidelines for Undergraduate Computing Courses (BRASIL, 2016), the SBC reference curricula (SBC, 2005), and the ACM/IEEE curriculum (CC2020 Task Force, 2020).

The most current references (ZORZO et al., 2017), despite being developed based on the notion of competence, present the core computing contents or disciplines. When compared to the knowledge areas in the ACM/IEEE curriculum (CC2020 Task Force, 2020) and the SBC reference curriculum disciplines (SBC, 2005), they may show variations but contain numerous similarities in the proposed disciplines necessary for a computing course.

Based on these references, highlighting the areas of agreement, we present the fundamental disciplines of Computer Science, as cataloged: Algorithms and Data Structures; Algorithm Analysis; Computer Architecture and Organization; Databases; Computer Graphics; Parallel and Distributed Computing; Software Engineering; Artificial Intelligence; Human-Computer Interaction; Programming Languages; Computer Networks; Computer Systems Security; Operating Systems; Theory of Computation; Graph Theory. It is important to note that these disciplines were listed based solely on the computational field; thus, even though Computing originates from Mathematics, disciplines from this area were not considered, nor were those from Physics, Electronics, and Social and Professional Context (SBC, 2005).

Therefore, we highlight the areas of computing that have been used in the development of computational thinking (CT). In this research, out of the 15 studies that mentioned using concepts other than programming, the subfields with specific topics addressed were Artificial Intelligence and Theory of Computation, with only one mention each, and Software Engineering, appearing in two studies. Concepts of data structures and logic also appeared, but they can somewhat be related to programming. In other words, despite the various subfields of Computer Science and the possibility of using diverse concepts from the area, programming remains the primary content addressed. But what about the other subfields that are fundamental to Computer Science? Aren't they essential for the development of CT? How many other possibilities could be explored through them?

How have the experimental interventions been conducted?

To enhance the arguments addressing the research questions that guided this MSL: Q1 - "How has computational thinking (CT) been approached in the teaching of computer science concepts in experimental research?" and the second part of Q2 – "For which educational levels and stages?", specifically regarding the execution of educational experiments, we investigated the duration of the activities in each study (intervention duration), in relation to other aspects of interest, such as the educational level and stage of the target audience, the computational concept addressed, teaching and learning methodologies, data analysis methods, experimental control, data collection instruments, and statistical procedures. Table 2 illustrates the quantities of activities at each educational level and stage in relation to the duration of the interventions.

Tabela 2 - Duration of the Intervention, Educational Levels, and Stages

	Childhood Education	EE (early years)	EE (final years)	High School	Higher Education	Initial Training	Continuing Education
Less than 3 hours	1	7	4	2	3	1	2
Between 3 and 10 hours	1	6	9	8	2	2	1
Between 11 and 20 hours	2	9	10	3	1	1	0
Between 20 and 30 hours	1	1	2	2	0	1	0
More than 30 hours	0	3	5	3	4	2	4
Not specified	0	7	9	0	7	3	6
Total	5	33	39	18	17	10	13

Source: Prepared by the authors.

It is worth noting that, within a single study, we found activities with different durations conducted with individuals at various educational levels and stages. We thus had a total of 135 records, considering educational levels and stages in relation to the activity durations. Based on guidelines for each educational level present in documents such as the BNCC (BRAZIL, 2017), Base Nacional Comum para a Formação Inicial de Professores da Educação Básica (BNC-Formação) (BRAZIL, 2019), and SBC guidelines (RAABE et al., 2017), we analyzed the characteristics of these activities.

For Early Childhood Education, we expected the activities to be more playful, supported by games and play. However, when investigating the durations of the activities and the actions taken, we found that three of the five studies had durations between 11 and 30 hours, indicating more continuous activities, which gave a more formal character to the activities. Additionally, we found that all activities involved programming, with four using robotics and one using another tangible artifact.

Some studies even had experimental and control groups for comparing the strategies used and possible indications of the development of Computational Thinking (CT). Tests to assess learning indicators appeared in three studies, and four of them employed some form of statistical tool.

Three actions deserve attention in the interventions conducted at this educational level: 1) manipulation of tangible objects with young children, which is essential for this age group; 2) use of methodologies such as project-based, game-based, and design-based approaches, present in three of the five studies; and 3) systematic execution of activities, with qualitative and quantitative analyses, as well as the division into control and experimental groups, which some of these studies took care to outline.

Unplugged computing activities, which allow greater interaction among participants, could be appealing to children at this age, as well as early exposure to technological devices (RAABE et al., 2017). In other words, at this age, children may not necessarily develop CT, but it would be an appropriate time to provide play and manipulation of objects aimed at interaction between children, offering new experiences and knowledge related to computational concepts.

In Elementary Education, early years, the activities presented in the selected studies were well-distributed in terms of duration, particularly for activities of up to 20 hours, with fewer interventions of longer duration. Activities involving programming and robotics were the most common, followed by unplugged computing and games. The teaching and learning methodologies were project-based and collaborative, with some qualitative and mixed analyses, and the composition of experimental and control groups in a few studies. Despite this, pre- and post-tests were present in various studies, accompanied by some statistical analyses (T-test and ANOVA). Studies that did not specify the duration of the activities generally maintained these characteristics.

We consider that the early years of Elementary Education is the appropriate time to understand the importance of computing and technology in daily life, and consequently, to introduce some computing concepts, beginning the development of Computational Thinking (CT) with activities aimed at problem analysis and resolution through playful visual languages, already incorporating basic concepts about information manipulation and storage and basic architecture of technological devices, as recommended by the SBC (RAABE et al., 2017). Robotics and programming can be introduced towards the end of the first stage of Elementary Education, but still in a playful manner, with project or problem-oriented proposals, such as the need to build a robot to help solve a particular problem. In the investigated studies, the majority used programming, with some incorporating robotics and games, but without a systematic approach to content and concepts related to the development of CT. The absence of a curricular subject on computing at this educational stage is noteworthy (SBC, 2018b).

In the final years of Elementary Education, nearly half of the activities had a duration between 3 and 20 hours, approaching 50% of the total. Programming with the Scratch tool and unplugged computing were present in several studies, with robotics coming next and some experiments involving simulation. Teaching and learning methodologies were scarcely utilized, with mentions of problem-based, project-based, game-based, and collaborative approaches. Mixed and qualitative data analysis methodologies were employed in nearly half of the studies, indicating a greater focus on evaluative metrics, as recorded by the data collection instruments that appeared considerably, such as pre- and post-tests, observation records, questionnaires, interviews, and diaries. The T-test and Cronbach’s Alpha coefficient were also present in almost one-third of the analyzed studies.

Studies that did not specify the duration, which accounted for about 23%, also followed these characteristics but did not use unplugged computing, with very few mentions of data analysis methodologies, and did not use interviews in data collection. According to the SBC guidelines (RAABE et al., 2017), in the final years of Elementary Education, students could further develop CT by understanding more specific computing concepts and their applications in daily life, such as the conception, use, and implications of internet access and available information, the relationship between hardware and software in a broader sense, as well as applying computational concepts in problem analysis and resolution, including concepts from other areas of knowledge. This educational perspective was generally not mentioned in the investigated studies.

For High School, the final stage of Basic Education, it is crucial that young people develop a more theoretical understanding of digital technologies and computing. The investigated studies targeting this audience conducted activities over various time periods. The interval between 3 and 10 hours was the most common, appearing in nearly 45% of the studies that reported the duration. Almost all of these studies addressed programming using tools and platforms aimed at developing Computational Thinking (CT). Few mentioned the teaching and learning methodologies employed (project-based, design, games, and problem-based), with some using analysis instruments (pre- and post-tests, questionnaires, and interviews) and statistical tools (Wilcoxon Test and Cronbach’s Alpha). Both qualitative and quantitative analyses were present in almost all studies. Other time intervals, with less frequency among the studies, generally maintained these same characteristics, which in our analysis was a critical point, as much more could be explored at this educational stage.

With CT developing in students, this would be the time to propose broad projects that encompass social aspects and curricular knowledge from other areas, using CT skills, computational tools, and techniques to implement solutions for the projects through longer-duration activities. The limits of computing should also be understood at this stage, as well as issues of security, intellectual property of data, and the impacts of technology use on society (RAABE et al., 2017). Another relevant aspect is that during this period of education, students may develop an interest in further studying computational concepts and consequently pursuing this field in higher education. Again, the lack of computing as a curricular component in Basic Education is considered a disadvantage for the development of CT in students.

In Higher Education, the time intervals for intervention activities were well distributed. Studies conducted in undergraduate courses that described the duration of their activities generally involved the use of programming for developing CT. Only one study included experimental and control groups; nearly half resorted to qualitative or quantitative analyses; few mentioned the teaching and learning methodologies (game-based, collaborative), use of pre- and post-tests, questionnaires, and observation for data collection, and very few statistical tools. Studies that did not specify the duration of activities, which accounted for nearly 44% of the total, were primarily different due to not reporting the data analysis methodologies used. Other characteristics remained similar to interventions in other educational stages.

Studies targeting undergraduate students in initial teacher education also showed a well-dispersed intervention duration. Overall, programming was again the predominant concept addressed, with one study specifically describing concepts related to Data Structures and Turing Machines (Theory of Computation). Teaching and learning methodologies were rarely referenced (problem-based learning in one study, peer-based and design-based in another).

Pre- and post-tests appeared in almost all studies, as well as various statistical analyses (T-test, ANOVA, MANOVA, Cronbach's Alpha). The data analysis in nearly all studies employed qualitative, quantitative, or mixed approaches. These characteristics were also maintained in studies that did not specify the duration of their activities. In undergraduate courses focused on teacher training, we expected activities with longer durations to facilitate the comprehensive development of

Computational Thinking (CT), as it is a useful component of pedagogical training in any area. However, this was only observed in two of the seven studies, as shown in Table 1.

Studies that offered activities aimed at developing CT for in-service teachers (continuing education) presented activities with a duration of more than 30 hours in four of the seven studies that reported duration. This demonstrates a concern, albeit late, for the broad development of CT. The overall scenario of these studies was similar to that of studies targeting initial training, concerning the computational concept used (programming), teaching and learning methodologies, data collection methods, and statistical tests. The studies identified here did not emphasize the use of data analysis methodologies, with only one study mentioning qualitative analysis. Additionally, there was a lack of experimental and control groups, which is somewhat curious given that the formation of these groups would be more suitable for experimental studies with longer activity durations. Studies that did not specify the duration of their activities had similar characteristics to the others.

Conclussions

Through the investigated documentary corpus, we extracted and analyzed information to contribute to answering the research questions guiding this study. After organizing the data into analytical categories, we sought to synthesize it through groups that directed the discussion. In general, programming emerged as one of the primary areas of computing explored in the development of Computational Thinking (CT). It appeared as a computational concept, whether as a means (tool) or as an end (concept), in almost 86% of the investigated studies. We consider that this factor may be related to the fact that programming has been present in the educational environment longer than other subareas of computing and, consequently, has various possibilities for application in teaching, such as in game creation, robotics, or even unplugged activities. One element that needs further clarification is the simplification of its learning process, possibly due to the numerous visual block-based programming platforms that have been significantly presented in the literature as facilitating tools. We are not claiming that these platforms do not contribute to learning; rather, we want to highlight the level of programming learning achievable through them, i.e., the depth of programming concepts they allow students to grasp. There is a significant difference between programming an object with three or four instructions on a platform and programming a complete system.

Next, we investigated the levels and stages of education where computing education was being promoted. The levels were distributed, with the highest presence in Higher Education and the final years of Elementary Education, followed by the initial years of Elementary Education, less in Secondary Education, and even less in Early Childhood Education. It is worth noting that the studies categorized under Higher Education were divided into initiatives in undergraduate programs, initial teacher training, and in-service teacher training. This division occurred precisely to check if pre-service teachers were receiving information about the development of Computational Thinking (CT), as they would be responsible for leading this learning in Basic Education.

Another point that deserves attention is how computing concepts have been addressed in undergraduate courses. Generally, one or two courses are offered, not integrated with others, and related to programming or general technology topics. We consider that, as Wing (2006) argued, CT should be a form of thinking developed by everyone, not just computing professionals.

Regarding the format of educational interventions used in the investigated studies, we did not identify any pattern concerning duration, experimental control methods, data analysis methodologies, data collection instruments, and statistical procedures, nor any relationship with the levels and stages of education. Additionally, we identified little support from pedagogical theories in the propositions and analyses of teaching and learning processes.

We hope that the results of this literature review (MSL) will provide new possibilities for research and investigations on the development of CT, beyond the extensive and diverse material available in the literature. We emphasize that programming is certainly a concept of fundamental importance in Computer Science, but we believe that other subareas can significantly contribute to the development of this form of thinking. As future work, we propose investigating how other fields of knowledge and ways of thinking could contribute to studies in the area of CT. One such area is theoretical thinking, widely researched by scholars and researchers from various fields (DAVYDOV, 1988; LIBÂNEO, 2004; VYGOTSKY, 2008; [AUTHOR REMOVED FOR REVIEW], 2019). Furthermore, in line with recent observations (OLIVEIRA; CAMBRAIA; HINTERHOLZ, 2021), we agree that it is imperative to conduct experimental research with longer durations on the development of CT, which, despite being widely investigated, still needs to strengthen ties with pedagogical theories in pursuit of teaching that promotes student learning and development.

References

AURELIANO, V. C. O.; TEDESCO, P. C. de A. R.; GIRAFFA, L. M. M. Desafios e oportunidades aos processos de ensino e de aprendizagem de programação para iniciantes. *In*: **Anais do XXIV Workshop sobre Educação em Computação**. SBC, 2016. p. 71-80.

BARR, V.; STEPHENSON, C. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? **Acm Inroads**, v. 2, n. 1, p. 48-54, 2011.

BEBRAS - International Challenge on Informatics and Computational Thinking [On-line], n. d. Disponível em: www.bebras.org. Acesso em: 2 fev. 2021.

BELL, T., WITTEN, I. H.; FELLOWS, M. Ensinando ciência da computação sem o uso do computador. 2011. Disponível em: https://classic.csunplugged.org/wp-content/uploads/2014/12/CSUnpluggedTeachers-portuguese-brazil-feb-2011.pdf. Acesso em: 10 jun. 2021.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular** (Terceira Versão). Ministério da Educação, Brasília, DF: MEC, 2017. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 2 fev. 2021.

BRASIL. MINISTÉRIO DA EDUCAÇÃO**. Resolução CNE nº 2, de 20 de dezembro de 2019**. Diretrizes Curriculares Nacionais para a Formação Inicial de Professores para a Educação Básica e Base Nacional Comum para a Formação Inicial de Professores da Educação Básica (BNC-Formação). Diário Oficial da União, 2019. Disponível em: http://portal.mec.gov.br/docman/dezembro-2019-pdf/135951-rcp002-19/file. Acesso em: 17 jul. 2021.

BRASIL. MINISTÉRIO DA EDUCAÇÃO. **Resolução CNE/CES nº 5, de 16 de novembro de 2016**. Diretrizes Curriculares Nacionais para os Cursos de Graduação em Computação (DCN16). Brasília, DF: MEC, 2016. Disponível em:

http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=52101-rces005-16-pdf&category_slug= novembro-2016-pdf&Itemid=30192. Acesso em: 30 maio 2021.

BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. *In*: **Proceedings of the 2012 annual meeting of the American educational research association**. Vancouver, Canada. 2012.

CC2020 Task Force. Computing Curricula 2020: Paradigms for Global Computing Education. Association for Computing Machinery, New York, NY, USA. 2020.

CSTA – Computer Science Teacher Association; ISTE – International Society for Technology in Education. Operational definition of computational thinking, 2011. Disponível em: https://www.iste.org/explore/computational-thinking/computational-thinking-all. Acesso em: 6 out. 2021.

CSTA. CSTA K-12 Computer Science Standards 2012, Revised 2017. **Computer Science Teachers Association**. Disponível em: http://www.csteachers.org/standards. Acesso em: 7 jul. 2021.

DAVYDOV, V. V. Problems of Developmental Teaching. The Experience of Theoretical and Experimental Psychological Research. Excerpts (Part III). **Soviet education**, v. 30, n. 10, 1988.

DÍAZ-LAUZURICA, B; MORENO-SALINAS D. Computational Thinking and Robotics: A Teaching Experience in Compulsory Secondary Education with Students with High Degree of Apathy and Demotivation. **Sustainability**. v. 11, n. 18, 2019. Disponível em: https://doi.org/10.3390/su11185109. Acesso em: 9 mar. 2023.

GROVER, S.; PEA, R. Computational thinking in K–12: A review of the state of the field. **Educational researcher**, v. 42, n. 1, p. 38-43, 2013.

HEINTZ, F.; MANNILA, L.; FÄRNQVIST, T. A review of models for introducing computational thinking, computer science and computing in K-12 education. **2016 IEEE Frontiers in Education Conference (FIE)**. p. 1-9, 2016. Disponível em: 10.1109/FIE.2016.7757410. Acesso em: 9 mar. 2023.

HSU, T.-C.; CHANG, S.-C.; HUNG, Y.-T. How to learn and how to teach computational thinking: Suggestions based on a review of the literature. **Computers & Education**, v. 126, p. 296-310, 2018. Disponível em: https://doi.org/10.1016/j.compedu.2018.07.004. Acesso em: 9 mar. 2023.

HUBWIESER, P., GIANNAKOS, M., BERGES, M., BRINDA, T., DIETHELM, I., MAGENHEIM, J., JASUTE, E. A global snapshot of computer science education in k-12 schools, 2015. ITICSE-WGR '15 **Proceedings of the 2015 ITiCSE on Working Group Reports**, p. 65-83, jul. 2015. Disponível em: https://doi.org/10.1145/2858796.2858799. Acesso em: 9 mar. 2023.

KALELIOGLU, F.; GULBAHAR, Y.; KUKUL, V. A framework for computational thinking based on a systematic research review. **Baltic J. Modern Computing**, v. 4, n. 3, p. 583-596, 2016.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in *software* engineering. **Technical Report EBSE** 2007-001, jul. 2007. Keele University and Durham University Joint Report.

KORKMAZ, Ö.; ÇAKIR, R.; ÖZDEN, M. Y. A validity and reliability study of the computational thinking scales (CTS), **Computers in Human Behavior**, v. 72, p. 558-569, 2017. Disponível em: https://doi.org/10.1016/j.chb.2017.01.005. Acesso em: 9 mar. 2023.

LIBÂNEO, J. C. A didática e a aprendizagem do pensar e do aprender: a Teoria Histórico-cultural da Atividade e a contribuição de Vasili Davydov. **Revista Brasileira de Educação**, v. 27, p. 5-24, 2004. Disponível em: https://doi.org/10.1590/S1413-24782004000300002. Acesso em: 9 mar. 2023.

LYON, J. A., MAGANA, A. J. Computational thinking in higher education: A review of the literature. **Comput Appl Eng Educ**. v. 28, p. 1174-1189, 2020. Disponível em: https://doi.org/10.1002/cae.22295. Acesso em: 9 mar. 2023.

NOURI, J.; ZHANG, L.; MANNILA, L.; NORÉN, E. Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. **Education Inquiry**, v. 11, n. 1, p. 1-17, jun. 2019. Disponível em: 10.1080/20004508.2019.1627844. Acesso em: 9 mar. 2023.

OCDE. **PISA 2021 Mathematics Framework (Draft)**. Paris: OECD, 2019. Disponível em: http://www.oecd.org/pisa/pisaproducts/pisa-2021-mathematics-framework-draft.pdf. Acesso em: 30 maio 2021.

OLIVEIRA, W.; CAMBRAIA, A. C.; HINTERHOLZ, L. T. Pensamento Computacional por meio da Computação Desplugada: Desafios e Possibilidades. *In*: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 29., 2021, Porto Alegre. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 468-477. Disponível em: https://doi.org/10.5753/wei.2021.15938. Acesso em: 12 mar. 2023.

PALTS, T.; PEDASTE, M. A model for developing computational thinking skills. **Informatics in Education**, v. 19, n. 1, p. 113-128, 2020.

PAPERT, S. **Mindstorms**: Children, computers, and powerful ideas. New York: Basic Books, Inc, 1980.

RAABE, A. L. A.; FRANGO, I.; RIBEIRO, L.; GRANVILLE, L. Z.; SALGADO, L.; CRUZ, M. K.; FORTES, S. Referenciais de formação em computação: Educação básica. **Sociedade Brasileira de Computação**, 2017. Disponível em: http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf. Acesso em: 7 jul. 2021.

ROMÁN-GONZÁLEZ, M.; PÉREZ-GONZÁLEZ, J.-C.; JIMÉNEZ-FERNÁNDEZ, C. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test, **Computers in Human Behavior**, v. 72, p. 678-691, 2017. Disponível em: https://doi.org/10.1016/j.chb.2016.08.047. Acesso em: 9 mar. 2023.

ROYAL SOCIETY (GREAT BRITAIN). Shut down or restart?: The way forward for computing in UK schools. **Royal Society**, 2012. Disponível em: https://royalsociety.org//media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf. Acesso em: 16 jul. 2021.

SBC. Currículo de Referência da SBC para cursos de Graduação em Bacharelado em Ciência da Computação e Engenharia da Computação (CR05). 2005. Disponível em: https://www.sbc.org.br/documentos-da-sbc/send/131-curriculos-de-referencia/760-curriculo-de-referencia-cc-ec-versao2005. Acesso em: 11 jun. 2021.

SBC. Diretrizes para ensino de Computação na Educação Básica, 2018a. **Sociedade Brasileira de Computação.** Disponível em: https://www.sbc.org.br/documentos-da-sbc/send/131-curriculos-de-referencia/1177-diretrizes-para-ensino-de-computacao-na-educacao-basica. Acesso em: 7 jul. 2021.

SBC. Itinerário Formativo de Computação. Porto Alegre, 2019. **Sociedade Brasileira de Computação**. Disponível em: https://www.sbc.org.br/documentos-da-sbc/send/203-educacao-basica/1216-itinerario-formativo-da-computacao. Acesso em: 7 jul. 2021.

SBC. Nota Técnica da Sociedade Brasileira de Computação sobre a BNCC-EF e a BNCC-EM, 2018b. **Sociedade Brasileira de Computação**. Disponível em: https://www.sbc.org.br/institucional-3/cartas-abertas/summary/93-cartas-abertas/1197-nota-tecnica-sobre-a-bncc-ensino-medio-e-fundamental. Acesso em: 19 jul. 2021.

SEITER, L.; FOREMAN, B. Modeling the learning progressions of computational thinking of primary grade students. *In*: **Proceedings of the ninth annual international ACM conference on International computing education research**. 2013. p. 59-66.

SELBY, C.; WOOLLARD, J. **Computational thinking**: the developing definition. 2013.

SHUTE, V. J.; SUN, C.; ASBELL-CLARKE, J. Demystifying computational thinking. **Educational Research Review**, v. 22, p. 142-158, 2017.

SIMON; LUXTON-REILLY, A.; AJANOVSKI, V. V.; FOUH, E.; GONSALVEZ, C; LEINONEN, J.; PARKINSON, J.; POOLE, M.; THOTA, N. Pass Rates in Introductory Programming and in other STEM Disciplines. *In*: **Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '19)**, 2019. Association for Computing Machinery, New York, USA, p. 53-71. Disponível em: https://doi.org/10.1145/3344429.3372502. Acesso em: 9 mar. 2023.

TASLIBEYAZ, E.; KURSUN, E.; KARAMAN, S. How to Develop Computational Thinking: A Systematic Review of Empirical Studies. **Informatics in Education**, v. 19, n. 4, p. 701-719, 2020.

TIKVA, C.; TAMBOURIS, E. Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review, **Computers & Education**,

v. 162, 2021. Disponível em: https://doi.org/10.1016/j.compedu.2020.104083. Acesso em: 9 mar. 2023.

VYGOTSKY, L. S. O desenvolvimento dos conceitos científicos na infância. **Pensamento e linguagem**, p. 93-95, 2008.

WING, J. M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006.

WU, B.; HU, Y.; RUIS, A. R.; WANG, M. Analysing computational thinking in collaborative programming: A quantitative ethnography approach. **J Comput Assist Learn**. v. 35, p. 421-434, 2019. Disponível em: https://doi.org/10.1111/jcal.12348. Acesso em: 9 mar. 2023.

ZHANG, L.; NOURI, J. A systematic review of learning computational thinking through Scratch in K-9, **Computers & Education**, v. 141, 2019. Disponível em: https://doi.org/10.1016/j.compedu.2019.103607. Acesso em: 9 mar. 2023.

ZORZO, A. F.; NUNES, D.; MATOS, E.; STEINMACHER, I.; LEITE, J.; ARAUJO, R. M.; CORREIA, R.; MARTINS, S. Referenciais de Formação para os Cursos de Graduação em Computação. **Sociedade Brasileira de Computação (SBC)**. 2017.

Recebido em: 18/05/2022

Aprovado em: 20/03/2023