Computational Thinking and Educational Robotics Integrated into Project-Based Learning

Albert Valls (albertvalls80@gmail.com)
La Salle Campus Barcelona-Ramon Llull Universtity

Research Article

Keywords: Computational Thinking, Educational Robotics, Project-Based Learning, STEAM

Posted Date: January 31st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1303413/v1

Abstract

In the context of STEAM discipline in education, subjects tend to fall into contextualized activities or projects. Educational robotics and computational thinking have the potential to be subjects, although not all educational programs provide for it. Although the use of technology and programming platforms is widespread, it is not common practice to integrate computational thinking and educational robotics into the official curriculum in the field of secondary school.

That is why this paper continues with the initial project of integrating computational thinking and educational robotics in La Salle Bonanova school, a Secondary school in Barcelona, Spain. This study presents a project-based learning where the main focus is the development of skills related to STEAM subjects and the acquisition of computational thinking knowledge in a 2nd year of use a block-based programming environment.

Introduction

The use of educational robotics (ER) as a pedagogical resource is a great source of research and is quite integrated in schools and high schools. (Benitti, 2012). Similarly, computational thinking (CT) as such (J. M. Wing, 2006) is being introduced in the pre-university stages (Grover & Pea, 2013). Educational robotics and computational thinking are being introduced either as subjects themselves or as part of learning within the environment of science, technology, engineering or mathematics (STEM subjects) (Stohlmann, Moore, & Roehrig, 2012); without neglecting creativity and Art-STEAM as an indispensable part of student development (Bequette & Bequette, 2012). Educational robotics and computational thinking, directly or indirectly evoke the use of technology based on ideas such as experimentation, design or programming using computers (Papert, 1980).

Virtual software-based programming has evolved into tangible programming (Horn & Jacob, 2007). The development of this learning has followed the pretexts of constructivism; either by considering the student as the builder of his own knowledge (Piaget, 1973) or considering the person or student as the result of a social and cultural interaction with the environment (Vigotsky, 1978). It is also necessary to consider that programming by tangible elements has been based on constructionism to elaborate proposals of learning (Papert & Harel, 1991).

Educational robotics covers many technological aspects, but is basically focused on the use of educational robots, basic programming knowledge and problem solving in the engineering environment (Danahy et al., 2014). On the other hand, computational thinking, at a pre-university level, is being focused on laying the foundations for future programmers, code developers and enhancing the skills needed for the 21st century (J. Wing & Wing, 2017), and therefore wants to promote activities or projects to work these types of logical thinking.

Concepts related to STEAM subjects, robotics and computing are an important basis for the growing demand or lack of technology professionals (Vázquez & Manassero, 2017) and a way to encourage motivation, increasingly scarce, towards these disciplines (Solbes Matarredona, 2010). The integration of STEAM subjects in the educational curriculum is the basis on which these technical disciplines can be strengthened (Stohlmann et al., 2012). This integration into curricula can be done through technological platforms, such as robotic kits or programming tools, which will prove to be the means for learning educational robotics and computational thinking. At the same time, the knowledge society promotes the use of ICT as a fundamental part of the development of the person in society (Cobo, 2009). For these two reasons, the connection between the world of education and social reality is the basis of learning in schools. Thus, increasingly, STEM or STEAM learning is based on project-based learning (PBL) (Sahin, 2013). The application and development of PBL as a methodology not only increases motivation and learning, but also promotes the development of the skills needed for the 21st Century (Bell, 2010). One of the aims is the learning of computational thinking through the use of technological programming platforms in the field of education, in a global and interconnected framework such as PBLs.

Although not all education systems incorporate this content into the official curriculum in secondary stage, most governments, organizations or teachers encourage initiatives so that students can participate in events where the skills and knowledge of these technological disciplines are developed. In recent years, events such as FIRST® LEGO® League, World Robotics Olympiad®, RoboCup® o VEX® Robotics Competition have encouraged their increased participation based on motivation and learning from STEAM disciplines. Although these competitions are not based on the official curricula of some states (as is the case in Spain), they do have common features of concepts and competencies of robotics and computational thinking. (Alimisis, 2013) (Atmatzidou & Demetriadis, 2016).

There are also numerous examples and studies that integrate the disciplines of robotics and computational thinking in different stages of education. Of note are the studies for the introduction to the use of computational thinking and a specific curriculum in early childhood education, both with robots and tangible interface programming (Bers, Flannery, Kazakoff, & Sullivan, 2014), and in computational language using programmable blocks (Portelance, Strawhacker, & Bers, 2016). In primary education it is necessary to emphasize the use of platforms such as LEGO® WeDo® (Scaradozzi, Sorbi, Pedale, Valzano, & Vergine, 2015) or LEGO® Mindstorms® (Dagdilelis, Sartatzemi, & Kagani, 2005) and block programming using Scratch™ (Resnick et al., 2009). Other studies also focus on applications from other platforms and have questioned the relationship between robot-man and the goal of using robots in different areas of learning, such as language, science, and technology (Mubin, Stevens, Shahid, Mahmud, & Dong, 2013).

It is in secondary education where the learning and development of STEAM subjects is mainly developed (Sanders, 2009) along with the application and use of robotics to develop a range of 21st century skills and competencies (Bellaca, 2010). In many of these cases, the basis for the development of the activities comes from the concept of powerful ideas (S. Papert, 1980) and the use of computers and robots to promote socialization and learning (Wooster & Papert, 2006).

The use of technology platforms such as LEGO® Mindstorms® or similar, and block programming such as Scratch™ encourage the learning of programming, educational robotics and computational language in educational settings. (Resnick et al., 2009) . These platforms applied in the high school stage encourage creativity in contextualized environments (Valls, Albó-Canals, & Canaleta, 2018). Both platforms have in common the connection between constructivism and constructionism. As a common feature, both educational theories evoke a construction of knowledge from previous ideas, where students are the creators of their own cognitive tools as well as their external realities. In the same way that a scientist's vision changes depending on the environment, students' knowledge creation is influenced by the learning environment. (Ackermann, 2001). The most significant difference is at a deeper level, where Piaget's interest in constructivism lies primarily in building internal stability, conserving, and reorganizing assets for learning, while Papert's constructivism is more interested in the dynamics of change or the discovery of novelty (Ackermann, 2001).

Currently, trends in teacher training also clearly mark the future of education and are of great importance within universities. For this reason, the TPACK Model (Mishra & Koehler, 2006) has been significantly important in terms of teacher training in the technological field. The use of technology (T) with pedagogical use (P) and focused on enhancing knowledge (K) without neglecting the teacher's environment (C), is a good theoretical framework to encompass the use of new technologies.

This training by teachers is the indispensable basis for the application and promotion of the use of technological platforms for knowledge. However, the TPACK model is able to go slightly beyond the teaching environment and can be approached in a very similar way to the profile and environment of the student. This approach, so far, is being carried out in the format of project-based learning and results in an approach of knowledge to the social and real environment of the student. All of these learning methodologies or models are not far removed from Piaget's constructivism (Piaget, 1973) but with a substantially different factor if one considers that in the middle of the 21st century society is hyperconnected. This connectivity affects many areas, both in social and daily life and in the acquisition and dissemination of knowledge. Connectivism (Siemens, 2005) has acted as an enhancer in obtaining knowledge and information.

The application of the TPACK model has been focused from the beginning on the environment of teacher training (Saengbanchong, Wiratchai, & Bowarnkitiwong, 2014), analysing and applying it only to the adult environment when they are trained to be teachers and focusing on their learning process (Schmidt et al., 2009). If only this circumstance is taken into account, it could be said that the TPACK model would be "for teachers". Given that it is a model applied to training and all related aspects (T, P, C, K), little potential is taken away from it if it is only applied to teacher training, as the great "target" of students is not in the postgraduate or masters' stages, but in compulsory education (mainly primary and secondary).

Project-Based Learning is another learning methodology that is at another level than TPACK model. Working with a PBL environment meets many of the requirements for developing skills of 21st century (Bell, 2010) and acquiring knowledge in a cross-cutting and interconnected way.

1.1 Background

With the desire to continue advancing in the development of a specific curriculum for educational robotics and computational thinking, La Salle Bonanova School (Barcelona, Catalonia) faces the 2nd year of the implementation of educational robotics and computational thinking as part of the curriculum of the Secondary stage. The secondary stage is distributed over four years ages as specified in the Government of Catalonia's own law (de Catalunya, G.,2009), and the curriculum details the contents and skills that students must achieve at the end of the educational stage and the digital age in which they are immersed (de Catalunya, G., 2013) (d'Ensenyament, D., 2015).

The subject of technology is included in this curriculum and therefore the contents and competencies are very well specified. In the case of computational thinking an educational robotics, there is no clear development of the curriculum, and a certain freedom is left to each school to develop the skills and contents according to their own resources and the methodology they consider most appropriate. In the case of La Salle Bonanova, the structure and methodology for the inclusion of these two aspects were well defined in the first year of implementation of the project (Valls et al., 2018). In this first year, the bases were defined so that all the students who attended secondary education developed their own abilities and competences in computational thought, educational robotics and STEAM subjects were strengthened. The main lines that were defined in terms of content were: knowledge related to technology, computer science, coding and the use of information technologies. In terms of methodology and transversal knowledge, the lines were marked based on learning with contextualized activities and the development of skills in five areas: Communication, Collaboration and Community Building, Context Creation, Creativity and Conduct or Behaviour (Bers, 2010).

Literature Review

2.1 Computational thinking

Computational Thinking involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental of computer science (J. M. Wing, 2006). Therefore, the inclusion of the CT in the school curriculum is a fundamental aim in the development of resources and tools for solving problems in STEAM subjects. As predicted, CT is used in schools as basic skills such as reading, writing and arithmetic (J. M. Wing, 2006). However, the inclusion of CT is not really a subject, but is usually linked to the subject of Technology or Robotics. One of the reasons for this is that there is little development of the dimensions or concepts that should be worked on within the discipline of CT. In this way, Brennan and Resnick suggest de following key dimensions of computational thinking framework (Brennan & Resnick, 2012) : computational concepts, computational practices and computational perspectives.

Brennan and Resnick (2012) separate seven computational concepts, four computational practices and three computational perspectives:

Computational concepts

-Sequences: When programming is very important, the activity or task is expressed as a sequence of individual steps or commands that can be performed by the computer. They must be followed gradually and in the correct order.

-Loops: Loops are a tool that allows us to run multiple sequences and also run sequences indefinitely or until some condition that ends the loop is met.

-Events: Events are an important element of other interactive elements as a trigger for a sequence. As an example, we can take the start button, which triggers music playback, changes the setting or causes a movement of an object.

-Parallelism: Parallelism is the implementation of several sequences simultaneously. There are two types of parallelism: parallelism between sprite and parallelism within a single facility. The parallelism between sprites is that several sequences are implemented in the same object. On the other hand, if each object has a series of different actions that are triggered by the same condition: the start button is clicked, in this case, a parallelism is used within a sprite.

-Conditionals: This is the ability to make decisions or take actions based on certain conditions. This is to determine the conditions under which a sequence takes place.

-Operators: Operators providing support mathematical and logical expressions and strings that allow to perform mathematical operations (addition, subtraction, division, multiplication, function, ...) and operations with strings (grouping, a length of the string, ...).

-Data: Data involves storing, retrieving, and updating values. We can use variables and assign the value of a number or string, and with lists that may contain a set of numbers.

Computational practices

-Being incremental and iterative: Designing a project is not a clean, sequential process of first identifying a concept, then developing a design plan, and then implementing it in code format. It is an adaptive process, in which the plan can change in response to the approach of a solution in small steps. It can be described as iterative cycles of imagining and constructing: developing a little, then trying it out, and then developing more, based on your experiences and new ideas.

-Testing and debugging: It is crucial for designers to develop strategies to deal with and anticipate problems, as things rarely work as imagined. A fundamental practice of programmers is test and debugging, which were developed through trial and error, transfer of other activities or with the support of other colleagues.

-Reusing and remixing: Relying on other people's work has been a long-standing practice in programming and has only been amplified by networking technologies that provide access to a wide range of other people’s work to reuse and remix. Reuse and remixing support the development of code-reading skills and help you find ideas and code to build on, which allows you to create much more complex things than you could have created on your own.

-Abstracting and modularizing: Abstraction and modularization, characterized as the construction of something large by assembling smaller pieces, is an important practice for all design and problem solving. Designers use multi-level abstraction and modularization, from the initial work of conceptualizing the problem to translating the concept into individual sprites and stacks of code. The modularization of the behaviors of the object makes it easier to think (try / debug) the different parts of the work or problem

Computational perspective

At a dimension not captured by the framework of concepts and practices, perspective is described by designers and programmers as the evolution of their understanding of themselves, their relationships with others, and the technological world around them.

-Expressing: People spend time surrounded by interactive media as simple consumers, performing activities that are important for learning to use technology, but that are not sufficiently developed as a computer thinker. A computational thinker sees computing as more than something to consume; computing is something that can be used for personal design and expression and is seen as a means to create and express one’s own ideas.

-Connecting: Creativity and learning are deeply social practices, and so designing computer media is surprisingly enriched by interactions with others. It has been observed as the wide variety of ways in which individual creative practice has benefited from access to others, through face-to-face or online interactions. Young developers described the power of accessing new people, projects and perspectives through these networks, a change of perspective succinctly expressed as, "I can do different things when I have access to others."

Creating with other people describes how they can do more than they can handle on their own, whether it's answering questions in online forums, studying and mixing others, or establishing intentional partnerships and collaborations. By creating for others, the value of the authentic audience is experienced. It was appreciated that others were getting involved and appreciating their creations, whether it was entertaining others, equipping others, involving others, or educating others.

-Questioning: Everyday life is increasingly regulated by complex technologies that most people do not understand or believe can do much to influence (Bandura, 2001).

With the computational perspective of questioning, we look for indicators to avoid feeling this disconnect between the technologies around us and the ability to negotiate the realities of the technological world. Young developers should feel empowered to ask questions about and with technology: "I can (use computing to) ask questions to make sense of (computational things) the world." Questioning involves questioning what is taken for granted and, in some cases, answering that question through design and development.

Although in our research we were focused on the computational concepts and computational perspective, the literature introduces us to some concepts, practice and perspective of CT that have been studied.

Computational concepts like sequences, loops and conditionals were identified were basics in young learners (Zhang & Nouri, 2019). As with students in grades 4th to grade 6th (age 9-12) a high percentage are able to complete sequence-related tasks (Franklin et al., 2017). The conditional concept was conceived by students, but not fully, particularly when the conditional block was combined with other blocks such as nested within a forever loop (Lye & Koh, 2018). One of the most difficult concepts for students in these stages is repetition (loops), this is because students may not know the condition and procedures needed to complete a loop (Grover, Pea, & Cooper, 2015). All these studies are developed in the stages of primary education, especially between age 9 and age 12 and them suggest that it cannot be assumed that students' understanding of CT concepts will necessarily improve when using a block-based programming environment. However, the research of this work focuses on ages 12-14, where students have already worked with a block-based programming environment (Valls et al., 2018) and have developed skills and concepts of computational thinking that can improve results.

The computational practice has also been analyzed as it has been found that framing computational thinking only around the concepts was insufficiently represented by other elements of the learning and participation of designers (Brennan & Resnick, 2012). One of the most similar conclusions to other learning is that computational practices focus on the process of thinking and learning, going beyond what you are learning to how you are learning. The CT practice is an element that raises more doubts than the CT concepts and questions arise such as: How to design a valid assessment tool for TC practices? can algorithmic thinking, abstraction, testing and debugging, and reuse and remixing be used at the 4^{th} to 6^{th} grade level? What do these assessment tasks tell us about the student? (Basu, Rutstein, Shear, & Xu, 2020). Other work in the STEAM high school disciplines has shown improved pre and post-test scores based on computational thinking practice (Arastoopour Irgens et al., 2020).

The third point of view of CT is perspective, where according to Brennan and Resnick (2012) it focuses on the evolution of their understanding of themselves, their relationships with others, and the technological world around them. These three aspects, according to their experience are related to the interaction that developers have with each other, especially in the aspects of how to express, contribute and ask each other peers. In this sense, there are few studies that have challenged this fact in the secondary stage, but instead other researchers have focused on the early stages of students (Bers, Flannery, Kazakoff, & Sullivan, 2014). These studies in other education stages have developed tools to be able to observe more objectively these concepts and where a series of competences such as communication, creativity or collaboration are defined (Bers, 2010), and which have a similarity with the definition from the perspective of Brennan and Resnick (2012).

Although not explicitly defined as CT perspective, other studies tend to define some skills related to CT (Atmatzidou & Demetriadis, 2016) as the capacity for abstraction or modularization, but following Brennan and Resnick (2012) these definitions coincide more with CT practices than with CT perspective.

CT and all its magnitude has been analyzed in many disciplines and has been related to other types of thinking, so that an attempt has been made to define its characteristics, the relationship between computer science and programming, the most appropriate tools for to develop it or the skills of the CT (Shute, Sun, & Asbell-Clarke, 2017).

2.2 Scratch

Scratch™ is one of the most popular and widely used block-based programming platforms. From the conception of the main idea (Resnick, M., Kafai, Y., Maeda, J., et al. 2003) as the post-development (Resnick et al., 2009) or the explanation of the environment (J. Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010), it has been used to develop and enhance many concepts of CT.

The literature shows us many studies, workshops and works related to Scratch™. In primary education, studies should focus in tutorials on how to teach and learn Scratch (Marji, 2014), computer games (Wilson, Hainey, & Connolly, 2013), mazes development (Ternik, Koron, Koron, & Šerbec, 2017) or evaluation programming concepts (Fatourou, Zygouris, Loukopoulos, & Stamoulis, 2018).

In many other cases, researchers have attempted to establish how many blocks they use in creative projects, either in science projects in fifth grade (Baytak & Land, 2011) at computer clubhouse (J. H. Maloney, Peppler, Kafai, Resnick, & Rusk, 2008) or at summer campus in middle school (Adams, 2010).

Regardless of the intention to use this platform for other uses, the main goal of Scratch™ is to be used to program and thus learn the concepts, practices and perspectives of computational thinking (Brennan & Resnick, 2012).

Scratch™, like any other programming platform, requires basic learning by students. In the event that students are relatively young and new to programming (like stages 4th), the main goal of the learning courses is to introduce the core features of Scratch and CT through content and programming activities that are significantly introductory to students (Fagerlund, Häkkinen, Vesisehano & Viiri, 2020). Projects such Wilson, Connolly, Hainey and Moffat (2011) shows how Scratch can be used with young children aged 8-9 years old to learn programming concepts through the introduction of game making.

Scarcth's learning and use is based on students' prior knowledge like any other discipline, subject or knowledge (Ausubel, Novak, & Hanesian, 1976). Just as constructivism evokes the student to be the protagonist of their own learning (Piaget, 1973) and constructionism proposes tangible learning elements (Seymour Papert & Harel, 1991), Scratch has emerged as one of the platforms to encourage self-learning based on elements that the student can create, imagine, design, modify... and learn computational thinking.

2.3 Project-Based Learning

Both, PBL and inquiry-based learning are active learning methods that are based on the philosophy of John Dewey, who believed that education begins to arouse students' curiosity (Dewey, 1938).

Sahin (2013) identifies similarities between these two methods are obvious if we consider that project-based learning is an educational approach that uses student-centre research processes to develop a product that has real-life connections and applications. Specifically, PBL contains research-based tasks that help students develop important content from the technological, social, and core curriculum . PBL has also been defined as a special case study (Slough & Milam, 2008), making STEAM PBL and inquiry-based learning go hand in hand in terms of instruction focused on student.

However, the differences are significant because in the inquiry-based learning method, questions and curiosity of students form the central part of the curriculum. Students are encouraged to ask questions and conduct research on topics that interest them to make their own discoveries (Kessler & Galvan, 2007). Inquiry-based learning begins with a question followed by research. This research may involve the collection of data and the development of new knowledge; and at the end of this process, students reflect on their new knowledge (Branch & Oberg, 2004).

Instead, STEM PBL starts with the final product in mind. Students are introduced to the PBL through a well-defined outcome that conveys clearly defined expectations and limitations for completing the task (Capraro & Slough, 2008). Development through STEM PBL differs from inquiry-based learning in the emphasis on the construction of artifacts or objects by students to represent what they have learned.

Another outstanding feature of PBL in STEM projects is that it is usually unstructured. This is due to the fact that the PBL normally works in small groups of students and each group has to organize its own work, materials, individual tasks and manage its own time (Sahin, 2013)

Thus, in STEM project-based learning, students take charge of their own learning and develop collaborative skills. At times, the less structured nature of project-based learning can make PBL classrooms seem disorganized or out of control.

One of the key factors in the successful use of PBL in STEM or STEAM disciplines is the interdisciplinary nature (Serkan, 2013). That is why the use of these methodologies is an increasingly constant fact in the current education of STEAM subjects, computational thinking or educational robotics and there is a desire to create a theoretical framework for design STEM projects (Slough & Milam, 2013) or a an integrated STEM approach (Capraro, Capraro, & Morgan, 2013), as well as focusing on the integration of STEAM subjects to enhance Informal education and community collaboration through engineering (Burrows, Lockwood, Borowczak, Janak, & Barber, 2018).

Course Design And Methodology

3.1 Educational context

In the first year and throughout a non-pandemic academic year, 1st, 2nd and 3rd grade of Secondary, 445 students took the introduction to educational robotics and computational thinking. This first contact with these disciplines develops in the corresponding role (Valls et al., 2018) and as a summary it consisted of:

-15 sessions to introduce block programming platforms such as Scratch™ and LEGO® Mindstorms® as basic tools for computational thinking and educational robotics

-Work in two different groups: In one of the two groups the activities were not contextualized and in the other they were contextualized.

-Differentiation of the sessions between one group and the other.

-Definition of the contents related to programming.

-Definition of the competencies to work grouped into five areas.

-Analysis of the results obtained by the two groups

In this second year of implementation, activities are still carried out in all the courses described above, but in this role the focus is on the project-based learning that takes place in the 2nd year of secondary education (N=160 students, ages 13-14). Students in this course took a subject in robotics for the first time in the previous year, and the authors of this paper decided to delve into the basic aspects of PBL to enhance computational thinking and knowledge of educational robotics. That is why the PBL decided to be part of the knowledge of Scratch™ acquired in the previous course.

3.1 Previous knowledge acquired

As in any other discipline in education, robotics and computational thinking, teaching content is based on prior knowledge (Ausubel et al., 1976), and help and resources are provided to achieve the learning objectives (Brunner, 1987). To develop the full potential of the project, students in these courses are based on the concepts they acquired in the previous course in the Scratch™ activity (Valls et al., 2018). Regardless of the group they were part of, the use of blocks programming platform such Scratch™ and knowledge about computational concepts were assimilated by the vast majority of students (M=5.31, SD=1.04 and M=6.00, SD=1.05).

These aspects are shown in more detail in Table 1 and are the same ones that have been worked during the development of the sessions in this research project.

Table 1. Evaluated items on Scracth™ platform

Conditional	- Understand the concept of conditional structures - Use different types of conditional structures
Loops	- Understand the concept or loop iteration - Use loops within the structure of the game
Events (Objects)	- Use various objects - Import objects outside environment Scratch - The motion control is done several ways - The use of objects follows criteria established
Events (Scenario or Dresses)	- Use different scenarios - Make changes in objects dresses
Parallelism	- Implement several sequences in the same object - Different actions for each object
Bloc Posts	- Use the blog post to give orders objects
Sequences (Text)	- Use language structures - Dialogues appear
Operators	- Use variables to make a counter + - Use variables to make a counter - - Conditions certain actions variables
Events (Music / Sounds)	- Use music blog - Use varied sounds - Use block sound conditioning in another action

3.2 Previous skills development

In today's education, concepts are still important, but student development based on their competencies, skills, and abilities is gaining momentum. As mentioned above, there is no specific robotics curriculum that develops the content, and in the same way there is no curriculum to develop the skills. There are several jobs in the field of early childhood education (Bers, 2010) in which a series of capabilities are defined that could be described as generic without being specific to a curriculum. These five areas are the ones used in the first year of application of this project (Valls et al., 2018), and they are the ones that continue to be worked on and developed in this role. These competencies are shown in more detail in Table 2.

Table 2. Competencies aspect evaluated

C1 Communication	C1.1 Exchange of ideas among group members C1.2 Expression of ideas and debate them C1.3 Demand for teacher support and is beneficial for the project
C2 *Collaboration and Community Building*	C2.1 Help peer group C2.2 The individual contributions make the group advance C2.3 Different work roles/ Tasks diversity
C3 *Context Creation*	C3.1 The activity follows a structure designed C3.2 Analysis of the errors in the process C3.3 Justification of the solution C3.4 Write the process of solution to the challenge
C4 *Creativity*	C4.1 Initiative to make further steps in programs C4.2 Use of various elements outside environment platform C4.3 Application of concepts from other disciplines
C5 Conduct	C5.1 Concentration activity C5.2 Following the rules of the classroom C5.3 Responsible use of the material C5.4 Behaviour with classmates and teacher C5.5 Motivation towards activity

In the case of competency development, the assessment was done following a Linkert scale of 1 to 5. The results of each group expressed with averages were C1: 2.68-3.29, C2: 2.48-2.96, C3: 2.25 -2.64, C4: 1.66-2.57 and C5: 3.73-3.95. The first number of each competency evaluated corresponds to the group in which it was worked without a contextualization of the activities and the second, corresponds to the group that did work with contextualized activities.

As the authors point out (Valls et al., 2018), there is a significant difference in competencies between groups. In this way, they reaffirm the conclusions of the first work in the use of contextualized activities to improve the abilities and skills related to educational robotics and computational thinking. These conclusions give value to the present study and thus, to be able to follow an even more inclusive methodology and apply a PBL in the learning of educational robotics and computational thinking.

Following the work of Brennan & Resnick (2012) in reference to CT perspective, the authors of this study have related the five competences assessed to the items of the CT perspective, making the relationship shown in table 3:

Table 3. Relationship between CT perspective and competences

Computational Perspective (Brennan & Resnik, 2012)	Competences (Bers, 2010) (Valls et al., 2018)
Expressing	C1 Communication
Questioning	C2 Collaboration and community building
Connecting	C3 Context creation C4 Creativity

This relationship can be used in future research to establish connections between competencies within the curricular framework and can help to improve both the design of activities and the evaluation of these.

3.3 Methodology

The work methodology used is that based on project-based learning. The main axis of the PBL is to apply in order to improve the skills of the 21st century and therefore follows a series of guidelines and premises so that students are the drivers of their learning and the teacher the guide or facilitator (Bell, 2010).

Students will present the PBL as if it were the final project of the Scape Room using the block programming platform Scratch™. As this is the 2nd year of use of this platform, they already start from previous knowledge and therefore it is not necessary to make a conceptual introduction or of the contents of the platform. During the work sessions, the students have to develop the history and the programming of the Scape Room, and the teacher serves as a guide to solve the doubts and to consider the way to follow in the elaboration of the final product.

In the PBL methodology, peer collaboration is very important, as it involves sharing one's own knowledge towards the benefit of the group to reach a common goal (Bell, 2010).

Trial error is also considered a very important factor. As this is a Scape Room, where prior to coding it has been necessary to create a thread and draw up a list of all the objects to be programmed (as best specified in section 3.3.2), it is this previous design may change during the development of the program. This fact is fully assumed for the teachers who teach the project but not for most students. It is precisely at this point, when substantial changes occur between the previous design and the material realization of the programming where the difficulties will appear and where a development of 21st century skills will be generated such as communication, respect for others, work in team and problem solving (Bell, 2010); and a content acquisition towards computational language and programming.

3.3.1 Criteria for the composition of working groups

All 2nd grade students were included in this study, a total of 160 students participated, and they were distributed in five group classes of 32 students each class. 60% of these 160 students were boys and 40% of them were girls. The age range was between 13 and 14 years.

In the current project, the work in the classroom is based on groups of four, encouraging peer collaboration and cooperative learning within the subject of technology. This cooperative work seeks to help students better identify concepts, perform better problem analysis, and foster peer relationships (Slavin, 1980).

In the previous study carried out by the same authors at La Salle Bonanova School (Valls et al., 2018), educational robotics and computational thinking were introduced. As already described in this study, two different working methodologies were differentiated and led to a significant difference in level in the assimilation of competencies between members of one group and another. Although the assimilation of the computational concepts was homogeneous in the two groups, this was not the case in the competences.

To correct for this difference, new criteria is proposed: the working groups of four students must be formed heterogeneously, selecting two students of one group of the five existing class group and the other two must be of the other group. Students from the same methodology group were randomly selected.

The time distribution allows applicators to work with middle class and therefore have 16 students (4 groups of 4 students) and be able to conduct two-hour sessions.

3.3.2 Development of the sessions

The PBL that is developed is a Scratch Room, where through the Scratch™ platform a Scape Room is designed. The distribution and description of the sessions are shown in Table 4 :

Table 4. Sessions distribution

Sessions	Developed
1	Groups of four students are formed and an explanation is given of what a Scape Room is. Students are asked to brainstorm a possible Scapes Rooms. A worksheet is handed out for each group to start drawing a Story Board from the chosen Scape Room.
2	The Story Board is finished, and a list is made of all the possible objects, characters and backgrounds that are needed. Start searching within the Scratch platform for the objects and backgrounds that are needed, otherwise can search it at the websites.
3, 4, 5	During these three sessions, students must develop the design and, above all, the programming of their Scratch Room. They must define the complexity of their project and thus establish their learning.
6	An oral presentation is given on each Project where students explain the operation of their Scratch Room. It is a checkpoint to improve and debug the project. In the open turn there are comments and observations from the rest of the groups.
7, 8, 9	During these three sessions students continue to develop their project and improve based on the observations from the previous session.
10	Presentation of the final project: oral presentation, demonstration and co-evaluation

Results

During the sessions, different aspects were evaluated, such as the competences and the contents related computational thinking and programming with the Scratch™ platform. In the evaluation of the PBL, the grades come from three fronts.

The first is based on the direct observation of each student with a rubric, either on the interaction with the members of his group or the individual development during the sessions, in this way the assessments of the competences C1, C2, C3, C4 and C5 are obtained.

Competence C5 is not typical of the CT (like shows table 3) but of a cooperative work and therefore, the authors consider that it is basic to be able to evaluate the work among equals.

Another point that has been valued is the oral presentation and how they have developed their project idea and justified the decisions made.

The third point is the assessment that has been made of the final project and the involvement of the members themselves through a coevaluation.

Although three different aspects have been assessed, the authors focus their attention on competencies and not so much on oral presentation and coevaluation. The reason for this is that one of the aims of the development of this project-based learning is to be able to observe if the development of certain competences related to computational thinking are favoured, as already described above, these competences are those defined in the first year of implementation and correspond to Positive Technological Development (PTD) (Bers, 2010).

It has already been mentioned that no special grouping has been done or any different methodology has been applied. Assuming that the groups were homogenized in relation to the first year of application of the project and with the corresponding weighting, the average of the five areas for all students following a Likert scale (1-5) are shown in Table 5.

Table 5. Mean and Standard Deviation of competences

Competences	C1		C2		C3		C4		C5	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
	3.22	0.62	2.98	0.72	3.17	0.71	3.19	0.70	3.98	0.66

Similarly, the evaluation of the CT Concepts with Scratch^{TM} platform follows the items listed in Table 1. In this case the evaluation was performed on a scale of 0 to 10, by quantifying the functional blocks corresponding to the concepts of CT, and the results are shown as Mean of all of them in Table 6.

Table 6. Mean and Standard Deviation of CT concepts with Scracth^{TM}

CT concepts	Scratch™	
	Mean	SD
	6.36	0.9

In this way, it is possible to check whether the results of the study carried out in the previous year are taken into account. In the last study the group that did not perform any active methodology and worked on robotics and computational thinking in a classical way obtained inferior results than the group in which a methodology was applied where the activities were contextualized, and competencies and the learning process were taken into account.

Table 7. Comparison 1st year vs 2nd year of competent areas

Competences	1st year					2nd Year	
	1st Group		2nd Group		1st & 2nd mixed group		
	Mean	SD	Mean	SD	Mean	Mean	SD
C1	2.68	0.50	3.29	0.69	2.98	3.22	0.62
C2	2.48	0.64	2.96	0.74	2.72	2.98	0.72
C3	2.25	0.77	2.64	0.80	2.45	3.17	0.71
C4	1.66	0.65	2.57	0.80	2.12	3.19	0.70
C5	3.73	0.86	3.95	0.77	3.84	3.98	0.66

Table 7 shows how the overall results of the 2nd year competencies are very similar or higher than those of the 2nd group of the 1st year. This is significantly relevant since in this 2nd year the groups were mixed and were formed by students who received, in the previous year, different methodologies. Despite the difference in the degree of acquisition of skills, the methodology used promotes rapid and meaningful learning of the most competent aspects of STEAM subjects, as well as educational robotics or computational thinking.

The other fundamental aspect of this work is the evaluation of concepts related to CT. The assessment of the use and acquisition of knowledge of CT has been revived with the Scratch™ platform. Table 6 shows the mean obtained in these concepts in this 2nd year of the project working with PBL format (M=6.36, SD=0.9) and contrasts with the results obtained in the 1st year (M=5.66, SD=1.03) where the work environment were contextualized activities (Valls et al., 2018).

Conclusion

The elaboration of this paper, and other previous studies related to STEAM subjects computational thinking and project-based learning, has brought to light a series of elements within the educational field to be taken into account.

First of all, it should be noted that an active methodology such as project-based learning can increase the performance and motivation of students, and thanks of these, they become the protagonist of their learning and generate a physical product as a result of their learning.

Students who have participated in this 2nd year of using block-based programming platforms with a PBL environment improve in their skills and acquisition of knowledge of computational thinking and in the use of educational robotics platforms. This improvement in the area of computational thinking is done in virtually every aspect: concepts, practice, and perspective; and also relating to the competencies previously defined in this study, such as communication between classmates, creativity, collaboration and community building and context creation, as seen in the results described in Table 7.

In relation to this last statement, the authors conclude that the relationship between TC practice and competences that shows on table 3 is a good first approximation, as both represent how students interact with each other within programming environments.

It remains to establish the competencies of the secondary school curriculum in the legal framework and to be able to describe it more accurately. It should be noted that both competencies and CT practices were described on very specific platforms (such as Scratch^{TM}) and should be globalized in order to be able to develop them without a specific programming platform.

Another positive aspect to highlight in the implementation of this study is the possibility of repeating and graduating. This type of PBL can be reproduced within the same school in other STEAM subjects and also in other schools. The possibility of repeating and graduating can be carried out by adapting the time distribution and the composition criteria of the work groups, adapting the level according to the previous knowledge and the educational stage, but maintaining the same curriculum and the objectives of learning of CT practice and concepts.

Finally, the authors of this study conclude and demonstrate that the integration of cross-curricular computational thinking into STEAM subjects is an added and necessary value for schools.

With the data obtained, it can be seen that for the La Salle Bonanova school, the development of the skills of the 21st century and the learning and development of the computational thinking of the students is an important aim and is the right direction for school's technological educational project. The La Salle Bonanova school project can be consolidated by encouraging the learning of ER and CT and designing a strategy to integrate it into the curriculum of STEAM subjects in secondary stage.

Declarations

Acknowledgments

First, we must thank La Salle Bonanova School for her willingness to implement this study as a school project.

We would also like to thank the collaboration, help, and above all, moral support of the fellow writers of this work.

Last but not least, special thanks to the students involved in the study, who with their enthusiasm have helped to overcome the difficulties that have arisen along the way.

Compliance with Ethical Standards

This study was conducted in a secondary school and during the development of a non-pandemic school year. This fact in itself implies that the parents received and accepted the corresponding informed consent at the beginning of the school year.

The data obtained are confidential and are treated in accordance with current data protection law, whether personal data or academic notes. The study also complies with the curriculum and laws of the Department of Education of the Government of Catalonia and is endorsed by La Salle-Ramon Llull University and therefore does not cause any conflict of interest in the possible disclosure.

References

Ackermann, E. (2001). Piaget ’ s Constructivism , Papert ’ s Constructionism : What ’ s the difference? *Future of Learning Groups Publication*.

Adams, J. C. (2010). Scratching middle schoolers’ creative itch. *SIGCSE’10 - Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. https://doi.org/10.1145/1734263.1734385

Alimisis, D. (2013). Educational robotics: Open questions and new challenges. *Themes in Science and Technology Education*, *6*, 63–71. Retrieved from http://earthlab.uoi.gr/theste/index.php/theste/article/view/119

Arastoopour Irgens, G., Dabholkar, S., Bain, C., Woods, P., Hall, K., Swanson, H., … Wilensky, U. (2020). Modeling and Measuring High School Students’ Computational Thinking Practices in Science. *Journal of Science Education and Technology*, *29*(1). https://doi.org/10.1007/s10956-020-09811-1

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students’ computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*. https://doi.org/10.1016/j.robot.2015.10.008

Ausubel, D. P., Novak, J. D., & Hanesian, H. (1976). Significado y aprendizaje significativo. In *Psicología educativa: un punto de vista cognoscitivo (Vol. 3)*.

Bandura, A. (2001). Social cognitive theory: An agentic perspective. *Annual review of psychology*, *52*(1), 1-26.

Basu, S., Rutstein, D., Shear, L., & Xu, Y. (2020). A principled approach to designing a computational thinking practices assessment for early grades. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*. https://doi.org/10.1145/3328778.3366849

Baytak, A., & Land, S. M. (2011). An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom. *Educational Technology Research and Development*, *59*(6). https://doi.org/10.1007/s11423-010-9184-z

Bell, S. (2010). Project-Based Learning for the 21st Century: Skills for the Future. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*. https://doi.org/10.1080/00098650903505415

Bellanca, J. A. (Ed.). (2010). *21st century skills: Rethinking how students learn*. Solution Tree Press

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers and Education*, Vol. 58, pp. 978–988.

Bequette, J., & Bequette, M. (2012). A place for art and design education in the STEM conversation. *Art Education*, (March), 40–48. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+Place+for+Art+and+Design+Education+in+the+STEM+conversation#0

Bers, M. U. (2010). Beyond computer literacy: supporting youth's positive development through technology. *New Directions for Youth Development*. https://doi.org/10.1002/yd.371

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education*, *72*, 145–157.

Branch, J., & Oberg, D. (2004). Focus on inquiry: A teacher's guide to implementing inquiry-based learning. *Canada: Alberta Education, Alberta*.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Annual American Educational Research Association Meeting, Vancouver, BC, Canada*.

Bruner, J.S. (1987). Acts of the meaning. Cambrigde: Harvard University Press.

Burrows, A., Lockwood, M., Borowczak, M., Janak, E., & Barber, B. (2018). Integrated STEM: Focus on informal education and community collaboration through engineering. *Education Sciences*, *8*(1). https://doi.org/10.3390/educsci8010004

Capraro, R. M., Capraro, M. M., & Morgan, J. R. (2013). *STEM Project-Based Learning an Integrated Science, Technology, Engineering, and Mathematics (STEM) Approach*. https://doi.org/10.1007/978-94-6209-143-6

Capraro, R. M., & Slough, S. W. (2008). Project based learning: An integrated science technology

engineering and mathematics (STEM) approach. Rotterdam, The Netherlands,

Cobo, J. C. (2009). El concepto de la informacion Benchamarking sobre las deginiciones de las TIC en la sociedad del conocimiento. *Zer*.

Dagdilelis, V., Sartatzemi, M., & Kagani, K. (2005). Teaching (with) robots in secondary schools: Some new and not-so-new pedagogical problems. *Proceedings - 5th IEEE International Conference on Advanced Learning Technologies, ICALT 2005*. https://doi.org/10.1109/ICALT.2005.255

Danahy, E., Wang, E., Brockman, J., Carberry, A., Shapiro, B., & Rogers, C. B. (2014). LEGO-based robotics in higher education: 15 years of student creativity. *International Journal of Advanced Robotic Systems*, *11*(1). https://doi.org/10.5772/58249

Decret 187/2015, de 25 d'agost, d'ordenació dels ensenyaments de l'educació secundària obligatòria. Diari Oficial de La Generalitat de Catalunya, 6945.

Dewey, J. (1938) Logic: The theory of inquiry. *The Later Works*, (May 1938).

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, *29*(1), 12-28.

Fatourou, E., Zygouris, N. C., Loukopoulos, T., & Stamoulis, G. I. (2018). Teaching concurrent programming concepts using scratch in primary school: Methodology and evaluation. *International Journal of Engineering Pedagogy*, *8*(4). https://doi.org/10.3991/ijep.v8i4.8216

Franklin, D., Skifstad, G., Rolock, R., Mehrotra, I., Ding, V., Hansen, A., … Harlow, D. (2017). Using upper-elementary student performance to understand conceptual sequencing in a blocks-based curriculum. *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*. https://doi.org/10.1145/3017680.3017760

Generalitat of Catalonia Department of Education. Curriculum of Secondary Education. Basic skills in the digital realm. Identification and deployment in compulsory secondary education. (2013)

Grover, S., & Pea, R. (2013). Computational Thinking in K–12. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, *25*(2). https://doi.org/10.1080/08993408.2015.1033142

Horn, M. S., & Jacob, R. J. K. (2007). Designing tangible programming languages for classroom use. *TEI'07: First International Conference on Tangible and Embedded Interaction*. https://doi.org/10.1145/1226969.1227003

Kessler, J. H., & Galvan, P. M. (2007). Inquiry in action—investigating matter through inquiry. 3rd.

Lye, S. Y., & Koh, J. H. L. (2018). Case studies of elementary children's engagement in computational thinking through scratch programming. In *Computational thinking in the STEM disciplines* (pp. 227-251).

Springer, Cham.

Llei 12/2009, del 10 de juliol, d'educació (DOGC núm. 5422, de 16.7.2009, p. 56.589)

Marji, M. (2014). *Learn to program with Scratch: A visual introduction to programming with games, art, science, and math*. No Starch Press.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice. *ACM SIGCSE Bulletin*, *40*(1). https://doi.org/10.1145/1352322.1352260

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, *10*(4), 1–15. https://doi.org/10.1145/1868358.1868363

Mishra, P., & Koehler, M. J. (2006). Technological pedMishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. Teachers College Record. doi:10.1111/j.1467-9620.2006.00684.xagogical content knowledge: A framework for teacher knowledg. *Teachers College Record*. https://doi.org/10.1111/j.1467-9620.2006.00684.x

Mubin, O., Stevens, C. J., Shahid, S., Mahmud, A. Al, & Dong, J.-J. (2013). A REVIEW OF THE APPLICABILITY OF ROBOTS IN EDUCATION. *Technology for Education and Learning*. https://doi.org/10.2316/journal.209.2013.1.209-0015

Papert, S. (1980). Computers for children. In *Mindstorms: Children, computers, and powerful ideas*.

Papert, Seymour, & Harel, I. (1991). Situating Constructionism. *Constructionism*.

Piaget, J. (1973). To understand is to invent: the future of education (G. Roberts, Trans.). *NY: Grossman Publishers, A DIVISION OF THE VIKING PRESS, NEW YORK*.

Portelance, D. J., Strawhacker, A. L., & Bers, M. U. (2016). Constructing the ScratchJr programming language in the early childhood classroom. *International Journal of Technology and Design Education*, *26*(4). https://doi.org/10.1007/s10798-015-9325-0

Resnick, M., Kafai, Y., Maeda, J., Rusk, N., & Maloney, J. (2003). A networked, media-rich programming environment to enhance technological fluency at after-school centers in economically-disadvantaged communities. *Proposal to National Science Foundation*.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., … Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

Saengbanchong, V., Wiratchai, N., & Bowarnkitiwong, S. (2014). Validating the Technological Pedagogical Content Knowledge Appropriate for Instructing Students (TPACK-S) of Pre-service Teachers. *Procedia -*

Social and Behavioral Sciences. https://doi.org/10.1016/j.sbspro.2014.01.252

Sahin, A. (2013). STEM project-based learning: Specialized form of inquiry-based learning. In *STEM Project-Based Learning an Integrated Science, Technology, Engineering, and Mathematics (STEM) Approach*. https://doi.org/10.1007/978-94-6209-143-6_7

Sanders, M. (2009). STEM, STEM education, STEMmania. *The Technology Teacher*.

Scaradozzi, D., Sorbi, L., Pedale, A., Valzano, M., & Vergine, C. (2015). Teaching Robotics at the Primary School: An Innovative Approach. *Procedia - Social and Behavioral Sciences*. https://doi.org/10.1016/j.sbspro.2015.01.1122

Özel, S. (2013). W3 of STEM Project-Based Learning: Who, Where, and When: Revisited. In *STEM project-based learning* (pp. 41-49). Brill Sense.

Schmidt, D. A., Baran, E., Thompson, A. D., Mishra, P., Koehler, M. J., & Shin, T. S. (2009). Technological pedagogical content knowledge (Track): The development and validation of an assessment instrument for preservice teachers. *Journal of Research on Technology in Education*. https://doi.org/10.1080/15391523.2009.10782544

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/J.EDUREV.2017.09.003

Siemens, G. (2005). Connectivism: A learning theory for the digital age. *International Journal of Instructional Technology and Distance Learning*.

Slavin, R. E. (1980). Cooperative Learning. *Review of Educational Research*. https://doi.org/10.3102/00346543050002315

Slough, S. W., & Milam, J. O. (2013). Theoretical framework for the design of STEM project-based learning. In *STEM Project-Based Learning an Integrated Science, Technology, Engineering, and Mathematics (STEM) Approach*. https://doi.org/10.1007/978-94-6209-143-6_3

Slough, S. W., & Milam, J. (2008). Theoretical framework for STEM project-based learning. *Project based learning: An integrated science technology engineering and mathematics (STEM) approach*, 19-37.

Solbes Matarredona, J. (2010). ¿Por qué disminuye el alumnado de ciencias? *Alambique : Didáctica de Las Ciencias Experimentales*.

Stohlmann, M., Moore, T., & Roehrig, G. (2012). Considerations for Teaching Integrated STEM Education. *Journal of Pre-College Engineering Education Research*, Vol. 2, pp. 28–34. https://doi.org/10.5703/1288284314653

Ternik, Ž., Koron, A., Koron, T., & Šerbec, I. N. (2017). Learning programming concepts through maze game in scratch. *Proceedings of the 11th European Conference on Games Based Learning, ECGBL 2017*.

Valls, A., Albó-Canals, J., & Canaleta, X. (2018). Creativity and Contextualization Activities in Educational Robotics to Improve Engineering and Computational Thinking. In W. Lepuschitz, M. Merdan, G. Koppensteiner, R. Balogh, & D. Obdržálek (Eds.), *Robotics in Education: Latest Results and Developments* (pp. 100–112). https://doi.org/https://doi.org/10.1007/978-3-319-62875-2_9

Vázquez, A., & Manassero, M. (2017). El declive de las actitudes hacia la ciencia de los estudiantes: un indicador inquietante para la educación científica. *Revista Eureka Sobre Enseñanza Y Divulgación De Las Ciencias*, *5*(3), pp. 274-292. Recuperado a partir de https://reuredc.uca.es/index.php/eureka/article/view/3740

Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard university press.

Wilson, A., Connolly, T., Hainey, T., & Moffat, D. (2011, October). Evaluation of introducing programming to younger school children using a computer game making tool. In *Proceedings of the Fifth European Conference on Games Based Learning* (pp. 639-649).

Wilson, A., Hainey, T., & Connolly, T. M. (2013). Using scratch with primary school children: An evaluation of games constructed to gauge understanding of programming concepts. *International Journal of Game-Based Learning*, *3*(1). https://doi.org/10.4018/ijgbl.2013010107

Wing, J. M. (2006). ComputationalThinking. *Magazine Communications of the ACM*. https://doi.org/10.1145/1118178.1118215

Wing, J., & Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*. https://doi.org/10.17471/2499-4324/922

Wooster, J. S., & Papert, S. (2006). Mindstorms: Children, Computers, and Powerful Ideas. *The English Journal*. https://doi.org/10.2307/816450

Zhang, L. C., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, *141*, 103607. https://doi.org/10.1016/J.COMPEDU.2019.103607

Abbreviations

ER: Educational Robotics

CT: Computational Thinking

PBL: Project Base-Learning

TPACK: *Technological Pedagogical and Content Knowledge*

STEAM: Science, Technology, Engineering, Arts and Mathematics