Designing a Socio-Cultural Approach for Teaching and Learning Computational Thinking

Francisco Buitrago-Flórez
Project coordinator, Faculty of Engineering, Universidad de los Andes

Giovanna Danies
Assistant Professor, Faculty of Architecture and Design, Universidad de los Andes

Javier Tabima
Postdoctoral scholar, Department of Botany and Plant Pathology, Oregon State University

Silvia Restrepo
Professor, Faculty of Engineering, Universidad de Los Andes

Carola Hernández
Associate Professor, Faculty of Engineering, Universidad de Los Andes
c-hernan@uniandes.edu.co

Abstract
In recent years, Computational Thinking (CT) has been claimed to be a powerful way for competencies and skill development for students at any stage. Programming has been the main tool for CT development; nevertheless, several issues associated with learning and teaching programming exist and have been widely described. In this study, we designed and tested in a qualitative way a set of active pedagogical strategies to implement an alternative way to immerse university students in the learning process of CT development. Such strategies were designed and analyzed under the socio-cultural vision of learning, in which individuals interact in communities to build significant knowledge. Results showed that our curriculum design engaged students in the active use of five key skills related to CT, which could be used as the base ground for further programming learning and high-level CT skills development. Additionally, data indicates that students were involved in reflective processes of learning, as well as in the development of key competencies such as teamwork, communication skills and creativity. Theoretical characteristics of the socio-cultural vision of education are subsequentially used to explain and discussed the scope and limitations of the implementation described herein.

Keywords
Computational thinking, active pedagogical strategies, socio-cultural education, skills, competencies

Introduction
Computational Thinking (CT), as defined by Jeannette Wing in 2015, “is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent” (Wing & Stanzione, 2016). The concept and definition of CT have been a topic of dis-
discussion and research over the past decade due to the capacity of CT to be used as a tool to
develop skills of a high level of complexity in students ranging from K-12 to bachelor pro-
grams and beyond (Bocconi, Chiocciariello, Dettori, Ferrari, & Engelhardt, 2016; Buitrago-
Florez et al., 2017; Curzon, Dorling, Ng, Selby, & Woollard, 2014; French Academy of Sci-
ences (FAS), 2013; Qualls & Sherrel, 2010; Wing, 2006). This paper does not pretend to
review the concept of CT in depth, but to present a discussion of the concept and to define
underlying concepts fundamental to CT. For additional information about CT, the fol-
lowing papers are recommended: Bocconi et al., 2016; Buitrago-Florez et al., 2017; Wing,
2006.

Wing’s (2011) definition of CT comprises three main characteristics. First, CT is a
thought process independent of technology (Bocconi et al., 2016). This suggests that to
teach or learn CT, computers are not necessary. Furthermore, CT is a problem-solving
approach that represents solutions that can be carried out by a human, a machine, or a
combination of both, meaning that the solution does not necessarily have to be processed
by a computer. Second, the use of CT involves a set of high-level skills used to identify and
solve a problem. Some of these skills comprise: abstraction (the process of making a situat-
ion more understandable by reducing unnecessary details), algorithmic thinking (a way in
which a solution is represented in a series of clear steps), decomposition (a way of thinking
about problems and solutions in separate components, reducing the complexity of the sys-
tem), debugging (a systematic approach to finding and resolving defects in the proposed
solution), and generalization (a way to solve upcoming problems based on previously crea-
ted solutions) (Bocconi et al., 2016). It is worth noting that the term CT may sometimes be
used indistinctly from Algorithmic Thinking. However, we view these two as different con-
cepts, in which CT is a powerful resource for problem solving, whereas algorithmic thin-
k ing is a specific skill to be used in CT (Syslo, 2014).

Third, programming has been referred to as the best method to teach CT. However, most
of the programming courses are initially taught by introducing the student to a coding lan-
guage in order to solve a problem, which will be carried out by a machine (Bennedsen &
Caspersen, 2007). Unfortunately, this approach presents several problems. On the one
hand, students that take a programming course are given tasks that require a set of abilities
for solving problems of exercises in class (CT skills). The problem is that teachers and pro-
fessors sometimes assume students have these abilities, although most of the time this is not
the case (Lahtinen, Mutka, & Jarvinen, 2005). On the other hand, students must learn a spe-
cific programming language to solve their in-class problems and exercises, leading to an
overwhelming number of syntaxes and semantics as a consequence of the diversity of
commands that are specific to each programming language (Jenkins, 2002). Both of these
tasks are high-level processes that might provoke high stress to students (Grandell, Pelto-
maki, Back, & Salaskoski, 2006), and can be compared to learning math concepts in an
unknown, foreign language.

On the other hand, numerous researchers in education have pointed out four key
competences that promote a substantial contribution to learners not only in specific fields
or disciplines, but in all occupations or walks of life (Ontario Ministry of Education., 2016).
Such competences comprehend Critical thinking, Collaboration, Communication and Cre-
ativity (P21 association, 2017), commonly known as the 4Cs. Concisely, critical thinking is
defined as a competence that allows one to critically evaluate information and arguments,
to solve complex problems, and to construct meaningful knowledge in order to subse-
quently put it into practice. Collaboration is the capacity to team up to work efficiently and
effectively, displaying team-related and interpersonal skills through the process. Communi-
cation is understood as the mastery of writing, speaking and digital skills for a wide range of audiences. Finally, creativity is explained as an individual being able to generate novel ideas and processes, as well as exhibiting the required skills to put them into practice (Fullan & Langworthy, 2014).

To ease the challenges students face while learning to program, several approaches such as video games, specialized software, LEGO Robots suites, and specialized websites such as Code.org have been developed (see more in Buitrago-Florez et al. (2017)). Nevertheless, these approaches confront the programming language learning issue but do not necessarily provide situations that foster the skills required to develop CT and resolve computational problems. Therefore, the aim of this study is to describe the findings obtained in regards of two research questions: i) How could active pedagogical strategies contribute the necessary components for CT curriculum design? And ii) How could an active pedagogical approach based on CT contribute to the fostering of critical thinking, collaboration, communication and creativity?

**Method**

**Theoretical Framework**

Socio-cultural approaches in education claim that learning is a complex set of activities, contexts, and socio-cultural factors involving the learner and their surroundings (Vygotsky, 1978). As such, it is plausible to state that the origins of human thinking may be linked to social and cultural practices (Hernández, Ravn, & Valero, 2015). This perspective of learning, as discussed by Radford (1997), proposes that knowledge is the product of the negotiation of meaning, framed by activities of individuals in a cultural context. In other words, learning occurs as a specific social process in which the learner improves progressively in cultural forms of thinking through the mastering of language, interactions, signs, and artifacts (Radford, 2008). This vision of learning could be easily seen when a professor complains about the inexperience or lack of knowledge of new graduate students. However, after a few years of working in the specific field, professors realize that students are transformed by learning in context (Wieman, 2007).

Embracing a socio-cultural perspective in education research indicates that researchers and practitioners must envision education processes as social activities developed in communities (Leach & Scott, 2003). Wenger (1998) therefore proposes that learning must not be considered as an isolated and individual process. Conversely, learning must be understood as a distribution and transformation among members of a community of practice. For Wenger (1998), the term *community of practice* derives from the notion of learning as active social participation, and lies in the idea that the process of learning occurs when people who share a common practice also share ideas, values, beliefs, languages and ways of doing over an extended period of time (Hernández et al., 2015).

From a socio-cultural perspective of education at the university level, students can get involved in authentic learning experiences through immersion in the routines, rituals and conventions of their professions. These processes allow the learners to acquire skills, language, values and knowledge associated with their specific practices (Sutherland, Scannlon, & Sperring, 2005), taking into account that the way in which members of specific professions interact and the way they use language, sign systems and tools is particular for each career (Northedge, 2002). Therefore, through this learning process the professional identity is also developed (Radford, 2008; Wenger, 1998). Consequently, the learning process can be seen as the learner moving from the outside of a practice community all
the way to its center, where each step transforms the practitioner into a member of this community as s/he learns and participates in each step taken along the way (Hernández et al., 2015).

Additionally, under the socio-cultural perspective of education, significant learning could be described as the process whereby the meaning of concepts related to a discipline becomes less and less dependent on the unique spatiotemporal context in which they are used (Wertsch, 1985). This means that in the transit of a person from the outside of a community of practice to its center, the actions and meanings of a person become less and less determined by the aspects from the context in which the actions or concepts were originally learned. Such a process is described and analyzed as recontextualization by van Oers (1998). Therefore, learning is not seen as a process of transfer itself, as described in other perspectives of education, but is described as the process in which students are able to recontextualize actions and concepts from an original situation to new scenarios. It is important to highlight that multiple tensions or contradictions could arise in students as they become participants in communities of practice, mainly driven by drastically changing their role from passive to active entities in the learning process. Such tensions and contradictions are a vital construct in socio-cultural perspectives as they trigger powerful moments of reflection in learners which favor their transit and learning inside the community.

Consequently, the development of CT can be described as the transit of a student into a community of practice. Such transit may be generalized in three different phases. First, students must be engaged in the general features of CT, offering an immersion in key skills needed in the field, as well as basics of language and concepts commonly used by initial processes of recontextualization. This would generate an initial identity as the students begin to participate and become peripheral participants of the community of practice by sharing artifacts and experiences related to CT. Once situated at the periphery, students could be involved in the basics of computer programming in a subsequent course in order to further develop skills related to CT as well as the basics of coding. Lastly, students may enroll in a robust course of computer programming in which participation and identity would be fully legitimated as the students progressively approach the center of the community.

Taking into account that the focus of this study lies in the transit from the first to the second phase, from being an outsider to moving to the periphery, we consider it important to clearly distinguish CT from computer programming. This distinction allows us to focus on CT skills and foster students’ involvement as peripheral participants in the CT community. As such, computational thinking can be seen as the active process of understanding and analyzing a problem, designing a solution to a given problem, and finally implementing the best possible solution for that problem by using the CT skills. In this journey, a person develops and uses skills related to CT without necessarily having to use a computer at all. Computer programming, on the other hand, is the process in which CT skills are used to solve problems via a computer. This process takes into account computational concepts such as variables, vectors, strings, arrays, hashes, and data structures, among several others, and of course a programming language to interact with the machine.

We wish to expand the differentiation between CT and computer programming by using a specific example of the former:

A friend has come up with a wonderful yet complex cooking recipe and s/he wants to share it with the world. To do so, s/he must write a step-by-step detailed set of instructions on how to prepare this meal. A CT process is the ideal way to help someone in this situation.
First, the person must remove all unnecessary details. In this case, it is not necessary to know where to buy the ingredients or the costs associated with the recipe (the recipe must be universal, not constrained by the availability of certain resources to a limited area). In order to eliminate these unnecessary details, abstraction is used to find all the details that should be avoided. Second, the instructions should be stated in as much detail as possible, avoiding common sense ideas. Algorithmic thinking may then be used to correctly indicate in detail the steps for cutting a piece of meat or how to marinate a specific piece of fish. Additional skills related to CT can be included in a third instance, where decomposition has to be used to get a successful recipe for a complex meal. To illustrate decomposition, let’s assume that the recipe contains three courses: an appetizer, a very complex entrée, and a dessert. One of the options available is to think of each course as parallel recipes that must be executed at the same time in order to get the meal served concurrently (nobody wants the main course to be cold, or wants to enjoy the dessert three hours after the meal is done). Computational Thinking also involves debugging the steps that are problematic and do not yield the expected results. In this case, going to another friend’s house and asking him/her to recreate the recipe exemplifies debugging. Under this situation, the original creator of the recipe would observe which steps are missing, which variables might change due to changes in the environment (e.g. a different oven that does not reach the required temperature), or which steps might be unclear. Finally, it turns out that your friend has more than one recipe available to share. Here, generalization will aid in quickly producing other recipes by using the information and the experience obtained through previous experiences.

In order to avoid confusion among readers, professors and researchers interested in this discussion, one must be aware that CT comprises additional thinking skills, concepts and processes, besides the ones mentioned in this document (Bocconi et al., 2016; Wing & Stanzione, 2016). Nevertheless, the use of a set of basic skills related to CT could be used as an immersion strategy for novice students by creating situations of recontextualization, described as follows.

Pilot Course Set-Up

Since this is a completely new pedagogical approximation in the development of skills related to Computational Thinking, it is important to execute a preliminary assessment and to report whether CT exercises are appropriate or inappropriate to accomplish for CT development (Teijlingen van, Rennie, Hundley, & Graham, 2001). Hence, to successfully test our novel approach, during the second semester of 2017 we developed a pilot course with 10 volunteer students from our institution. We made a call via e-mail looking for volunteers, from which we received 55 responses for participation. We subsequently formed a group of 10 students comprising five female and five male students ranging from 17 to 22 years old. Selected students belonged to the undergraduate programs of medicine (2), microbiology (1), business and administration (1), law (1), design (2), geology (1) and engineering (2). All the students were studying first, second or third semester of their studies. The only parameter used to select our participants was that none of the students should have taken courses related to computational programming as part of their coursework. Our institution is a renowned private university located in Bogotá, Colombia, with more than 70 undergraduate and postgraduate programs usually taught in Spanish. We decided to implement a pilot course to obtain valuable information towards a subsequent massive implementation.

The students were subsequently engaged in a 16-hour pilot course divided into three main stages. The first stage consisted of a two-hour lecture in which the professor explained
the concept of CT as well as the definition and characteristics of the five basic skills related to CT (abstraction, algorithmic thinking, decomposition, debugging and generalization). An in-depth explanation of CT and several examples were used in order to create awareness in the students about the differences between CT and computational programming.

Later in the pilot course, two Problem-Based Learning (PBL) exercises (each PBL exercise defining stages two and three of the course) were executed in each of the following stages to foster the five basic CT skills. We decided to use the PBL approach to develop skills related to CT due to the large set of benefits that this way of teaching and learning has shown in recent decades. As stated by Capon and Kuhn (2004), the PBL strategy displays a superior acquisition of new material, a superior recall of new material, and a superior integration of new material with existing knowledge in students. Hence, in PBL exercises students could use the problem case or scenario to enter the periphery of a CT community of practice by displaying recontextualization and conscious awareness of the situations in which each of the key five CT skills appears as critical aspects to solve a specific situation. Additionally, the use of PBL in this specific case would not only tackle the development of CT skills, but also teamwork, sharing of information, independent responsibility for learning, communication skills and respect for others (Wood, 2003).

Therefore, the second stage of the course was a two-moment PBL exercise designed with the objective of encouraging the students to use LEGO bricks in order to perform a controlled and discrete immersion in CT. To successfully carry out the first moment of the PBL, each student was equipped with a standard 720-piece LEGO bricks box and was instructed to build a structure of 10 to 15 bricks accompanied by a set of instructions (algorithm) so another student would be able to build the same structure by using the same materials later on. Three specific recommendations were given before starting this part of the exercise: i) the structure and the algorithm should be elaborated in 90 minutes; ii) there were no restrictions for the structure in shape, length, height or any other instruction besides the number of bricks; iii) to annotate each step of construction in the algorithm as it was elaborated. Once all the structures and algorithms were made, the students were told to dismantle the structures and then to share the algorithm and blocks used with a classmate. Hence, students were arranged in pairs to build the exchanged structure. Students were instructed to annotate and describe the steps in which they had issues while assembling the bricks in the algorithm provided by their partner. To conclude this first part of the activity, students were instructed to debug the algorithm they made given the recommendations of their working pair. This first step of the PBL exercise was executed in two sessions of two hours each.

The second part of the first PBL exercise consisted of instructing the students to modify their algorithms in 60 minutes so that another student would be able to build an identical structure using the same materials while blindfolded. To do this, the professor shared two recommendations: i) to decompose the query structure into simpler structures to be subsequently assembled; and ii) to abstract the information that could be useful for a blind person to build the structure. Once the students ended their modifications, all students were put into pairs to build the exchanged structure. Students were instructed to annotate and describe the steps in which they had issues while assembling the algorithm provided by their partner. To conclude this first part of the activity, students were instructed to debug the algorithm they made given the recommendations of their working pair. This first step of the PBL exercise was executed in two sessions of two hours each.
Finally, the third stage of the course was a PBL exercise that comprised the construction of a 7-step Rube-Goldberg Machine. These machines are deliberately complex contraptions in which a series of devices that perform simple tasks are lined up to produce a domino effect (Rankin, Gooch, & Gooch, 2008). Hence, students were randomly arranged into two groups of three students and one group of four students, and each group was equipped with one construction kit that consisted of strings, springs, wood blocks and sticks of different sizes, metal and wood spheres, pulleys, clay, glue, moving cars, motors, batteries, cables, switches and small light bulbs (for a detailed list of materials, see Appendix 1). Students were allowed to use the LEGO brick boxes used in the previous PBL exercise as well. Additionally, three recommendations were given to the groups beforehand: i) to explore all the materials given for the activity first; ii) to discuss, sketch and construct the final step of the Rube-Goldberg machine after material recognition; and iii) to discuss and sketch all subsequent steps of the machine prior to its construction. This PBL exercise was executed in three sessions of 2 hours each.

Each of the steps in the PBL exercises were carefully designed and implemented to create a conscious awareness of the characteristics, concepts and benefits of each of the CT skills in the students through different activities performed in iterative cycles. Table 1 describes a curricular alignment matrix for each PBL activity in function of the CT skills.

Table 1. Curricular alignment matrix for PBL exercises.

| PBL activity | Skill(s) in action |
|--------------|--------------------|
| PBL 1 Stage 1. Creating an algorithm to build a LEGO structure. | Abstraction and algorithmic thinking. |
| PBL 1 Stage 1. Testing of the steps of the algorithm given recommendation of assessment in pairs. | Abstraction, algorithmic thinking and debugging. |
| PBL 1 Stage 2. Modification of algorithm for a blindfolded student. | Abstraction, algorithmic thinking, decomposition and generalization. |
| PBL 1 Stage 2. Real-time debugging of algorithm. | Algorithmic thinking and debugging. |
| PBL 2. Visualization of non-common uses of materials provided. | Abstraction and generalization. |
| PBL 2. Designing and building each of the steps of the Rube-Goldberg machine. | Abstraction, algorithmic thinking and decomposition. |
| PBL 2. Testing of the steps of the Rube-Goldberg machine. | Debugging. |
| PBL 2. Assembling of Rube-Goldberg machine. | Abstraction, debugging and generalization. |

Data collection and analysis
A qualitative approach of data collection was used to determine the benefits of the methodology in the development of CT skills. A total of three different qualitative instruments were used to maximize the amount of data: A written reflection made by the students at the end of each of the PBL exercises, a diary written by the professor, and a focus group discussion conducted at the end of the pilot course. As such, reflections allow students to evaluate their performance and contextualize their actions to build significant meaning, while providing an enormous amount of qualitative data (Agouridas & Race, 2007). The professor’s field diary provides reflection and specific details about students’ interactions from another point of view since the professor perceives learning processes in a different way (Merriam
Finally, the focus group retrieves additional data by generating dynamic conversation between the participants via a logical sequence of questions (Lecanda & Garrido, 2003).

Consequently, written reflection 1 was designed to inquire into how students elaborate their conscious awareness in the use of CT skills in relation to PBL exercise 1. On the other hand, reflection 2 was designed to engage students in the understanding of the use of CT skills in PBL exercise 2, as well as how these skills could be extrapolated into everyday scenarios. Moreover, in field diary the professor recorded relevant information on how students developed their use of CT skills, how they communicated and approached problem-solving situations, and how they interacted with the materials and instructions provided. Furthermore, the focus group questions provided insights into how the students perceived the development of conscious awareness of the use of CT skills throughout the whole pilot course, as well as inquired about the perception of the scope and limitations they observed regarding the course design. The pilot course and the qualitative instruments were developed in Spanish, with a subsequent translation of relevant information to English. For the detailed set of questions included in the reflections and the focus group discussions, see Appendix 2. Additionally, students signed an informed consent in which they authorized the use of anonymized written, visual and audio data for academic purposes.

Data from all three qualitative sources was subsequently triangulated to discern if the qualitative methods would expose different meaningful issues regarding the pedagogical approach, as well as to validate qualitative data by comparing interpretations of results from different data-collection methods (Oliver-Hoyo & Allen, 2006). Information coming from each of three qualitative instruments was organized and coded into five different pre-established categories, which correspond to the five CT basic skills: abstraction, algorithmic thinking, decomposition, debugging and generalization. As described by Patton (1999), in this study we used the majority strategy for aggregating information while triangulating. This means that the analysis of the pre-established categories was derived from the direction from which the majority of the sources agreed by cross-validation. Additionally, some data from the field diary elucidate four emergent categories highlighting a series of aspects considered important from the pilot course: learning accompaniment and reflection processes, teamwork, communication and creativity. Information regarding these categories was subsequently retrieved in depth from the focus group.

Results

Abstraction

Triangulation of the information evidenced a progression in the conscious awareness and performance regarding the skill of abstraction. At the beginning, students struggled with the concept of abstraction since it is not very common that professors or students deal with abstraction in a direct way. Curiously, the professor’s diary showed that most of the students thought that abstraction was restricted to the field of art. Nonetheless, data showed that students recognized quickly that abstraction is a skill used almost every day in an unconscious manner. Likewise, students were aware that abstraction was a key skill in order to develop the PBL exercises proposed in the pilot course in several actions in order to determine the nature of a problem, to propose alternative solutions, as well as to extrapolate and adapt concepts and ideas into new scenarios. Some perceptions from reflections and focus group regarding the first PBL exercise were:
St1: “Abstraction was important to identify the characteristics of the LEGO pieces (shape, color, size) to make the first structure”.

St4: “I used abstraction to imagine what figure I could make given the time and the pieces”.

St7: “I think abstraction was important to determine the characteristics of the LEGO bricks that would be useful to create the instructions to assemble the structure while blindfolded”.

St10: “Abstraction was very useful to imagine, select or discard pieces while I was making my classmate’s structure blindfolded”.

In the same way, students were able to recognize abstraction as a pillar in the construction of the Rube-Goldberg machine. Some perceptions from the focus group discussions were:

St8: “Abstraction, I think, was present all the time (while making the machine) since we had to imagine what to do with all the materials we had available in order to accomplish the objective”.

St2: “I think that, for example, when my group decided to use a spring as channel for a sphere, abstraction was the skill that enabled us to do so. Then, we abstract multiple times to imagine other, less common uses for all the pieces that we had available for building the machine”.

Algorithmic Thinking
The data indicates that algorithmic thinking was the skill best recognized by the students throughout the development of the PBL exercises. Students were very familiar with the concept given that the development of algorithms is the basis of several mathematics courses in schools. Thus, students recognize the times in which algorithmic thinking was useful to approach the PBL situations relatively easily. Nevertheless, in several cases it was noticeable that students struggled when not enough specific instructions were provided. In other words, when the students wrote their own instructions, they realized that specific steps were ignored because they were thought to be logical somehow. However, their classmates did not necessarily consider those steps ‘logical’. Our results show that students became more aware of the importance of avoiding common sense at the time of algorithm design, which is ultimately critical for computational programming and CT regarding the way in which ideas are transmitted to people or machines. Some perceptions from the focus group supporting these ideas are:

St6: “I think there were two phases of algorithmic thinking in the first PBL exercise, when you are building the structure and when you are writing the steps. Hence, it was a key skill to be able to transmit the information to the classmates”.

St5: “While building the machine it was very interesting since we had to imagine the algorithm in our heads first”.

St9: “Since we had several sessions to assemble the machine, we had to use a lot of algorithmic thinking to remember the steps we made in the previous sessions, otherwise, we had to imagine everything again”.

St1: “I think something very important was that at the end of the LEGO exercise, the professor was very insistent on the level of detail we had to include in the instructions. So when we worked on the machine, all the members revised measurements, shapes, colors and other details all the time”.
Decomposition
Decomposition is a very useful skill that allows the division of complex problems into a group of well-described subset of problems to be solved independently. The ability resides in being able to determine the specific point of division of the main problem, so the independent solutions increase the efficiency and efficacy of the problem-solving approach. Data shows that students were not familiar with the concept or characteristics of decomposition at the beginning of the course. Nevertheless, students quickly realized how decomposition could be a powerful resource to facilitate the process of making algorithms and communicate sets of ideas to their peers. Moreover, cross-validation of data indicates that students were able to extrapolate the benefits of decomposition to another situations. For example, in the focus group some students recalled:

St10: “Decomposition was useful to fragment my LEGO model into small steps, making it easier to write the algorithm later on.”

St2: “I think that decomposition was present since we had to make a step-by-step instruction for assembling the LEGO structure while blindfolded.”

St6: “I think decomposition was very useful in the machine exercise, since we actually decomposed some steps in order to make it easier to reassemble in subsequent sessions.”

St9: “This skill would be extremely helpful to divide, for example, a monograph essay, since one could propose sub-topics to approach in an easy way.”

Debugging
Debugging is a critical skill in CT since it provides strategies to find and eliminate errors in a problem-solving approach, allowing students to be responsible for a high-quality product at the end of the PBL exercises. Data demonstrated that students were not particularly aware of using this skill very often in their classes or in their daily lives, making their learning experiences successful or unsuccessful without the opportunity of correcting or improving their mistakes in real or asynchronous time. Triangulation of the information allowed us to state that being able to correctly debug situations related to the PBL activities developed a deep sense of learning awareness in students. As recounted by some students in the written reflections:

St8: “I found it very interesting when I got feedback from my classmate in the first part of the first exercise. I realized that several key information points were missing in the instructions.”

St2: “I could not believe how many steps I did not write in the algorithm. The information I did not include initially, I had found logical. However, this was not the case for my classmate once he tried to build the structure.”

St6: “I think debugging was very explicit when I had to change some steps in real time so my classmate could make my structure while blindfolded.”

St7: “Debugging was very present in the first exercise because we had to make corrections in that specific moment. Once I realized that some instructions were missing or unclear, I had to debug fast.”

From the professor’s perspective, students improved their ability to debug in the construction of the Rube-Goldberg machine given that the exercise *per se* is an iterative process of
real-time debugging, which enhanced greatly the conscious awareness perspective of the students with regards to this skill. Some perceptions from the focus group discussion regarding this idea were:

St10: “I think we debugged each time we tested the machine as a set of steps”.

St2: “It was very interesting to be able to correct the steps in real time and watch how they worked later. This is not an activity one does frequently in class or any other everyday situations”.

St8: “Every time we were testing the functioning of the machine, we had to see whether things actually hit or moved other things all the time”.

**Generalization**

Generalization implies the ability to modify and adapt solutions previously established to a new specific problem. The data indicates that students did not realize how generalization could be used in PBL situations at the beginning of the course, which can be seen as the product of isolated activities of traditional school education. However, after being in contact with the LEGO bricks, students started to generalize different concepts. First, they recognized the material, systematized their brainstorming activity, and used less time to develop instructions. Additionally, it was clear that students generalized multiple times when they sketched the steps of the Rube-Goldberg machine. This led to an increase in the efficiency and efficacy of their actions. Likewise, from the professor’s perspective, students developed a conscious awareness of the benefits of generalizing situations to reduce time and resource consumption. Students described their experiences as:

St7: “I think generalization was quite present because, for instance, in the LEGO structure I constructed, which looked like a tower, the instructions became a repetition of steps, that seemed like a loop”.

St1: “Well I consider that once we were in the blindfolded activity, generalization was very useful to perceive, as time passed, the instructions our classmate actually understood”.

St7: “(in the creation of the Rube-Goldberg machine) we designed some walls using LEGO bricks to generate a chain reaction, these walls were like generalized loops”.

St3: “In my case I struggled to understand generalization, but when my group used the moving car, we realized the car’s displacement was always the same after trying several times. Actually, I remember the displacement was 1.40 meters”.

St5: “We learned to use one of the springs as a channel from a previous (failed) attempt”.

**Learning Accompaniment and Reflection Processes**

Students claimed that two additional actions helped them increase their awareness and performance of CT skills. Students appreciated that the professor was always reminding them in which specific moments to use CT skills as they solved the PBL exercises. Moreover, students found it very useful to write reflections about their processes, which allowed them to realize what actions they carried out, what concepts they learned, and what they needed to improve in the future. Some perceptions regarding this category were:

St9: “I think that the reflection activity was extremely helpful since I was able to link all the things I had done in the exercises with concepts we were reviewing in class”.

St10: “I think it is valuable since it is like an auto-evaluation. It allowed me to reflect if I had accomplished the objectives of the course. For example, in the first exercise I struggled a lot since a decided to build a very complex structure. However, in the second exercise I was focused on accomplishing the steps to build the machine. I realized that in the second exercise I was able to see what the problem was, and thus was able to solve it”.

St5: “I think all students really appreciated the accompaniment, as we developed the exercises, the professor was observing frequently and making clear the exact moment in which we used a specific skill”.

St4: “In fact, I think that the professor’s guidance was one of the most powerful strategies in this course”.

Teamwork
Students recognized that teamwork was enhanced in the second PBL exercise because it was extremely complex to arrive at an agreement in several actions. Nonetheless, as time passed and the professor helped solve issues, students managed to work as a team in an appropriate way. Some perceptions regarding teamwork were:

St1: “I think I improved my teamwork skills since we had to agree on the steps for the machine. Sometimes things did not work as we planned, but after our group discussions we eventually came up with a solution. For instance, we had several issues handling the pulley and the motion car, nonetheless we debugged multiple times with the help of the professor until we were able to come up with a solution”.

St7: “Let’s say that my group was a little bit individualist at the beginning, but as we worked on the machine, all members of my group started brainstorming, discussing and working as a team. Actually, I am a very closed person in terms of teamwork, nevertheless, this time I realized that I needed my classmates to successfully accomplish the objective”.

St5: “There was something very remarkable in my group. At the beginning, one team member did not fit very well, but as time passed that person overcame the issues and was very happy and enthusiastic participating in the exercise. I think that over time I understood the person much better and we were able to interact easily”.

St2: “In fact, the simple action of brainstorming enhances teamwork, that was a key aspect for accomplishing the exercise in the time provided with the guidance of the professor”.

Communication
Students were very emphatic about the development of communication competences throughout the pilot course. They realized that interdisciplinary communication could be challenging, as is communicating ideas in a specific way, avoiding common sense actions. They agreed that this pedagogical approach helps them enhance oral communication skills. Some perceptions regarding communication were:

St7: “I think that in the making of the algorithms we had to exercise and improve our communication. One thinks that by expressing actions as we normally do, other people will understand, but that is not the case. As I was saying before, in the case of the blindfolded classmate, it was not possible to say put this piece above the red one, we had to be very specific and avoid a lot the common sense”.

St6: “I learned that not everyone has the same background. That other people do not think as I do. Then, one must give instructions and stop assuming others are ignorant, one must not suppose anything. Once I realized this it was a breakthrough for me”.
St9: “I think we naturally assume that everyone has our same background. In the university I have noticed that students outside of my discipline cannot understand me when I talk about specific topics in my field of study. Then, this exercises showed me that I need to be very specific sometimes in order to communicate effectively”.

St1: “After each activity I realized that we are not detail-oriented and we assume a lot of things when we try to communicate with others. I found it very challenging to communicate in an interdisciplinary group at the beginning. I had to try several strategies to communicate what I imagined in the exercises”.

Creativity
Students commented that creativity was required throughout the course in order to solve problems in different ways. To imagine non-common uses of the different materials provided to achieve objectives in the PBL exercises allowed them to be creative in several moments. Perceptions regarding creativity were:

St10: “I think creativity was present all the time. At the start of the exercise, we had a lot of extremely diverse materials allowing me to imagine a lot of ideas for each single step of the machine”.

St8: “I think creativity was present for instance in the moment we needed a ball to fall in a specific point. We looked at all the materials and realized that we could use them in different ways to solve the problem. We worked all the time finding new and innovative solutions”.

St6: “It is like thinking outside of the box. It is like thinking about all possible functions a material could have. Besides the creativity we displayed in the machine, I think I had to be extremely creative in designing the algorithm for the blindfolded classmate. For instance I struggled a lot, but in the end I decided to use coordinates to guide the assembly of the structure. When I reflected about the process, I realized that the use of coordinates was very resourceful and efficient”.

Discussion
Triangulation of the information shows that throughout the development of the pilot course, students became familiar with CT skills as a problem-solving approach. As stated by Vihavainen, Airaksinen, and Watson (2014), the set of basic skills related to CT allows any person to approach a problem, analyze it, and propose a solution. From the perspective of the students, being aware of the definitions, concepts, and uses related to CT skills allowed them to effectively solve the PBL exercises provided. Hence, the development of skills through our immersion strategy to CT offers a scenario to develop and reinforce CT skills before any type of computer programming is involved in the process of learning.

Overall, the use of active pedagogical strategies centered in the learning process of the students showed development, enhancement and awareness of CT skills in participants of this study. Results from this pedagogical process can be interpreted by two specific sociocultural learning concepts. First, students were able to share experiences, concepts, situations, expressions and practices in order to solve a set of problems given in the course, allowing them to participate actively in the process. Second, students were able to transform the concepts related to the five CT skills into artifacts at the time they were developing solutions for the PBL exercises. The process of transforming abstract information into real artifacts, called reification by Wenger (1998), allowed students who are beginners to understand, clarify and raise awareness about what they were learning. Therefore, as proposed by Wenger (1998), when participation and reification processes interact a negotiation of mea-
ning environment is generated, an environment in which students were able to acquire new knowledge and skills (Radford, 2008).

Data supporting both socio-cultural concepts allow us to interpret that students who participated in the pilot course successfully developed an identity as peripheral members of a CT community of practice. Students were able to create an identity as novices in CT via two main processes. On the one hand, students were able to interact with a CT expert under a community of learning in which basic concepts, signs, and language forms from the CT field were shared among members. On the other hand, students got involved in active pedagogical strategies by engaging in PBL activities in which concepts and definitions explained at the beginning of the course were recontextualized. This process of recontextualization enables students to create a truly conscious awareness towards a reflective use of the five CT skills, at the same time that a new perspective around becoming part of the CT community was crafted.

As CT can be seen as a large community of practice in which expert programmers lie at the center of the community, novices should be involved in basic routines to generate a first step to enter as a peripheral participant. The pedagogical approach described in this study shows promising data about how peripheral participation can be developed by deploying a strategy focused on fostering a set of five CT skills. Thus, once a good level of CT skills has been obtained, professors could guide the path throughout the CT community by using videogames, LEGO Robots, or specialized software and platforms to immerse students into computer programming, as well as the use of advanced concepts such as variables, strings, arrays and hashes, to increase critical abilities related to CT (Buitrago-Florez et al., 2017). This would allow students to travel to the center of the community of practice by getting involved in formal computer programing via programming languages, and aiming to develop high-level CT skills such as sequencing flow and control, automation, efficiency and effectiveness constrains, and iterative and recursive thinking.

Results also revealed how reflection processes can act as a twofold action in the process of the creation of meaning. Students were able to recognize and internalize successful practices and experiences, reinforcing concepts and skills related to CT. Besides, students were able to identify unsuccessful actions, allowing them to obtain insights into what may have done wrong in order to avoid now-known pitfalls (Agouridas & Race, 2007). Consequently, by using appropriate guidance, reflective processes can turn into effective forms of participation and reification, because students abstract their role and performance in the learning and use of CT skills and concepts, transforming their thoughts into written words that could be seen as artifacts.

Furthermore, the students recognized the accompaniment of the professor as a critical step in creating an active learning atmosphere. As proposed by Northedge (2002), in order to involve the students in the use of specialized knowledge, artifacts and language, the professor must assume the role of a representative member of a specialized community who generates strategies to involve students in specialized actions. In other words, the professor must provide guidance along the process of learning by becoming a bridge that facilitates the transit of the students from a colloquial to a specialized way of communication.

Moreover, results showed that CT framed in a socio-cultural vision of education, as proposed by Vygotsky (1978), allowed students to participate actively under PBL situations and take part in socially interactive experiences. Thus, emerging categories indicate that students were able to enhance and reflect over 21st century competences such as teamwork, communication and creativity. The development of these competences have been discussed as major objectives in higher education, aiming to respond to the demands of complex
phenomena such as globalization, technological advances, the development of new specializations, the need for interdisciplinary work, among several others (Becker, 2006; Gallo-
way, 2007). Therefore, our pedagogical approach could not only be used to involve students in CT as a problem-solving approach, but also to create a scenario for competence development.

Conclusions

We can summarize the findings of our efforts in three main points. First, our active pedagogical strategies based on PBL contributes broadly to the establishment of a CT curriculum course, which could be used as a twofold learning approach by offering a first immers-
sion for freshmen students in CT, as well as a setting for the fostering of 21st century competencies by offering a first step to becoming a peripheral participant. Second, students and professors must be aware of the definitions and situations in which each of the CT skills are used. This will allow the students to reflect on their practices and their performance, becom-
ing aware of a powerful tool for CT development. Third, teachers and professors can increase the use and difficulty of each of the suggested skills related to CT by changing the instructions or scenarios of the exercises. Different problem-solving situations will lead to different uses, iterations, and development of CT skills. Thus, professors are encouraged to experiment, randomize the scenarios, and reflect on the products obtained to develop different situations for teaching CT through programming.

The findings described and discussed herein represent a methodological framework that provides new mechanisms not only to develop CT skills in novice programmers, but also to foster four competences widely described as key elements for 21st century professionals. We are fully aware that these ideas must be tested, experimented, and further evaluated. Comparative research of how students involved in the approach documented in this study perform in subsequently programming courses will generate important insights into how this approach equips novices with tools to transit from peripheral actions to formal computer programming. Nonetheless, we believe that this novel strategy to approaching to CT could aid in the development of computational skills and key competencies for a wide range of educational programs, and expand into a new line of research focused in education, formalization, and expansion of techniques for CT development.

Acknowledgements: We would like to thank to the National Foundation of Science, Technology and Innovation of Colombia (COLCIENCIAS) for providing the resources for developing this project. Additionally, we would like to thank to the volunteer students who made part of the computational thinking pilot for their active participation and valuable data and feedback.

References

Agouridas, V., & Race, P. (2007). Enhancing Knowledge Management in Design Education Through Systematic Reflection Practice. Concurrent Engineering: Research and Applications, 15, 63–76. https://doi.org/10.1177/1063293X07076267

Becker, F. (2006). Globalization, curricula reform and the consequences for engineers working in an international company. European Journal of Engineering Education, 31, 261–272. https://doi.org/10.1080/03043790600644749

Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. SIGCSE Bulletin, 39(2), 32–36. https://doi.org/10.1145/1272848.1272879
Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing computational thinking in compulsory education – Implications for policy and practice. EUR 28295 EN. https://doi.org/10.2791/792158

Buitrago-Florez, F., Casallas, R., Hernandez, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a Generation’s Way of Thinking: Teaching Computational Thinking Through Programming. Review of Educational Research, 87(4), 834–860. https://doi.org/10.3102/0034654317710096

Capon, N., & Kuhn, D. (2004). What’s so good about Problem Based Learning? Cognition and instruction, 22, 61–79. https://doi.org/10.1207/s1532690Xci2201_3

Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). Developing computational thinking in the classroom: a framework. Computing at School (CAS) Releases. Retrieved from https://eprints.soton.ac.uk/369594/

French Academy of Sciences (FAS). (2013). Teaching computer science in France, tomorrow can't wait. Institut De France – Académie des Sciences. Retrieved from http://www.academie-sciences.fr/pdf/rapport/rads_0513gb.pdf

Fullan, M., & Langworthy, M. (2014). A rich seam: How new pedagogies find deep learning: London: Pearson.

Galloway, P. (2007). The 21st-Century Engineer: A Proposal for Engineering Education Reform. Reston, Virginia: ASCE Press.

Grandell, L., Peltoniäki, M., Back, R.-J., & Salakoski, T. (2006). Why Complicate Things? Introducing Programming in High School Using Python. Proceedings of the 8th Australasian Conference on Computing Education – Volume 52, 71–80.

Hernandez, C., Ravn, O., & Valero, P. (2015). The Aalborg University PO-PBL Model from a Socio-cultural Learning Perspective. Journal of Problem Based Learning in Higher Education, 3(2), 16–36. https://doi.org/10.5278/OJS.JPBLHE.V0I0.1206

Jenkins, T. (2002). On the difficulty of learning to program. Proceedings of 3rd Annual LTTSN ICS Conference, 53–58. Retrieved from http://www.psy.gla.ac.uk/~steve/localed/jenkins.html

Leach, J., & Scott, P. (2003). Individual and Sociocultural Views of Learning in Science Education. Science & Education, 12, 91–113. https://doi.org/10.1023/A:102266551

Lecanda, R., & Garrido, C. (2003). Introducción a la metodología de investigación cualitativa. Revisa te psicodidáctica, 14, 5–40. Retrieved from http://www.redalyc.org/articulo.oa?id=17501402

Merriam, S. B., & Tisdell, E. J. (2015). Qualitative Research: A Guide to Design and Implementation: Wiley. Northedge, A. (2002). Organizing excursions into specialist discourse communities: A sociocultural account of university teaching:. Wiley Online Library.

Ontario Ministry of Education. (2016). 21st century competencies: Foundation document for discussion, phase 1: Towards defining 21st century competencies for Ontario. Ministry of Education of Canada. P21 association. (2017). Framework for 21st century learning. Partnership for 21st competencies development.

Patton, M. (1999). Enhancing the quality and credibility of qualitative analysis. Health Ser Res, 34, 1189–1208.

Qualls, J., & Sherrel, B. (2010). Why computational thinking should be integrated into the curriculum. Journal of computer science colleges, 25, 66–71.

Radford, L. (1997). On Psychology, Historical Epistemology, and the teaching of Mathematics: Towards a socio-cultural History of Mathematics. International journal of Mathematics Education, 17, 26–33. Retrieved from http://www.jstor.org/stable/40248219

Radford, L. (2008). The ethics of being and knowing: towards a cultural theory of learning: Sense Publishers.

Rankin, Y., Gooch, A., & Gooch, B. (2008). The impact of game design on students' interest in CS. GDCSE conference 08, 59–63. https://doi.org/10.1145/1463673.1463680
Sutherland, L., Scanlon, L., & Sperring, A. (2005). New directions in preparing professionals: examining issues in engaging students in communities of practice through a school–university partnership. *Teaching and Teacher Education, 21*, 79–92. [https://doi.org/10.1016/j.tate.2004.11.007](https://doi.org/10.1016/j.tate.2004.11.007)

Soysło, M. M. (2015). From Algorithmic to Computational Thinking: On the Way for Computing for all Students. *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education – ITiCSE ’15*, 1–1. [https://doi.org/10.1145/2729094.2742582](https://doi.org/10.1145/2729094.2742582)

van Teijlingen, E. R., Rennie, A.-M., Hundley, V., & Graham, W. (2001). The importance of conducting and reporting pilot studies: The example of the Scottish Births Survey. *Journal of Advanced Nursing, 34*(3), 289–295. [https://doi.org/10.1046/j.1365-2648.2001.01757.x](https://doi.org/10.1046/j.1365-2648.2001.01757.x)

van Oers, B. (1998). The Fallacy of Detextualization. *Mind, Culture and Activity, 5*(2), 135–142. [https://doi.org/10.1207/s15327884mca0502_7](https://doi.org/10.1207/s15327884mca0502_7)

Vihavainen, A., Airaksinen, J., & Watson, C. (2014). A systematic review of approaches for teaching introductory programming and their influence on success. *Proceedings of the tenth annual conference on international computing education research*, 19–26. [https://doi.org/10.1145/2632320.2632349](https://doi.org/10.1145/2632320.2632349)

Vygotsky, L. (1978). *Mind in society: Cambridge, Ma.: Harvard University Press.*

Wenger, E. (1998). *Communities of practice: Learning meaning and identity.* Cambridge university press.

Wertsch, J. (1985). *Vygotsky and the social formation of mind.* Cambridge, MA: Harvard University Press.

Wieman, C. (2007). Why Not Try a Scientific Approach to Science Education? *Change: The Magazine of Higher Learning, 39*(5), 9–15. [https://doi.org/10.3200/CHNG.39.5.9-15](https://doi.org/10.3200/CHNG.39.5.9-15)

Wing, J. (2006). Computational thinking. *Commun. ACM, 49*(3), 33–35. [https://doi.org/10.1145/1118178.1118215](https://doi.org/10.1145/1118178.1118215)

Wing, J., & Stanzione, D. (2016). Center for computational thinking. *Communications of the ACM, 59*, 10–11. [https://doi.org/10.1145/2933410](https://doi.org/10.1145/2933410)

Wood, D. F. (2003). ABC of learning and teaching in medicine: Problem based learning. *BMJ, 326*(7384), 328–330. [https://doi.org/10.1136/bmj.326.7384.328](https://doi.org/10.1136/bmj.326.7384.328)

**Appendix**

**Appendix 1.** List of materials for the development of PBL exercises. The list represents a construction kit that each of the teams had as equipment to work.

- 1 Lego brick standard box with 720 pieces
- 1 blue container of 120 L for all materials
- 1 set of 10 weights ranging from 10 g to 100 g
- 1 single pulley
- 1 double pulley
- 1 triple pulley
- 2 packs of clay
- 6 wooden sticks
- 1 set of 5 springs (5, 10, 15, 20, 30 cm)
- 1 magic spring
- 1 metallic 5 cm sphere
- 1 wooden 10 cm sphere
- 1 scissors
- 1 scalpel
- 3 markers
- 1 ruler
- 10 standard marbles
- 2 Motion cars with auto propulsion
Appendix 2. Guide questions for reflections 1, 2, and focus group for the pilot program.

Reflection 1
1. Describe how you experienced the Lego activity.
2. Describe the difficulties you encountered in solving the activity.
3. In which moments do you think the 5 skills related to CT were useful in solving the activity (Abstraction, algorithmic thinking, decomposition, debugging and generalization)?

Reflection 2
1. Describe what you understand by Computational Thinking.
2. Describe three differences between CT and computational programming.
3. Describe the five skills related to CT and explain how you used them to solve the Rube-Goldberg machine activity.
4. Given the five CT skills, provide an example in which each one could be used to approach an everyday situation.

Focus group
1. Do you think that the theoretical explanation that was made at the beginning of the course is clear?
2. Do you consider that the theoretical and practical difference between CT and computer programming is clear?
3. Do you consider that the exercise performed with LEGO bricks clearly addresses the 5 basic skills of computational thinking? (Abstraction, algorithmic thinking, decomposition, debugging, and generalization). Why?
4. Do you consider that the exercise carried out with Rube-Goldberg machines clearly addresses the 5 basic skills of computational thinking? (Abstraction, algorithmic thinking, decomposition, debugging, and generalization). Why?
5. Do you think the reflection process was useful in this course?
6. Do you think that this course helps the development or increases your ability to solve problems? What about Teamwork, Creativity and Communication?
7. Please specify what you think was good or innovative in this course.
8. Please specify something that you think could be improved in this course.
9. If this course is completed, it would last 16 weeks, each week with 2 classes of 1.5 hours. Do you consider that it could have a positive effect on the development of skills that could be useful in the academic and/or personal field for students of the Universidad de los Andes? Would you take the course? Would you recommend it to your friends/classmates?