IEEE TRANSACTIONS ON
EMERGING TOPICS IN COMPUTING

Received 7 January 2021; revised 16 March 2022; accepted 25 March 2022.
Date of publication 5 April 2022; date of current version 6 March 2023.

Digital Object Identifier 10.1109/TETC.2022.3163650

Engaging Primary and Secondary School Students in Computer Science Through Computational Thinking Training

RAFAEL HERRERO-ÁLVAREZ, GARA MIRANDA, COROMOTO LEÓN, AND EDUARDO SEGREDO, (Member, IEEE)

The authors are with the Departamento de Ingeniería Informática y de Sistemas of Universidad de La Laguna, 38200 San Cristóbal de La Laguna, Spain
CORRESPONDING AUTHOR: RAFAEL HERRERO-ÁLVAREZ (rherrero@ull.edu.es.)

This work was supported by the project Piens@ Computacion@ULLmente under Grant REF 21120050. Programa educativo para el fomento del pensamiento computacional a través de la realización de actividades que permitan su desarrollo y su inclusión en el currículo. Cabildo de Tenerife. Fundación General de la Universidad de La Laguna.

ABSTRACT Although Computer Science has grown to become one of the most highly demanded professional careers, every year, only a small percentage of students choose a degree directly related to Computer Science. Perhaps the problem lies in the lack of information that society has about Computer Science itself, and particularly about the work computer scientists do. No one doubts the role of Mathematics or Languages as core subjects in every primary and secondary education syllabus; however, Computer Science plays a negligible role in most current syllabuses. Only in a few countries have governments paid special attention to content related to Computer Science and to learning to analyze and solve problems the way computer scientists do (Computational Thinking). In this article, we present *Piens@ Computacion@ULLmente*, a project that provides a methodology to promote Computer Science through Computational Thinking activities among primary and secondary education students. The results obtained from an exhaustive statistical analysis of the data we collected demonstrate that the perception of Computer Science that pre-university students have can be improved through specific training. Moreover, we can also confirm that the performance of pre-university students involving Computational Thinking skills is independent of gender, particularly at the primary education level.

INDEX TERMS Computer science, computational thinking, primary education, secondary education, syllabus

I. INTRODUCTION

Computational Thinking refers to the skills involved in analysis and problem solving through the application of Computer Science concepts. Although the first mention of the term Computational Thinking appeared in 2006 [1], previous papers introduced a procedural way of thinking, especially applied in geometry problems [2], [3]. In spite of the above, there is still no consensus for a formal definition of Computational Thinking. Some authors propose that it is the persistence for working with complicated problems or the ability to handle ambiguity [4]. Others go far beyond computers, claiming that it involves three areas, namely the concepts of computing as used by programmers (sequences, loops, events, etc.), practices that are developed with programming skills (problem solving, reusing and combining different projects, etc.), and perspectives on the world around them (self-expression, engaging with others, questioning ideas, etc.) [5]. The idea that Computational Thinking is not thinking like a computer is also emphasized, since computers do not *think* as such; rather, the correct definition should be to think like computer scientists, since it is their problem-solving skills that constitute Computational Thinking [6]. Computational Thinking is not a mere support tool; it plays an important role in the way we understand the world and the problems around us [7]. Moreover, Computer Science is a key element for offering solutions to problems in many disciplines. Because of this, training in this area will be essential if future generations are to reason computationally, improve their problem-solving skills and apply these abilities to transform the world around them [8].

Since Computational Thinking is a cross-disciplinary ability that allows problem solving, designing systems and understanding human behavior using fundamental concepts in Computer Science, some authors believe in the possibility of integrating Computational Thinking into cross-curricular practices [4], [9]. A report from the European Union [10] shows that, although some efforts have been made to integrate Computer Science content into the pre-university educational curricula of different countries, some do not contain specific content related to Computational Thinking. In particular, only ten out of twenty one countries include *Computing and Coding skills*. Some countries, such as the United Kingdom, have made efforts to include a subject in the educational curriculum, referred to as *Computing*, that takes into account the importance of Computational Thinking, and does not just deal with programming [11]. Others, like Austria, have also analyzed the need to replace the subject called Information and Communication Technology (ICT), as well as Informatics, with one that encompasses new content related to Computer Science, which is called *Basic Digital Education* [12]. Switzerland has developed a national curriculum called *Lehrplan 21*, which requires primary and secondary school students to receive education in Computer Science, and created a mandatory pre-service teacher Computer Science Education course in order to properly implement said curriculum [13].

In the case of Spain, there is no official document that deals with Computational Thinking at the national level [14]; instead, each autonomous region decides what content should be taught in its educational curriculum. An analysis involving 12 autonomous regions concluded that most of them do not mention Computational Thinking as an ability that every citizen should have. All of these autonomous regions reflect content related to programming and robotics in their curricula, but almost all of them do so at the secondary education level, while only three of them (Comunidad Foral de Navarra, Comunidad de Madrid and Cataluña) do it at the primary education level [15].

Other countries outside Europe, such as the United States, do not have a nationally-defined curriculum, and therefore each state has its own laws in this regard. For example, 34 out of the 50 states have K-12 Computer Science standards. Moreover, in 38 states, teachers are required to have a certification in Computer Science to teach this subject. Finally, 19 states have a program to train teachers in this field [16]. Furthermore, a wide range of public and private initiatives related to Computational Thinking, such as CODE.org [17] and the one proposed by [18], among others, have been proposed by different entities in this country. Another country that has made great strides is Australia, which has proposed a clear program focused on algorithmic thinking, Computational Thinking and programming [19].

Many experts have supported the idea of introducing Computational Thinking in pre-university studies as a way of improving students' notion of Computer Science [1], [20]. At this point, we should note that some studies have shown that not only are young people unaware of what Computer Science really is, but that this misconception has a direct impact on their interest in Computer Science, and consequently, on their affinity for this specific academic field [21]. Some experiments have confirmed that giving students a simple introductory course to Computer Science, lasting only six weeks, could change their opinion of it [22]. Bearing the above in mind, students could have an idea of what Computer Science entails as a professional career.

Since the aforementioned misconception could considerably reduce the interest in this academic field [21], in this paper we present *Piens@ Computacion@ULLmente* [23], which is a project designed to disseminate and promote Computer Science through the development of Computational Thinking skills.

The main aim of *Piens@ Computacion@ULLmente* is to provide a methodology to introduce concepts related to Computational Thinking, and therefore to Computer Science. The Computational Thinking training phase of said methodology consists of a set of both plugged and unplugged Computational Thinking activities, which have been designed and scheduled in five sessions lasting four hours each, involving primary (8-9 years old) and secondary education (12-13 years old) students.

The Cabildo de Tenerife (the island council) is the institution in charge of offering this innovative educational project to schools, which have signed up for this voluntarily, and where the different teachers have taught hours in their fields over the course of the sessions.

The main hypothesis that we posit in this paper is that one of the reasons why young people are not interested in Computer Science is because they do not know exactly what it is. As a result, we would like to test whether providing specific training on Computational Thinking, which is directly related to Computer Science, influences their interest or not. In order to measure the impact that the aforementioned training has on students' interest in Computer Science, a questionnaire is used to gauge their opinion. At the same time, the students' improved Computational Thinking abilities are measured by means of a test specifically designed by Román-González *et al.* [24], [25] for this purpose.

The rest of this paper is organized as follows. A further description of the hypotheses and research goals is given in Section 2. The complete methodology involved in the design and development of the *Piens@ Computacion@ULLmente* project is described in Section 3. Afterwards, Section 4 presents and discusses the results. Finally, Section 5 is devoted to the conclusions and areas of additional research.

II. RESEARCH GOALS

The main hypothesis considered in this paper is that the poor interest in Computer Science shown by young people is mainly due to their misconception about this particular field. In the case of the local University, in the 2019-2020 academic year, only 14.9% of the total number of new students (4,438) enrolled in engineering degrees [26]. Only 159 out of those new 4,438 students (3.6%) enrolled in the Computer Science Degree. We would also like to increase girls' interest in Computer Science since the number of girls enrolled in engineering degrees is low [27], with the stereotypes they have about the people who work in this field being one of the reasons [28].

Bearing the above in mind, Computational Thinking training would also allow girls to become much more interested in Computer Science [29]. At this point, we should note that, due to the low participation of women in engineering degrees, the questionnaires and activities designed were analyzed from a gender perspective. We would like to prove that no gender differences exist when conducting training on Computational Thinking. A consequence of the above would be that gender differences are not due to greater capacities, but to other external factors, such as the little support they receive at home from their parents to study a scientific career, when compared to that received by the boys [30]. Also, recent studies have shown that girls tend to align with stereotypes related to subjects of a more verbal nature, while boys excel in Mathematics and Science. This difference occurs mainly in adolescence [31], [32]. Some studies focused on gender differences in Computer Science have shown that these differences occur before college, mainly in high school, and that having previously taken CS-focused courses is an important aspect when making a decision about the studies they undertake [33]. This is why differences in the perception of Computer Science between genders are not expected in primary education, but they are in secondary education.

Taking into account the aforementioned context, one of the main motivations behind this project is to present and analyze a methodology to make Computer Science much more appealing to young people through specific training on Computational Thinking. The specific research hypotheses analyzed herein are as follows:

H1: The perception of Computer Science that pre-university students have is improved through specific training.

H2: The learning methodology applied during the training influences the students' perception.

H3: In primary education, girls and boys have a similar perception of Computer Science.

H4: In secondary education, girls and boys have a different perception of Computer Science.

H5: There are differences in how Computer Science is perceived based on age.

H6: Computational Thinking skills are improved after completing the proposed training.

H7: The Computational Thinking skills or performance of students is independent of gender.

III. METHODOLOGY

In order to interest students in Computer Science through the principles of Computational Thinking, we designed a comprehensive methodology which involves the following experimental stages:

1) Measure the students' perception of Computer Science before they receive any training.
2) Train the students by performing a specific set of activities to develop Computational Thinking.
3) Measure the students' perception of Computer Science, and check their Computational Thinking skills after providing the training.

To see if the proposed training in Computational Thinking influences the students' perception, a survey, prepared by the authors (see Section 3.1) will be used at two different times: once before the start of the training, and then again at its conclusion. This way, we can analyze if performing specific activities involving Computational Thinking helps improve how students feel about Computer Science. To test the influence of the training on the students' Computational Thinking skills, a validated and consolidated Computational Thinking Test will be used (see Section 3.2). In this case, two groups of students will be created. The first group will receive the training, and then their skill level will be checked; while in the second group, their skill level will be assessed before any training is provided. This lets us analyze how Computational Thinking activities improve certain problem-solving skills.

We implemented this methodology through the following steps:

- Design of a tool—questionnaire—to measure the students' perception of Computer Science. This tool will be necessary to evaluate hypotheses **H1** to **H5**.
- Identify specific tests for measuring Computational Thinking skills. This will allow us to assess hypotheses **H6** and **H7**.
- Design activities to develop Computational Thinking skills.
- Design and carry out an activity to implement this methodological proposal.
- Analyze the results in order to ascertain the effect of Computational Thinking training.

A. MEASURING STUDENTS' PERCEPTION OF COMPUTER SCIENCE

To study whether the proposed training influences the students' perception of Computer Science, we designed a questionnaire based on Likert scales [34] for the students to complete twice, once before and once after the training. The questionnaire used in the pre-training stage is denoted as *pre-test*, while the questionnaire administered after the Computational Thinking training is called the *post-test*. All the students, regardless of the methodology they had followed when carrying out the activities, answered the same questionnaires. In both cases, students can select a numerical response from 1 to 5, where a low value means that they do not like or have no knowledge about the question, while a high value implies that they do like or have knowledge about the question. Specifically, the questionnaire contains six questions (the sixth is answered only on the post-test) that are intended to ascertain the students' notion of Computer Science. The questions were formulated in simple terms, in keeping with the educational level of the students:

I1: *How much do you like Computer Science?*

I2: *How much do you know about Computer Science?*

I3: *Do you think Computer Science is hard or difficult to learn?*
I4: *Do you think Computer Science is important?*
I5: *How much do you think you need to learn about Computer Science?*
I6: *Did you like the Computational Thinking activities that were presented?*

B. COMPUTATIONAL THINKING TEST

The level of Computational Thinking skills was studied and analyzed using the Computational Thinking Test (CTT), version 2.0, of November 2014 proposed by Román-González *et al.* [24], [25]. This test consists of twenty-eight multiple-choice questions of increasing difficulty. The questions involve different programming concepts and use exercises that have to be solved using block-based visual programming. The test also includes two items on self-efficacy that students must rate between 0 and 10:

- *How do you think the test went?*
- *What is your level of confidence regarding computers and computing?*

The programming concepts evaluated in this paper are the basic instructions, which in this case consist of movements in four directions (4 specific items), loops (12 items), conditionals (8 items) and functions (4 items), as well as nesting. These are embodied in three types of tasks: sequencing, completion, and debugging. A student's test score is a number ranging from 0 to 28 (which is the total number of answers), and which corresponds to the total number of correct answers. This categorization follows the one proposed by Román-González in the CTT.

In order to ascertain the effectiveness of the activities presented during the project, a group of students took this test before the training (in the first session), while the other group took it at the end of the Computational Thinking activities (last session). The groups were organized using logistical reasons, since all the students have the same educational level, the same age and some even are in the same school, although in different groups. This was done like this due to the amount of time required for each student to take the test, and because the time that can be devoted to the project is limited, as it takes time away from mandatory activities.

C. COMPUTATIONAL THINKING TRAINING

In this section, we propose a methodology that identifies the steps to take when integrating Computational Thinking into pre-university education. This model is a combination of the findings identified in the literature and our practical experience as educators. In addition to using widely accepted methodological strategies, this proposal is useful for clarifying the elements of Computational Thinking, its relationship with the problem-solving process and the practices most often used to develop it.

The specific training on Computational Thinking is based on a set of activities that were carefully selected for different age ranges and adapted for gender inclusiveness. The aim of the activities is to introduce Computational Thinking fundamentals, but also to put these skills into practice for general problem solving. The set of Computational Thinking activities includes both plugged and unplugged activities. The complete training involves 20 hours of activities. The training was divided into five sessions lasting four hours each. Half of each session was conducted face-to-face in the schools, which allowed us to introduce and carry out the activities directly with the students. The other half of each session was done in school under the supervision of the teachers, or at home by the students through self-study. We also designed two alternative training roadmaps based on two different learning perspectives or methodologies:

- *Guided learning*: this methodology focuses on developing Computational Thinking by introducing basic concepts and principles, in such a way that an example is introduced and solved step by step. Once the fundamentals are explained through an example, the students can try to apply a similar process to solve another problem.
- *Discovery learning*: this methodology focuses on tools that can be used to put Computational Thinking into practice. It provides students greater freedom to carry out the exercise, in such a way that they learn to use these tools autonomously, through trial and error mechanisms.

We decided to follow these two methodologies since theories like constructivism, discovery and guidance have been studied by many experts, creating a debate around which is most effective, from a cognitive point of view [35], and for Computer Science, the studies offer no clear conclusion [36].

Also, we could find no work in which the differences between students who carried out activities on Computational Thinking were analyzed using one learning methodology or another. Then, our goal is to present an initial approach to this subject by studying if there are significant differences between the results obtained by students who have followed either methodology.

Tables 1 and 2 show the distribution of the activities—and also the questionnaires or tests—performed in each session. In the primary education activities, for the guided methodology the students followed a course named "*Course 2*" from CODE.org. Since the students had no previous programming experience, it was important to use a platform like CODE.org, where the exercises are solved using a block-based visual programming language designed using Blockly [37]. In this modality, they also worked with the *Code & Go robot* [38], a programmable mouse-shaped device with two motors and buttons that can move around a board. For the discovery modality in primary education, physical devices were used, such as a *Makey Makey* board [39]. In this activity, the students, divided into groups of four, have to design a guitar with cardboard, adding buttons made with aluminum foil and connecting them to the board. Once they finish the guitar, they have to design and code a program in Scratch [40] to

TABLE 1. Description of the primary education activities performed.

Guided learning methodology				
First session	Second session	Third session	Fourth session	Fifth session
- Pre-test - Code&Go robot	- CODE.org (Maze)	- CODE.org (Artist)	- CODE.org (Bee)	- CODE.org (Loops, debugging) - Post-test
Discovery learning methodology				
First session - Pre-test - Makey Makey demo	Second session - Desing and build the guitar	Third session - Pong game in Scratch	Fourth session - Guitar program in Scratch	Fifth session - Guitar program in Scratch - Post-test

TABLE 2. Description of the secondary education activities performed.

Guided learning methodology				
First session	Second session	Third session	Fourth session	Fifth session
- Pre-test - CODE.org (Maze)	- CODE.org (Artist)	- CODE.org (Farmer)	- Guided Scratch (Pong game)	- CODE.org (Functions, conditionals) - Post-test
Discovery learning methodology				
First session - Pre-test - mBot demo	Second session - Desing and build the circuit	Third session - Pong game in Scratch	Fourth session - Pong game in Scratch - Program circuit in mBlock	Fifth session - Program and test the mBot - Post-test

play a different sound depending on the key that is pressed on the guitar. In addition, they have to simulate the famous *Pong game* in Scratch.

For the secondary education activities, in the guided learning methodology, the students completed a course named "*Accelerated Intro to CS Course*" on the CODE.org platform. This course, which lasts twenty hours, introduces concepts such as functions and conditionals. In addition, if the solution proposed is not optimal, the platform advises the student to try to find the best possible solution. For the learning by discovery method, the *mBot* [41] robot was used. The students had to program it to travel through a black course, designed and built by them using cardboard, using sensors to distinguish whether the robot is on a black or white background, such that if it veers off the course, it can turn around and continue walking through it. Additionally, an ultrasound sensor is used to detect if there are any obstacles in front of it. All the programming was done using the *mBlock* program [41], which combines the Scratch and Python programming languages.

D. STUDENT SAMPLE

The experimental evaluation was carried out at multiple local primary and secondary schools. A total of 558 students participated in this project: 276 from primary education (8-9 years old) and 282 from secondary school (12-13 years old). Table 3 provides information on the students involved in this project, broken by grade, gender and the learning methodology followed during the training. This table shows the number of students who completed the set of activities once incomplete or erroneous data —involving the tests or questionnaires— were discarded.

E. DATA ANALYSIS

Before the data were analyzed and studied, they were processed to eliminate questionnaires that were incomplete (only one of the questionnaires answered, and only in the case of the tests on the perception of Computer Science), that were duplicated or that had erroneous data, such as a student answering that they were a male in the pre-test and female in the post-test. As a result, although the same students were involved, the number of questionnaires analyzed for the two studies are different.

Once this process was complete, two different types of analytic studies were employed. In the case of the pre- and post-tests, the variation in the average scores before and after the specific training was taken into account. This variation was analyzed taking into account the age and gender of the students, but also the learning method used during the training activities. The scores obtained in the tests were analyzed with a significance level of 95% ($p < 0.05$), as per the Student's t-distribution study, to look for significant differences and accept or reject the hypotheses posed.

Regarding the measurement of the skills provided by the Computational Thinking Test, the Kolmogorov-Smirnov test

TABLE 3. Quantitative description of the study group.

PRIMARY EDUCATION				SECONDARY EDUCATION			
Guided		Discovery		Guided		Discovery	
148 students		128 students		160 students		122 students	
69 girls	79 boys	72 girls	56 boys	82 girls	78 boys	53 girls	69 boys

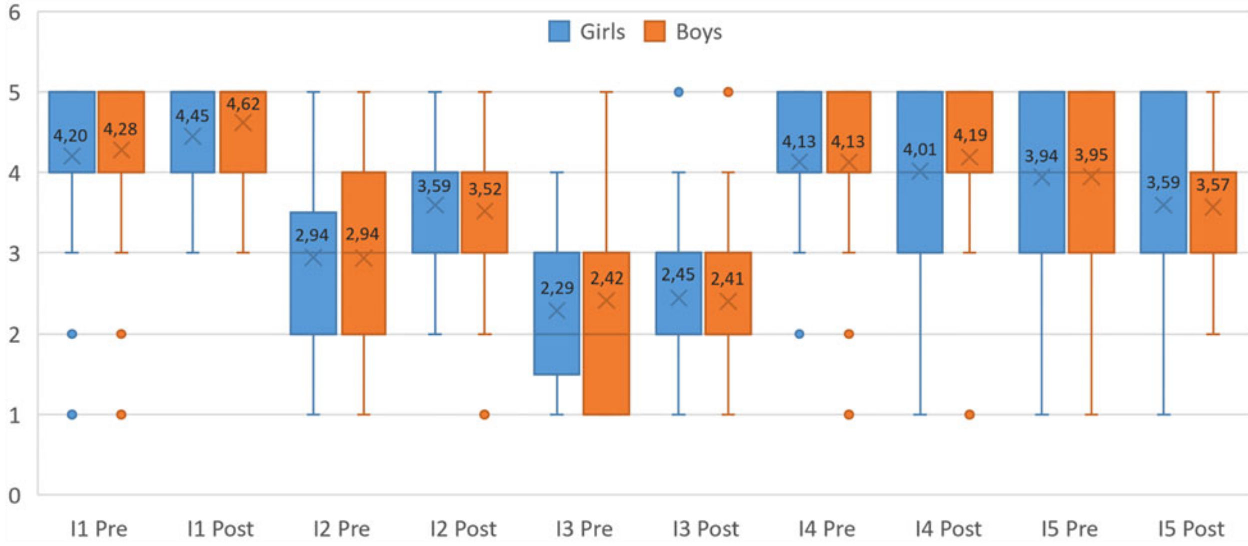

FIGURE 1. Results of the Pre-Test and Post-Test for the Guided Methodology in Primary Education, shown using a boxplot with quartiles Q1 and Q3, and the whiskers for the minimum and maximum values

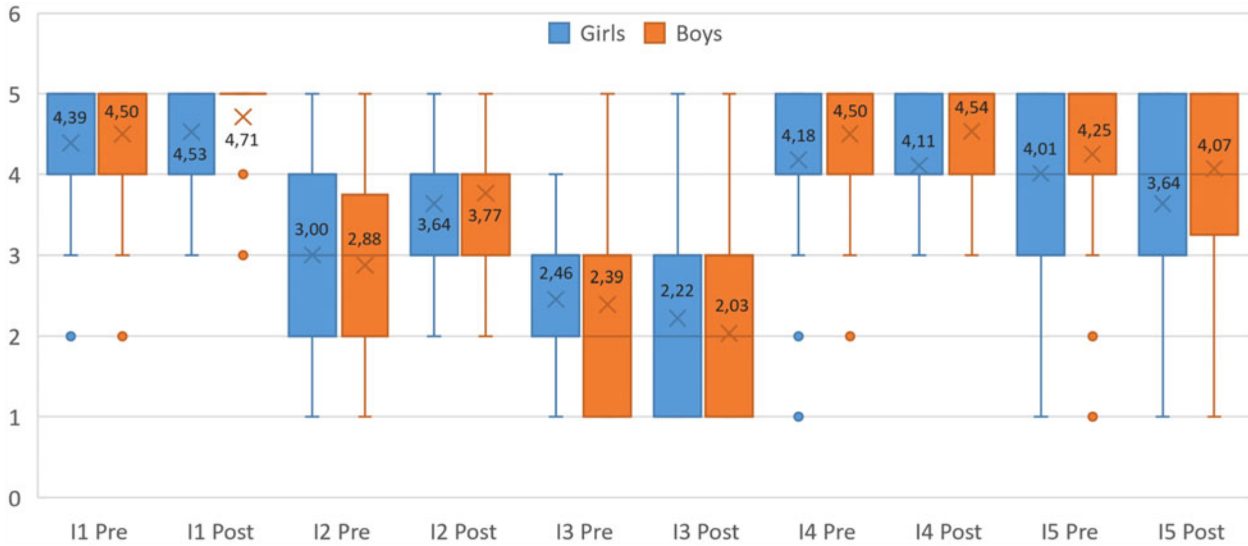

FIGURE 2. Result of the Pre-Test and Post-Test for the Discovery Methodology in Primary Education, shown using a boxplot with quartiles Q1 and Q3, and the whiskers for the minimum and maximum values.

was performed in order to verify normality. To verify that there were no significant differences in the results, the Student's t-distribution study was performed with a significance level of 95% ($p < 0.05$). Finally, Pearson's tests were also carried out to study the correlation between some of the questions and their answers. Just as with the variations between the pre- and post-tests, the average scores were also analyzed for the Computational Thinking Test.

IV. RESULTS AND DISCUSSION

In this section, the results for the questionnaires (pre-test and post-test) and for the Computational Thinking Test are presented and discussed. In accordance with the hypotheses posited, the analysis takes into account the grade (primary or secondary education), the gender and the learning methodology.

A. PRE-TEST AND POST-TEST

The results collected —before and after the training— through the questionnaires are shown using boxplot representations, with quartiles Q1 and Q3 for the scores and the whiskers for the minimum and maximum values for each inquiry (question or item in the questionnaire). Figures 1 and 2 show the results obtained in the pre-test and post-test for primary education students for both the guided and discovery methodologies. The results are separated by inquiries (in the x axis) and gender (with colors for differentiation). For the first and second question "*How much do you like Computer*

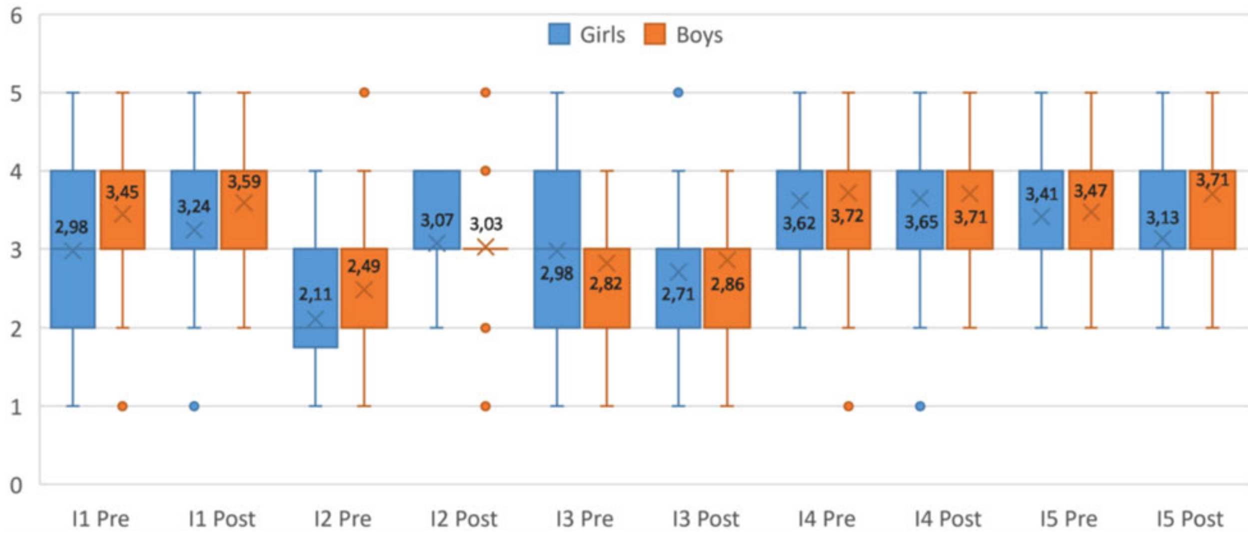

FIGURE 3. Result of the Pre-Test and Post-Test for the Guided Methodology in Secondary Education, shown using a boxplot with quartiles Q1 and Q3, and the whiskers for the minimum and maximum values.

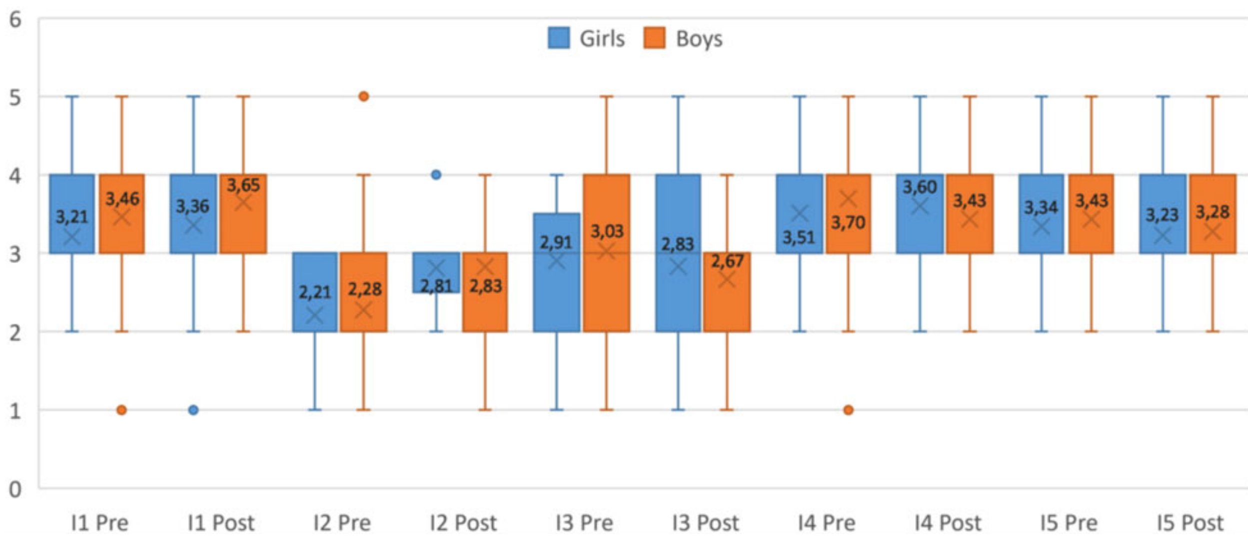

FIGURE 4. Result of the Pre-Test and Post-Test for the Discovery Methodology in Secondary Education, shown using a boxplot with quartiles Q1 and Q3, and the whiskers for the minimum and maximum values.

Science?" and "*How much do you know about Computer Science?*", we can see how, after the training, the feeling of the students in this area changed favorably. In every case, independently of gender and methodology, the score for these inquiries was higher in the post-test. For the third question, "*Do you think Computer Science is hard or difficult to learn?*", the results show that students viewed Computer Science as a complicated field before the training and afterwards, independently of the methodology. Going deeper into the fundamentals of a field does not usually make it look simpler, but instead produces an awareness of how broad or powerful it can be. Concerning the last two questions, "*Do you think Computer Science is important?*" and "*How much do you think you need to learn about Computer Science?*", we see that the students believe that Computer Science is a relevant field, meaning they will need to know about it.

Similar information is shown in Figures 3 and 4 for secondary education students taking part in both the guided and discovery-based methodologies. In this case, when analyzing the first two questions, "*How much do you like Computer Science?*" and "*How much do you know about Computer Science?*", there is a clear difference between gender, with the girls scoring lower on average than the boys. Regarding the third question, "*Do you think Computer Science is hard or difficult to learn?*", both girls and boys think thought it was easier after the training. The last two questions, "*Do you think Computer Science is important?*" and "*How much do you think you need to learn about*

Computer Science?" also reveal a difference between the answers of the girls and boys.

When analyzing the graphs, the differences between primary and secondary education are evident. In the case of primary education, note that the scores are higher compared to secondary school for the questions "*How much do you like Computer Science?*" and "*How much do you know about Computer Science?*", which reflects how Computer Science appeals more to the younger students. We also see that boys like Computer Science more than girls, although this difference is not significant in primary school. This is completely different in secondary school, with larger variations. These questions also indicate that girls' perception of Computer Science is worse than the boys'. Regarding the question "*Do you think Computer Science is important?*", in the post-test, the average score for this question decreased for the girls, while it increased for the boys, reflecting a change in perception. In the case of the secondary school students, the average scores are the opposite, with the girls' scores increasing and the boys' decreasing. However, with the question "*Do you think Computer Science is hard or difficult to learn?*", the secondary education students said it was less complicated than the primary education students, which means they have a better notion of Computer Science. In the case of the question "*How much do you think you need to learn about Computer Science?*", the primary school students thought they had to learn more than secondary school students. We also note that in the post-test, both groups of students in the guided and the discovery methodologies thought they did not need to learn too much. Despite the fact that both groups of students believed that they need to learn less after completing the activities, the greatest variation in this question was in primary education, so it is at this age when this type of content has the greatest influence. Regarding how much they thought they had to learn, we see a lack of confidence in the girls in this area, since their opinion changed favorably, meaning their scores were lower, after the Computational Thinking training.

In order to accept or reject the hypotheses posited, we conducted a statistical analysis in order to identify any significant differences among the results presented above. Table 4 shows, for each item in the questionnaire, the p-values obtained from comparing the pre-test and post-test results for primary and secondary education. Values lower than 0.05 are shown in bold. Note that in three out of five items, for both primary (**I1**, **I2** and **I5**) and secondary education (**I1**, **I2** and **I3**), there were statistically significant differences in the students' perceptions before and after the Computational Thinking training. Hence, **H1** is accepted, since specific training on Computational Thinking yields a change in how the students perceive the Computer Science field.

Table 5 shows statistical differences between the learning methodologies for both primary and secondary education. In the case of primary education, we note that there were no differences at all in the pre-test results, i.e., before the training. In the case of the post-test, i.e., after the training, statistically

TABLE 4. P-values obtained when comparing the results achieved for the different items of the pre-test and post-test to assess hypothesis h1 (the perception of computer science that pre-university students have is improved through specific training), which is accepted.

	I1	I2	I3	I4	I5
Primary	**0.0004**	**8.2E-18**	0.2760	0.7682	**0.0005**
Secondary	**0.0164**	**3.2E-23**	**0.0280**	0.5534	0.3044

significant differences between the two learning methodologies arose only in the case of **I3**, meaning that students that did the guided learning methodology considered Computer Science more difficult or hard to learn after specific Computational Thinking training. In the case of secondary education, there were no statistically significant differences in the pre-test results. If we consider the post-test results, statistically significant differences were found only for item **I2**, meaning that students that followed the guided learning methodology were more confident in their abilities and in what they thought they knew about Computer Science after the training, in comparison to those students who underwent the discovery learning methodology. In any case, hypothesis **H2** is rejected, since the majority of comparisons yielded no —statistically significant— differences.

Table 6, which presents the differences from a gender perspective, shows that the number of statistically significant differences between boys and girls are greater for secondary school students (items **I1**, **I2** and **I5**) than primary school students (items **I1** and **I4**), particularly for those questionnaire items involving a measure of what the students knew or felt about Computer Science. As a result, hypotheses **H3** and **H4** are accepted. We can conclude that, for primary education, there were no significant differences between boys and girls regarding their preferences for Computer Science, while in secondary education, those differences between boys are girls were more noticeable.

Finally, Table 7 shows the differences in how Computer Science is perceived by primary and secondary education students, i.e., according to age. We see that there were statistically significant differences for all the items in the questionnaire. Specifically, the younger students were much more enthusiastic about Computer Science and had fewer preconceptions about the difficulties involved in studying this subject. Bearing the above in mind, hypothesis **H5** is also accepted, and as a result, we conclude that early intervention can help modify the preconceptions that students have regarding this field of knowledge.

B. PERCEPTION OF THE ACTIVITIES

Students answered the question "*Did you like the Computational Thinking activities that were presented?*" once they had carried out all the interventions. Table 8 shows the mean values of the answers, arranged by educational level, methodology followed and gender.

TABLE 5. P-values obtained when comparing the results achieved for the different items of the questionnaire, for the guided learning methodology and the discovery learning methodology, to assess hypothesis h2 (the learning methodology applied during the training influences the students' perception), which is rejected.

	I1 Pre	I1 Post	I2 Pre	I2 Post	I3 Pre	I3 Post	I4 Pre	I4 Post	I5 Pre	I5 Post
Primary	0.0793	0.3628	0.9598	0.1288	0.5879	**0.0197**	0.0680	0.0689	0.1977	0.0641
Secondary	0.1937	0.3295	0.6197	**0.0102**	0.4868	0.6988	0.5696	0.1282	0.6421	0.1452

TABLE 6. P-values obtained when comparing the results achieved for the different items of the questionnaire, considering a gender perspective, to assess hypotheses h3 (in primary education, girls and boys have a similar perception of computer science) and h4 (in secondary education, girls and boys have a different perception of computer science), both of which are accepted.

	I1 Pre	I1 Post	I2 Pre	I2 Post	I3 Pre	I3 Post	I4 Pre	I4 Post	I5 Pre	I5 Post
Primary	0.5125	**0.0238**	0.6175	0.9554	0.8110	0.5059	0.2327	**0.0090**	0.4727	0.2281
Secondary	**0.0004**	**0.0035**	**0.0118**	0.6676	0.7819	0.9062	0.1678	0.6371	0.5107	**0.0019**

TABLE 7. P-values obtained from the comparison of the results achieved for the different items of the questionnaire, according to age, to assess the hypothesis h5 (there are differences in how computer science is perceived based on age), which is accepted.

	I1 Pre	I1 Post	I2 Pre	I2 Post	I3 Pre	I3 Post	I4 Pre	I4 Post	I5 Pre	I5 Post
Primary & Secondary	**2E-36**	**8.7E-49**	**3.3E-17**	**2.6E-23**	**3.4E-10**	**2.2E-08**	**2.7E-15**	**1.5E-14**	**3.8E-12**	**4.4E-05**

The results analyzed show that there is a considerable difference between primary and secondary school, with the younger children liking these activities more. The p-values obtained for this question reveal large differences between primary and secondary school, with a value of $2.78e-23$. Despite this, there are no significant differences between following one learning methodology or the other in either primary or secondary school, or between genders in primary and following a discovery methodology in secondary, but there are between the students that followed a guided methodology in secondary, with a p-value of 0.0066.

C. COMPUTATIONAL THINKING TEST

This study of Computational Thinking skills was conducted in two subgroups: one before the training and another afterward. The first subgroup consisted of 65 primary education students and 80 secondary education students. The other subgroup consisted of 76 primary school students and 130 students from secondary education. To study the normality of the sample, a Kolmogorov-Smirnov test was applied. P-values equal to 0.553 and 0.341 were obtained for primary school girls and boys, respectively, when comparing results obtained before and after the training. P-values equal to 0.723 and 0.459 were obtained in the case of secondary school students. According to the above data, the normality of the samples is accepted.

TABLE 8. Mean values for the sixth question, "Did you like the computational thinking activities that were performed?".

	Mean	Method.	Mean	Gender	Mean
Primary	4.7138	**Guided**	4.7230	**Girls**	4.6957
				Boys	4.7468
		Discovery	4.7031	**Girls**	4.6111
				Boys	4.8214
Secondary	4.0319	**Guided**	4.0000	**Girls**	3.8049
				Boys	4.2051
		Discovery	4.0738	**Girls**	4.0000
				Boys	4.1304

Figures 5 and 6 show the results for the Computational Thinking Test, divided according to the two subgroups: the one that took the test before the training and the one that took it afterward. The results are also differentiated by gender. In the case of primary education, the average score—out of 28.0—for the group that took the test before the training was 10.2 for girls and 9.7 for boys, while for the group that took the test after the training, the average score was 13.5 for girls and 14.6 for boys. As for secondary education groups, the average score for the group that took the test before the training was 15.6 for girls and 16.2 for boys, while for the group that took it once the training ended, the average score was 17.2 for girls and 19.4 for boys. These differences were analyzed using a Student's t-distribution with a significance level equal to 95% ($p < 0.05$). The P-values were equal to $3.48e-09$ and 0.0008 for primary and secondary education, respectively. As a result, there were statistically significant differences when the test was taken before and after the specific training, and therefore hypothesis **H6** (Computational Thinking skills are improved after completing the proposed training) is accepted.

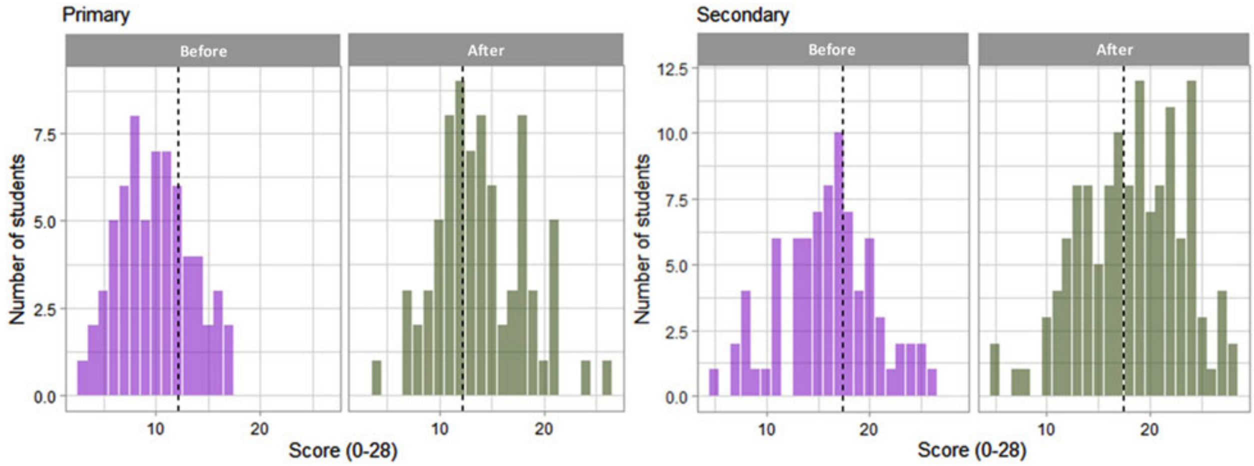

FIGURE 5. CTT results shown using a bar chart, with the average score shown for both cases.

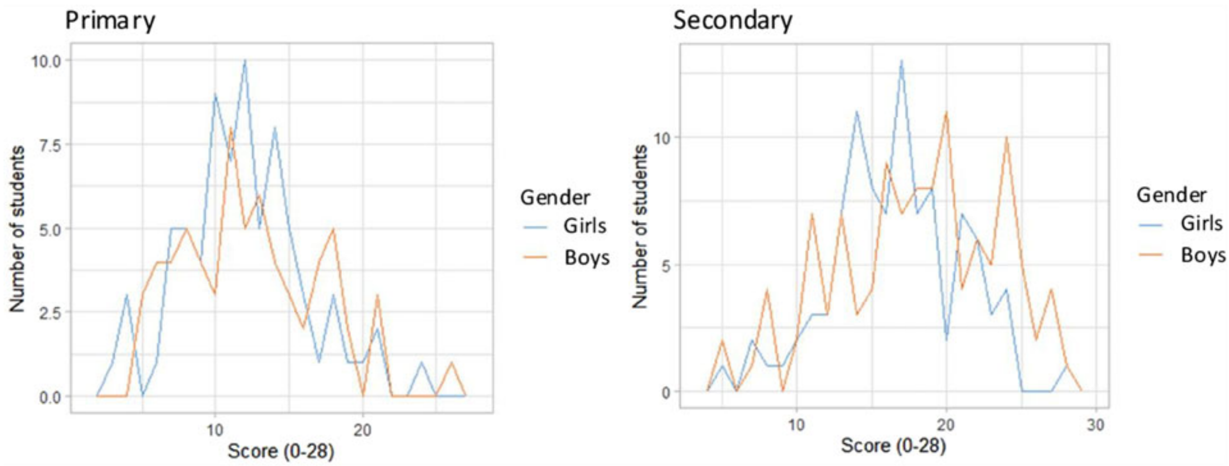

FIGURE 6. CTT results separated by gender displayed using a frequency polygon, based on score and number of students.

The concepts (directions, loops, conditionals and functions) that were worked on using exercises in this Computational Thinking Test are analyzed in seven different categories, following the one proposed by Román-González in the CTT:

1) *Loop until*: exercises with loops that are repeated until a condition is met.
2) *Loop times*: exercises with loops that are repeated a specified number of times.
3) *Conditional compound*: exercises with conditionals with various requirements.
4) *Conditional while*: exercises with conditionals in which the conditional is executed while something happens.
5) *Simple conditional*: exercises in which only one requirement has to be met.
6) *Routes*: exercises in which an object has to follow a series of basic directions.
7) *Functions*: exercises that include functions.

The results of the concepts analyzed are collected in Figure 7, which shows the average scores for each of the aforementioned concepts. These results are grouped by gender and by the period when the test was carried out (before or after the training). These results reflect a favorable change for all ages and genders, in addition to all the concepts that were worked on. However, there was a difference in primary education of 1.1 points and of 2.2 points in secondary. Also, a Student's t-distribution study showed that, when comparing results by gender, there were no statistically significant differences with respect to primary education (p-value = 0.5526), but there were differences regarding secondary education (p-value = 0.03543). Therefore, hypothesis **H7** (The Computational Thinking skills or performance of students is independent of gender) is accepted for primary education, while it is rejected in the case of secondary education.

In addition to the study of Computational Thinking skills, two additional questions were asked about self-efficacy. The first question ("*How do you think the test went?*") let us know how the students thought they had done on the test. According to Figure 8, primary school students were more self-confident: the average scores they thought they had received, for both genders, was over 7 points (out of 10), despite the fact that the actual results did not reflect these scores. In secondary school, most students did not have much self-confidence, since they thought they have received a score below 7 points (out of 10),

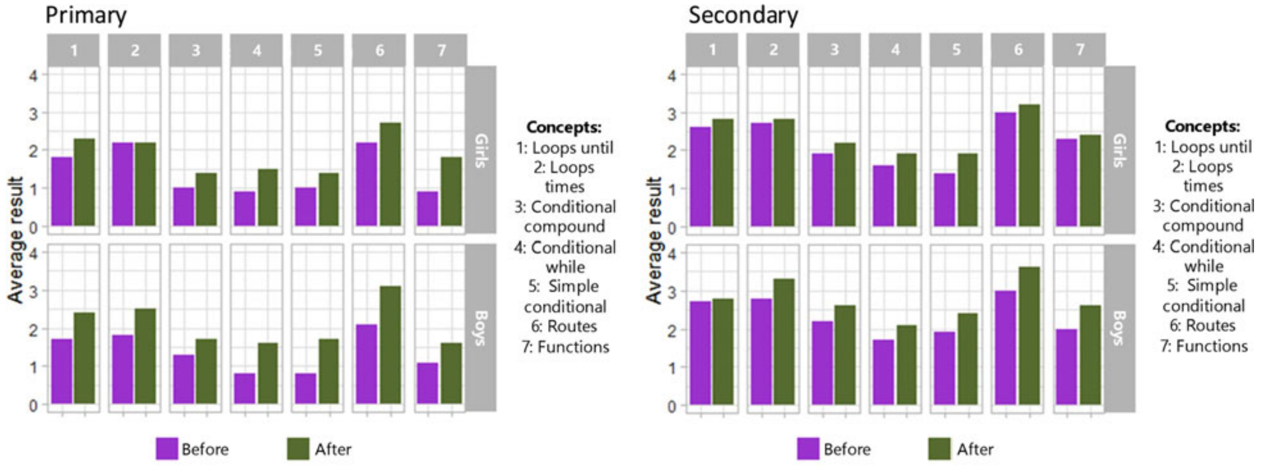

FIGURE 7. **Average result of Computational Concepts shown using a bar chart, differentiating between the answers of the group that took the test before and those that did it later, as well as by gender.**

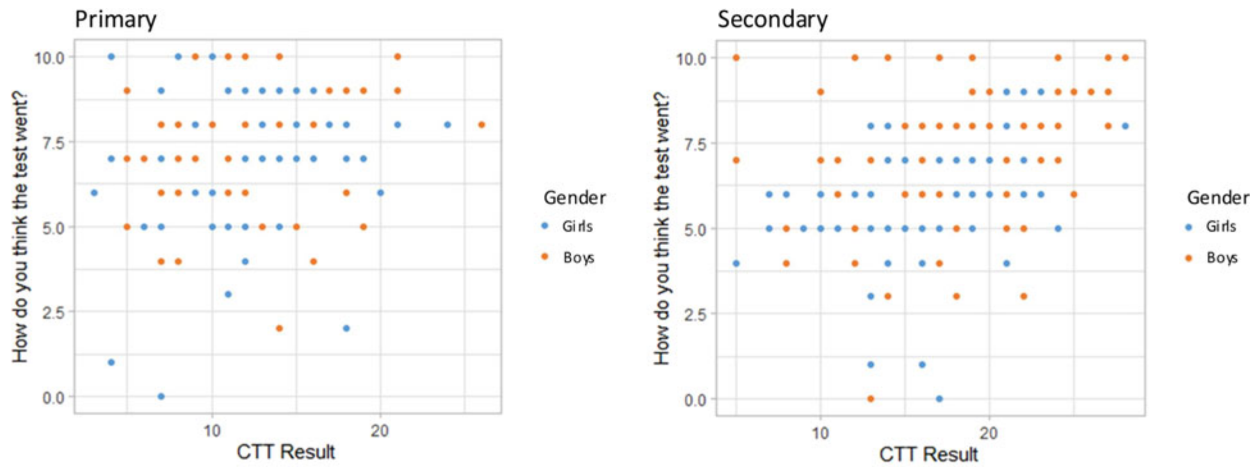

FIGURE 8. **Relationship between the Computational Thinking Test Results and the answer to the question "*How do you think the test went?*" segregated by sex, displayed using a point cloud.**

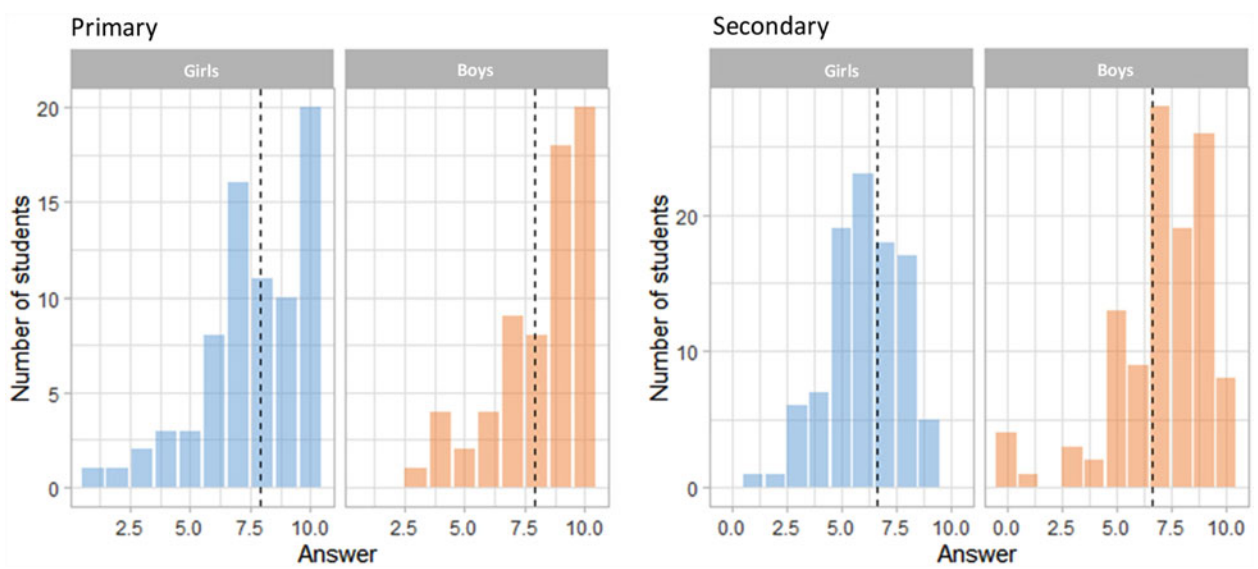

FIGURE 9. **Relationship between the Computational Thinking Test results and the answer to the question "*What is your level of confidence regarding computers and computing?*" segregated by sex, displayed using a bar chart, with the average score shown for both genders.**

even though the actual results reflect the opposite. In addition, the boys showed more self-confidence than the girls. The second question ("*What is your level of confidence regarding computers and computing?*") can be analyzed through Figure 9. We see that the average value for primary school students (7.95 out of 10.0) is higher than for secondary school students (6.6 out of 10.0). When considering the gender, the average values are lower for girls in primary education, as well as in secondary education.

The differences between taking the test before and after the training reveal the improvement in Computational Thinking skills for all ages after completing the exercises. This difference is greater in primary education. In the case of secondary education, there is a difference between boys and girls, with the latter obtaining an average score of more than two points lower. Regarding the concepts, the differences between primary and secondary education are evident, with the latter obtaining a higher score in all concepts. However, in the conditionals, both the youngest and the oldest students received the lowest scores, with loops and basic directions receiving the highest. The boys obtained better scores in every concept compared to the girls, and the difference after doing the activities was also more significant for the boys.

V. CONCLUSION

The methodology proposed in this paper seeks to provide more Computer Science education, and more specifically Computational Thinking, to pre-university students, since the interest shown in this particular area is much lower in comparison to other fields of knowledge. This initial approach yields some ideas about the different ways in which this topic could be implemented in the educational curriculum. The analyses presented over the course of this paper show how, with relatively little specific training on Computational Thinking, the existing perception of Computer Science can be improved, as can the skills related to Computational Thinking. This improvement is more noticeable in primary education, meaning that activities involving younger students could have more of an effect in helping to avoid misconceptions about Computer Science. Our analyses also revealed that younger students are more self-confident, and that differences between girls and boys are smaller at earlier stages. As a result, it is necessary to emphasize this type of training before the gap between gender and self-confidence arises.

At the same time, one of the hypotheses posited in this paper involved the existence of differences when the specific Computational Thinking training was provided following a guided learning methodology or a discovery learning methodology. However, those differences were not clearly discernible. This may be due to the time that was devoted to the interventions, carried out over only five 2-hour sessions, so it would be interesting to analyze this with students who can, for example, participate in this type of training over a full academic year. Consequently, as an area of future research, other types of activities that combine both methodologies will be designed and tested. In short, the results offer a promising insight on how to deal with preconceptions involving Computer Science, so it is necessary to continue designing experimental methodologies, training activities and other types of interventions, particularly those that focus on younger students, even as early as kindergarten. In any case, the independent activities proposed offer a good starting point to introduce Computational Thinking and the concepts of Computer Science into classrooms, since, as some authors have noted, introducing it into an educational curriculum as an interdisciplinary approach to increase interest in these areas is a complex task that involves different institutions [4], [9], [42].

We were able to demonstrate that specific training in Computational Thinking can change students' attitude towards the field of Computer Science and their Computational Thinking skills. However, the way in which this proposal was implemented requires certain resources to be available: on the one hand, qualified personnel to design the methodology and specific activities, as well as teachers interested in being part of these initiatives. Although the training hours required are not excessive, it may not be easy for schools - or their teachers - to reorganize their schedule to accommodate this type of proposal, especially in countries where this content has not been officially included in the curriculum.

In addition, one of the main obstacles faced by different educational institutions, whether primary or secondary, in countries with or without laws that require the inclusion of Computational Thinking, is teacher training. This was already noted by Papert in 1980 [2] and even debated in the 1960s, as presented in Caeli and Yadav [43]. After more than half a century, teacher training remains one of the great outstanding questions involving research in Computational Thinking. Because of this, the proposal presented in this paper can offer a good starting point, since it allows teachers continuous training by providing experts who carry out the activities. In addition, as is evident from our results, we were successful in capturing the interest of the students, who then asked for and requested this type of content.

There are numerous initiatives that attempt to analyze the status of Computational Thinking in the educational curricula of different countries, such as the Reviewing Computational Thinking in Compulsory Education report of the Joint Research Centre, at the European level [10]. Locally, we are also working on new editions of this and other projects, such as "*C**4. Computational Thinking in the Canarian Educational System: Diagnosis and Roadmap for its Incorporation into the Curriculum*" [44], with which we seek to ascertain the state of Computer Science in the various educational centers of the Canary Islands from the point of view of the students, teachers and the centers themselves so that we can define a roadmap for effectively integrating Computational Thinking as a cross-disciplinary skill in the educational curriculum. Another project is the Open Course Ware (OCW) on "*Computational Thinking digital booklets*" [45], with which we seek to freely and openly disseminate Computational Thinking so that anyone can work on it.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Aula Cultural de Pensamiento Computacional de la Universidad de La Laguna and the Unidad de Cultura Científica y de la Innovación (Cienci@ULL) de la Fundación General de la Universidad de La Laguna.

REFERENCES

[1] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006.
[2] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. Sussex: The Harvester Press, 1980.
[3] S. Papert, "An exploration in the space of mathematics educations," *Int. J. Comput. Math. Learn.*, vol. 1, no. 1, pp. 95–123, 1996.
[4] V. Barr and C. Stephenson, "Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?," *ACM Inroads*, vol. 2, no. 1, pp. 48–54, Feb. 2011.
[5] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," in *Proc. Annu. Amer. Educ. Res. Assoc. Meeting*, 2012, pp. 1–25.
[6] S. Grover and R. Pea, "Computational thinking in K-12: A review of the state of the field," *Educ. Researcher*, vol. 42, no. 1, pp. 38–43, 2013.
[7] E. Segredo, G. Miranda, and C. León, "Towards the education of the future: Computational thinking as a generative learning mechanism," *Educ. Knowl. Soc.*, vol. 18, no. 2, pp. 33–58, 2017.
[8] E. Segredo, G. Miranda, C. León, and A. Santos, "Developing computational thinking abilities instead of digital literacy in primary and secondary school students," in *Smart Education and e-Learning*, V. L. Uskov, R. J. Howlett, and L. C. Jain, Eds., Berlin, Germany: Springer International Publishing, 2016, pp. 235–245.
[9] J. Voogt, P. Fisser, J. Good, P. Mishra, and A. Yadav, "Computational thinking in compulsory education: Towards an agenda for research and practice," *Educ. Inf. Technol.*, vol. 20, no. 4, pp. 715–728, 2015.
[10] Joint Research Centre, "Reviewing computational thinking in compulsory education," Publications Office of the European Union, Luxembourg (Luxembourg), Scientific analysis or review KJ-06–22-069-EN-N (online), 2022, Accesed: Mar. 15, 2022. [Online]. Available: https://publications.jrc.ec.europa.eu/repository/handle/JRC128347
[11] S. Sentance and A. Csizmadia, "Computing in the curriculum: Challenges and strategies from a teacher's perspective," *Educ. Inf. Technol.*, vol. 22, no. 2, pp. 469–495, 2016.
[12] P. Micheuz, S. Pasterk, and A. Bollin, "Basic digital education in austria – one step further," in *Proc. Adv. Inf. Commun. Technol.*, 2017, pp. 432–442.
[13] A. Lamprou and A. Repenning, "Teaching how to teach computational thinking," in *Proc. Conf. Innov. Technol. Comput. Sci. Educ.*, 2018, pp. 69–74.
[14] S. Bocconi *et al.*, "Developing computational thinking in compulsory education," European Commission, Joint Research Centre, Tech. Rep. JRC104188, 2016.
[15] INTEF, "Programación, robótica y pensamiento computacional en el aula. situación en españa," Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, Tech. Rep., Jan. 2018, Accessed: Mar. 15, 2022. [Online]. Available: http://code.intef.es/wp-content/uploads/2017/09/Pensamiento-Computacional-Fase-1-Informe-sobre-la-situaci%c3%b3n-en-Espa%c3%b1a.pdf
[16] B. Standl, J. Rosato, A. Leftwich, and V. Fragapane, "(How) do they bring computer science to everyone? Pre-service computer science teacher programs in Germany and in the USA," in *Proc. Annu. Conf. Soc. Inf. Technol. Teacher Educ.*, 2020, pp. 631–637.
[17] Code.org, "Code.org," 2012. Accesed: Mar. 15, 2022. [Online]. Available: https://studio.code.org/courses
[18] C. Angeli *et al.*, "A K-6 computational thinking curriculum framework: Implications for teacher knowledge," *Educ. Technol. Soc.*, vol. 19, no. 3, pp. 47–57, Jul. 1, 2016.
[19] K. Falkner and R. Vivian, "Coding across the curriculum: Resource review," Computer Science Education Research Group, The University of Adelaide, Tech. Rep. D15/608355, 2015.
[20] A. Funke, M. Berges, and P. Hubwieser, "Different perceptions of computer science," in *Proc. Int. Conf. Learn. Teach. Comput. Eng.*, 2016, pp. 14–18.
[21] J. Henry and B. Dumas, "Perceptions of computer science among children after a hands-on activity: A pilot study," in *Proc. IEEE Glob. Eng. Educ. Conf.*, 2018, pp. 1811–1817.
[22] S. Grover, R. Pea, and S. Cooper, "Remedying misperceptions of computer science among middle school students," in *Proc. ACM Tech. Symp. Comput. Sci. Educ.*, 2014, pp. 343–348.
[23] R. Herrero, E. Segredo, G. Miranda, and C. León, "El proyecto Piens@Computacion@LLmente," in *Proc. Actas del V Congreso Internacional sobre Aprendizaje, Innovación y Cooperación*, 2019, pp. 573–578, Accesed: Mar. 15, 2022. [Online]. Available: https://zaguan.unizar.es/record/84633
[24] M. Román-González, "Computational thinking test: Design guidelines and content validation," in *Proc. Int. Conf. Educ. New Learn. Technol.*, 2015, pp. 2436–2444.
[25] M. Román-González, J.-C. Páerez-González, and C. Jiménez-Fernández, "Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test," *Comput. Hum. Behav.*, vol. 72, pp. 678–691, 2017.
[26] ULL, "Alumnado de nuevo ingreso en estudios de grado y primer y segundo ciclo," Universidad de La Laguna, Tech. Rep. 4987, 2019, Accesed: Mar. 15, 2022. [Online]. Available: https://riull.ull.es/xmlui/bitstream/handle/915/4987/Evo_matricula_nues_grado.pdf
[27] R. Strachan, A. Peixoto, I. Emembolu, and M. T. Restivo, "Women in engineering: Addressing the gender gap, exploring trust and our unconscious bias," in *Proc. IEEE Glob. Eng. Educ. Conf.*, 2018, pp. 2088–2093.
[28] A. Master, S. Cheryan, and A. N. Meltzoff, "Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science," *J. Educ. Psychol.*, vol. 108, no. 3, pp. 424–437, Apr. 2016.
[29] O. Seneviratne, *Making Computer Science Attractive to High School Girls with Computational Thinking Approaches: A Case Study*, Cham, Switzerland: Springer International Publishing, Apr. 25, 2017, pp. 21–32.
[30] J. D. Miller and L. G. Kimmel, "Pathways to a stem profession," *Peabody J. Educ.*, vol. 87, no. 1, pp. 26–45, Jan. 1, 2012.
[31] I. Plante, M. Théorêt, and O. E. Favreau, "Student gender stereotypes: Contrasting the perceived maleness and femaleness of mathematics and language," *Educ. Psychol. (Dorchester-on-Thames)*, vol. 29, no. 4, pp. 385–405, Jul. 1, 2009.
[32] B. Kurtz-Costes, K. E. Copping, S. J. Rowley, and C. R. Kinlaw, "Gender and age differences in awareness and endorsement of gender stereotypes about academic abilities," *Eur. J. Psychol. Educ.*, vol. 29, no. 4, pp. 603–618, Dec. 2014.
[33] J. Wang, H. Hong, J. Ravitz, and M. Ivory, "Gender differences in factors influencing pursuit of computer science and related fields," in *Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, 2015, pp. 117–122.
[34] A. Joshi, S. Kale, S. Chandel, and D. Pal, "Likert scale: Explored and explained," *Brit. J. Appl. Sci. Technol.*, vol. 7, no. 4, pp. 396–403, Jan. 10, 2015.
[35] P. A. Kirschner, J. Sweller, and R. E. Clark, "Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching," *Educ. Psychol.*, vol. 41, no. 2, pp. 75–86, Jun. 1, 2006.
[36] M. Guzdial, "What's the best way to teach computer science to beginners?," *Commun. ACM*, vol. 58, no. 2, pp. 12–13, Jan. 28, 2015.
[37] Blockly, "Blockly," 2012. Accesed: Mar. 15, 2022. [Online]. Available: https://developers.google.com/blockly/
[38] Learning Resources, "Code & Go Robot Mouse Activity Set," 2016. Accesed: Mar. 15, 2022. [Online]. Available: https://www.learningresources.com/code-gor-robot-mouse-activity-set
[39] JoyLabz, "Makey makey," 2012. Accesed: Mar. 15, 2022. [Online]. Available: https://makeymakey.com/
[40] M. Resnick *et al.*, "Scratch: Programming for all," *Commun. ACM*, vol. 52, no. 11, pp. 60–67, Nov. 2009.
[41] Makeblock, "mBot," 2015. Accesed: Mar. 15, 2022. [Online]. Available: https://www.makeblock.com/
[42] C.-C. Lin, M. Zhang, B. Beck, and G. Olsen, "Embedding computer science concepts in k-12 science curricula," in *Proc. 40th ACM Tech. Symp. Comput. Sci. Educ.*, 2009, pp. 539–543.

[43] E. N. Caeli and A. Yadav, “Unplugged approaches to computational thinking: A historical perspective,” *TechTrends*, vol. 64, no. 1, pp. 29–36, Jul. 6, 2019.
[44] ULL, “C**4. el pensamiento computacional en el sistema educativo canario: Diagnóstico y hoja de ruta para su incorporación en el currículo,” Mar. 2021, Accesed: Mar. 15, 2022. [Online]. Available: https://portalciencia.ull.es/proyectos/47401/detalle
[45] ULL, “Cuadernillos digitales de pensamiento computacional,” 2020. Accesed: Mar. 15, 2022. [Online]. Available: https://campusvirtual.ull.es/ocw/course/view.php?id=153

RAFAEL HERRERO-ÁLVAREZ received the BS degree from Universidad de La Laguna, Spain, in 2017, and the MS degree from Universitat Oberta de Catalunya, Spain, in 2019. He is currently working toward the PhD degree in computer science related to the Development of Computational Thinking through an Emotion-Based Adaptive Learning Support System.

GARA MIRANDA received the BS, MS, and PhD degrees from Universidad de La Laguna, Spain, in 2002, 2004 and 2009, respectively, all in computer science. She is currently an associate professor with the Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna, Spain. Her research work has been linked mainly to combinatorial optimization and software development of frameworks for solving real problems through the application of heuristics, meta-heuristics and hyper-heuristics. She is also interested in Computational Thinking and other issues related to Computer Science teaching. Along these lines, she has published more than 100 scientific papers. Such publications currently report more than 700 citations in Google Scholar which results in an H-index of 16.

COROMOTO LEÓN received the MS degree in mathematics, and the PhD degree in computer science, both from the Universidad de La Laguna, Tenerife, Spain, in 1990 and 1996, respectively. She currently teaches programming languages and paradigms in graduate and master’s programs, with the Departamento de Ingeniería Informática y de Sistemas. She has led dozens of doctoral courses and multiple seminars and has directed several research projects and contracts for innovation and industry transfer. She has published more than 30 journal papers and refereed conferences. Her research interests include programming languages, algorithmic techniques, optimization, parallel programming and Computational Thinking.

EDUARDO SEGREDO (Member, IEEE) received the BS, MS, and PhD degrees from the Universidad de La Laguna, San Cristóbal de La Laguna, Spain, in 2006, 2008, and 2014, respectively, all in computer science. He is currently a lecturer with the Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna. He has authored or coauthored more than 50 technical papers and book chapters, including more than 15 journal papers. His publications currently report more than 400 citations in Google Scholar with an H-index of 13. His current research interests include single/multi/many-objective optimization, evolutionary algorithms, meta-heuristics, machine learning and Computational Thinking. He currently serves on the editorial board of multiple international conferences.