A literature review of computational thinking in early ages

Edelberto Franco Silva, Bruno Jose Dambogurski and Gustavo Silva Semaan

Computer Science Department, Universidade Federal de Juiz de Fora, Juiz de Fora, Brazil; Universidade Rural do Rio de Janeiro, Seropedica, Brazil; Universidade Federal Fluminense, Niteroi, Brazil

ABSTRACT

Nowadays, technology has become dominant in the daily lives of most people around the world. Technology is present from children to older people, helping in the most diverse daily tasks and allowing accessibility. However, many times these people are just end-users, without any incentive to develop computational thinking (CT). With advances in technologies, the abstraction of coding, programming languages, and the hardware resources involved will become a reality. However, while we have not progressed to this stage, it is necessary to encourage the development of CT teaching from an early age. This work will present the state of the art concerning teaching initiatives and tools on programming, robotics, and other playful tools for the development of CT in the early ages, explicitly filling the gap of CT at the kindergarten level. We present a systematic literature review evaluating more than 60 papers from 2010 to December 2020. The paper’s amount was classified in taxonomy to show CT’s principal tools and initiatives applied to children early. To conclude this paper, an extensive discussion about the future trends in this field is present.

1. Introduction

In this century, people are interested in being consumers of technology and producing them. Different from the past decades, today, the children grow up as digital natives. However, to motivate people to not grow only as end-user, it is necessary to drive them and offer tools to develop computational thinking (CT) as soon as possible. The development of CT is essential to create future scientists or engineers, but it can also enhance many cognitive and intellectual skills, allowing people to solve real problems. For example, to ‘find the best path from his/her house to the market’, or to ‘calculate the trajectory of an object’.

The CT was credited to Papert (1980), but only in Wing (2006) was popularised the term and sparked the international community’s interest. In the last decade, many types of research were conducted to understand and propose new strategies to develop CT (Bers 2017; Bers 2019; Palmer 2017; Ehsan et al. 2018). However, a tiny part of these initiatives focuses on early ages, creating a gap to apply STEM education.
(science, technology, engineering, and math) and develop the CT for the next generations. An overview and update about CT, in general, can be found in Yasar (2017).

The importance of developing the CT at an early age is discussed by actual papers and books Bers (2020); Sanford and Naidu (2016). Its primary motivation is to provide a familiarity with CT from an early age, incorporating the benefits of thinking logically in real-life challenges. However, to exist an incentive to propose and evaluate new curricula and tools from children at an early age, the papers published about this topic are needy. More than this, it is necessary to investigate the state-of-the-art in this field and discuss its characteristics and benefits.

To fill this gap, in this paper, we introduce a systematic review of curriculum initiatives and tools applied to CT from an early age. This work is the first to present the state-of-the-art in this field of knowledge, focusing only on kindergarten and children with 2–5 years old. To conduct our research was considered the decade from 2010 to 2020. Besides presenting the most relevant papers published in the last decade, a growth trend analysis of this research field is introduced as a taxonomy of the main tools used to develop CT at an early age.

A systematic review has cleared the emergence of this research topic, showing an increasing number of published papers, a growth of more than three times in the last five years. The proposal of a new curriculum and the development of new tools to support their topics need to be discussed. Many of these are playable using robotic and tangible parts. Another relevant topic is the prevalence of papers in journals concerning conferences, creating the necessity to investigate this topic at symposiums and scientific events. Lastly, the number of documents with case studies and surveys is twice compared to paper without them, reinforcing the need for validation and feedback by education researchers.

Figure 1 is shown the most common words across all papers: computational, thinking, programming, and robotics. As aforementioned, we will see in this
article that the main initiatives are related to robotics and programming to create playful tools and incentivize the CT at an early age. The word cloud is vital to show the main words and their relevance – where more significant words represent more occurrences on the selected papers than small ones. It is relevant to comment on the limitation of this work. In this paper, we only consider the gap of CT for the kindergarten level concerning teaching initiatives and tools on programming, robotics, and other playful tools. We do not consider the impact of each initiative applied from an early age to the rest of life (i.e. young and adult life) Tables 1 and 2.

Table 1. Frequency of words most used.

| Id | Word         | Count | (%) |
|----|--------------|-------|-----|
| 1  | Computational| 15    | 44.1|
| 2  | Thinking     | 15    | 44.1|
| 3  | Programming  | 12    | 35.2|
| 4  | Robotics     | 10    | 29.4|
| 5  | Teaching     | 8     | 23.5|
| 6  | Coding       | 7     | 20.5|
| 7  | Development  | 7     | 20.5|
| 8  | Preschool    | 7     | 20.5|
| 9  | Learning     | 6     | 17.6|
| 10 | Childhood    | 6     | 17.6|
| 11 | Teacher      | 5     | 14.7|
| 12 | Kindergarten | 5     | 14.7|
| 13 | Early        | 5     | 14.7|
| 14 | Children     | 5     | 14.7|
| 15 | Educational  | 4     | 11.7|
| 16 | Computer     | 4     | 11.7|
| 17 | Activities   | 4     | 11.7|
| 18 | Education    | 4     | 11.7|
| 19 | Skills       | 3     | 8.8 |

Table 2. Ranking of most relevant words by year.

| Word (%) | 2017 | 2018 | 2019 | 2020 |
|----------|------|------|------|------|
| Computational | 50   | 33   | 35   | 25   |
| Thinking   | 50   | 33   | 35   | 25   |
| Programming| 50   | 67   | 24   | 0    |
| Robotics   | 17   | 0    | 29   | 50   |
| Teaching   | 0    | 67   | 29   | 25   |
| Coding     | 17   | 0    | 29   | 25   |
| Development| 17   | 0    | 24   | 25   |
| Preschool  | 33   | 33   | 24   | 100  |
| Learning   | 33   | 33   | 18   | 0    |
| Childhood  | 0    | 0    | 29   | 0    |
| Teacher    | 17   | 33   | 18   | 0    |
| Kindergarten| 0   | 0    | 0    | 0    |
| Early      | 0    | 0    | 24   | 0    |
| Children   | 17   | 33   | 0    | 25   |
| Educational| 17   | 0    | 6    | 50   |
| Computer   | 0    | 0    | 18   | 25   |
| Activities | 0    | 0    | 24   | 0    |
| Education  | 17   | 33   | 6    | 0    |
| Skills     | 17   | 0    | 6    | 25   |
Thus, the main question to be answered by this article is: ‘Which tools and strategies were adopted to encourage the CT development at early ages?’

The remainder of this article is structured as follows: Section 2 presents the methodology approach used on this paper and how the documents were selected; Section 3 describes the leading tools and classify than following a taxonomy proposed by the authors; Section 4 conclude this article by suggesting a debate and possible future works.

2. Methods – systematic review

In this work, a mapping of state of the art was conducted based on a systematic review. This method is responsible for collecting the most relevant papers in the last decade, respecting the keywords chosen to answer the question created on the protocol step. The following paragraph explains step by step how we constructed the base of this article. Kitchenham and Charters (2007) outline three phases to conduct a systematic review: (i) planning, (ii) execution, and (iii) summarization/reporting. In the first phase, it is necessary to identify the need for a review and create a review protocol containing the systematic review’s important information. The second phase identifies and selects relevant primary studies, performs the data extraction, and synthesises the extracted data. Finally, in the third phase, the systematic review results are summarised and published to the community. Similar to Kitchenham and Charters (2007), the PRISMA 2020.1 All steps were conducted using the software StArt (Fabbri et al. 2016) that implements all requirements described.

Four indexed libraries were selected, namely:

- Scopus – Elsevier
- ACM
- IEEE
- Web of Science

Using the logical keywords combination and restricting the search only to articles and proceedings, written in English and the range of from January 2010 to December 2020, the selection was started. The search string used on all libraries follows the logic:

```
("computational think*") AND ("early age") OR ("kindergarten") OR ("pre-school") AND ("playful tool") OR ("programming") OR ("robotic"), beyond the restrictions already mentioned.
```

Thus, 72 papers were recovered in the selection step, where 13 articles were excluded as duplicate entries, and 20 rejected them not to be in the scope of ‘early ages’ (dealing with children older than five years). With the 39 selected papers, the extraction stage was initiated. In this step, 31 papers were chosen based on the exclusion of 8 articles because they are outside the standards of full articles in English. Finally, three relevant papers were included in the qualitative inclusion stage, totalling 34 papers. Considering the quantitative selection was considered only papers which fulfil the language and the scope of our research. For qualitative selection, we consider papers publish in relevant vehicles (e.g. Scopus Q1 and Q2) with more than 10 citations.

With this final selection, 7 were identified as having a general review approach to the concept of CT at an early age, 8 deal with the proposal for the new curriculum, and 19
modern tools, with or without case studies. It is interesting to note that the case study proves to be of great value in this context and all works, where 22 articles of the total amount present validation per case study, representing approximately 64% of the papers. Figure 2 shows the steps described in this section and the proportion of paper with (Yes) or without (No) use cases in Figure 4. The ‘use cases’ term are considered when research with people was conducted to validate the proposal/idea.

It is noteworthy that most papers on journals than conferences are essential, as shown in Figure 3. This kind of behaviour is not observed in many other research fields. It creates a feeling about the necessity to encourage all people involved in CT to create more forums to discuss the topic, e.g. at symposiums or conferences.

Another relevant preliminary analysis is the relevance of use cases. The majority of the papers present a kind of validation, as through surveys or observations. The distribution of documents with or without use cases can be seen in Figure 4.

Figure 2. The phases of systematic review following the Kitchenham and Charters (2007) and PRISMA (Moher et al. 2011) guidelines.
As previously mentioned, we will present the main initiatives for the development of CT at an early age. This article is the first research that we know that addresses the topic precisely at an early age. However, we can see that the number of studies and publications has grown in recent years, according to Figure 5. The last five years are more than three times bigger than the other years. It is possible to see the growth trend of this research field.

**Figure 3.** Number of papers published on conference versus journals.

**Figure 4.** Distribution of paper with (Yes) or without (No) use cases.

As previously mentioned, we will present the main initiatives for the development of CT at an early age. This article is the first research that we know that addresses the topic precisely at an early age. However, we can see that the number of studies and publications has grown in recent years, according to Figure 5. The last five years are more than three times bigger than the other years. It is possible to see the growth trend of this research field.
3. Results

The following sections will present the main existing tools, guiding the reader on the selected papers, the articles with the proposals and evaluations of the school curriculum, and describing a taxonomy for all of the other documents.

3.1. Tools

As part of the initiatives and efforts to develop computational thinking in early ages, tools such as robotics and computer programming initiatives are increasingly being encouraged among researchers and early childhood educators. Robotics and computer programming in this context can support a range of cognitive and social aspects. We will introduce the leading technology solutions that were cited in the selected researches. There is a great incentive to use block-based programming in the early ages, a playful way to encourage the child. There is also an incentive to use tiny robots that memorise simple commands like moving forward, side or back, to create a spatial vision and cognition capacity. There are web-based tools, as we will see, but these are more complex for children at an early age, and there are block solutions that use both the tangible form and the aid of a device such as a tablet or iPad.

As a way of organising the presentation and tools’ classification, Figure 6 is presented where it is possible to see BeeBot, Kibo, ScratchJr, and a group of several other tools that appear less frequently in the verified studies. Thus, we will present each one of them below.
3.1.1. Bee-Bot

The Bee-Bot\(^2\) is a simple robot based on simple commands focusing on early ages children. This tool can improve the children’s skills and develop the CT at an early age, only using a couple of sequences. Their main idea is to enhance the cognition capacity and future view of steps through directional language as a programming sequence: forward, backward, left, and right 90 degree turns. In Figure 7 is possible to see the robot and their commands on top of them. Thus, the sequence is created by the child before clicking on the start. After it, the tool enables them to run all commands sequentially storage.

This robot works as a playful tool once it improves the CT skills through the necessity to memorise the steps and imagine how the robot will run the actions. It is possible to create a puzzle for children using square paper or a map with obstacles. An example of this kind of problem is shown in Figure 8.

3.1.2. KIBO

The KIBO robotics kit is a tool (Sullivan, Elkin, and Bers\(^2\)\(^\)\()\) that proposes to engage young children in both building and programming. The KIBO is a kit developed by the DevTech Research Group at Tufts University\(^3\) and commercialised by KinderLab Robotics,\(^4\) an enterprise created to enhance STEM. Although the KIBO is designed for young children ages 4–7 to learn foundational engineering and programming content, it can be applied to help develop CT at an early age. Figure 9 can see its main components, the robot, and its blocks with several actions. Once the blocks are read, the robot can run the steps sequentially. It is possible to create another logical programming with the blocks beyond to go forward or change direction; one of these examples is the loop block, mainly used to develop more complex procedures.
Elkin, Sullivan, and Bers. (2016) present a case use with children three years old, where the tool is used to validate a proposal to incorporate than on an urban public preschool in Rhode Island, US.

### 3.1.3. Scratchjr

The ScratchJr\(^5\) is a programmable web-based tool free of charge that can be used from an early age. But this solution is not so simple as KIBO or Bee-Bot needing more help from an adult to understand and manipulate the computer/tablet. Even focusing on children from 5 to 7 years old, it is common to find many use cases to adopt this tool.

Its motivation is to improve logic programming from an early age. Like its version for young people, this tool has the purpose of creating funny stories using a sequence of blocks. In Figure 10 is possible to see its web interface where a chain of blocks is put on the way to create the story. Although to be more complex, it is possible to create more simple stories with a limited number of blocks, making it possible to be applied at an early age.

### 3.1.4. Others

Other tools are cited by many papers, from prototypes (e.g. an incipient real version of the proposal) of robots to tangible and web-based block-based solutions. Examples of robotic tools are LEGO WeDo and EV3, both are solutions of LEGO company, and their objective is to evaluate the CT for STEM. Despite being in this work field, we consider both of them more indicated to be incorporated in the curriculum for children with more than eight years old because of the necessity to manipulate small parts and have the necessary knowledge to manage the computer. Other papers cited the mBot, but it is more like LEGO WeDo. Different from the tools mentioned, other work proposes a
prototype of a playful robotic tool called Robotito. This initiative follows the same KIBO concepts, but today it is only in an initial stage. Another new robot is BlueBot that does the same things as Bee-Bot, but with Bluetooth connection support. Other initiatives as LightBot follow ScratchJr, doing possible programming with blocks on a web-based application, and can be used for children with five years, for example. More interest to the early ages in this section is the Happy Maps, an unplugged activity. The children can create simple algorithms as a set(s) of instructions to move a character through a maze using a single command. Figure 11 shows an example of a maze that can be solved by arrows that indicate how the character finds the fruit.

There are many tools and prototypes in the literature, and it does not enhance the discussion at this moment. About it, the other tools will be suppressed at this moment. If a tool that was not introduced in this section and appeared ahead, it will be presented in its context.

After knowing about the leading tools in the literature, it is possible to introduce the papers where a curriculum strategy was proposed or validated.
Figure 9. KIBO robot tool with the main blocks reading the code bar by infrared (Elkin, Sullivan, and Bers 2016).

Figure 10. ScratchJr.
3.2. Taxonomy

In this section, we propose a taxonomy as one of the main contributions of this paper, and that can be seen in Figure 12. In tools, there are four possible types: block, robot, block and robot, and prototype. Whenever there is a tool based on blocks, it may be tangible or web. Despite having the taxonomy organised that way, it is possible to have papers only classified in one, another, or more than one kind of tool.

3.2.1. Block

3.2.1.1. Web. In Papadakis, Kalogiannakis, and Zaranis (2016), a case study carried out in 2016 on the ScratchJr tool is presented. The study sample consisted of 43 pre-school children (22 boys, 21 girls) who were attending classes in public and a private kindergarten in the region of Crete, Greece, during the school year 2014–2015. Findings reveal that ScratchJr enhances student interest by making the learning experience fun. Similarly, animated scenarios showed high levels of engagement among students. Specifically, ScratchJr allowed children to engage in deep reflection as they solved problems and collaborated with their peers, both of which activities enhanced their learning experience.
Part of the authors from Papadakis, Kalogiannakis, and Zaranis (2016) published another paper in the same way in Papadakis and Kalogiannakis (2019). This new paper focused on the future teacher of kindergarten, to create the CT on the people will learn in the future for the kids. The authors adopted Scratch as the introductory programming language for a semester in the Department of Preschool Education at the University of Crete. The aim of using Scratch was to excite students’ interest and familiarise them with the basics of programming. For 13 weeks, students were introduced to the main Scratch concepts and were asked to prepare their projects afterward. For the projects, they were required to develop a game to teach specific concepts about Mathematics or Physical Science or present an Aesop myth to pre-school-age students. The results we obtained were more satisfactory than expected and, in some regards, encouraging.

The authors of Lowe and Brophy (2019) look at the intersection of CT and computer science in first-grade learners who are developing computational solutions involving literacy tasks. Students retell a story by animating characters in ScratchJr by breaking down the story, creating an animation storyboard, and finally implementing the plan in ScratchJr. For most of the participants, this is their first time using ScratchJr or any programming language. Therefore, their early experience with technology means they are working on analyzing a story using literacy skills, considering a visual representation of the story, and learning how to realise its expression using a computer language. Despite not being directly applied to early-age children, it is an excellent motivation for kindergarten teachers.

In Ciftci and Bildiren (2020), the researchers evaluated an experimental study to put forth the impact of computer programming courses’ problem-solving and cognitive abilities on 4–5-year-old pre-school children. This study uses a pretest-posttest control group experiment model. According to their results, there is an increase in the non-verbal cognitive skills of children in the experiment group with no statistically critical problem-solving skills. The ‘course A’ from code.org was applied to the evaluation with the help of unplugged activities (e.g. happy maps) and programming block-based.

3.2.2.2. Tangible and web. In Clarke-Midura et al. (2019), the authors examine three block-based coding tools applying a framework developed based on Gibson’s theory (Adolph and Kretch 2015) of affordances and Palmer’s external representations (Palmer 1978). They intend to verify known tools for children at an early age, motivated by more work and solutions for the development of computational thinking only for children in K-12. It is considered an exciting concept that facilitates the child’s interaction with the TUI programme (Tangible User Interface). The tools compared are ScratchJr, Osmo Coding Awbie⁶, and KIBO.

3.2.2. Robot
In González and Munoz-Repiso (2017), the authors show three different robotic technology applied to early school age. There are six groups of students, with 131 students and eight teachers distributed in first, second, and third kindergarten. With the ease of access to groups of students and teachers, the total population will be used for the study. On the conclusions, the authors received positive results regarding the acceptance and motivation to use educational activities mediated by programmable robots in students.
Roussou and Rangoussi (2020) presents research based on a case study investigating the impact of robotics on the cultivation of CT skills in early childhood through an educational intervention implemented in a typical public kindergarten in Athens, Greece. On the material, the authors adopt the tool Code & Go Robot Mouse Activity Set from Learning Resources, similar to the Bee-Bot tool. The authors investigate using a pre-test, intervention, and post-test, proving the benefits of robotics in kindergarten.

3.2.3. Block and robot

3.2.3.1. Tangible. In Urlings, Coppens, and Borghans (2019), sixty-five kindergarteners received assignments to go through a maze with a programmable robot, the Bee-Bot. The authors conducted this study via observation, quantifying which time and errors occurred, measuring how the increase of comprehension and evolution from a child with a mean age equals 6. The results were satisfactory, with a low rate of errors and a significant increase in the child’s motivation.

Bers, González-González, and Armas-Torres (2019) evaluated a ‘coding as a playground’ experience in keeping with the Positive Technological Development (PTD) framework with the KIBO robotics kit, specially designed for young children. The research was conducted with pre-school children aged 3–5 years old (N = 172) from three Spanish early childhood centres with different socio-economic characteristics and teachers of 16 classes. Results confirm that it is possible to start teaching this new literacy very early (at three years old). The results show that the strategies used promoted communication, collaboration, and creativity in classroom settings. Teachers had excellent experience enhancing their motivation and confidence to motivate the CT on the students. The concept of ‘coding as a playground’ as a new literacy is a new language for children to learn to code at a young age through fun, play, and creativity Bers (2017).

3.2.3.2. Web. In Strawhacker, Lee, and Bers (2018), the authors investigated the little is known about the relationship between a teacher’s unique instructional style and their students’ ability to explore and retain programming content. The study focuses on children aged 5–8 years. In this mixed-methods study, quantitative and qualitative data were collected from 6 teachers and 222 kindergartens through second-grade students at six schools across the United States from 2 months in 2014. All participants engaged in a minimum of two lessons and a maximum of seven lessons using the ScratchJr programming environment to introduce coding. Teachers reported on their classroom structure, lesson plan, teaching style, and comfort with technology.

3.2.3.3. Tangible and web. In Rial-Fernández and Santacruz-Valencia (2019), the authors describe the case study carried out in a classroom of Early Childhood Education, with students of 5 years of age, with which basic programming concepts have been worked out. The results obtained indicate that they have been able to master the new vocabulary, assimilate the concepts, and work for themselves with the chosen tool to carry out the intervention. The tools evaluated were: Bee-Bot, Happy Maps, and another sheet with arrows that they have to cut out. All tools were applied to be playful tools using blocks to increase the CT in programming.

In Otterborn, Schonborn, and Hulten (2020), it was conducted systematically investigates how Swedish pre-school teachers implement programming activities in their
teaching practice. Data were collected through a national online survey with 199 participants. Findings revealed a range of apps and resources used in combination with tablets, where activity integration takes place as unplugged programming, digital programming, or a combination of the former. On the survey, the most tools cited by the teachers of kindergarten were: Bee-Bot, Blue-Bot, LightBotJr,8 and ScratchJr. Another relevant result is that the study showed that nine pre-school teachers use unplugged programming to introduce digital programming.

In Kanbul and Uzunboylu (2017), the aim is to reveal the importance of coding education and robotic applications for achieving twenty-first-century skills in North Cyprus.

The skills from pre-school level mapped by the authors to help in CT development are: (i) Skills: putting in order, separating into little pieces, giving order; (ii) Software: ScratchJr, Code.org, the first two periods of Kodable, The foos; (iii) Robotic tools: BeeBots and KIBO. The authors strongly recommend the use of tools from early ages to increase de CT.

Pugnali, Sullivan, and Bers (2017) investigated the impact of the interface to help and adopt technology tools in CT. Children from 4 to 7 years old participated in the survey, evaluating ScratchJr at iPad and KIBO as the tangible tool. Results suggest that type of user interface impacts children’s learning but is only one of many factors that affect positive academic and socio-emotional experiences. Tangible and graphical interfaces each have qualities that foster different types of education.

3.2.4. Prototype

3.2.4.1. Robot. In Tejera et al. (2019), the authors present a prototype solution called Robotito, which comprises programming and robotics skills. The proposal has a more technical bias and uses ROS (Robot Operating System) standard mechanisms. Despite the work elucidating computational thinking as a focus, its presentation is more technical than discussing the impact of learning at an early age.

The authors of Coiro et al. (2020) propose an open-source robot platform called ProRobot. They aim to foster CT abilities in pre-school children, including a first approach on the built platform, an embedded processing unit for not requiring any additional equipment, and development to minimise costs. The paper is more technical but very important to increase the options of robot solutions for CT.

3.2.4.2. Block / tangible. In Wang, Wang, and Liu (2014), the authors present an economic tangible programming tool called T-Maze for children aged 5–9 to build computer programmes in maze games by placing wooden blocks. Through computer vision technology, T-Maze provides a live programming interface with real-time graphical and voice feedback. However, as highlighted by the authors, the focus is on children aged 5–9 years, which imposes difficulties to be used by children at an early age.

3.2.4.3. Block / web. Koracharkornradt (2017) presents a programming game called Tuk Tuk that aims to assist in developing CT in children in kindergarten (a junior version of the game). It allows the child to organise the blocks to create the steps of a car until they complete a race, accumulate points, and complete a detailed task. The idea is to develop an understanding of algorithms. There is the version for older children, a rating similar to ScratchJr and Scratch.
Kanaki and Kalogiannakis (2018) present the computational environment PhysGramming, which was designed to be used by children of early childhood age, between 4 and 8 years old. PhysGramming deploys a hybrid schema of visual and text-based programming techniques, emphasising object-orientation, to introduce elementary programming concepts in early childhood education. The solution can provide three kinds of games: puzzles, matching games, and group games. In each match, instructions about its functionality are given through the use of an animated cat. An example of application is the teacher creating digital games to help the students learn the names of some animals of the jungle and their nutritional habits. The authors have the ambition to present basic concepts of object-oriented programming playfully. Despite being a recent prototype, tools in the same vein are excellent candidates for help in developing the CT.

Baratè, Ludovico, and Mauro (2019) show in this paper a recent evolution of a web prototype conceived initially to teach music and CT to pre-school and primary school learners through a gamification approach. The software tool, Legato, is based on the metaphor of building blocks whose characteristics (e.g. position in space, shape, and colour) can be associated with basic music parameters (e.g. pitch, rhythm, and timbre). Legato is a web app written using standard languages, such as HTML5, CSS, and JavaScript; besides, it adopts the Web MIDI API to produce sounds. The prototype is made publicly available for evaluation and uses in an educational context. The paper did not restrict the age of the child to use it. However, it is a technical specification. The main objectives induce to be used only for kids after 5-years old, once it is necessary to manipulate the blocks to construct the music on a web-based platform.

CodyColor (Klopfenstein et al. 2019) is a simplified coding game that uses basic programming instructions to represent movements (e.g. turn left and turn right) and describe them with colour blocks. In contrast to most other coding games, colour-coded programming relies on no symbolic interpretation on the player’s part to be approachable by very young players. This initiative is based on Hour of Code CodyRoby, and it is a new version of CodyColor with massive multiplayer support.

4. Discussion

This paper presented state of the art in CT at an early age based on a systematic review. We considered only children between 2 and 5 years old highlighted the main tools and curriculum proposals and evaluation. A taxonomy was introduced to show the most gaps and advances in the last decade (2010–2020).

It is clear that this field needs attention, and the last five years prove that a growing number of researches, papers, and new initiatives have flourished. The significant number of articles in journals led us to believe it is an excellent opportunity to motivate the participants to offer conferences and events to discuss these new finds more thoroughly. Another important observation was the effort conducted by many countries, such as the USA and Spain, to incorporate, in kindergarten’s curriculum, the CT as an improvement for children’s capabilities in a variety of fields in their life. Finally, the most common way to validate a proposal, tool, etc., is to conduct a use case following pre-test and post-test in different cities and countries.

This research’s main question (i.e. ‘Which tools and strategies were adopted to encourage the CT development at early ages?’) was answered. It is clear to see that there are
many initiatives. However, a considerable part of them is limited in terms of impact on people’s life. Considering it, the authors suggest conducting more experiments and evaluations with tools and strategies for CT development at early ages. These future works need to consider the actual and long-term impacts in the life of these students’.

Furthermore, there is an ongoing discussion about gender disparity in STEM fields, where men outnumber women (Sullivan and Bers 2019). Usually, interventions aiming to increase women’s interest in science-related areas are performed during high school and college with subpar results. New evidence shows that introducing an appropriate curriculum with CT can increase girls’ interest in engineering.

Another critical issue that must be discussed is how technology will affect the future of the next generation of kids. One of the primary drivers of change in the current world is Artificial Intelligence (AI) and automation. Job characteristics and required skills will change drastically because of it. These bring us a variety of opportunities related to social and economic mobility, which could lead us to a better society. But to reap all these possible benefits, we must sow our future generations with this new knowledge.

The World Economic Forum, in Perisic (2018), discussed how artificial intelligence is shaking up the job market and two trends were perceived: one showing the continuous need for tech jobs and skills; and one pointing to human-centric skills, which directly depends on human qualities. Also, the impact of AI is not just theoretical anymore, and its influence can be observed across industries and jobs worldwide. There is way more information about this matter in different media types, but this discussion is not the centre of this work.

Thus, the points mentioned above reinforce the need for an early introduction of CT to prepare this future generation for a new reality regarding the job market. To conclude, the authors believe that the first step to improving CT at an early age is to start using these resources as soon as possible.

Complementary, an analysis of the most influential words observed during this research is present. Based on the 34 selected works from the Literature Systematic Review, it was needed to perform a pre-processing step, where data organisation, enrichment, consolidation, and formatting are performed. Specifically, the titles of all works were fragmented in words, occurrence, and next, exclusion of stop-words (words without semantic relevance) such as conjunction and prepositions. The terms of interest were unified through a process of radical reduction or equivalence to the most frequent similar word. As a complementary way to this paper, the presentation of this ranking is only to try to help people that are beginning in this research field to find new words, and highlight the most relevant keywords.

Notes
1. http://www.prisma-statement.org/ Moher et al. (2011) statement and its checklist and flow diagram.
2. http://bee-bot.us.
3. https://sites.tufts.edu/devtech/research/kibo-robot/.
4. https://kinderlabrobotics.com/.
5. https://www.ScratchJr.org/.
6. https://www.playosmo.com/en/coding/.
7. https://www.learningresources.com/code-gor-robot-mouse-activity-set.
Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Edelberto Franco Silva http://orcid.org/0000-0002-0058-9260
Bruno Jose Dembogurski http://orcid.org/0000-0002-6719-0537
Gustavo Silva Semaan http://orcid.org/0000-0003-2873-2628

References

Adolph, K., and K. S. Kretch. 2015. “Gibson’s Theory of Perceptual Learning.” International Encyclopedia of the Social and Behavioral Sciences 10: 127–134.
Baratè, A., L. A. Ludovico, and D. A. Mauro. 2019, September 18-20. “A Web Prototype to Teach Music and Computational Thinking Through Building Blocks.” Proceedings of the 14th International Audio Mostly Conference: [A] Journey in Sound, 227–230. Nottingham, UK: ACM. doi:10.1145/3356590.3356625
Bers, M. U. 2017. Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom. New York: Routledge.
Bers, M. U. 2019. “Coding as Another Language: A Pedagogical Approach for Teaching Computer Science in Early Childhood.” Journal of Computers In Education 6 (4): 499–528.
Bers, M. U. 2020. Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom. New York: Routledge.
Bers, M. U., C. González-González, and M. B. Armas-Torres. 2019. “Coding as a Play-Ground: Promoting Positive Learning Experiences in Childhood Classrooms.” Computers and Education 138: 130–145.
Ciftci, S., and A. Bildiren. 2020. “The Effect of Coding Courses on the Cognitive Abilities and Problem-Solving Skills of pre-School Children.” Computer Science Education 30 (1): 3–21.
Clarke-Midura, J., V. R. Lee, J. F. Shumway, and M. M. Hamilton. 2019. “The Building Blocks of Coding: A Comparison of Early Childhood Coding Toys.” Information and Learning Sciences 120 (7-8): 505–518.
Coiro, F., M. A. Solis, C. J. Nettle, and A. Chila. 2020. “Pre-robot: An Open-Source Educational Robotics Platform for Preschoolers.” CEUR Workshop Proceedings 2564: 18–23.
Ehsan, H., T. M. Dandridge, I. H. Yeter, and M. E. Cardella. 2018. “K-2 students’ Computational Thinking Engagement in Formal and Informal Learning Settings: A Case Study (fundamental)”. ASEE Annual Conference and Exposition, Conference Proceedings 2018-June.
Elkin, M., A. Sullivan, and M. U. Bers. 2016. “Programming with the Kibo Robotics kit in pre-School Classrooms.” Computers in the Schools 33 (3): 169–186.
Fabbri, S., C. Silva, E. Hernandes, F. Octaviano, A. Di Thommazo, and A. Belgamo. 2016. “Improvements in the Start Tool to Better Support the Systematic Review Process”. In Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering, pp. 1–5.
González, Y. A. C., and A. G. V. Munoz-Repiso. 2017. “Development of Computational Thinking and Collaborative Learning in Kindergarten Using Programmable Educational Robots: A Teacher Training Experience”. ACM International Conference Proceeding Series Part F132203.
Kanaki, K., and M. Kalogiannakis. 2018. “Introducing Fundamental Object-Oriented Programming Concepts in pre-School Education Within the Context of Physical Science Courses.” *Education and Information Technologies* 23 (6): 2673–2698.

Kanbul, S., and H. Uzunboylu. 2017. “Importance of Coding Education and Robotic Applications for Achieving 21st-Century Skills in North Cyprus.” *International Journal of Emerging Technologies in Learning* 12 (1): 130–140.

Kitchenham, B., and S. Charters. 2007. “Guidelines for Performing Systematic Literature Reviews in Software Engineering”.

Klopfenstein, L. C., S. Delpriori, R. Maldini, and A. Bogliolo. 2019. “Codycolor: Design of a Massively Multiplayer Online Game to Develop Computational Thinking Skills”. *CHI PLAY 2019 - Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play*, 453–458.

Koracharkornradt, C. 2017. “Tuk tuk: A Block-Based Programming Game”. *IDC 2017 - Proceedings of the 2017 ACM Conference on Interaction Design and Children*, 725–728.

Lowe, T. A., and S. P. Brophy. 2019. “Identifying Computational Thinking in Storytelling Literacy Activities With Scratch jr.”. *ASEE Annual Conference and Exposition, Conference Proceedings*.

Moher, D., D. G. Altman, A. Liberati, and J. Tetzlaff. 2011. “Prisma Statement.” *Epidemiology* 22 (1): 128.

Otterborn, A., K. J. Schonborn, and M. Hulten. 2020. “Investigating Preschool Educators’ Implementation of Computer Programming in Their Teaching Practice.” *Early Childhood Education Journal* 48 (3): 253–262.

Palmer, S. 1978. “Fundamental Aspects of Cognitive Representation.” *Cognition and categorization*, edited by E. Rosch & B. L. Lloyd, 259–303. Lawrence Erlbaum Associates.

Palmer, H. 2017. “Programming in Preschool-with a Focus on Learning Mathematics.” *International Research in Early Childhood Education* 8 (1): 75–87.

Papadakis, S., and M. Kalogiannakis. 2019. “Evaluating a Course for Teaching Introductory Programming with Scratch to pre-Service Kindergarten Teachers.” *International Journal of Technology Enhanced Learning* 11 (3): 231–246.

Papadakis, S., M. Kalogiannakis, and N. Zaranis. 2016. “Developing Fundamental Programming Concepts and Computational Thinking with Scratchjr in pre-School Education: A Case Study.” *International Journal of Mobile Learning and Organisation* 10 (3): 187–202.

Papert, S. 1980. *Mindstoons*. New York: Basic Books 607.

Perisic, I. 2018. *How Artificial Intelligence is Shaking up the Job Market*.

Pugnali, A., A. Sullivan, and M. U. Bers. 2017. “The Impact of User Interface on Young Children’s Computational Thinking.” *Journal of Information Technology Education: Innovations in Practice* 16: 171–193.

Rial-Fernández, B., and L. P. Santacruz-Valencia. 2019. “The Teaching of Programming is Not the Future but the Present”. In 2019 *International Symposium on Computers in Education (SIIE)*, pp. 1–6.

Roussou, E., and M. Rangoussi. 2020. “On the use of Robotics for the Development of Computational Thinking in Kindergarten: Educational Intervention and Evaluation.” *Advances in Intelligent Systems and Computing* 1023: 31–44.

Sanford, J. F., and J. T. Naidu. 2016. “Computational Thinking Concepts for Grade School.” *Contemporary Issues in Education Research (CIER)* 9 (1): 23–32.

Strawhacker, A., M. Lee, and M. U. Bers. 2018. “Teaching Tools, Teachers’ Rules: Exploring the Impact of Teaching Styles on Young Children’s Programming Knowledge in ScratchJr.” *International Journal of Technology and Design Education* 28 (2): 347–376.

Sullivan, A., and M. U. Bers. 2019. “Investigating the use of Robotics to Increase Girls’ Interest in Engineering During Early Elementary School.” *International Journal of Technology and Design Education* 29 (5): 1033–1051.

Sullivan, A., M. Elkin, and M. U. Bers. 2015. “Kibo Robot Demo: Engaging Young Children in Programming and Engineering”. In *Proceedings of the 14th international conference on interaction design and children*, pp. 418–421.
Tejera, G., G. Amorin, A. Sere, N. Capricho, P. Margenat, and J. Visca. 2019. “Robotito: Programming Robots From Pre-School To Undergraduate School Level”. In 2019 19th International Conference on Advanced Robotics (ICAR).

Urlings, C. C., K. M. Coppens, and L. Borghans. 2019. “Measurement of Executive Functioning Using a Playful Robot in Kindergarten.” Computers in the Schools 36 (4): 255–273.

Wang, D., T. Wang, and Z. Liu. 2014. “A Tangible Programming Tool for Children to Cultivate Computational Thinking.” The Scientific World Journal 2014: 10, Article ID 428080. doi:10.1155/2014/428080

Wing, J. M. 2006. “Computational Thinking.” Communications of the ACM 49 (3): 33–35.

Yasar, O. 2017. “The Essence of Computational Thinking.” Computing in Science Engineering 19 (4): 74–82.