


Article

Assessing the Computational Thinking of Pre-Service Teachers: A Gender and Robotics Programming Experience Analysis

Lourdes Villalustre ^{1,*} and Marisol Cueli ^{2,*} ¹ Department of Educational Sciences, University of Oviedo, 33005 Oviedo, Spain² Department of Psychology, University of Oviedo, 33003 Oviedo, Spain

* Correspondence: villalustrelourdes@uniovi.es (L.V.); cuelimarisol@uniovi.es (M.C.)

Abstract: Computational thinking has provided a new logical approach to teaching programming. However, certain variables, such as gender and previous experience in robotic programming, may influence its development. The present study was aimed to (1) characterise 164 infant and primary education pre-service teachers in terms of their level of computational thinking, (2) analyse whether there are statistically significant differences according to gender and previous experience in robotics programming, and (3) identify profiles that would allow us to analyse the differences between them and the dimensions of computational thinking. For this purpose, the Computational Thinking Test (CTT) was used. The results of the *t*-tests revealed that men obtained better results than women and that previous experience in robotics programming is a determining factor in the level of development of the CTT. Furthermore, by means of a two-stage cluster analysis, three profiles were identified among the prospective teachers. The statistical analysis revealed that women who had previous experience and had used programming languages at some point had the best results in the CTT. These findings underline the importance of integrating computational thinking into the training of prospective teachers and adapting teaching strategies according to the specific needs of each group of students.

Keywords: computational thinking; gender; robotics programming



Citation: Villalustre, L.; Cueli, M. Assessing the Computational Thinking of Pre-Service Teachers: A Gender and Robotics Programming Experience Analysis. *Educ. Sci.* **2023**, *13*, 1032. <https://doi.org/10.3390/educsci13101032>

Academic Editor: Michael Simonson

Received: 3 July 2023

Revised: 29 September 2023

Accepted: 12 October 2023

Published: 16 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Computational thinking involves a process of problem solving by applying basic knowledge of science, technology, logic, and mathematics [1,2]. The term emerged from Papert's [3] work on constructionist programming environments based on design and algorithm-based problem solving. It is a form of emergent literacy that seeks to foster the learning of programming progressively, using basic computer coding principles. In this approach, students at different educational levels create sequences of instructions and algorithms to control the actions of an object in a virtual or physical space. For Want et al. [4], the development of computational thinking involves various skills and abilities, such as organising and logically analysing information, identifying, investigating, implementing possible solutions, etc. while strengthening algorithmic thinking.

Algorithmic thinking is a fundamental problem-solving skill, which involves devising a step-by-step solution to address a given problem. Bers et al. [5] state that, unlike coding, which refers to the technical skills needed to write code in a specific programming language, algorithmic thinking focuses on the conceptual process of designing an algorithm. Along these lines, for Selby [6], algorithmic thinking is distinguished from coding and is considered an independent skill prior to coding. It involves the ability to plan and organise the actions needed to solve a problem sequentially and effectively. These skills are considered important elements of computational thinking and are fundamental in the development of programming and computational problem-solving skills [7,8].

There is a broad consensus about the close relationship between computational thinking skills and programming. Different authors [9,10] argue that computational thinking

has provided a new logical framework for teaching programming. Computational thinking involves a set of cognitive and problem-solving skills that are fundamental to the field of robotic programming. These skills include breaking down complex problems into simpler steps, identifying patterns and regularities, designing algorithms to solve problems, and the ability to think in a logical and structured way. Programming, in turn, provides a practical platform for students to exercise and apply these computational thinking skills in solving specific problems.

Existing studies have shown that learning and practising robotic programming contributes significantly to the development of computational thinking in students [11,12]. By participating in programming activities, learners are challenged to analyse and understand how computer systems work, to identify and resolve bugs, and to improve their solutions. These experiences foster skills such as logical reasoning, abstraction, creativity, and collaboration, which are fundamental to both computational thinking and programming [13]. Recently, Su and Yang, in their extensive review of the literature on computational thinking, concluded that the abstraction and generalization of patterns, the systematic processing of information, and the decomposition of the structure of problems, among others, are basic elements that are integrated in computational thinking and should be part of the curriculum, given their cognitive contributions [14].

The development and improvement of computational thinking associated with robotic programming have undergone a growing evolution in recent years. However, it is true that this progress has not had the same impact on women as on men and that there is a certain disparity in interest in robotic programming between the two groups. It is important to recognise that this situation may be influenced by cultural factors and socialisation processes experienced from childhood that may motivate men more than women to get involved in these fields [15].

Along these lines, gender stereotypes may have been constructed by associating science, technology, engineering, and mathematics (STEM) with masculine characteristics. For Chan [16], these stereotypes may create barriers and disincentives for women to explore and actively participate in these sectors. Furthermore, as Charlesworth [17] points out, the gender roles and expectations that are transmitted in society can also influence people's choices and preferences from an early age. It is therefore essential to promote gender equality in science and technology fields as well as in the development of robotics programming. This implies fostering inclusive and equitable education that provides opportunities and resources for all people to expand their interests and skills in these fields, regardless of their gender [18].

Strong computational thinking skills can be a motivating factor for students to be interested in STEM-related careers. However, it is not currently integrated into curricula in a broad and systematic way like other subjects such as mathematics or science [19,20]. The lack of a specific curriculum that lays the foundations for computational thinking from an early age represents a challenge in education. There are few models or approaches that highlight computational thinking in current educational practices [21]. Despite the importance of computational thinking, little attention has been paid to its development in the training of future teachers. Most research and experiences prioritise its teaching at the compulsory secondary education stage [12], leaving a gap in the training of future teachers.

However, there is a European and worldwide trend to introduce computational thinking into the curriculum. For example, in Spain, some institutionalized initiatives have been developed for the incorporation of computational thinking in both primary and secondary education [22] such as the School of Computational Thinking and Artificial Intelligence created by INTEF [23], which is aimed to explore the possibilities of computational thinking and Artificial Intelligence for learning in the classroom. To achieve this goal, they have created different educational and training resources for interaction in non-university classrooms. In addition, there is a national effort to include computational thinking in subjects related to technology and computer science. However, it is likely that students who do not attend these subjects will not have access to the benefits that computational thinking can

bring them [24]. For this reason, the measures taken at the national level for the integration and development of computational thinking are mainly focused on infant, primary, and secondary education. At the university level, to the best of our knowledge, there are no unified and institutionalized measures for the development of computational thinking.

To address this situation, it is necessary to promote greater integration of computational thinking in the training of future teachers. This implies the incorporation of content and methodologies related to computational thinking in teacher training programmes, as well as the development of appropriate resources and materials for its teaching [25]. It is also important to take into account the level of computational thinking that students possess. Carrying out diagnostic assessments to find out their level of competence in this area can help to identify areas for improvement and adapt training actions according to the needs of each group of students. In addition, it is important to consider variables such as gender and previous experience in robotics programming, as they can influence the development of computational thinking [26]. This will allow for more tailored and inclusive training actions.

Along these lines, several studies have addressed the relationship between students' computational thinking skills and gender, as well as with previous experience in robotics programming. Thus, the work carried out by Atmatzidou et al. [27] showed that there are no significant gender differences in the development of computational thinking skills among students aged 15–18 years who participated in an Educational Robotics programme. The same holds for the studies conducted by Bati [28] and Kanaki et al. [29] with infant education pre-service teachers. These findings suggest that gender does not significantly influence the development of these skills in these contexts. Master et al. [30] found that elementary school students with programming experience reported higher technical interest and self-efficacy compared to those without robotic programming experience.

In short, most of the current studies that have focused on analysing gender and programming experience in relation to computational thinking have been conducted on students at non-university levels. However, the way in which these factors may influence future teachers has not been explored in depth. Therefore, the present work stated three aims. First, to characterise the pre-service teachers (first-year university students in infant and primary education) in terms of the level of computational thinking based on four dimensions of study (simple, looping, conditional, and advanced conditional algorithms). The second aim was to analyse whether there are statistically significant differences between the level of computational thinking according to gender and the previous experience in robotic programming languages that pre-service teachers have. And the third aim was to identify the existence of profiles that allows us to analyse the differences between them with the dimensions of computational thinking in order to obtain a more solid understanding of how these factors may influence students' computational thinking.

2. Materials and Methods

2.1. Participants

A total of 164 infant and primary education pre-service teachers participated in the study; 145 of them were female (88.4%) and 19 were male. All of them were university students enrolled in the first year of infant and primary education degrees, which comprises a total of four years. The age of the participants was mostly (72.3%) between 18 and 21, followed by those aged between 22 and 25 (23.1%), and those between 26 and 29 years, with 4.6%. The choice of the sample followed a non-probabilistic paradigm, following criteria of convenience and accessibility [31].

2.2. Analysis Instrument

Research on the assessment of computational thinking agrees that there is no specific and accepted method [32]. Computational thinking is a skill that is related to others such as logic, creativity, or problem solving, and, consequently, its assessment could be performed in several different ways. Thus, the instrument Fairy Assessment in Alice has relied on both

assessing objects programmed by students and analysing their responses to predesigned programming questions to try to measure abstraction, conditional logic, and algorithmic thinking [33]. Also, the Computational Thinking Framework (CTF) is a framework for assessing both computational thinking learning materials and resources as well as student skills [34]. Both instruments focus on the secondary and high school stages. To address advanced vocational education, Lee et al. [35] designed a test consisting of a selection of 15 task problems extracted from the International Bebras Contest.

On the other hand, the Computational Thinking Test (CTT) is aimed to measure the computational thinking of the subjects [36]. It consists of twenty-eight questions with four response choices, only one of which is correct. The instrument does not require any previous programming experience. The questions show different scenarios in which logical-mathematical thinking skills must be applied in order to solve them correctly. The CTT has been translated into several languages and used by the international scientific community in multiple investigations [37,38].

Despite the available instruments, there is a notable absence of tests designed specifically for the university education stage that have undergone a complete validation process. Consequently, given the lack of standardized instruments for measuring computational thinking at the university level, the CTT, validated by Román-González [36], was used in terms of its content, reliability and criterion, and concurrent validity. The CIT has been used at the university level in several studies and has been extensively validated in the Spanish context, with high levels of concurrent validity [39–41]. The *Computer Science Teachers Association* (CSTA) proposes computational thinking among the curricular standards and considers three development levels [42]. At the third level, it equates high school and first year of university. Thus, the CSTA considers that high school and first year of university have the same level of curricular development in terms of computational thinking. Therefore, the use of the CTT instrument for the present study can be considered appropriate.

Although the instrument had been validated, an analysis of its psychometric properties was performed. Thus, the corrected item–total correlation (r_{i-t}) was positive for all items, indicating that they all contributed to measuring the general construct measured by the instrument and in the same direction. Cronbach's Alpha was adequate (0.705) with a mean item score of 0.76. The Kaiser–Meyer–Olkin measure of sampling adequacy was 0.599, and Barlett's test of sphericity was significant ($p < 0.001$), with a chi-square value of 0.599 and 231 degrees of freedom.

To achieve the stated objectives, three additional data were collected. Firstly, we determined the level of prior programming knowledge that students considered they had, using a Likert-type scale from 1 to 5 (where 1 equals Very Low and 5 equals Very High). Secondly, students were asked whether they had previous experience in robotics programming through a dichotomous question (Yes or No), in which they were asked if at any time they had used programming tools such as Scratch, Arduino, Lego, Alice, etc. Finally, the performance level of the university students was used as a reference measure. This level of performance was based on the grade obtained by the students in the subject of Information and Communication Technologies (ICT) applied to the education of the infant and primary education pre-service teachers, since several authors consider the grades obtained as a valid indicator to measure academic performance [43,44], in this case, in the field of technologies. The ICT subject evaluates both theoretical and practical knowledge through four types of activities: (1) exploration and analysis activities, aimed at promoting reflection on ICT through group dynamics; (2) evaluation and diagnosis activities, especially of digital and audiovisual materials; (3) design activities of digital and multimedia materials (short animation films, creation of digital games, etc.); and (4) activities of the didactic integration of ICT. With these activities, students put into practice problem-solving skills, planning, organization, creative and deductive abilities, etc. The final grade of the course is based on the achievement in the written test, in which theoretical and practical knowledge are evaluated, as well as on the results of the laboratory practices.

2.3. Procedure

The study was conducted in accordance with the World Medical Association Code of Ethics (Declaration of Helsinki), which reflects the ethical principles for research involving human subjects [45]. Prior to conducting the CTT, students were informed that their participation was voluntary. Participants were required to give informed consent prior to completing the questionnaire.

The students completed the CTT digitally, accessing it through the link sent to their corporate emails at the university before carrying out the training actions on robotics programming within the framework of the aforementioned subject. Therefore, an exploratory study was carried out in order to determine the initial state of the incoming students to the degrees in Infant Education and Primary Education in order to generate training actions that support their academic performance in the field of STEM competences.

Once the answers had been collected, those that were answered incorrectly were assigned a value of 0, and those that were correct were assigned a value of 1. Subsequently, the items were grouped according to the type of skills required for their correct resolution, giving rise to four dimensions, as follows: (1) simple algorithms (SA), made up of seven linear programming scenarios; (2) loop algorithms (LA), made up of seven practical scenarios where the programming sequence is repeated several times; (3) conditional algorithms (CA), made up of seven scenarios based on stipulating certain basic conditions for the programming blocks to be executed; and (4) advanced conditional algorithms (ACA), made up of seven scenarios and, like the previous one, focus on establishing advanced conditions for solving the problems posed. Finally, the scores in each of the four dimensions were weighted, obtaining scores from 0 to 10 in each of them.

2.4. Data Analysis

Firstly, to analyse the existence of differences in the four dimensions that make up computational thinking according to gender and students' previous experience in programming languages, *t*-tests for independent samples were carried out.

A two-stage cluster analysis was used to obtain the student profiles. This analysis provided an automatic procedure of the optimal number of clusters, allowing clusters with categorical and continuous variables [46]. The variables included in the cluster were "gender", "previous experience in robotic programming languages", "level of knowledge that students consider possessing about programming", and "level of performance", measured from 0 to 10 and calculated from the final grade obtained in the subject. The model complies with the assumptions of independence between its variables, the continuous variables follow a normal distribution, and the categorical qualitative variables follow a multinomial distribution. Even if these assumptions were not met, Rubio and Vilá [46] indicate that this procedure is robust enough to be applied. To analyse the distance between clusters, the log-likelihood measure was used. The Bayesian Schwartz Criterion (BIC) was used to determine the number of clusters. To minimise order effects, the cases were randomly ordered. The results yielded a satisfactory model consisting of three clusters.

Finally, once the students' profiles were obtained, a one-factor ANOVA with post hoc tests was used to analyse each of the four dimensions of computational thinking (simple algorithms, loop algorithms, conditional algorithms, and advanced conditional algorithms) in terms of these profiles.

The data obtained were analysed using SPSS 27, with a *p*-value < 0.05 as a reference for significance. Effect sizes were assessed using partial eta squared, considering that the effect size is small when $\eta_p^2 = 0.01$, medium when $\eta_p^2 = 0.059$, and large when $\eta_p^2 = 0.080$ [47].

3. Results

Table 1 presents the descriptive statistics of the variables as well as the Pearson correlation matrix. The values of skewness and kurtosis lie within the limits indicating a normal distribution (values of less than 3 for skewness and less than 10 for kurtosis), [48].

Table 1. Descriptive statistics and correlation (r) between the variables that make up computational thinking.

	SA	LA	CA	ACA
SA	-	-	-	-
BA	0.333 *	-	-	-
CA	0.329 *	0.374 *	-	-
ACA	0.205 *	0.461 *	0.413 *	-
Skewness	-1.618	-0.692	-0.188	-0.269
Kurtosis	2.242	0.163	-0.800	-0.698
Mean	9.268	8.049	6.171	5.988
Standard Deviation	1.170	1.647	2.400	2.771

Note: SA = simple algorithms, LA = loop algorithms, CA = conditional algorithms, and ACA = advanced conditional algorithms. * $p < 0.001$

Regarding the scores, the highest means are found for simple and loop algorithms, with questions on applying mathematical-logical knowledge of conditionals and advanced conditionals obtaining the lowest scores. Regarding correlations, the results showed statistically significant and positive correlations between the variables under study in all cases.

On the other hand, as shown in Table 2, only 21.3% of the study sample has previous experience in programming languages, considering that more than half of them have a low level in this type of knowledge. However, the average level of academic performance in the subject and the level of computational thinking skills derived from the CTT are of around 7.

Table 2. Descriptive statistics in relation to previous experience in programming languages, the level of programming that the students believe they possess, the students' level of performance in the subject, and the result in the CTT.

	YES	NO			
PE	21.3%	78.7%			
	Very Low	Low	Medium	High	Very High
PKL	38.4%	51.2%	10.4%	-	-
	MI	MA	M	SD	
PL	2.8	9.3	7.05	1.332	
CTT	3.6	10	7.36	1.471	

Note: PE = previous experience in programming languages. PKL = level of programming knowledge that students believe they possess. PL = students' level of performance in the subject. CTT = overall score obtained by the students in the Computational Thinking Test. MI = minimum. MA = maximum. M = mean. SD = standard deviation.

To determine the differences in the dimensions that make up computational thinking based on gender and students' previous experience in programming languages, t -tests were performed for independent samples. The results showed that the differences were statistically significant between the scores obtained in the simple algorithms dimension and that of gender ($t_{162} = -1.87$, $p < 0.001$, $n_p2 = 1.16$), with a large effect size. The results indicated that males ($M = 9.73$; $SD = 0.624$) scored higher than females ($M = 9.20$; $SD = 1.21$).

Similarly, statistically significant results are obtained between the scores obtained in the simple algorithm dimension ($t_{162} = 2.30$, $p < 0.001$, and $n_p2 = 1.15$) according to the students' previous experience in programming languages, with a large effect size. The results indicated that students with prior experience ($M = 9.66$; $SD = 0.97$) scored higher than those without ($M = 9.16$; $SD = 1.19$).

Significant results were also obtained in the loop algorithm dimension ($t_{162} = 2.34$, $p = 0.04$, and $n_p2 = 1.62$) according to students' prior experience in programming languages,

with a large effect size. The results indicated that students with prior experience ($M = 8.61$; $SD = 1.24$) scored higher than those without ($M = 7.89$; $SD = 1.71$).

In terms of obtaining student profiles from a cluster, three groupings were obtained with the four variables with a “good” result, indicating, in line with Kaufman and Rousseeuw [49], that the data reflect reasonable and strong evidence of a cluster structure. The variable that carries the most weight is the level that students believe to possess in programming (with a predictor significance of 1 out of 1), followed by previous experience in programming languages (with a significance of 0.59 out of 1), gender (with a significance of 0.29 out of 1), and lastly, level of performance in the subject (with a significance of 0.01 out of 1). The characteristics of the three clusters are as follows:

Cluster 1: A cluster made up of 31.7% of the sample, where 48.1% declare that they have a low level of knowledge of programming languages, 67.3% declare that they have previous experience in programming languages, 63.5% are women, and their average level of performance in the subject is 7.06. We will call this cluster “initiates in robotics programming” (IRP), and it is made up of those students who have had experience and have used programming at some point.

Cluster 2: A cluster made up of 32.3% of the sample. All of them (100%) declare that they have a very low level of knowledge of programming languages, they have no previous experience in programming languages, they are all women, and their average level of performance in the subject is 7.13. We will call this cluster “novices in robotics programming” (NRP), made up of those students who have had very little contact with programming.

Cluster 3: A group formed by 36% of the sample with the following characteristics: all (100%) have a low level of knowledge about programming languages, have no previous experience in programming languages, all are female, and their average level of performance in the subject is 6.98. We will call this cluster “explorers in robotic programming” (ERP), and it is composed of those students who have started to investigate programming languages.

To analyse the differences between the profiles obtained with the students’ knowledge in the dimensions that make up computational thinking, a one-factor ANOVA analysis was carried out using Welch’s procedure. There, statistically significant differences were found in “Simple algorithms” according to the cluster type ($F_{w 3, 726} = 3.92$, $p < 0.022$), with a mean effect size ($\eta_p^2 = 0.46$). This was also the case in “Loop algorithms” ($F_{w 3, 952} = 4.34$, $p < 0.015$), with a medium effect size ($\eta_p^2 = 0.51$).

Similarly, the overall level of computational thinking possessed by university students was analysed. Significant differences were obtained according to the type of cluster ($F_{w 3, 699} = 3.65$, $p < 0.028$), with a medium effect size ($\eta_p^2 = 0.43$).

Subsequently, the Bonferroni procedure was used to analyse the post hoc multiple comparisons, the results of which are shown in Table 3:

Table 3. Post hoc multiple comparisons of “Simple Algorithms” according to the profile type.

	<i>M</i>	<i>SD</i>	<i>CM</i>	<i>DM</i>	<i>p</i>	<i>IC 95%</i>
IRP	9.55	0.940	IRP-NRP	0.229	0.88	[−0.30, 0.75]
NRP	9.32	1.12	IRP-ERP	0.620	0.01 *	[0.07, 1.16]
ERP	8.93	1.35	NRP-ERP	0.391	0.22	[−0.13, 0.91]

Note: IRP = initiates in robotic programming, NRP = novices in robotic programming, and ERP = explorers in robotic programming. * $p < 0.001$.

The post hoc analysis regarding the results obtained by the university students in those questions that measure knowledge related to loop algorithms are shown in Table 4:

Table 4. Post hoc multiple comparisons of “Loop Algorithms” according to the profile type.

	<i>M</i>	<i>SD</i>	<i>CM</i>	<i>DM</i>	<i>p</i>	<i>IC 95%</i>
IRP	8.46	1.39	IRP-NRP	0.325	0.87	[−0.41, 1.06]
NRP	8.13	1.51	IRP-ERP	0.914	0.01 *	[0.15, 1.67]
ERP	7.54	1.89	NRP-ERP	0.588	0.16	[−0.15, 1.32]

Note: IRP = initiates in robotic programming, NRP = novices in robotic programming, and ERP = explorers in robotic programming. * $p < 0.001$.

Similarly, the post hoc analysis regarding the results obtained by the university students in the Computational Thinking Test according to the profiles defined in the cluster analysis are shown in Table 5:

Table 5. Post hoc multiple comparisons of “Level of computational thinking” according to the profile type.

	<i>M</i>	<i>SD</i>	<i>CM</i>	<i>DM</i>	<i>P</i>	<i>IC 95%</i>
IRP	7.80	1.41	IRP-NRP	0.538	0.15	[−0.12, 1.20]
NRP	7.26	1.45	IRP-ERP	0.738	0.03 *	[0.05, 1.42]
ERP	7.06	1.47	NRP-ERP	0.199	1.0	[−0.46, 0.86]

Note: IRP = initiate in robotic programming, NRP = novices in robotic programming, and ERP = explorers in robotic programming. * $p < 0.001$.

In all cases, the cluster “Initiates in robotic programming” (IRP), formed by those students who have had experience and have used programming at some point, obtained better scores than the profile of students called “Explorers in robotic programming” (ERP), formed by those students who have started to investigate programming languages and who state that they have some level of knowledge in programming languages. It is noteworthy that those who have no previous experience in programming languages perform less well in the CTT. This result confirms the need to address this type of knowledge in the training of future teachers to strengthen STEM competences.

4. Discussion and Conclusions

Computational thinking is characterised by a specific way of thinking that favours the analysis and relation of ideas for the organisation and logical representation of information. In recent years, there has been a growing need to develop it in education in order to respond to the needs of an increasingly technological society. We are thus dealing with the emergence of a new literacy that seeks to encourage the learning of programming in a progressive way by using basic principles of computer coding through the use of algorithms. Therefore, the general objective of this study is to analyse the level of computational thinking of university students in order to determine the initial state of incoming students to infant and primary education pre-service teachers. Specifically, three specific aims were defined: (1) characterise the participants in the sample in terms of the level of computational thinking; (2) to study the existence of differences in the level of computational thinking according to gender and previous experience of university students in robotic programming; and (3) to identify the existence of profiles that allow us to analyse the differences between these profiles with the dimensions of computational thinking. Regarding the first objective, pre-service teachers obtain higher scores in simple and loop algorithms, being the questions related to applying mathematical-logical knowledge about conditionals and advanced conditionals the ones that obtain lower scores. Therefore, and according to the results obtained, the study sample has a low/medium level of computational thinking with basic algorithmic notions presenting weaknesses in conditional logic. In this way, it is possible to identify which skill each student needs to reinforce learning. The data collected allow us to define the infant and primary education pre-service teachers in terms of their level of computational thinking, establishing which skills require a greater

focus from a first diagnosis in order to subsequently promote training proposals adapted to the needs of each student.

In relation to the second objective, it was possible to confirm the existence of differences according to gender, in line with previous studies collected in the review conducted by Kiliç [50]. In the present study, men obtaining better scores in the simple algorithms dimension. These results are in line with those obtained by other studies [15,51], among others, that found that men showed greater conceptual understanding, using more advanced strategies and algorithms than women in robotic programming processes. And some studies have even found that, despite having a high level of thinking, women lack interest in programming [52], indicating that they may lack the necessary motivation. It is therefore essential to pay attention to these differences when carrying out teaching-learning processes, since knowing the interests of students can develop activities that increase their motivation and have an impact on improving academic results. In this sense, as Kanny et al. [53] point out, by promoting gender equality, challenging stereotypes, and providing equal opportunities, we can encourage the participation of women, harnessing the talent and creativity of all individuals, regardless of their gender.

Similarly, differences were found according to students' prior experience with robotic programming languages, with those who had prior experience performing better on the simple and loop algorithms dimensions. Sun et al. [26] found that those students who had learned to program at an early age had a higher level of computational thinking. Therefore, there is a strong need for appropriate teaching strategies at different stages of learning. Moreover, computational thinking and programming are closely linked, and teaching them together can provide students with powerful tools to understand and address the challenges of the digital age.

Along the same lines, the third objective sought to identify the existence of profiles and to analyse the differences between them in terms of the dimensions of computational thinking. The results of the study reflect the existence of three student profiles. In a cluster called IRP, where 48.1% of the students declared to have a low level of knowledge of programming languages, 67.3% had previous experience in programming languages, 63.5% were women, and their average level of performance in the subject was 7.06. This cluster has the highest levels of computational thinking. The second cluster called NRP is characterised by the fact that all students report a very low level of knowledge of programming languages and have no previous experience in programming languages; they are female, and their average level of performance in the subject is 7.13. This cluster has an average level of computational thinking. And, finally, the so-called ERP, characterised by those students who have a low level of knowledge of programming languages, have no previous experience in programming languages, are female, and their average level of performance in the subject is 6.98. This group has the lowest level of computational thinking.

These results highlight the complexity of the interaction between the variables analysed. However, we found a group of students, mostly women with previous experience in programming languages, who obtained high levels of computational thinking, especially in the dimensions of simple algorithms and loops. This result supports the idea proposed by Kelleher and Pausch [54] that scaffolding-based instructional design is at the heart of programming instruction in training planning. In this line, the design of training actions that take into account students' starting profiles may be the key to developing computational thinking, cultivating talent among trainee teachers, and enhancing STEM skills. Thus, in order to develop computational thinking, it is necessary to know and understand the mechanisms involved in its implementation in a simple and structured way, understanding how problems are solved through logical systems. The results obtained show the need to offer educational proposals that contribute to strengthen the teaching-learning processes through didactic resources and materials that provide dynamism, flexibility, and innovation. Thus, it is proposed to adapt to the training program of future teachers, the TangibleK programme [55], which uses robotic programming through algorithms as a tool to stimulate computational thinking among university students.

In this context, we propose to adapt and integrate in the training of future teachers the basic principles defined by Bers [55]: (1) making visible the design process and the computational thinking required to program the robot—in other words, posing a problem, researching, planning, developing a prototype, testing, redesigning, and analysing the solutions; (2) enhancing collaboration by creating teams that share resources and materials; (3) generating technology conversation groups to share the robotic programming projects developed by the different groups of students, which provides the opportunity to correct mistakes as a community; and (4) transferring the projects beyond the classroom to make the learning visible, testing it through demonstrations, exhibitions, etc., or applying it in classrooms, by making it visible in the classroom, or by applying it in infant and/or primary classrooms as future teachers. In this way, through a simple sequencing, focusing on identifying the starting knowledge, introducing the project through games, proposing a programming task, sharing it through the technology conversation groups, and evaluating the task, future teachers are provided with the tools and skills necessary to foster computational thinking in the classroom.

Undoubtedly, studies on computational thinking are developing and deserve more research, especially among trainee teachers, as they are the basis for enhancing computational thinking in infant and primary school classrooms. Although this research has provided some important results on the levels of computational thinking among university students and the variables that may condition these levels, some limitations must be taken into account. Firstly, the size of the sample and having opted for a cross-sectional design. Future research can dynamically track the evolution of students' computational thinking by using longitudinal data and expanding the study sample to make more robust causal inferences. Secondly, the nature of the sample belongs to two very specific contexts. In order to verify better results, the study is to be directed towards students from other education-related degrees. Finally, this study has mainly used quantitative data for analysis, and future research may add qualitative data such as interviews to complement the research results.

Author Contributions: Conceptualization, L.V. and M.C.; methodology, formal analysis, investigation, data curation and writing—original draft preparation, L.V.; writing—review and editing, M.C.; supervision, L.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study was conducted in accordance with the guidelines of the Ethics Committee of the Principality of Asturias and the Ethics Committee of the University of Oviedo (reference: 1_RRI_2023). All procedures complied with relevant laws and institutional guidelines.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: All the research data are reported in the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tsarava, K.; Moeller, K.; Román-González, M.; Golle, J.; Leifheit, L.; Butz, M.V.; Ninaus, M. A cognitive definition of computational thinking in primary education. *Comput. Educ.* **2022**, *179*, 104425. [[CrossRef](#)]
2. Wing, J.M. Computational thinking. *Commun. ACM* **2006**, *49*, 33–35. [[CrossRef](#)]
3. Papert, S. *Mindstorms: Children, Computers, and Powerful Ideas*; Basic Books: New York, NY, USA, 1980; ISBN 978-0465046744.
4. Wang, C.; Shen, J.; Chao, J. Integrating computational thinking in STEM education: A literature review. *Int. J. Sci. Math. Educ.* **2022**, *20*, 1949–1972. [[CrossRef](#)]
5. Bers, M.U.; Flannery, L.; Kazakoff, E.R.; Sullivan, A. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Comput. Educ.* **2014**, *72*, 145–157. [[CrossRef](#)]
6. Selby, C. Promoting computational thinking with programming. In Proceedings of the 7th Workshop in Primary and Secondary Computing Education, Hamburg, Germany, 8–9 November 2012; ACM: New York, NY, USA, 2012; pp. 74–77. [[CrossRef](#)]
7. Hurt, T.; Greenwald, E.; Allan, S.; Cannady, M.A.; Krakowski, A.; Brodsky, L.; Collins, M.A.; Montgomery, R.; Dorph, R. The computational thinking for science (CT-S) framework: Operationalizing CT-S for K–12 science education researchers and educators. *Int. J. STEM Educ.* **2023**, *10*, 1. [[CrossRef](#)]

8. Majeed, B.; Jawad, L.; ALRikabi, H. Computational Thinking (CT) Among University Students. *Int. J. Interact. Mob. Technol.* **2022**, *16*, 244–252. [CrossRef]
9. Tikva, C.; Tambouris, E. Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Comput. Educ.* **2021**, *162*, 104083. [CrossRef]
10. Berland, M.; Wilensky, U. Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *J. Sci. Educ. Technol.* **2015**, *24*, 628–647. [CrossRef]
11. Lu, C.; Macdonald, R.; Odell, B.; Kokhan, V.; Demmans, C.; Cutumisu, M. A scoping review of computational thinking assessments in higher education. *J. Comput. High. Educ.* **2022**, *34*, 416–461. [CrossRef]
12. Israel, M.; Pearson, J.; Tapia, T.; Wherfel, Q.; Reese, G. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Comput. Educ.* **2015**, *82*, 263–279. [CrossRef]
13. Popat, S.; Starkey, L. Learning to code or coding to learn? A systematic review. *Comput. Educ.* **2019**, *128*, 365–376. [CrossRef]
14. Su, J.; Yang, W. A systematic review of integrating computational thinking in early childhood education. *Comput. Educ.* **2023**, *4*, 100122. [CrossRef]
15. Angeli, C.; Valanides, N. Developing young children’s computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Comput. hum. Behav.* **2020**, *105*, 105954. [CrossRef]
16. Chan, R. A social cognitive perspective on gender disparities in self-efficacy, interest, and aspirations in science, technology, engineering, and mathematics (STEM): The influence of cultural and gender norms. *Int. J. STEM Educ.* **2022**, *9*, 37. [CrossRef]
17. Charlesworth, T.; Banaji, M. Gender in science, technology, engineering, and mathematics: Issues, causes, solutions. *J. Neurosci.* **2019**, *39*, 7228–7243. [CrossRef] [PubMed]
18. Wang, M.; Degol, J. Gender gap in science, technology, engineering, and mathematics (STEM): Current knowledge, implications for practice, policy, and future directions. *Educ. Psychol. Rev.* **2017**, *29*, 119–140. [CrossRef]
19. Sneider, C.; Stephenson, C.; Schafer, B.; Flick, L. Computational thinking in high school science classrooms. *Sci. Teach.* **2014**, *81*, 53–59. [CrossRef]
20. Acevedo-Borrega, J.; Valverde-Berrocoso, J.; Garrido-Arroyo, M. Computational thinking and educational technology: A scoping review of the literature. *Educ. Sci.* **2022**, *12*, 39. [CrossRef]
21. Repenning, A.; Webb, D.; Ioannidou, A. Scalable game design and the development of a checklist for getting computational thinking into public schools. In Proceedings of the 41st ACM Technical Symposium on Computer Science Education, Milwaukee, WI, USA, 10–13 March 2010; ISBN 978-1-60558-885-8.
22. Roig-Vila, R.; Moreno-Isac, V. El pensamiento computacional en educación. Análisis bibliométrico y temático. *RED Rev. Educ. Distanc.* **2020**, *63*, 1–24. [CrossRef]
23. INTEF, Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado. *Escuela de Pensamiento Computacional e Inteligencia Artificial*; Ministerio de Educación y Formación Profesional: Madrid, Spain, 2018; Available online: <https://code.intef.es/epcia/> (accessed on 11 October 2023).
24. Caballero-González, Y.A.; y García-Valcárcel, A. ¿Aprender con robótica en Educación Primaria? Un medio de estimular el pensamiento computacional. *Educ. Knowl. Soc.* **2019**, *20*, 15. [CrossRef]
25. Ung, L.; Labadin, J.; Mohamad, F. Computational thinking for teachers: Development of a localised E-learning system. *Comput. Educ.* **2022**, *177*, 104379. [CrossRef]
26. Sun, L.; Hu, L.; Zhou, D. Programming attitudes predict computational thinking: Analysis of differences in gender and programming experience. *Comput. Educ.* **2022**, *181*, 104457. [CrossRef]
27. Atmatzidou, S.; Demetriadis, S. Advancing students’ computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robot. Auton. Syst.* **2016**, *75*, 661–670. [CrossRef]
28. Bati, K. A systematic literature review regarding computational thinking and programming in early childhood education. *Educ. Inf. Technol.* **2022**, *27*, 2059–2082. [CrossRef]
29. Kanaki, K.; Kalogiannakis, M. Assessing Algorithmic Thinking Skills in Relation to Gender in Early Childhood. *Educ. Process Int. J.* **2022**, *11*, 44–59. [CrossRef]
30. Master, A.; Cheryan, S.; Moscatelli, A.; Meltzoff, A. Programming experience promotes higher STEM motivation among first-grade girls. *J. Exp. Child Psychol.* **2017**, *160*, 92–106. [CrossRef]
31. Etikan, I.; Musa, S.A.; Alkassim, R.S. Comparison of convenience sampling and purposive sampling. *Am. J. Theor. Appl. Stat.* **2016**, *5*, 1–4. [CrossRef]
32. Sun, L.; Hu, L.; Zhou, D.; Yang, W. Evaluation and developmental suggestions on undergraduates’ computational thinking: A theoretical framework guided by Marzano’s new taxonomy. *Interact. Learn. Envir.* **2022**, *1*, 1–23. [CrossRef]
33. Werner, L.; Denner, J.; Campe, S.; Kawamoto, D. The fairy performance assessment: Measuring computational thinking in middle school. In Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, Raleigh, NC, USA, 29 February–3 March 2012; pp. 215–220. [CrossRef]
34. Gouws, L.; Bradshaw, K.; Wentworth, P. First year student performance in a test for computational thinking. In Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference, East London, South Africa, 7–9 October 2013; pp. 271–277. [CrossRef]

35. Lee, G.; Lin, Y.; Lin, J. Assessment of computational thinking skill among high school and vocational school students in Taiwan. In Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications, Tampere, Finland, 23–27 June 2014; Volume 1, pp. 173–180.
36. Román-González, M. Computational Thinking Test: Design Guidelines and Content Validation. In Proceedings of the International Conference on Education and New Learning Technologies EDULEARN, Barcelona, Spain, 6–8 July 2015. [CrossRef]
37. Chan, S.; Looi, C.; Sumintono, B. Assessing computational thinking abilities among Singapore secondary students: A rasch model measurement analysis. *J. Comput. Educ.* **2021**, *8*, 213–236. [CrossRef]
38. Guggemos, J.; Seufert, S.; Román-González, M. Computational Thinking Assessment—Towards More Vivid Interpretations. *Technol. Knowl. Learn.* **2022**, *28*, 539–568. [CrossRef]
39. Cachero, C.; Meliá, S.; Barra, P. Pensamiento computacional y género: Un estudio observacional. In Proceedings of the Actas de las XXV Jornadas sobre la Enseñanza Universitaria de la Informática, Murcia, Spain, 3–5 July 2019; Asociación de Enseñantes Universitarios de la Informática: Murcia, Spain, 2019; pp. 263–270.
40. Sánchez Román, G.; Guerrero, J.; Martínez Mirón, E. Perfil del alumno de computación para el diseño de un sistema tutor. *CertiUni J.* **2019**, *5*, 19–26.
41. Viale, P.; Deco, C. Introduciendo conocimientos sobre el pensamiento computacional en los primeros años de las carreras de ciencia, tecnología, ingeniería y matemáticas. *Energeia* **2019**, *16*, 73–78. Available online: <https://repositorio.uca.edu.ar/handle/123456789/9824> (accessed on 11 October 2023).
42. Seehorn, D.; Carey, S.; Fuschetto, B.; Lee, I.; Moix, D.; O’Grady-Cunniff, D.; Verno, A. *CSTA K—12 Computer Science Standards: Revised 2011*; ACM: Danvers, MA, USA, 2011; ISBN 978-1-4503-0881-6.
43. Álvaro Paje, M. *Hacia un Modelo Causal del Rendimiento Académico*; CIDE: Madrid, Spain, 1990.
44. Ben, A.; Dahmani, M.; Ragni, L. ICT use, digital skills and students’ academic performance: Exploring the digital divide. *Information* **2022**, *13*, 129. [CrossRef]
45. Morris, K. Revising the Declaration of Helsinki. *World Rep.* **2013**, *381*, 1889–1890. [CrossRef]
46. Rubio, M.J.; Vilà, R. El análisis de conglomerados bietápico o en dos fases con SPSS. *REIRE Revista d’Innovació i Recerca en Educació* **2017**, *10*, 118–126. [CrossRef]
47. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed.; Lawrence Erlbaum Associates, Publishers: Mahwah, NJ, USA, 1988; ISBN 0-8058-0283-5.
48. Kline, R.B. *Principles and Practice of Structural Equation Modeling*; Guilford Press: New York, NY, USA, 2011; ISBN 9781462551910.
49. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons: Hoboken, NY, USA, 2009; ISBN 9780471878766.
50. Kiliç, S. Tendencias towards computational thinking: A content analysis study. *Particip. Educ. Res.* **2022**, *9*, 288–304. [CrossRef]
51. Fennema, E.; Carpenter, T.; Jacobs, V. A longitudinal study of gender differences in young children’s mathematical thinking. *Educ. Researcher.* **2016**, *27*, 6–11. [CrossRef]
52. Zhong, B.; Wang, Q.; Chen, J.; Li, Y. An exploration of three-dimensional integrated assessment for computational thinking. *J. Educ. Comput. Res.* **2016**, *53*, 562–590. [CrossRef]
53. Kanny, M.; Sax, L.; Riggers-Piehl, T. Investigating forty years of STEM research: How explanations for the gender gap have evolved over time. *J. Women Minor. Sci. Eng.* **2014**, *20*, 127–148. [CrossRef]
54. Kelleher, C.; Pausch, R. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Comput. Surv.* **2005**, *37*, 83–137. [CrossRef]
55. Bers, M. The TangibleK robotics program: Applied computational thinking for young children. *Early Child. Res. Pract.* **2010**, *12*, n2. Available online: <https://files.eric.ed.gov/fulltext/EJ910910.pdf> (accessed on 11 October 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.