Computers and Education Open 10 (2026) 100366

Contents lists available at ScienceDirect

Computers and Education Open

journal homepage: www.sciencedirect.com/journal/computers-and-education-open

Coding, robots, computational concepts, and machine learning using the microbit card and the Maqueen and Nezha kits. A study in initial teacher training

José-Manuel Sáez-López [a,*], Arturo-Simon García-Jiménez [b], Sebastián de Lara García-Cervigón [c]

[a] Universidad Nacional de Educación a Distancia (UNED), Spain
[b] Universidad de Castilla-La Mancha, Spain
[c] Junta de Comunidades de Castilla-La Mancha Spain

ARTICLE INFO

Keywords:
Educational technology
Higher education
Machine learning
Programming
Robotics

ABSTRACT

This research assesses the design and implementation of innovative educational robotics activities in higher education through 105 subjects in the Primary Education degree from two Spanish universities. The activities involved simple activities, visual block programming, machine learning, and the construction of sensors and actuators, leading to the learning of computational concepts and the implementation of interesting, practical tasks. The experiment included a control group of 39 students. A test (Coding, Robotics, and Machine Learning Test CRMT) and three scales were used to generate descriptive data and statistical inference through a quasi-experimental design with a student t-test in the first dimension and a Mann-Whitey U test in the second. Students in the experimental group improved their performance and results with the practical intervention of the study. It is concluded that there are training gaps in the use of programming in higher education, and that coding is essential in initial teacher training, with a greater presence of robotics programming needed in the training. Significant improvements are noted in active learning, creativity, the development of computational concepts, and machine learning, and improvements in effective implementation in mathematics and art. Ultimately, it is feasible to implement these pedagogical designs in the training of future primary school teachers, fostering an understanding of machine learning and the use of coding and robotics in the university training of future teachers.

1. Introduction

Several authors, including one of Papert's disciples, Mitchel Resnick [1], emphasize that computer programming is an extension of writing, so there are compelling reasons for everyone to learn to program. Resnick was responsible for leading the creation of the Scratch block-based programming language at the Massachusetts Institute of Technology (MIT), which underlies many of the robots and programmable boards currently available for educational use.

This way of working, in which students acquire technological skills and are able to create and manage their projects, assumes a constructionist and constructivist framework with effective activation of prior knowledge, promoting metacognition through greater information elaboration, greater understanding, and placing learning in a real-life context [2].

Visual block-based programming environments offer a considerable advantage over languages intended to provide novices with their first introduction to computing, allowing users to 'write' by dragging and dropping graphic blocks to compose simple programs that, in turn, allow them to create games, interactive stories, or simulations [3–5]. In elementary education, they can be easily used in different curricular areas, creating all kinds of multidisciplinary activities and projects with a shallow learning curve that allows increasingly complex projects.

There is evidence highlighting the effectiveness and benefits of implementing visual block-based programming in areas encompassed in the current curriculum [6] and in interactions that support reflection [7,

* Corresponding author.
E-mail addresses: jmsaezlopez@edu.uned.es (J.-M. Sáez-López), ArturoSimon.Garcia@uclm.es (A.-S. García-Jiménez), sebasdelara@gmail.com (S.L. García-Cervigón).

https://doi.org/10.1016/j.caeo.2026.100366
Received 31 July 2025; Received in revised form 13 April 2026; Accepted 15 April 2026
Available online 16 April 2026

8]. Several studies have found positive results related to attitudes toward computer science and computer science [9,10].

Several studies and authors [11–15] highlight advantages in educational processes in elementary education, with statistically significant improvements in the understanding of computational concepts and practices. These projects are especially interesting in Art Education, Natural Sciences, and Social Sciences, allowing the creation of colorful, dynamic, and motivating projects from an active perspective.

Comparing the perspective of the NMC/CoSN Horizon report: 2016 K-12 edition of the Horizon K12 report [16] with the current 2025 EDUCAUSE Horizon report teaching and learning edition [17], there is an increasing emphasis on the cognitive processes of student learning. It is observed that technology is changing student cognition, that students who work with projects are more independent, manage different information resources, become more adaptable to change, and are better able to work as team members [2]. In addition, technology is favoring new ways of documenting student learning and success [18,17].

While these studies demonstrate the benefits of programming and educational robotics for the development of computational thinking in primary and secondary education, most of them focus primarily on learning outcomes in school students rather than on the preparation of future teachers. Consequently, relatively little empirical research has examined how preservice teachers can acquire the necessary computational knowledge, pedagogical strategies, and technological confidence to integrate these tools effectively into their future classrooms. In addition, existing studies rarely incorporate emerging topics such as introductory machine learning within teacher education contexts, despite their growing relevance in contemporary digital literacy frameworks. Therefore, there remains a need for empirical research examining integrated training approaches that combine visual block programming, educational robotics, and introductory machine learning within initial teacher education programs

Today, there is a wide variety of programmable boards. These are electronic devices that can be programmed to perform different tasks according to the instructions we provide. They are like small computers that, depending on their programming, can control lights, sensors, motors, screens, interactive elements, and many other components. The benefits of these are multiple, such as the development of computational thinking [19], fostering creativity [3,5] and innovation [20], improving problem-solving [6], promoting active and meaningful learning [8], and encouraging collaborative work [21], among others.

This intervention uses the Microbit environment's block-based programming with a simplicity that allows successful pedagogy for students. It is then progressively linked and sequenced with activities related to robotics kits (Maqueen and Nezha), installing their plugins and making these devices work with visual block programming (https://makecode.microbit.org/).

It is important to articulate more explicitly the specific training gaps that motivate the study. These appear to include, for instance, the limited prior exposure of preservice teachers to visual block programming environments; insufficient conceptual understanding of fundamental computational thinking constructs such as sequencing, loops, and conditional logic; a lack of familiarity with basic robotics hardware (sensors, actuators, and programmable boards); and minimal experience integrating programming and robotics into curricular areas beyond technology-specific subjects. Additionally, the low level of awareness and practical understanding of emerging topics such as machine learning suggests a further gap between current teacher preparation and contemporary educational demands. Explicitly framing these deficiencies as the starting point of the research would strengthen the rationale of the intervention and allow the objectives and outcome measures to be more clearly positioned as responses to clearly defined educational needs.

In this study, these theoretical frameworks directly inform both the design of the intervention and the formulation of the research objectives. Meaningful learning and scaffolding principles guided the sequencing of activities and the progressive introduction of computational concepts, while constructionist theory shaped the emphasis on hands-on creation, problem solving, and reflective practice. Accordingly, the selected outcome variables focus on preservice teachers' understanding of core computational concepts, their attitudes toward programming and machine learning, and their perceived capacity to integrate these approaches into educational practice.

Although prior research has demonstrated the educational potential of visual block programming and robotics in school contexts, fewer studies have systematically examined their integration within initial teacher training, particularly from a constructionist perspective and with the inclusion of introductory machine learning concepts. Existing studies tend to focus either on computational thinking development in compulsory education [12,18] or on programming skills in preservice teachers, often without addressing how emerging topics such as machine learning can be meaningfully incorporated into teacher education curricula [22,23]. This study addresses this gap by examining the impact of an integrated robotics and programming intervention on preservice primary teachers' computational understanding and attitudes, thereby directly informing the objectives formulated below.

1.1. Objectives

Main objective is to evaluate the impact of a constructionist-based educational robotics intervention on preservice primary teachers' acquisition of computational concepts and attitudes toward programming and machine learning.

The specific objectives are:

- To measure changes in preservice teachers' knowledge of core computational concepts (sequence, loops, conditionals, and sensors) as assessed by the CRMT.
- To determine the effectiveness of visual block programming in supportingzrobotics learning, as evidenced by post-intervention performance and self-reported skill development.
- To assess changes in attitudes toward the integration of programming and machine learning in primary education curricula using validated Likert-scale instruments.
- To examine preservice teachers' perceptions regarding the feasibility and educational usefulness of integrating robotics and block-based programming activities into primary education curricula, based on attitudinal scale responses.
- To analyze perceived impacts of the intervention on active learning, creativity, and interdisciplinary curricular application.

1.2. Research questions

RQ1 (Knowledge outcomes). To what extent does a constructionist-based educational robotics intervention influence preservice primary teachers' understanding of core computational concepts and introductory machine learning, as measured by the CRMT?

RQ2 (Attitudinal outcomes). What differences are observed between preservice teachers who participate in the educational robotics intervention and those who do not with respect to attitudes toward programming, educational robotics, and the integration of machine learning in primary education?

2. Materials and methods

2.1. Procedure and dimensions of the study

The study was carried out in two Spanish universities as part of the training of future primary education teachers. A quasi-experimental design was implemented with an experimental group and a control group. The overall structure of the study was organized into two main dimensions (Table 1), each addressing a specific aspect of learning and

Table 1
Variables and instruments used in each dimension of the study.

Dimensions	Variables	Instruments and Statistical Analyses
1. Knowledge of computational concepts, robotics and machine learning	Machine learning Sequence Iteration (looping) Conditional statements Sensors	Coding Robotics and Machine Learning Test (CRMT) Descriptive analysis Student t-test
2. Attitudes regarding block programming and educational robotics in teacher training	Programming mastery Visual block programming and robotics Computational concepts Application in curricular areas	Scale (attitude measurement, Table 3) Descriptive analysis Mann-Whitney U Test

perception related to educational robotics and computational thinking.

The first dimension focused on students' knowledge and understanding of computational concepts, educational robotics, and machine learning. This was assessed using a custom-designed questionnaire test, detailed in the appendix, consisting of ten items (CRMT).

The second dimension examined students' attitudes and perceptions toward the use of educational robotics and block-based programming in teacher training. To this end, three Likert-type scales were used, administered after the intervention (Table 3). The first scale focused on attitudes toward programming and machine learning, including items related to perceived usefulness, interest, and confidence in using these technologies. The second scale examined perceptions of the pedagogical value of robotics and programming activities in teacher education, addressing aspects such as educational innovation and curricular integration. The third scale measured perceptions related to active learning and creativity, including items that explored the potential of robotics-based activities to promote student engagement and creative problem-solving.

The intervention consisted of a series of educational robotics activities integrated into the coursework of the experimental group. These included visual block-based programming, the use of Unity with Machine Learning Agents (ML-Agents), and the development of simple robotic systems using sensors and actuators. The control group followed the same theoretical content through lectures but did not participate in the robotics-based activities.

Students in the control group received instruction covering similar theoretical topics related to programming, educational robotics, and emerging technologies in education. However, this instruction was delivered through traditional lecture-based sessions and class discussions rather than through hands-on activities. The theoretical sessions addressed basic concepts of computational thinking, visual programming environments, and the potential educational applications of robotics and artificial intelligence in primary education.

2.2. Participants

The study population consisted of first-year undergraduate students enrolled in the Bachelor's degree in Primary Education at three Spanish public universities. The sample was non-probabilistic and purposive, based on students' enrollment in regular course groups. The experimental group included 105 students from the University of La Laguna and the University of Oviedo, while the control group consisted of 39 students from a public university in the Community of Madrid. Assignment to experimental and control groups was determined by existing class groupings rather than random allocation, in order to preserve the regular organization of teaching activities.

The experimental group consisted of 90.5% female students and 9.5% male students, reflecting the gender distribution commonly observed in teacher education programs, where female students represent a large majority. The average age of participants was 19.14 years, and the group was relatively homogeneous in terms of age and educational level.

Participants did not have a specialized background in computer science or engineering. Prior to the intervention, their exposure to technology-related content was mainly limited to general digital competence courses focusing on basic computer use, information search, and educational technologies. Most participants had little or no formal experience with programming, robotics, or computational thinking, and only a minority reported previous contact with visual block-based programming environments.

To examine group comparability prior to the intervention, pre-test CRMT scores were compared using an independent samples t-test. No statistically significant differences were found between the experimental and control groups, indicating baseline equivalence and supporting the internal validity of the study.

Participants were recruited using purposive sampling, based on their enrollment in first-year Primary Education degree programs at three Spanish public universities. Assignment to experimental and control groups was determined by existing course groupings rather than random allocation, in order to preserve regular teaching organization and avoid disruption to academic schedules. The two universities where the intervention was implemented constituted the experimental group ($n = 105$), while students from the third university formed the control group ($n = 39$), all following comparable curricular content in educational technology. To examine group comparability prior to the intervention, pre-test CRMT scores were compared between groups using an independent samples t-test, with no statistically significant differences observed, indicating equivalence at baseline and supporting internal validity.

Confidence intervals (CIs) were calculated and are reported for all primary outcome measures to provide greater precision in the interpretation of the observed effects. The inclusion of CIs allows for a clearer assessment of the reliability and practical significance of the estimated differences between groups and strengthens the transparency and reproducibility of the statistical analysis.

2.3. Implementation

The Coding and Educational Robotics sessions were held in April and May 2025 with undergraduate students in Primary Education at two Spanish public universities. The nine sessions focused on preparing a Maqueen robot, building a traffic light barrier using the Nezha kit, and using the Microbit board for programming (One-hour sessions). These tasks used visual block programming with Microbit Makecode, allowing students to prepare the robots and program the projects. We focused on problem-solving and planning to engage and motivate university students. Sequencing and scaffolding were implemented, starting with basic computational concepts (sequences, loops, conditionals, and events), using the devices and robots.

The intervention was implemented by instructors involved in the research team and experienced in educational technology and programming education within teacher training programs. In both participating universities, the sessions were delivered following the same instructional design developed for the study.

To ensure consistency across institutions, all instructors followed a shared teaching guide that specified the learning objectives, sequence of activities, programming tasks, and use of technological resources for each session. The same materials, robotics kits, and programming environments were used in both institutions. This common framework helped maintain instructional fidelity and ensured that students in the experimental groups experienced comparable teaching conditions throughout the intervention.

The intervention was implemented using a set of accessible educational technologies commonly employed in introductory programming and robotics contexts. The Micro:bit board served as the core programmable device, using the MakeCode visual programming environment, which allows learners to create programs through block-based interfaces without prior experience in text-based coding. This environment supports fundamental computational constructs such as sequencing, loops, conditionals, events, and sensor input.

For robotics activities, two kits were used. The Maqueen robot, designed for beginner-level educational robotics, was employed to introduce motor control, sensor-based decision making, and basic autonomous behavior through infrared and light sensors. The Nezha kit, based on modular construction components, enabled the development of more complex robotic systems involving multiple actuators, sensors, and parallel processes, supporting advanced block-based programming concepts.

Introductory machine learning concepts were addressed through the "Shy Panda" project, developed using Scratch 3.0 and the Machine Learning for Kids platform. This activity introduced supervised learning through image classification, guiding students through dataset creation, labeling, model training, and real-time testing. The focus was placed on conceptual understanding of the machine learning workflow rather than on algorithmic or mathematical complexity, ensuring accessibility for preservice teachers. Detailed descriptions of each instructional session and associated activities are provided in the Appendix to support replicability and instructional transfer.

During the practical sessions, students in the experimental group worked in small, stable teams in order to promote collaboration and peer learning. Each team consisted of three to four students, which allowed all participants to actively engage in programming, construction, and problem-solving tasks.

The first sessions began with simple projects using Microbit. A clear example is the dice using random numbers (see Fig. 1).

In this session, a model is trained using images and the possibilities of machine learning are demonstrated. Scratch 3.0 and Machine Learning for Kids are used (https://machinelearningforkids.co.uk/). The Shy Panda project was developed, a resource available on the website and an original idea by Cassie Evans (https://codepen.io/cassie-codes/pen/jKaVqo). In this project, you interact with a shy panda bear (Fig. 2). If the character does not feel watched, it will dance, but if it detects that you are watching it, it will become embarrassed and stop dancing.

Ultimately, in this session, university students can understand machine learning and how it is trained with numerous images in two categories, "I see you" (*te veo*) and "I don't see you" (*no te veo*), to recognize whether you are looking at the panda or covering your eyes. It is a simple and fun project. It allows the students to understand image detection and visual block programming, exploring computational concepts and practices (Fig. 2).

In later, more advanced sessions, we work with Microbit and the Maqueen kit in the Light Car project, through the light sensor, the motors, and programming with a conditional. The result is that if you point the mobile phone's flashlight at the car (a lot of light), the light sensor of the Microbit card obtains a high value and the car moves by activating both motors (Fig. 3).

Another key project in the sessions is the automatic traffic light barrier. The Nezha kit, which uses Lego Technic blocks, is used and programmed with a Microbit card. Consistent construction is important, as is detailing parallelisms, sensors, and actuators in the programming (Fig. 4).

All these projects foster work that enhances logical and analytical thinking and problem-solving and attempts to debug and correct errors (trial and error). Using devices, programming, and machine learning, students learn about how machines work and develop digital competence.

From an economic perspective, the implementation of the intervention required relatively low-cost and widely available educational resources. The approximate cost of a Micro:bit board is 20–25 EUR, while Maqueen robotic kits range from 50 to 70 EUR per unit, and Nezha robotics kits from approximately 80 to 200 EUR, depending on configuration and supplier.

2.4. Data analysis

The survey technique and questionnaire were proposed as an instrument consistent with the study objectives; the test and scales were adapted from published studies [24,13]. The data collected in this experiment were characterized using descriptive statistics, including measures of central tendency and dispersion, as well as percentages of responses in specific response ranges. Statistical analyses were conducted using IBM SPSS Statistics (version 30) and Microsoft Excel.

For the first dimension, understanding of computational concepts, robotics, and machine learning, the results of the CRMT are analyzed by applying the student's t-test. Scores from the experimental group were compared using t-test, statistical significance defined as $p < 0.05$. A priori power analysis was conducted using G*Power to determine the required sample size for detecting medium effect sizes (Cohen's $d = 0.5$), with a significance level of $\alpha = 0.05$ and a desired power of 0.80. An allocation ratio of 0.4 between groups was assumed, based on the expected composition of student groups participating in the study. The analysis indicated that a total of 128 participants ($n = 93$ experimental; $n = 35$ control) would be sufficient; this requirement was exceeded. Reliability calculated using Cronbach's alpha coefficient offers 0.81 in this first dimension, which is considered acceptable [25].

On the other hand, in the second dimension, block programming and

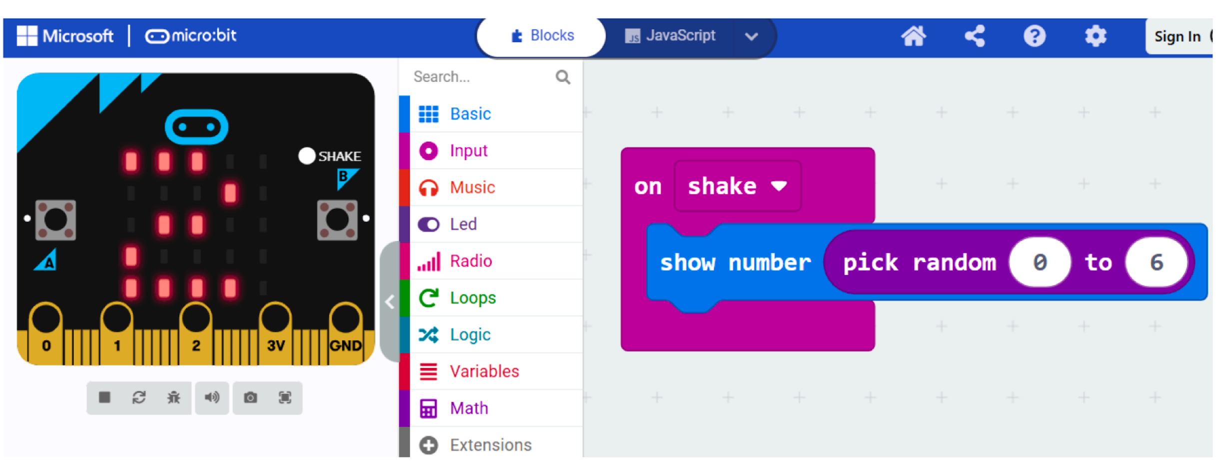

Fig. 1. Dice project programmed with Microbit using visual blocks.

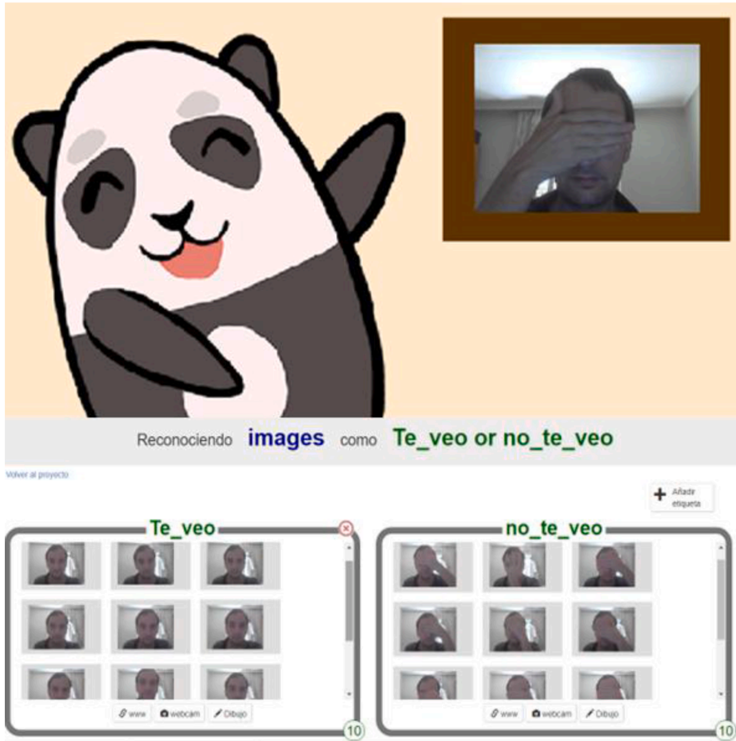

Fig. 2. Shy panda machine learning activity developed with Scratch 3.0 and Machine Learning for Kids.

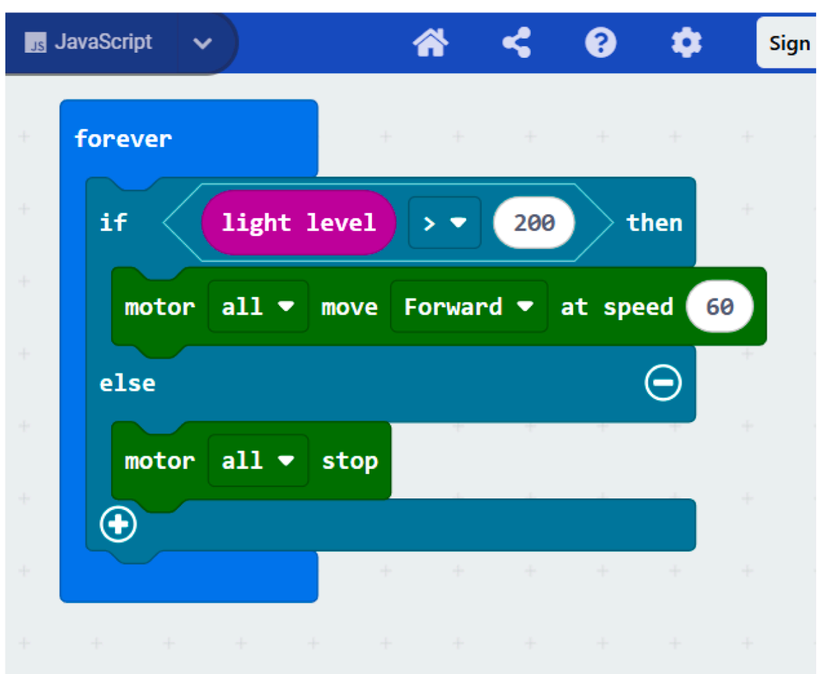

Fig. 3. Light Car robotics project programmed with microbit and the maqueen kit.

educational robotics in teacher training, which focused on attitudes and perceptions regarding educational robotics and programming, responses were collected using a four-point Likert scale ranging from "strongly disagree" (1) to "strongly agree" (4). Due to the ordinal nature of the data and the non-normal distribution, confirmed by the Shapiro-Wilk test, non-parametric analyses were performed. Specifically, the Mann-Whitney U test was used to compare responses between the experimental and control groups ($p < 0.05$).

This information allows us to determine whether there are significant differences between the control group (without any intervention) and the experimental group that underwent the intervention and practiced with educational robotics. A priori power analysis was conducted in G*Power to determine the required sample size to detect medium-sized effects (Cohen's $d = 0.5$), using a significance level of $\alpha = 0.05$ and a desired power of 0.80. An allocation ratio of 0.4 was assumed, in line with the expected group composition. Under these conditions, the

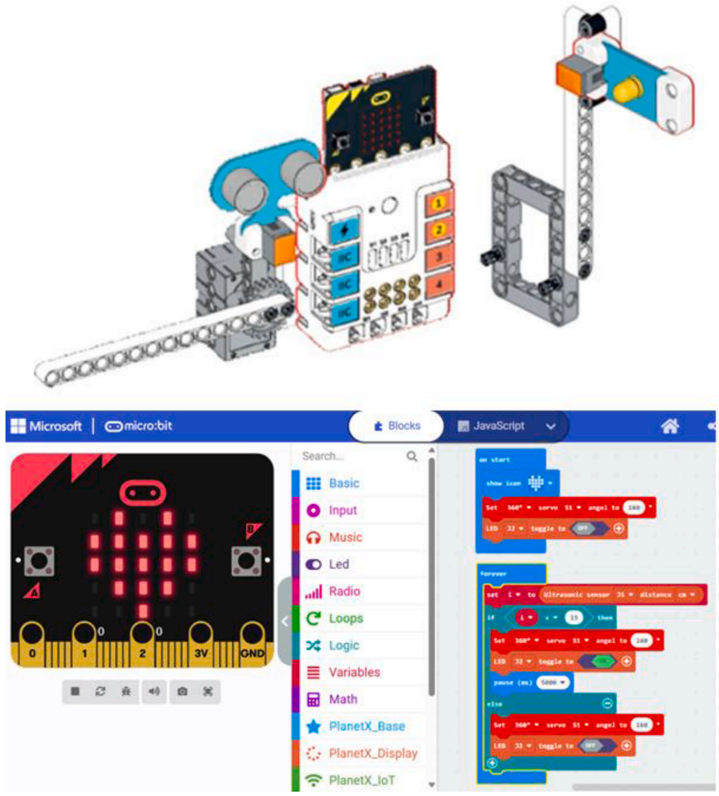

Fig. 4. Construction of the automatic barrier and traffic light project using the Nezha robotics kit.

required total sample size was estimated at 134 participants. Given that the actual sample size was 144 (n = 105 experimental; n = 39 control), the analysis was sufficiently powered. Reliability calculated using Cronbach's alpha coefficient yields an average of 0.911 for the scales in the second dimension, always higher than 0.6, and is therefore considered acceptable [25].

Different analyses and tests were used to determine whether there is evidence to support the validity of the results and minimize error variance [26]. Additionally, complementary post hoc power analyses were conducted for statistically significant comparisons in both dimensions. These analyses are reported in the results section to illustrate the sensitivity of the study and further support the interpretation of significant findings.

The CRMT was developed following a structured process aligned with the objectives and content of the intervention. Item generation was based on core computational concepts addressed during the training, namely sequencing, loops, conditional logic, sensors, and introductory machine learning, and was informed by assessment approaches. The survey technique was implemented using different instruments aligned with the two study dimensions: a knowledge test (CRMT) for the first dimension and three Likert-type scales for the second dimension, all adapted from previously published studies [24,13].

The CRMT consisted of 10 multiple-choice items designed to evaluate preservice teachers' understanding of core computational concepts addressed during the intervention. The items were distributed across five content areas: sequencing (2 items), loops/iteration (2 items), conditional logic (2 items), sensors and input-output interaction (2 items), and introductory machine learning concepts (2 items).

Each item included four response options with a single correct answer. Correct responses were scored as 1 point and incorrect responses as 0 points, resulting in a total possible score ranging from 0 to 10. Higher scores indicate a greater level of understanding of computational concepts related to programming, robotics, and introductory machine learning.

The selection of these constructs was guided by the learning objectives of the intervention and by established frameworks of computational thinking, particularly those proposed by Brennan and Resnick [6]. Sequencing, loops, and conditionals represent core programming structures, while sensor interaction reflects fundamental robotics functionality. The inclusion of introductory machine learning items aimed to assess participants' conceptual understanding of basic AI processes introduced during the intervention.

The CRMT instrument was developed by the research team responsible for the study, composed of researchers with experience in educational technology, computational thinking, and teacher education. The initial pool of items was designed based on the learning objectives of the intervention and on established frameworks of computational thinking. To ensure content validity, the preliminary version of the instrument was reviewed by a panel of seven experts in the fields of educational technology and teacher training.

The items were reviewed by experts in educational technology and initial teacher training to ensure content relevance, clarity, and alignment with the curricular level of preservice teachers. The instrument was administered within the study context, and its internal consistency demonstrated acceptable reliability (Cronbach's $\alpha = 0.81$), supporting its adequacy for evaluating preservice teachers' knowledge of computational concepts.

3. Results

3.1. Dimension 1: knowledge of computational concepts, robotics, and machine learning

In this dimension, a quasi-experimental design was applied, analyzing data using an independent samples Student's t-test. The Coding, Robotics, Machine, Test (CRMT test was administered, comparing the means of the 105 students in the experimental group who worked on the training implementation described above with a control group of 39 students from the Complutense University of Madrid, who were following a theoretical program related to ICT in education. This statistical contrast allows us to verify whether there were significant improvements between the control group and the experimental group to whom the intervention was applied (Table 2). The scores on the ten-item test emphasize the importance of an educational design that includes visual programming language and the implementation of educational robotics to promote the acquisition of computational concepts.

The control group of university students obtained an average score of 4.51. The values obtained in the experimental group provide a mean of 6.05 after implementation of the intervention. The Levene test for equality of variances is not significant ($P = .647$), so equal variances are assumed in both groups. Considering that equality of variances is assumed, the result of the administered Student's t-test provides a value ($t = 3.401$; $P = .001$) that allows us to affirm that there are significant improvements: the implemented program does improve the students' ability to understand visual programming and robotics. The effect size details a Cohen's d of 0.638, which is considered significant [27]. The Hedges correction and Glass's delta provide a value of 0.635, consistent with the results obtained.

3.2. Dimension 2: attitudes regarding block programming and educational robotics in teacher training

Unlike the first dimension, which followed a quasi-experimental pre-test/post-test design, the second dimension was based on a post-test-only comparison between the experimental and control groups. No baseline equivalence was established for attitudinal and perceptual variables prior to the intervention. The results for the three scales were highly positive, as reflected in the length of the blue and green bars in Figs. 5 and 6, as well as in the percentages presented in Table 3.

It is important to note that the results reported in Dimension 2 are based on participants' self-reported perceptions and attitudes rather than objective measures of learning outcomes. While these findings provide valuable insight into how preservice teachers perceive the educational value of robotics-based activities, they should be interpreted as indicators of perceived benefits rather than direct evidence of learning gains.

In the analysis of Scale 1 Machine Learning and Programming, coding knowledge is considered essential for training future teachers (item 1.1), with a positive rating (agree or strongly agree) from 80 percent of the sample. However, there are no significant improvements in this variable, as the control group also rates this item very highly. Furthermore, 61.9 percent claim to know what machine learning is (item 1.2), and 83.8 percent state that it is important to teach this concept in higher education (item 1.3). Meanwhile, 90.4 percent of students state that coding and robotics enhance the development of creativity (item 1.4), which represents the highest descriptive rating on the scale. Moreover, 46.7 percent of the subjects have worked with visual block programming at university (item 1.5), which is interpreted as an interesting value and should be worked on for improvement in higher education settings (Table 3).

Regarding statistical inference on this scale, a statistically significant improvement is observed using the Mann-Whitney U test, with a significance level of $\alpha = 0.05$ in items 1.2 and 1.4. Therefore, because of this research intervention, subjects improved their understanding of machine learning (item 1.2) and their understanding that programming and robotics stimulate creativity (item 1.4).

Regarding scale 2, computational concepts, approximately 80 percent of the subjects in the experimental group know what a sequence, a loop, and a conditional are (items 2.1, 2.2, and 2.3), demonstrating some understanding of basic computational concepts. Furthermore, statistically significant improvements were obtained in these three items, indicating improvement and learning compared to the control group (Table 3).

Furthermore, 89.5 percent of participants considered that the programming and robotics approach fosters educational innovation processes (item 2.4), a high descriptive value, although no statistically significant improvements were obtained. Therefore, it is inferred that the control group also considers the promotion of coding in educational innovation to be important and positively values it.

Regarding scale 3, active learning in curriculum areas, the subjects believe that a great deal of content is learned with visual block programming and robotics (item 3.1). Content can be implemented in Primary Education through visual block programming (item 3.2). Positive ratings (agree and strongly agree) were obtained from around 90 percent of subjects. Therefore, visual block coding and programming are highly valued in educational contexts. Furthermore, a statistically significant improvement was observed in both items compared to the control group, indicating that the intervention has a significant impact on the process.

Analyzing the results in the curriculum areas in depth, 93.3 percent of subjects believe that programming and robotics enhance mathematical learning (item 3.3) and that they help with learning art and music (items 3.6), with a statistically significant improvement in both items compared to the control group. Regarding natural sciences (item 3.4) and social sciences (item 3.5), descriptive values above 80 percent were obtained, but they did not show statistically significant improvements compared to the control group.

Most subjects (92.4%) believed that the coding approach made the topic studied more interesting (item 3.7), and 90.4 percent believed that programming actively engaged students (item 3.8). A statistically significant improvement was observed in both items, with an α significance level of 0.05, when the program and detailed intervention were implemented.

To calculate the effect size of the Mann-Whitney U test for the three scales, the Z statistic obtained was used and applied to the formula $r = Z / \sqrt{N}$, where N is the total sample size. Item 2.2, regarding understanding a loop in programming, obtained a high effect size (−0.847), according to Cohen's parameters. Item 3.8, which refers to the fact that programming actively engages students, obtains a high value (−0.673). Item 2.3, which refers to conditionals (−0.659), and item 3.3, which refers to coding being positive in mathematics (−0.631), obtain acceptable effect sizes.

To complement the significance testing and assess the sensitivity of the findings, post hoc power analyzes were conducted using G*Power for comparisons that showed statistically significant differences. In the first dimension, the observed effect size for the CRMT post-test was Cohen's d = 0. 647. Using the actual group sizes (n = 105 experimental; n = 39 control; allocation ratio ≈ 0.37), the achieved statistical power

Table 2
Results of the coding, robotics, and machine learning test (CRMT).

Group	n	Mean (M)	SD	t	p	95% CI	Cohen's d
Experimental	105	6.05	2403	3.401	.001***	[0.64, 2.42]	0.638
Control	39	4.51	2416				

Note. Independent samples Student's t-test. Effect size interpreted according to Cohen [27].
* $p < .05$.
** $p < .01$.
*** $p < .001$.

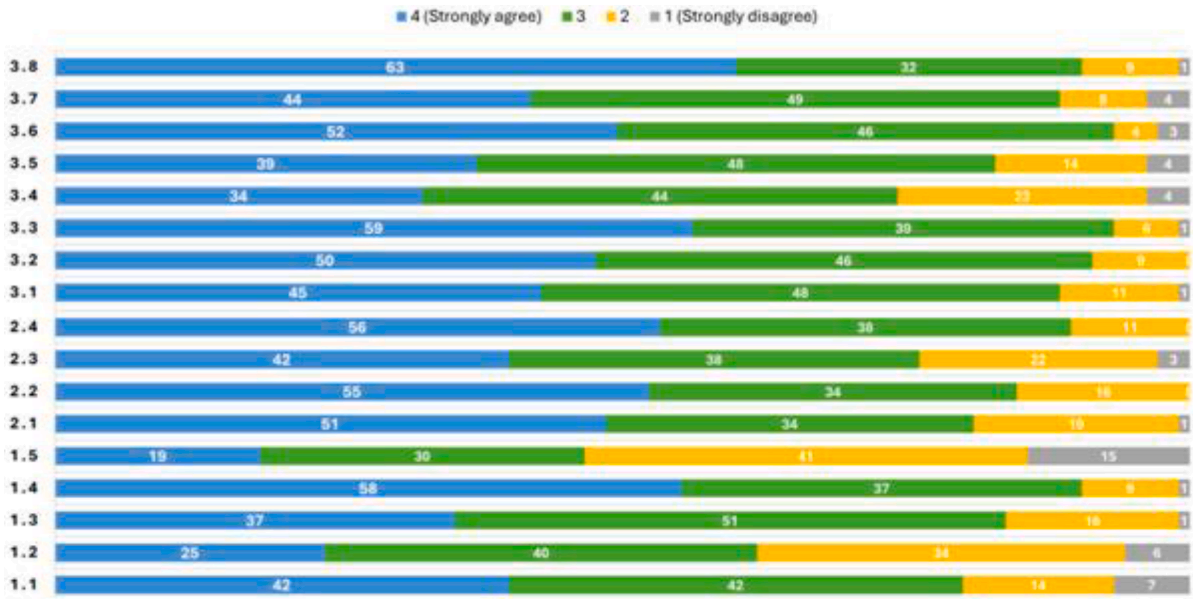

Fig. 5. Frequency of responses of the experimental group to each item in the questionnaire about programming and robotics. Responses from strongly agree (4) to strongly disagree (1).

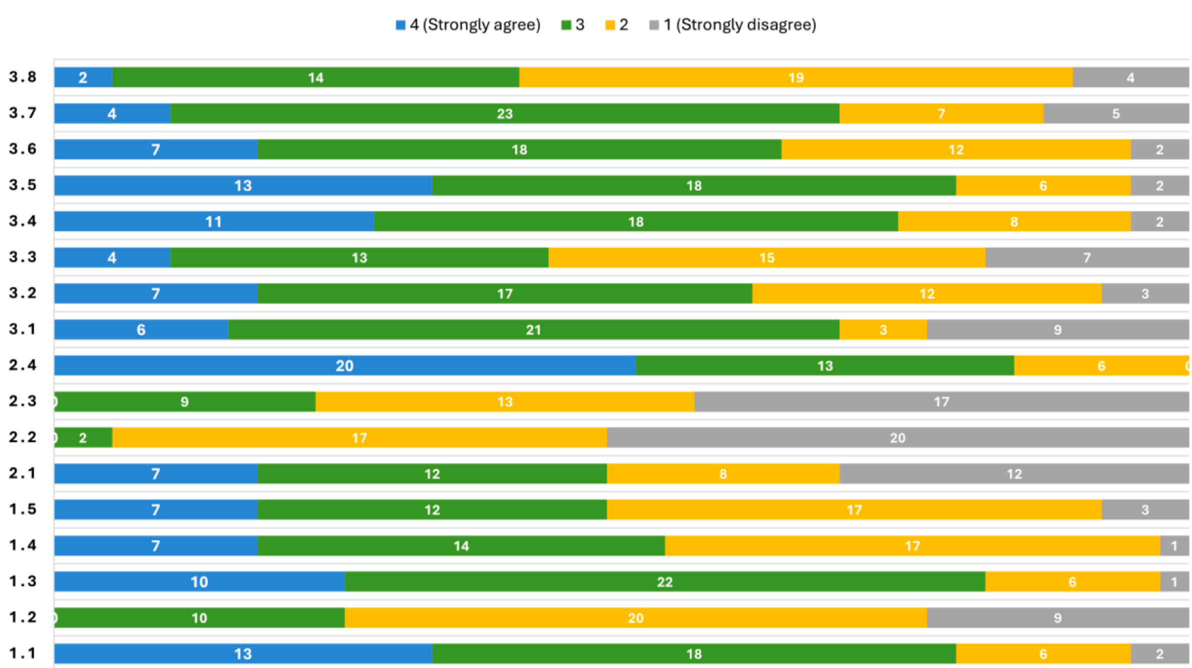

Fig. 6. Frequency responses from the control group to each item in the questionnaire about programming and robotics. Responses from strongly agree (4) to strongly disagree (1).

was 0. 96. This indicates a very low probability of Type II error and reinforces the robustness of the observed difference.

In the second dimension, several statistically significant differences were also supported by medium to large effect sizes. These were tested in G*Power under the same parameters. The results reinforce the reliability of the observed effects and help confirm that the sample size was adequate to detect meaningful differences.

4. Limitations

Several limitations of this study should be acknowledged. First, the use of purposive, non-random sampling and group assignment based on existing class groupings may limit the generalizability of the findings and introduce potential selection bias. Although pre-test equivalence between the experimental and control groups was verified, the absence of randomization restricts the ability to establish full causal inference. Future studies should consider randomized or stratified sampling designs to enhance external validity.

Second, the notable gender imbalance within the sample, reflective of the demographic composition typical of teacher education programs, may influence the interpretation of the results, particularly regarding attitudes toward programming, robotics, and technology-related self-efficacy. While this distribution accurately represents current enrollment patterns, it limits the capacity to explore potential gender-related

Table 3
Descriptive statistics and Mann–Whitney U test results for attitudinal scales.

Scale	Items	%				Mann-Whitney U test
		1	2	3	4	
1. Machine learning and programming	1. Knowledge of programming or coding is essential for training future teachers.	6.7	13.3	40	40	.601
	2. I know what machine learning is.	5.7	32.4	38.1	23.8	.001***
	3. In initial university education, it is important to work on machine learning concepts.	0	15.2	48.6	35.2	.341
	4. The programming and robotics approach enhances the development of creativity.	1.0	8.6	35.2	55.2	.001***
	5. I have worked with visual block programming in university.	14.3	39.0	28.6	18.1	.639
2. Computational concepts	1. I know what a sequence is in programming.	1.0	18.2	32.4	48.6	.001***
	2. I know what a loop is in programming.	0	15.2	32.4	52.4	.001***
	3. I know what a conditional is in programming.	2.9	21.0	36.2	40.0	.001***
	4. The programming and robotics approach fosters educational innovation processes.	0	10.5	36.2	53.3	.672
3. Active learning in curricular areas	1. A lot of content is learned with visual block programming and robotics.	1.0	10.5	45.7	42.9	.001***
	2. The content can be addressed in PE through a PVBlotes (blocks-based learning model).	0	8.6	43.8	47.6	.001***
	3. Programming and robotics enhance mathematical learning.	1.0	5.7	37.1	56.2	.001***
	4. Programming and robotics foster learning in natural sciences.	3.8	21.9	41.9	32.4	.739
	5. Programming and robotics help learning in social studies.	0	13.3	45.7	37.1	.593
	6. Programming and robotics support learning in art and music.	2.9	3.8	43.8	49.5	.001***
	7. With a coding approach, the subject matter is more interesting.	0	7.6	46.7	41.9	.001***
	8. Programming actively engages students.	1.0	8.6	30.4	60.0	.001***

Note. Scale ranged from 1 (strongly disagree) to 4 (strongly agree). Mann–Whitney U test.
* $p < .05$.
** $p < .01$.
*** $p < .001$.

differences in responses to the intervention. Subsequent research could benefit from more balanced samples or from explicitly examining the interaction between gender and learning outcomes in educational robotics contexts.

Third, another limitation of the study relates to the potential influence of institutional differences, since the experimental and control groups were drawn from different universities. Although all participating institutions were part of the same competitive research project and followed a common research design, it is possible that contextual or institutional factors may have influenced the results to some extent.

Another limitation of the study relates to the use of self-reported attitudinal data to assess several outcomes, particularly those included in the second dimension of the study. Although these measures provide valuable insight into participants' perceptions of programming, robotics, and technology-based learning, self-reported responses may not always fully reflect actual knowledge acquisition or behavioral changes. Future studies could complement these measures with additional objective assessments or observational data in order to obtain a more comprehensive evaluation of learning outcomes.

It should be noted that the findings related to Dimension 2 are based on participants' self-reported perceptions. Therefore, statements regarding potential improvements in subject learning (e.g., mathematics or arts) should be interpreted as perceived educational benefits rather than as direct evidence of learning outcomes.

A further limitation of the study concerns the potential influence of institutional differences between the experimental and control groups. Because participants were drawn from different universities, some of the observed differences may partly reflect contextual or institutional factors rather than the intervention alone. Although the participating institutions were involved in the same competitive research project and followed a shared research framework, it is still possible that local teaching contexts or institutional characteristics influenced the outcomes. Future research could address this limitation by implementing randomized group assignments within the same institutional setting or by including a larger number of participating institutions.

Finally, the study was conducted within a limited number of institutions and over a short intervention period, which may restrict the extent to which longer-term learning retention or transfer to authentic classroom practice can be assessed. Longitudinal designs and multi-site studies are recommended to further validate and extend the present findings.

5. Discussion

Several precedents and research studies that analyze the potential of visual block programming in educational contexts can be considered, showing positive results like those found in our study [12,14,15].

These approaches teach basic computational concepts and improve the students' skills in mathematics and the arts. The subjects show training gaps in the use of coding and robotics that must be addressed for the adequate implementation of computational thinking in higher education educational processes.

To address this problem, this study provides a training proposal based on the simplicity of visual programming and robotics, with a sequenced implementation of more complex concepts, so that future teachers have the strategies and knowledge to perform their work as active and engaged professionals. These conclusions are supported by several studies [12,14,28], demonstrating the acquisition of basic computer programming concepts in elementary education, where future teachers will work. University education and initial teacher training must adapt to the demands of today's society, considering the emerging trends that young professionals will encounter in their immediate future.

It is essential to propose a pedagogical design that considers robotics, machine learning, and visual coding to understand computational concepts in different curricular areas and to develop an innovative and dynamic educational practice as education professionals [15] through active approaches.

Based on these approaches focused on visual block-based programming and a simple application of robotics and coding, we propose to analyze the impact that these emerging innovations generate in the initial training of future teachers, as well as the real presence of these key trends in the university training context. In elementary education contexts, several investigations coincide with the results of this study [3,5,12,13,18,29–32]. It is suggested that educational authorities recommend and implement robotics, an understanding of machine learning, and programming in initial teacher training, so that the student teachers can design and develop practices when they are teachers in educational environments in elementary education.

This study makes several relevant contributions to the field of educational technology and initial teacher training. First, it offers an innovative approach by integrating educational robotics with early machine learning activities within a coherent, constructionist instructional framework specifically designed for preservice teachers. While robotics and visual block programming are increasingly present in school-based research, their systematic connection with introductory machine learning concepts in teacher education remains limited. This combined approach represents a novel contribution, demonstrating not only its pedagogical viability but also its positive impact on the development of computational concepts and professional attitudes toward emerging technologies.

Second, the study employs validated assessment instruments and a robust methodological approach, including power analysis, reliability testing, effect size estimation, and both parametric and non-parametric statistical procedures. This methodological rigor strengthens confidence in the reported outcomes and provides a model for future intervention-based research in teacher training contexts.

Finally, the findings demonstrate the practical feasibility of implementing scalable robotics and programming interventions in university settings without the need for complex infrastructure or high-cost technologies. The use of accessible platforms, such as Microbit and visual programming interfaces, enables institutions to integrate similar pedagogical designs into existing curricula, supporting systematic development of computational thinking and foundational machine learning literacy among future educators.

Together, these contributions position the present work as a meaningful addition to the educational technology literature, offering both empirical evidence and practical guidance for the incorporation of robotics and artificial intelligence concepts into initial teacher training programs.

6. Conclusions

This research provides interesting findings regarding the use of robotics and coding in university contexts for learning computational concepts and machine learning today, as well as the potential that coding offers in the initial training of future teachers. It is interesting to see the advantages of robotics and coding in different curricular areas, as well as the possibilities for innovation and motivation, and the enthusiasm of participants. Ultimately, these active practices contribute to the acquisition of computational concepts and practices through creativity and enthusiasm and with careful implementation of robotics and coding in simple and structured activities.

Based on the results of the study and considering data from different sources and tests, the following conclusions are drawn in a structured manner.

1. The results suggest that participation in robotics-based activities was associated with higher levels of computational knowledge. Participants improve their performance and results with the study intervention. Based on the scores of the CRMT administered to the subjects, it is evident that the scores obtained are not particularly high, demonstrating the difficulty of the content for the participants; however, the experimental group obtained statistically significant improvements compared to the control group. Therefore, the university students' ability to understand computational concepts, programming logic, the implementation of educational robotics, and machine learning is improved (dimension 1).
2. Coding is essential in teacher training. Eighty percent of the subjects consider coding knowledge essential for training future teachers (item 1.1) and for working with machine learning concepts (item 1.3), although these variables are not statistically significant. Furthermore, 60 percent of the sample know what machine learning is (item 1.2), and 90 per cent state that programming and robotics develop creativity (item 1.4). These two variables show statistically significant improvements: the intervention has therefore provided them with an understanding of the concept of machine learning and a sense of its value for creativity in these processes. Just under 50 percent of the sample has worked with visual block programming at university (item 1.5), highlighting this as an area in need of improvement in higher education settings. Thus, the subjects emphasize the great importance of teacher training in the use of programming in educational settings.
3. Computational concepts are learned. Basic computational concepts, sequences, loops, and conditionals (items 2.1, 2.2, and 2.3), are familiar to 80 percent of the respondents. The subjects have learned these concepts through the intervention, shown by the significant improvements obtained.
4. Pedagogical benefits of the use of coding are detailed. Subjects appreciate that learning through coding and robotics (item 3.1) involves active participation and is interesting (items 3.7 and 3.8), with descriptive values close to 90 percent in these three items and with statistically significant improvements in the three variables. Therefore, subjects appreciate benefits in the active approach, learning, and student interest. On the other hand, most subjects (90%) affirm that coding and robotics foster innovation (item 2.4), although no significant improvements were obtained in this variable.
5. Visual programming and educational robotics are feasible in the curricular areas. Around 90 percent, with significant improvements, consider these approaches optimal for learning mathematics (item 3.3) and the arts (item 3.6). In natural sciences (item 3.4) and social sciences (item 3.5), descriptive values (around 80 percent) are obtained, but there are no significant improvements. With appropriate selection and sequencing of content, adjusted depending on the students' prior knowledge and progress, implementation in different areas is feasible, especially in mathematics and arts education.

The conclusion, with high scores and significant improvements, is that the approach of integrating programming and robotics in the classroom fosters active learning, creativity, and interest among university students. It is emphasized that it is feasible to implement these pedagogical designs in the training of future primary school teachers, fostering an understanding of machine learning and the use of coding and robotics in the university training of future teachers.

All authors have read and agreed to the published version of the manuscript. Please address all correspondence to: José Manuel Sáez López jmsaezlopez @edu.uned.es

We confirm that this manuscript has not been published elsewhere and is not under consideration by another journal. All authors have approved the manuscript and agree with its submission to *Computers & Education Open.*

We would like to have the manuscript considered for publication. We believe our findings would appeal to the readership of *Computers & Education Open.* We look forward to hearing from you at your earliest convenience.

Funding

Creative programming in primary education. Development of materials and proposals for block coding, game engines, machine learning, and robotics. (PID2022–136442OB-I00) Knowledge Generation Projects 2022. (MICINN). Ministry of Science, Innovation and Universities of Spain.

CRediT authorship contribution statement

José-Manuel Sáez-López: Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Arturo-Simon García-Jiménez:** Validation, Supervision, Methodology, Investigation, Data curation, Conceptualization. **Sebastián de Lara García-Cervigón:** Writing – review & editing, Writing – original draft, Validation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

JOSE MANUEL SAEZ LOPEZ reports financial support was provided by National Distance Education University. JOSE MANUEL SAEZ LOPEZ reports a relationship with National Distance Education University that includes: funding grants. Creative programming in primary education. Development of materials and proposals for block coding, game engines, machine learning, and robotics. (PID2022- 136442OB-I00) Knowledge Generation Projects 2022. (MICINN). Ministry of Science, Innovation and Universities of Spain. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix I

Session 1 – Introduction to computational thinking and visual programming

This initial session introduces the theoretical foundations of computational thinking within a constructionist learning framework. Students explore the concepts of sequencing, algorithmic thinking, and trial-and-error problem solving through guided activities in the Micro:bit MakeCode visual programming environment. A simple "digital dice" project is developed to familiarize participants with block categories, event handling, and random number generation, supporting hands-on experimentation and immediate feedback.

Learning focus: Sequencing, algorithmic logic, familiarization with block-based programming.

Practical activities:

1. **Exploration of the MakeCode interface:** Students navigate block categories, test simulator functions, and examine basic event blocks to gain technical fluency with the environment.
2. **Dice simulation project:** Participants create a digital dice generator using random number blocks triggered by motion events (shake function) to introduce sequencing and event-driven programming.
3. **Debugging practice:** Students modify and repair intentionally flawed programs to identify common beginner errors and practice iterative problem-solving strategies.

Session 2 – Basic programming structures: sequences and loops

Building on Session 1, this session focuses on the explicit manipulation of programming sequences and iteration constructs (loops). Students design short routines involving repeating LED patterns and timed animations using Micro:bit. Instruction emphasizes logical structuring, pattern recognition, and debugging strategies consistent with computational thinking practice. The learning objective is to develop fluency with basic control structures and program flow.

Learning focus: Repetition, control flow, pattern development.

Practical activities:

1. **LED animation design:** Students create repeating light patterns using loop blocks to visualize iterative program execution.
2. **Timed sequence challenges:** Learners sequence actions (lights and pauses) to simulate countdown timers or rhythmic signaling patterns.
3. **Algorithm optimization activity:** Programs are rewritten to reduce block redundancy by replacing repeated sequences with loop structures, encouraging efficient algorithm design.

Session 3 – Conditional logic and sensor-based programming

Participants are introduced to conditional statements ("if–then" logic) and basic sensor inputs (light and temperature). Students construct programs that respond dynamically to environmental stimuli, such as displaying sensor readings or triggering visual responses when threshold values are reached. The session emphasizes real-time data processing and decision-making logic, reinforcing the relationship between physical input and computational output.

Learning focus: Decision-making structures, input-output interaction.

Practical activities:

1. **Environmental reading display:** Students program the Micro:bit to display temperature or light intensity when Button A is pressed.
2. **Threshold-trigger alert systems:** Programs are developed to activate visual or sound alerts when sensor readings exceed or drop below set values.
3. **"If–Then" scenario challenges:** Small programming tasks require students to design conditional responses to given real-world contexts (e.g., simulate weather-based warnings).

Session 4 – Data collection and analysis using sensors

In this session, Micro:bit devices are used to collect real-world data concerning light intensity and temperature from various campus locations. Students design data-logging programs, conduct structured field measurements, and compile findings for basic statistical analysis (e.g., averages, comparisons). The activity connects computational thinking to data literacy and interdisciplinary application, particularly in mathematics and environmental science education.

Learning focus: Sensor data acquisition and quantitative interpretation.
Practical activities:

1. **Campus temperature mapping:** Learners collect temperature readings from five locations using Micro:bit devices and record results in data tables.
2. **Light-level comparison analysis:** Groups measure ambient light in classrooms and outdoor spaces and graph differences.
3. **Statistical reflection exercise:** Students calculate mean values and discuss environmental variation, linking computational tools to mathematics education standards.

Session 5 – Introduction to robotics with the maqueen kit

Students transition from board-based programming to physical robotics using the Maqueen robot. After assembling the robot, participants program basic motor commands to execute controlled movements along predefined trajectories. Key concepts include directional control, sequencing of actions, and spatial reasoning. The pedagogical goal is to link abstract code to embodied robotic behavior.
Learning focus: Motor control, physical execution of algorithms.
Practical activities:

1. **Robot assembly workshop:** Students physically construct the Maqueen robot and integrate the Micro:bit controller.
2. **Movement sequencing programs:** Learners code straight-line movement, turns, and timed stops to navigate predefined pathways.
3. **Obstacle navigation challenge:** Students design programs to reach a destination using solution-oriented sequencing and trial-and-error debugging.

Session 6 – Sensor integration and conditional robotic control

This session introduces obstacle detection and line-following challenges using Maqueen's infrared sensors. Students construct programs that incorporate conditional logic to adjust motor responses based on sensor input (e.g., turn left or right upon detecting white or black surfaces). The activities reinforce algorithmic decision-making, debugging, and multi-variable problem-solving within tangible robotics tasks.
Learning focus: Real-time robotic decision-making.
Practical activities:

1. **Line-following programming:** Students design algorithms that use infrared sensors to detect surfaces and maintain track alignment.
2. **Light-driven "Car Project":** Programs are developed whereby robots move only when strong light is detected, illustrating sensor thresholds and conditionals.
3. **Error-debugging trials:** Groups experiment with incorrect sensor settings and revise logical structures to improve functional performance.

Session 7 – Introductory machine learning through visual programming

Students engage with the Machine Learning for Kids platform and Scratch 3.0 to explore supervised learning models using image classification. The "Shy Panda" project guides participants through dataset creation, model training, and real-time inference. Emphasis is placed on understanding the core ML workflow—data labeling, training, testing, and refinement—rather than technical implementation details, thus introducing AI concepts at an accessible level.
Learning focus: Supervised learning concepts, AI awareness.
Practical activities:

1. **Dataset generation:** Students capture and label images for two classification categories ("I see you / I don't see you") within the Machine Learning for Kids platform.
2. **Model training and testing:** Learners train models and observe classification accuracy through live webcam testing in the "Shy Panda" project.
3. **Algorithm refinement:** Groups modify datasets and retrain models to observe changes in prediction reliability and discuss model bias.

Session 8 – Robotics engineering with the Nezha kit

Participants construct and program a multi-component robotic model using the Nezha kit, culminating in the development of an Automated Traffic Barrier system equipped with sensors, lights, and actuators. Students implement parallel processes, event triggers, and conditional logic to coordinate mechanical actions. The session integrates engineering design principles with advanced block programming constructs, highlighting the complexity attainable through visual coding.
Learning focus: Advanced robot construction, coordination of multiple systems.
Practical activities:

1. **Mechanical assembly:** Students build the automated traffic barrier and traffic light system using Nezha components.
2. **Parallel programming design:** Learners write programs coordinating sensors, motor actuators, and LED signals simultaneously.
3. **System testing and optimization:** Groups identify malfunctions, refine logic sequences, and improve functional synchronization.

Session 9 – Educational design integration and pedagogical reflection

The final session centers on didactic transfer and curricular application. Students collaborate in groups to design short learning activities linking robotics and programming to primary curriculum areas such as mathematics, science, art, or physical education. Each group presents its lesson plan, justifies its pedagogical choices using constructivist and constructionist principles, and reflects on classroom feasibility, learner engagement, and assessment strategies. The session concludes with administration of the post-intervention instruments (CRMT and attitudinal scales).

Learning focus: Curriculum transfer, didactic reasoning.

Practical activities:

1. **Lesson-plan development:** Groups design classroom activities connecting robotics to curricular areas (mathematics, science, art, or PE).
2. **Micro-teaching presentations:** Each group presents and justifies its lesson using constructionist pedagogical principles.
3. **Self-assessment and reflection:** Students complete reflective questionnaires and evaluate their confidence in applying robotics and programming in primary education contexts.

Appendix II

Table 4
Structure of the CRMT.

Construct / Computational concept	Number of items	Description of assessed knowledge
Sequencing	2	Understanding the correct order of programming instructions
Loops / Iteration	2	Ability to identify and apply repetition structures in block-based programming
Conditional logic	2	Understanding decision-making structures (if–then conditions)
Sensors and input-output	2	Knowledge of how robots interact with environmental inputs
Introductory machine learning	2	Basic understanding of training data and classification concepts

References

[1] Resnick M, Maloney J, Hernández AM, Rusk N, Eastmond E, Brennan K, Millner A, Rosenbaum E, Silver J, Silverman B, Kafai Y. Scratch: programming for all. Commun ACM 2009;52(11). http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf.

[2] Sonmez D, Lee H. Problem-based learning in science. ERIC Clearinghouse for Science, Mathematics, and Environmental Education; 2003.

[3] Chevalier M, Giang C, El-Hamamsy L, Bonnet E, Papaspyros V, Pellet JP, Audrin C, Romero M, Baumberger B, Mondada F. The role of feedback and guidance as intervention methods to foster computational thinking in educational robotics learning activities for primary school. Comput Educ 2022;180:104431. https://doi.org/10.1016/j.compedu.2022.104431.

[4] Maloney J, Kafai Y, Resnick M, Rusk N. Programming by choice: urban youth learning programming with scratch. In: 39th SIGCSE Technical Symposium on Computer Science Education, Portland, Oregon; 2008. p. 367–71. https://doi.org/10.1145/1352135.1352260.

[5] Noh J, Lee J. Effects of robotics programming on the computational thinking and creativity of elementary school students. Edu Technol Res Develop 2020;68:463–84. https://doi.org/10.1007/s11423-019-09708-w.

[6] Brennan K, Resnick M. New frameworks for studying and assessing the development of computational thinking. In: Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada; 2012.

[7] Kolodner J, Camp P, Crismond D, Fasse B, Gray J, Holbrook J, Puntambekar S, Ryan M. Problem-based learning meets case-based reasoning in the middle-school science classroom: putting learning by design into practice. J Learn Sci 2003;12(4):495–547.

[8] Papert S. Mindstorms: children, computers, and powerful ideas. Basic Books; 1980.

[9] Lambert L, Guiffre H. Computer science outreach in an elementary school. J Comput Sci Colleges 2009;24(3):118–24.

[10] Lin JMC, Yen LY, Yang MC, Chen CF. Teaching computer programming in elementary schools: a pilot study. In: National Educational Computing Conference; 2005.

[11] Clark J, Rogers MP, Spradling C, Pais J. What, no canoes? Lessons learned while hosting a scratch summer camp. J Comp Sci Colleges 2013;28:204–10.

[12] Israel M, Pearson JN, Tapia T, Wherfel QM, Reese G. Supporting all learners in school-wide computational thinking: a cross-case qualitative analysis. Comp Edu 2015;82:263–79. https://doi.org/10.1016/j.compedu.2014.11.022.

[13] Sáez-López JM, del Olmo-Muñoz J, González-Calero JA, Cózar-Gutiérrez R. Exploring the effect of training in visual block programming for preservice teachers. Multim Technolog Interact 2020;4(3):65. https://doi.org/10.3390/mti4030065.

[14] Sáez-López JM, González-Calero JA, Cózar-Gutiérrez R, del Olmo-Muñoz J. Scratch and unity design in elementary education: a study in initial teacher training. J Comput Assist Learn 2023;39(5):1528–38. https://doi.org/10.1111/jcal.12815.

[15] Sáez-López JM, Román-González M, Vázquez-Cano E. Visual programming languages integrated across the curriculum in elementary school. A two-year case study using scratch in five schools. Comput Edu 2016;97:129–41. https://doi.org/10.1016/j.compedu.2016.03.003.

[16] Adams Becker S, Freeman A, Giesinger Hall C, Cummins M, Yuhnke B. NMC/CoSN horizon report: 2016 K-12 edition. The New Media Consortium; 2016.

[17] Robert J, Muscanell N, McCormack M, Pelletier K, Arnold K, Arbino N, Young K, Reeves J. 2025 EDUCAUSE horizon report. teaching and learning edition. EDUCAUSE; 2025.

[18] Giannakoulas A, Xinogalos S. Studying the effects of educational games on cultivating computational thinking skills to primary school students: a systematic literature review. J Comput Edu 2024;11:1283–325. https://doi.org/10.1007/s40692-023-00300-z.

[19] Wing J. Computational thinking. Commun ACM 2006;49(3):33–5. https://doi.org/10.1145/1118178.1118215.

[20] Bers MU. Blocks to robots: learning with technology in the early childhood classroom. Teachers College Press; 2008.

[21] Blikstein P. Digital fabrication and 'making' in education: the democratization of invention. In: Walter-Herrmann J, Büching C, editors. FabLabs: of machines, makers and inventors. Transcript Publishers; 2013. p. 203–22.

[22] Houngue P, Hountondji M, Dagba T. An effective decision-making support for student academic path selection using machine learning. Int J Adv Comput Sci Appl 2022;13(11):727–34. https://doi.org/10.14569/IJACSA.2022.0131184.

[23] Sperling K, Stenliden L, Jörgen N, Heintz F. Still w(AI)ting for the automation of teaching: an exploration of machine learning in Swedish primary education using Actor–Network Theory. Europ J Educ 2022;57(4):584–600. https://doi.org/10.1111/ejed.12526.

[24] Cózar-Gutiérrez R, Sáez-López JM. Game-based learning and gamification in initial teacher training in the social sciences: an experiment with MinecraftEdu. Int J Edu Technol Higher Edu 2016;13(2). https://doi.org/10.1186/s41239-016-0003-4.

[25] Hair JF, Anderson RE, Tatham RL, Black WC. Multivariate data analysis. 5th edition. Prentice Hall; 1998.

[26] Goetz JP, LeCompte MD. Ethnography and qualitative design in educational research. Academic Press; 1988.

[27] Cohen J. Statistical power analysis for the behavioral sciences. 2nd edition. Erlbaum; 1988.

[28] Sáez-López JM, Cózar R. Visual programming by blocks in primary education: learning and creating content in social sciences. Rev Complut Educ 2017;28(2):409–26. https://doi.org/10.5209/rev_RCED.2017.v28.n2.49381.

[29] Rich PJ, Bartholomew S, Daniel D, Dinsmoor K, Nielsen M, Reynolds C, Swanson M, Winward E, Yauney J. Trends in tools used to teach computational thinking through elementary coding. J Res Technol Edu 2022;56(3):269–90. https://doi.org/10.1080/15391523.2022.2121345.

[30] Wei X, Lin L, Meng N, Tan W, Kong SC. The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy. Comput Educ 2021;160:104023. https://doi.org/10.1016/j.compedu.2020.104023. Article.

[31] Wong GKW. Amplifying children's computational problem-solving skills: a hybrid-based design for programming education. Edu Inform Technol 2024;29:1761–93. https://doi.org/10.1007/s10639-023-11880-9.

[32] Wu S-Y, Su Y-S. Visual programming environments and computational thinking performance of fifth- and sixth-grade students. J Edu Comput Res 2021;59(6):1075–92. https://doi.org/10.1177/0735633120988807.