

## Research Article

# A Mobile Application That Allows Children in the Early Childhood to Program Robots

Kryscia Ramírez-Benavides, Gustavo López, and Luis A. Guerrero

Universidad de Costa Rica, San José, Costa Rica

Correspondence should be addressed to Kryscia Ramírez-Benavides; kryscia.ramirez@ucr.ac.cr

Received 30 May 2016; Revised 10 October 2016; Accepted 13 October 2016

Academic Editor: Laurence T. Yang

Copyright © 2016 Kryscia Ramírez-Benavides et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Children born in the Information Age are digital natives; this characteristic should be exploited to improve the learning process through the use of technology. This paper addresses the design, construction, and evaluation process of TITIBOTS, a programming assistance tool for mobile devices that allows children in the early childhood to create programs and execute them using robots. We present the results of using TITIBOTS in different scenarios with children between 4 and 6 years old. The insight obtained in the development and evaluation of the tool could be useful when creating applications for children in the early childhood. The results were promising; children liked the application and were willing to continue using it to program robots to solve specific tasks, developing the skills of the 21st century.

## 1. Introduction

The Information Age is a period in human history in which the use of technological tools extensive and almost every human activity is based on information computerization [1]. Children born in the Information Age are called *digital natives* [2]. Incorporating activities that promote the 21st century skills [3] in the learning process helps digital natives to develop abstract thinking abilities and apply them in an organized way [4–6].

Many authors have discussed the importance of programming as a capability for digital natives. Papert [4, 5] described programming as a tool that develops a comprehensive set of interconnected capabilities such as problem solving, teamwork, persistence, logical-mathematical thinking, abstraction, and creativity. Resnick [7] considers programming the new literacy. He states that, “in addition to writing and reading, programming helps organize thoughts and express ideas.” Furthermore, the skills gained with programming and robotics are a key aspect in the development of children and their future career [3, 8].

There is a global deficit of science, technology, engineering, and mathematics (STEM) professionals [9]. Therefore, countries are challenged to promote them. STEM concepts

are complex; however, they might be presented in stimulating ways, such as robotics [10] and mobile applications.

This paper describes the design, construction, and evaluation process of TITIBOTS, a mobile programming assistance tool (PAT) [11] that allows children in the early childhood to develop programs and execute them using robots. TITIBOTS has an icon-based interface and it integrates visual programming, robotics, and mobile devices in one tool. Moreover, the main issues and lessons learned during this process are described.

TITIBOTS was developed and evaluated applying several Human-Computer Interaction techniques such as participatory-design, experience prototyping [12], and usability testing [12, 13].

The main research question driving this work was to assess the possibility of children aged between 4 and 6 years to use a PAT based on mobile interfaces and robots. A major difficulty in this research work was to create a graphical interface to be used by kids between 4 and 6 years old. Similar works have been conducted in the past years. However, these projects are focused mainly in older children. The proposed mobile PAT was combined with robots to promote a fun exploration of complex concepts involving sensory, motor, and socioemotional skills [8, 10].

The evaluation of TITIBOTS was conducted in Costa Rica and had the participation of over 50 children in the early childhood during the different evaluation stages. This evaluation showed that children in the early childhood are able to program robots using mobile applications through TITIBOTS.

## 2. Robot Programming Assistance Tools for Children

This section presents the summarized results of a systematic literature review conducted to find PATs to program robots usable by children under 12 years old. The review was conducted in three academic databases: ACM and IEEE digital libraries and Springer Link.

The search query included the following keywords: programming environment, programming kit, programming interface, programming language, robots, robotics, and children. The selection criteria were established to find original research papers presenting empirical studies in real contexts.

In the 1970s, Papert and his students at MIT created Logo, a programming language and robot [4]. Logo acquired a great popularity in the 1980s. In 1985 MIT researchers changed the robot for a graphical representation. Later that decade, in 1988 Resnick and Ocko developed at MIT Media Lab one sensor and actuator system called LEGO TC Logo. This idea was a commercial success and the tool reached thousands of classrooms [14].

MicroWorlds was released in 1993 by LCSI [15]. At one point MicroWorlds also incorporated the possibility of programming a robot. In 1994, RCX brick (a programmable device and PAT) was released. This PAT was based on icons and allowed the users to create diagrams (programs) that controlled the RCX [14].

LEGO WeDo set, designed by LEGO in 2005, consisted of a set of mechanical parts used to build and design LEGO models. LEGO WeDO had an easy to use, icon-based PAT [16]. LEGO Mindstorms NXT brick was released in 2006. It was a visual programming environment that allows a novice programmer to easily create programs [17]. Smaller versions of NXT were available commercially. These bricks were called Pico-Cricket and they have an associate PAT called PicoBlocks. However, this brick was discontinued in 2010 [18]. MIT also approached their programmable brick with a puzzle based programming interface called LogoBlocks [14].

In 2007 miniBloq appeared. It was an open source multi-platform graphic PAT based on C++. It uses symbols to visualize, in real time, the possible coding errors [19]. Enchanting appeared in 2010. Enchanting is a programming environment for LEGO Mindstorms NXT. It is based on Scratch and supports leJOS NXJ [20]. MoWay, released in 2012, is a small autonomous robot with its own programming language [21].

The brick EV3 of LEGO Mindstorms appeared in 2013. It can be controlled using a remote control and a mobile application called "Robot Commander" which is available for iOS and Android devices [22].

In 2013, Wonder Workshop was presented. Wonder's main goal is that two small robots, Dash and Dot, teach children over 5 years old basic programming concepts through

TABLE 1: Robot programming tools ordered by age.

Programming tool	Ages	Publication year
PRIMO	4-7	2013
KIBO	4-7	2014
Wonder Workshop	5-8	2013
LEGO WeDo	6-12	2005
Logo	7-12	1970
LEGO TC Logo	7-12	1988
LogoBlocks	7-12	1994
LEGO RoboLab RCX	7-12	1994
PicoCrickets y PicoBlocks	7-12	2006
LEGO Mindstorms NXT	7-12	2006
LEGO Mindstorms EV3	7-12	2013
MoWay	7-12	2012
miniBloq and RobotGroup	8-12	2008
Enchanting	8-12	2010
MicroWorlds EX Robotics	10-12	1994

interactive games [23]. Also in 2013, Yacob et al. presented Primo. Primo is a tangible programming interface for children between 4 and 7 [24].

KIBO appeared in 2014. KIBO is a robotic kit for children between 4 and 7 years old that uses a tangible programming interface [25]. KIBO is the result of the research conducted by Marina Bers and her research group DevTech at Tufts University.

Table 1 summarizes the results of the systematic literature review. The programming tools most similar to TITIBOTS are KIBO, PRIMO, Wonder, and Software LEGO WeDo, since they are focused on children under 6 years old. KIBO and PRIMO use tangible interfaces instead of mobile systems, WeDO do not use robots, and Wonder app is restrictive of the robot that can be used. TITIBOTS uses an open set of commands that allows the use of any robot, and it was designed using an icon-based interface to allow children that do not know how to read to use it.

Most of the systems presented in this section are commercialized by private companies. Therefore, there is no evidence of their performance in academic literature. Moreover, longitudinal studies have not been conducted to assess their overall impact in the educative process.

## 3. TITIBOTS

TITIBOTS is a mobile programming assistance tool developed for children in the early childhood. It allows children to create programs and execute them using a robot.

Children between 4 and 6 years old are our goal user. Usually, at these ages children are still unable to read or write. TITIBOTS provides a simple graphical user interface using an iconographic approach. The design is intended to be intuitive and usable for children. It is fit for mobile devices and was designed to allow the use of any robot with an open set of commands.

Figure 1 shows the main components of TITIBOTS: the robot and the mobile application. We implemented an easy to

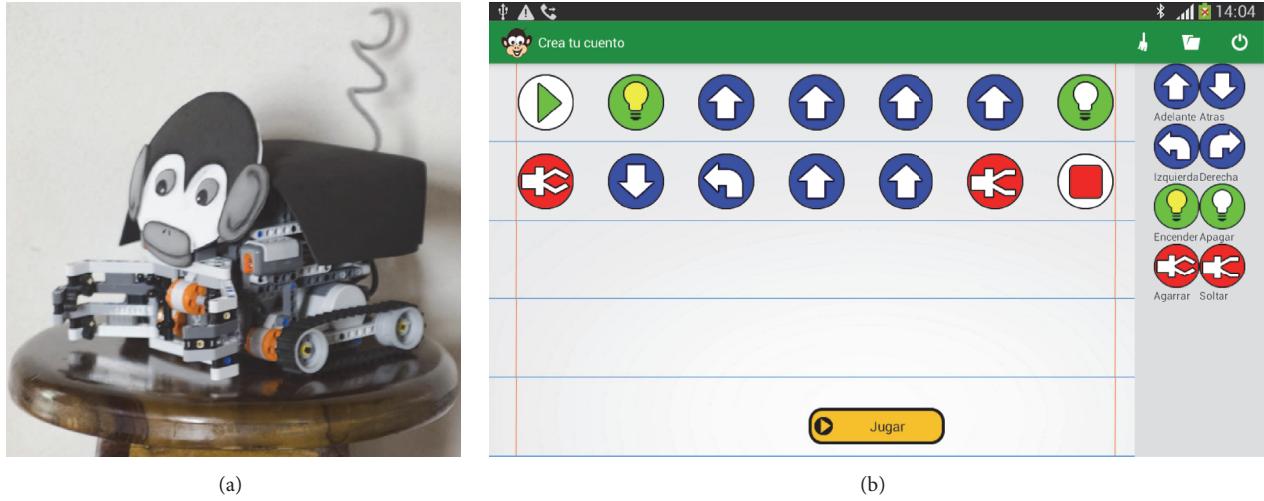


FIGURE 1: TITIBOTS robot (a) and mobile user interface (b).

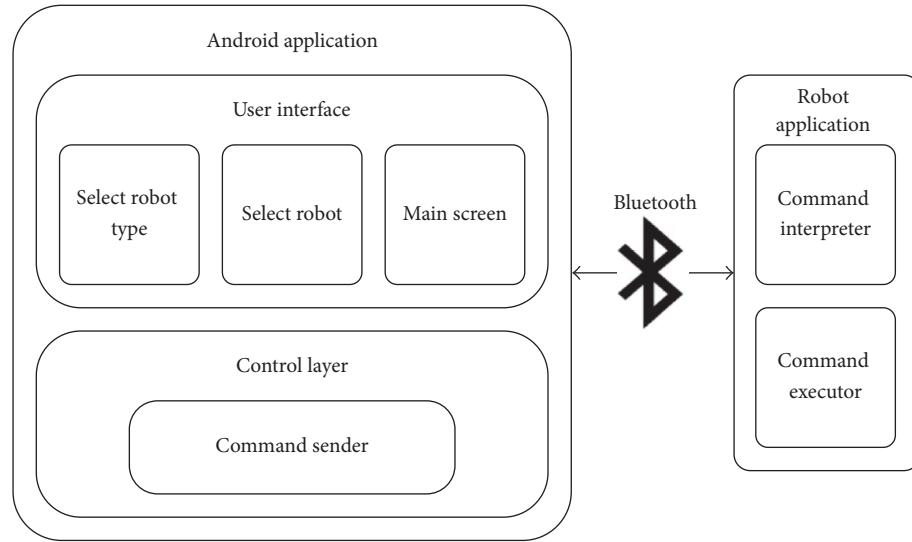


FIGURE 2: TITIBOTS architecture.

use wizard that allows configuration and connection between the robot and the mobile application. The communication between the tablet and the robot is via Bluetooth. The teacher can configure the system and add or remove robots. This version of the system runs on Android devices. Once the tablet is connected with the robot, the child can create a program dragging and dropping the available commands, and lastly he/she can send it to the robot for execution. TITIBOTS architecture (see Figure 2) consists of two main components:

- (1) A mobile application runs on the tablet and shows a simple interface commands and the workspace of the PAT. Commands include the following:
  - (i) Control: start and end (white commands, see Figure 1)
  - (ii) Movement (locomotion): forward, backward, left, and right (blue commands, see Figure 1)

(iii) Action (manipulation): turn on, turn off, grasp, and release (green and red commands, see Figure 1)

- (2) A program in the robot interprets and executes commands sent through the Bluetooth connection from the mobile device to the robot.

The teacher specifies the available commands for each work session. A stage approach was developed allowing the teacher to use a gesture to unlock new commands once he or she thinks the child achieved a certain level of effectiveness with the available commands (see Figure 3).

The basic programming concepts addressed with the use of the tool are algorithms and sequentialization. An algorithm is a set of instructions (steps) to perform a task. It is important to emphasize that the algorithms are the core of programming. Thus, each program is simply a list of

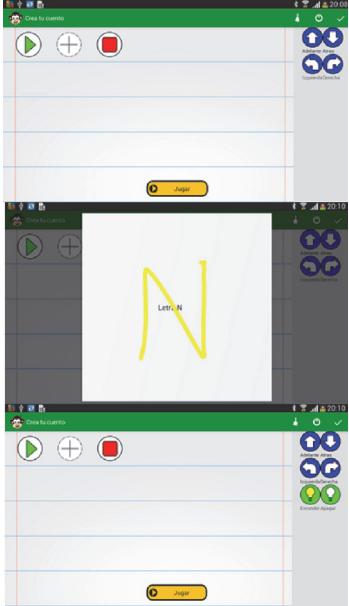


FIGURE 3: Stage approach used to increasingly add commands using teacher's gestures.

instructions that the computer must follow in a certain order (sequentialization).

Among the concepts of robotics addressed are effectors, actuators, locomotion, and handling. Effectors and actuators are components of a robot. An effector corresponds to any device that affects or modifies the environment. An actuator is any mechanism that allows the effector perform an action, for example, servos and lights. The locomotion system allows the robot to move through the ambient while handling system allows the robot to articulate and reach objects in the environment.

In order to evaluate TITIBOTS software interface, we designed and constructed a Mindstorm NXT robot. Our robot was able to move, turn a light on and off, and was provided with a claw that could be opened and closed. We designed and evaluated our system in collaboration with the Omar Dengo Foundation (FOD), which is a nonprofit Costa Rican organization created in 1987. Education experts working at FOD provided a set of functional and nonfunctional requirements in a participatory design process. Based on the gathered requirements the system should

- (i) use a metaphor that allows a clear understanding for the kids,
- (ii) provide a set of commands (8 to 10) that the robot can execute,
- (iii) implement control structures [26],
- (iv) store the last programmed routine locally,
- (v) guide the user in a corrective programming process,
- (vi) connect via Bluetooth with the robot,
- (vii) allow Text to Speech (TTS) capability,
- (viii) have actuators: servos and lights, at least.

Additionally, the preschool teachers defined several tasks to be programmed by children. Most of the tasks are movement sequences (i.e., move the robot from one place to another or move objects).

#### 4. Evaluation

TITIBOTS was designed using a user-centered design following the ISO 13407 [27]. Furthermore, we performed a concept validation and a prototype evaluation. The evaluation process consisted of four stages, which can be observed in Figure 4.

We conducted a group sketching activity [28] to design a preliminary version of TITIBOTS. The first stage of the evaluation process was a concept validation with experts and preschool teachers, in which iconography and interaction were validated. When we achieved a consensus, the best interaction proposals were evaluated with children.

The second stage was the iconography and interaction validation with children. At this stage, we created a sketch that represents the interface, iconography, and interaction patterns. With the information gathered we developed the first functional prototype of TITIBOTS.

In the third stage, we evaluated TITIBOTS functionality with children. For this purpose, we created a set of challenges for the children and used observations to evaluate their behavior. The main goal of this evaluation was to see the children's reaction with the tool and to find difficulties.

Finally, the prototype of TITIBOTS was evaluated in a real scenario. Our testing scenario was a workshop at FOD with 4 to 5 years old kids. The last two stages were performed with children between 4 and 5 years old, because we wanted to test the prototype with the younger end users. We follow the methodology proposed by Nielsen to conduct a usability test with users [29].

Table 2 shows the associated metrics with usability attributes and acceptance criteria for each attribute [29, 30]. To collect quantitative and qualitative data, we used the following measuring instruments [29, 30]:

- (i) Semistructured interview: aimed at obtaining user profile information
- (ii) Memory test: a questionnaire for measuring the number of successfully memorized system functions
- (iii) Recordings: video and audio recordings of the pilots
- (iv) Evaluator's booklet: a booklet in which the experimenter that conducts the assessment procedure takes notes, describes identified problems, and fills in information about average completed tasks and time spent on task
- (v) Satisfaction questionnaire: a questionnaire with two points used for users' subjective evaluation.

In the following subsections, we will introduce the instruments and procedures used in the evaluation of TITIBOTS as well as the participants and results.

Table 3 shows the participants in the evaluation of TITIBOTS per activity, distributed by gender and age, and Figure 5 shows the graphic user profile information obtained

TABLE 2: Usability attributes and associated metrics.

Usability attribute	Metrics	Usability goal
Learnability	(i) Average time used to complete a challenge the first time. (ii) Average time of training.	(i) The average time to complete one challenge for children should be between 10 and 30 minutes. (ii) The average time of training should be between 30 and 60 minutes.
Efficiency	(i) Total and percentage of successful challenges. (ii) Average time to complete a challenge.	(i) Successful challenges completions should be above 70%. (ii) The average time to complete one challenge for children should be between 10 and 20 minutes.
Memorability	Total and percentage of correct answers about the application.	The novice users should memorize at least half of the functionalities of the system, whereas the experienced users should memorize over 80%.
Errors	(i) Total and average unsuccessful attempt. (ii) Total and average of recovery unsuccessful attempt.	(i) The average of errors be between 5 and 10 errors. (ii) The average recovery errors by user should be above 60% of total errors.
Satisfaction	(i) Like or not like. (ii) Difficulty level.	Percentage of satisfaction usually should be above 65%.

by semistructured interview, which was conducted in a 13-children scenario-based usability testing (third and fourth stage).

**4.1. Validation.** In order to validate the interface design, iconography, and interaction, we created a set of instruments and validated the form, color, and possible interaction with 40 children. To perform the validation we carried out a laboratory test [31]. To achieve a good validation, we performed highly controlled observations and qualitative data.

**4.1.1. Participants.** This experiment was carried out at a Costa Rican Primary School. Two groups of children were included in the validation, both groups with children aged 4 to 6 years old. The selection of participants was made through a nonprobabilistic sampling and at an intentional mode based on the availability of the teachers. Three evaluators conducted this validation and one person was responsible of the logistic aspects.

**4.1.2. Setting and Instruments.** The validation was carried out in the schoolyard, 10 feet away of the classroom entrance. The setting consisted of four desks for toddlers at a distance of five feet from each other. The children were inside the classroom until they were called by the one responsible of logistics and showed to a desk with one evaluator that conducted the validation.

The elements used in the validation were a physical prototype of the interface and a validation guide. The evaluator used the guide to take notes and follow the process. She checked on the questionnaire according with the children reactions. The physical interface consisted of a sheet with the designed icons for programming, three sheets with the versions of the sequence options evaluated, and three sheets with the interaction patterns evaluated.

Figure 6 shows the interaction patterns and sequence options evaluated. Figure 6(a) shows the insert interaction. In this interaction, the user presses the place in which he/she

wants to insert a command and the options appear. Figure 6(b) shows the drag and drop interaction. The commands are selected and dragged to the place in which the user wants to insert it. Finally, Figure 6(c) shows the sequence options: monkey paw prints, dotted line, and no guide.

**4.1.3. Procedure.** The activity lasted four hours, at a rate of approximately 24 minutes per child, three children at the time. The evaluator first introduced herself and asked the child his/her age. Then, the instruments were presented and the process explained. Three different validations were conducted: icon validation (10 icons), interface guides design, and interaction modes.

With the help of a graphical designer and FODs professionals we constructed 10 icons that allowed the children to execute the actions described in the requirements. The researcher asked the child the meaning of each icon and made annotations either if the child got it right or wrong and any other interpretation stated by the child.

The second validation was the interface guides design. Three different approaches were designed in order to determine how to guide the children in the interface: a dotted line, a monkey paw prints, and an interface without guidelines (see Figure 6(c)). The three options were presented to each child and they were asked to explain the sequence order of a set of icons. Every evaluator presented the three possibilities in random order.

The third validation focused on the interaction. Again, three options were available: insert (Figure 6(a)), drag and drop (Figure 6(b)), and a guided mode. The insert interaction required the child to point the selected place, then the set of possibilities appeared, and the child selects the action or command to insert. The drag and drop option allowed the children to point an icon, drag it to the desired place, and drop it. The guided interactions allowed two options every time. This validation activity was performed only once per child (i.e., 13 validations per interaction mode were performed).

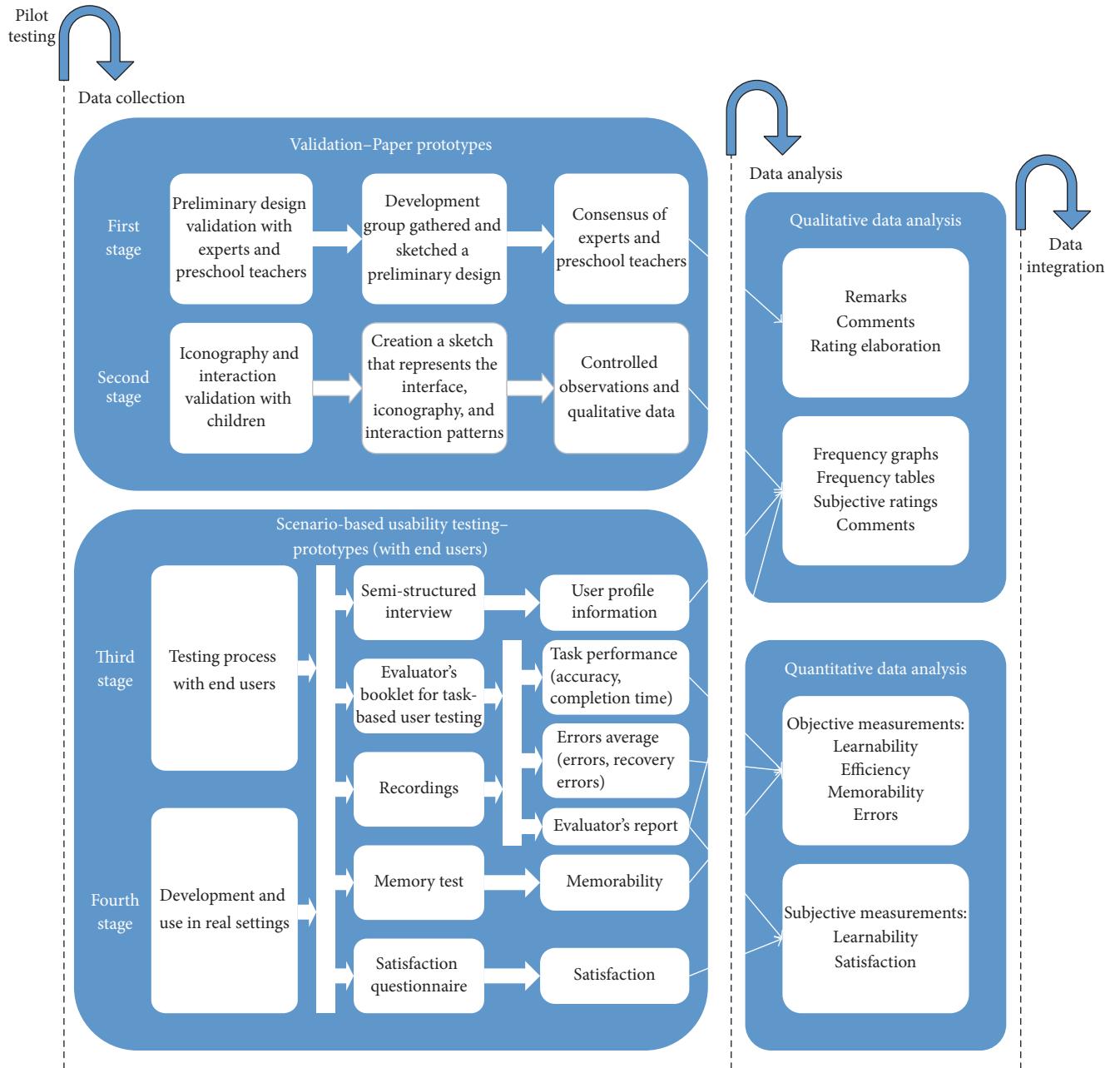


FIGURE 4: Evaluation approach.

TABLE 3: Children that participated in the evaluation of TITIBOTS per activity.

Activity	Number participants	Gender		Age		
		Male	Female	4	5	6
2nd stage: iconography and interaction validation	40 (76%)	20 (50%)	20 (50%)	11 (27.5%)	19 (47.5%)	10 (25%)
3rd stage: testing process with end users	7 (13%)	3 (43%)	4 (57%)	7 (100%)	0%	0%
4th stage: deployment and use in real setting	6 (11%)	2 (33%)	4 (67%)	4 (67%)	2 (33%)	0%
Total (percentage)	53 (100%)	25 (47%)	28 (53%)	22 (42%)	21 (40%)	10 (18%)

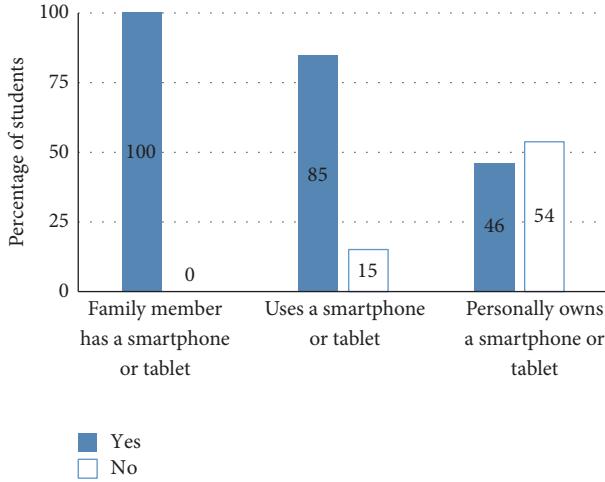


FIGURE 5: User profile information.

TABLE 4: Results of the evaluation of icons, interface, and interaction.

Icon	Age			Correct answers
	4	5	6	
Start	0%	0%	13%	0%
End	0%	0%	13%	0%
Forward	40%	90%	100%	42%
Backward	40%	90%	100%	42%
Left	40%	100%	100%	50%
Right	40%	100%	100%	50%
Turn on	50%	80%	88%	50%
Turn off	50%	80%	88%	42%
Grasp	0%	20%	63%	17%
Release	0%	20%	63%	17%
Interface				
Dotted line	30%	20%	63%	8%
Monkey paw print	30%	30%	63%	17%
No-guide	30%	40%	50%	17%
Interaction				
Task completed	100%	100%	67%	50%
Drag & Drop	67%	100%	67%	100%
Insert	100%	67%	100%	50%

**4.1.4. Results.** We were focused on the children's ability to understand the interface and to interact with the tool. We divided the results by age; 30% of the children were considered as a control group. Therefore, we have 25% of children with 4 years old, 25% with 5 years old, 20% with 6 years old, and 30% control.

Table 4 shows the results of this first validation. The start and end icons were not recognized at all. The grasp and release had a 23% of correctness. All the other commands had over 60% recognition. As for the interface and interaction design we found that the results were not significantly different.

Using the results of validation several changes in the design of the tool were implemented, the most significant changes were

- (i) reducing graphical load of the interface (i.e., decreasing background colors and figures, removing visual distractions to allow focus on the relevant elements),
- (ii) performing visual closures (for attention), and use primary and secondary colors,
- (iii) enlarging icon size.

**4.2. Testing Process with End Users.** This evaluation was conducted in a workshop with 7 children. We gave one tablet and one robot to each child. Then, children were asked to try to accomplish three different tasks.

**4.2.1. Participants.** Following Nielsen [32] recommendation of evaluating usability with 3 to 5 participants, we designed two groups of four children each one. However, due to a last minute problem, one child did not participate in the evaluation. The evaluation was conducted by a total of 11 people including 7 observers, 1 mediator, 3 robot experts, and one cameraman. The mediator explained the process to the children. The observers watched while children attempted to achieve the given goals. The robot experts were present to fix the robots if necessary.

**4.2.2. Setting and Instruments.** The evaluation was conducted at FOD's Central Office, in a robotics laboratory. Four working spaces were marked using adhesive tape in the floor, each separated by 78 inches (see Figure 7). The central space was used to set cameras and for the observers to move without interfering with the work of children. We defined tutorials and observation guides for every activity.

**4.2.3. Procedure.** The evaluation process was carried in two sessions (an hour and twenty minutes each one). In the first session, a technical mediation was performed. A teacher explained the general functions of TITIBOTS, the interaction, and the meaning of each instruction. In the second session, a recreational mediation was conducted; in this mediation the teacher played with the children, and each game introduced the instructions that TITIBOTS allowed. Before each session the teacher in charge did a welcome activity; each member of the crew was introduced. After the introduction, the teacher asked each child if they had tablets (user profile).

The evaluation was designed to assess if the children were able to achieve each task in a 30 minutes' period. The tasks were as follows:

- (1) Move the robot from the start point, in a straight line, going through a tunnel (must turn the light on inside the tunnel).
- (2) Move the robot from the start point, in a straight line, grab a ball in the end of the game area, and return to the start point.
- (3) Move in a straight line to the ball, grab it, turn left, move, and release the ball.

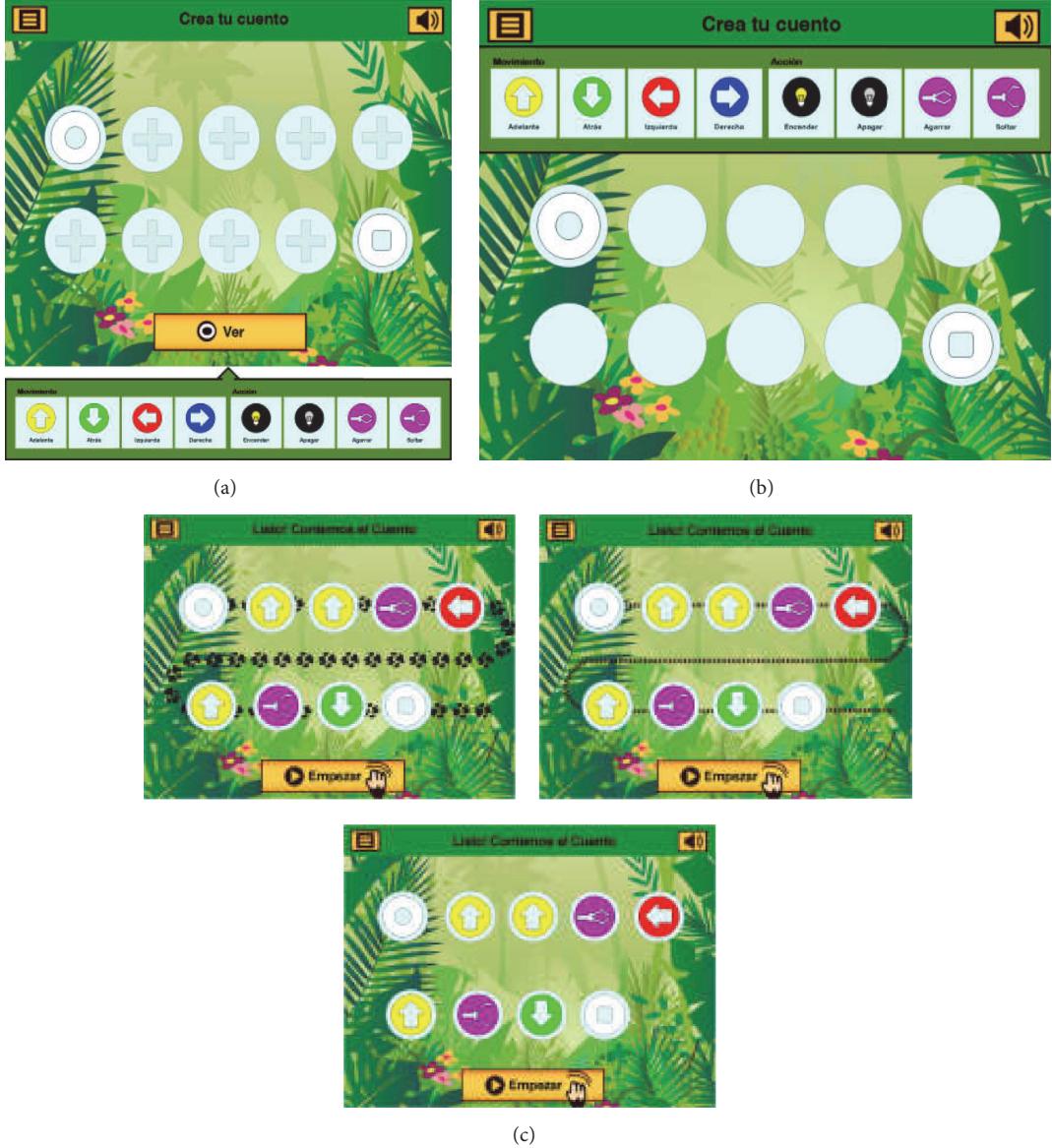


FIGURE 6: Interaction patterns: (a) Insert and (b) Drag & Drop. (c) Sequence options evaluated: claws, dots, and none.



FIGURE 7: Children working in the respective workspace.

After the mediation activity started and the challenges were explained, each child started the challenges, and the

observer looked the activity and took notes. At the end of the session a satisfaction questionnaire was conducted.

**4.2.4. Results.** Observations were performed in order to evaluate the usability of the software [33], to determine the necessity of a teacher's intervention when using TITIBOTS, and whether it helps or not in the learning process to have a strong guidance. Usability metrics are shown in Table 5.

Three challenges were designed for the children to solve. Nevertheless, we did not expect that they concluded all three challenges in the time frame provided. The evaluation showed that the application has a good usability. The interface showed to be simple and intuitive. Moreover, all the participants showed interest in the application and want to keep using it after the activity.

We observed that recreational mediation has a strong influence in the use of the tool and the level of achievement

TABLE 5: Usability goal outcomes.

Usability attribute	Usability goal	Usability outcome
Learnability	Average time to complete a challenge the first time = 11.7 minutes (between 10 and 30 minutes)	Successful
	Average time of training = 30 minutes (between 30 and 60 minutes)	Successful
Efficiency	Percentage of successful challenges = 33% (>70%)	Unsuccessful
	Average time to complete one challenge = 11.6 minutes (between 10 and 20 minutes)	Successful
Memorability	Percentage of correct answers about the application = 93% (>80%)	Successful
	Average of errors = 4.3 (between 5 and 10 errors)	Successful
Errors	Average recovery errors = 0.7, 16% of total errors (>60%)	Unsuccessful
	Like = 100% (>65)	Successful
Satisfaction	Easy = 64% (>65)	Unsuccessful

in the challenges. In the recreational mediation session, the use of the application was simpler. Otherwise, in the technical mediation session, the children used the application as a remote control (i.e., placing a command and sending it several times to the robot). This was exactly what we did not want the children to do, because we want them to create the sequence of steps in their minds and write the whole sequence on the tool before sending it to the robot.

Two of the participants in the nonmediated session ended frustrated and did not finish the challenges. On the other hand, all the children in the mediated session continued trying until the time frame finished. As expected, the children were only able to finish the first challenge, half of the children in the nonmediated session and all of them in the mediated session. The main difference was that those who received the mediation succeeded with fewer attempts and less time.

We found several problems in our system during the evaluation. For instance, robot's claws smashed easily. This forced a redesign in the software to avoid an open command if the claw was already opened. Bluetooth connection was unstable; the software was redesigned to reconnect automatically. Commands needed to be placed in pairs (e.g., on and off, catch and release) were easier to understand by the children than those in the form of keyboard (Forward and Backward, Left and Right). The software interface was redesigned to allocate all the commands in pairs. Finally, real time feedback is required from the application to let the user know what is happening.

**4.3. Deployment and Use in Real Setting.** Once we tested the proper functionality and design of TITIBOTS, we arranged with FOD to create a 4-5-year-old robotic workshop in which children use the tool to be introduced with programming concepts.

**4.3.1. Participants.** The workshop was designed for 6 children. Originally, the children were three boys and three girls; however, due to a last minute problem a girl substitutes one of the boys. The girls ranged between four years and three months to five years and six months. The two boys were four years and four months and four years and ten months.

**4.3.2. Setting, Instruments, and Procedure.** The workshop was carried at FOD's Central Office, and it was designed for 8

hours distributed in 4 days. Figure 8 shows some pictures of the workshop during different activities.

During the workshop, the teacher played with the children, and each game introduced the instructions that TITIBOTS tool allowed each day. The first day the teacher asked each child if they had tablets (user profile). The last day was for the evaluation; it consisted to make three challenges over a period of 60 minutes:

- (1) Move the robot from the start point, in a straight line, going through a tunnel (the player must turn the light on inside the tunnel).
- (2) Move the robot from the start point, in a straight line, grab a ball in the end of the game area, and return to the start point.
- (3) Move in a straight line to the ball, grab it, turn left, move, and release the ball. We had the participation of two observers during the whole workshop, watching children's actions and expressions. The challenges were explained, each child started the challenges, and the observer recorded the activity and took notes. At the end of the session a satisfaction questionnaire was conducted asking for the robot and the application.

**4.3.3. Results.** The most important result of the workshop (obtained from evaluator's report, recordings, and usability metrics) was that the children were always happy and attentive with the PAT. They found it easy to use and fun, according to the satisfaction questionnaire. Furthermore, they did not have problems understanding the commands or another miscellaneous buttons such as clear screen, load program, and disconnect (according to usability attribute: memorability). Usability metrics obtained in the test are shown in Table 6.

FOD's experts considered this workshop implementation a success, because they felt that all the participants achieved the basic knowledge intended for the activity. Besides, teacher and observers consider the events in which each child mimicked the robot and acted the commands were crucial to the learning process.

We found that older participants achieved an exceptional success rate. We think that the tasks were too easy for them. The younger participants (4 years old) had difficulties; however, the interaction with the older ones helped. This

TABLE 6: Usability goal outcomes.

Usability attribute	Usability goal	Usability outcome
Learnability	Average time to complete a challenge the first time = 10.9 minutes (between 10 and 30 minutes)	Successful
	Average time of training = 43.7 minutes (between 30 and 60 minutes)	Successful
Efficiency	Percentage of successful challenges = 88% (>70%)	Successful
	Average time to complete one challenge = 18 minutes (between 10 and 20 minutes)	Successful
Memorability	Percentage of correct answers about the application = 100% (>80%)	Successful
	Average of errors = 3.4 (between 5 and 10 errors)	Successful
Errors	Average recovery errors = 2.3, 69% of total errors (>60%)	Successful
	Like = 100% (>65)	Successful
Satisfaction	Easy = 83% (>65)	Successful
		Successful

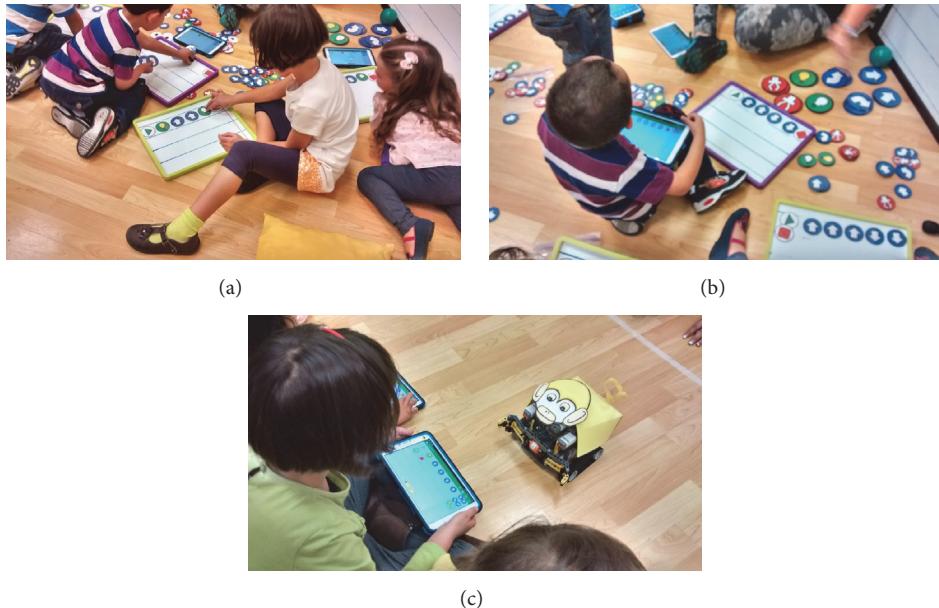


FIGURE 8: (a) Children planning the solution of challenge in a small whiteboard. (b) Child implementing the proposed solution using TITIBOTS. (c) Child watching a robot executes a program send using TITIBOTS.

leads us to believe that a workshop integrating 4- to 5-year-old children is beneficial. The time frame of each session (two hours) is considered as the limit for the children, because at the end of the session they were exhausted.

## 5. Discussion

The evaluation process showed problems of design, usability, and functionality. All these problems were resolved, obtaining a more useful PAT. The evaluator's report, the recordings, and the usability metrics show that TITIBOTS: it complies with the requirements given by the experts and it is easy and pleasant to use for children.

When comparing the third and fourth stage of the evaluation, we determined that the errors caused by the PAT and the robots were reduced significantly. With this result, we consider that our system is ready to be deployed in regular learning activities.

Results show that TITIBOTS allow children to play and to program robots in order to solve specific tasks. Moreover,

metrics that are under the expected rate are caused by the complexity of the tasks and not by the use of TITIBOTS.

Satisfaction metrics shows that 100% of the kids liked the programming tool and robot (the 13 children in the third and fourth stages). Besides, 31% found the tool difficult to use (although in observations and recordings this was not reflected). Instead, 77% of the children state that the robot was easy to use. As part of the satisfaction questionnaire, children were asked to draw what they liked best about the activity, 62% of the children drew a robot or a monkey, 8% drew a tablet, and 31% drew a tablet and a robot. When asked to describe their drawings, they said that they loved their monkey's robots and liked to tell them what to do.

Even though the problems solved by children are basic, moving the robot in a straight line or turning lights on and off, we believe that problems with a higher level of difficulty can also be solved. However, the time spent in this evaluation does not provide enough empirical evidence to support that claim. Further evaluation will allow us to fully determine the impact of our tool used by children since the age of 4. We

proved that programming concepts can be taught to children since age 4 and probably younger. However, at this age the amount of effort required to keep the children focused and trying to solve a problem is high enough.

Moreover, it is difficult for them to apply what they learned in real-life problems, because we are trying to develop in them abstract thinking by programming robots. Our goal is to teach children to think and solve problems in a structured way, using algorithms that will benefit them later to resolve any problems that are present in their life, but the latter cannot be proven. In addition, programming is the future literacy and we are working on it from an early age.

One clear impact that could be perceived in the period of time in which we performed this evaluation was that teachers were able to keep the attention of children for longer periods of time by using technology. Several studies support this discovery [4, 7, 8, 32, 33]; however, we applied it not only to keep the attention of children but also to teach them concepts that are usually too difficult to be understood by children that age, when presented in other ways.

The contribution to Mobile Information System is the design and evaluation of mobile interfaces for children to early childhood programmed robots in an intuitive and easy way.

TITIBOTS enables programming and robotics to teach preschoolers while having fun playing; which promotes the development of skills such as problem solving skills, logical-mathematical thought, abstraction, and creativity in the medium to long term.

## 6. Conclusions

In this paper we presented TITIBOTS, a PAT that allows children in the early childhood to create programs using tablets and execute them with robots (allows the use of any robot). We followed an extensive design and evaluation process. We used several techniques to evaluate the usability, including participatory-design, experience prototype, and usability testing.

We consider the development of the project was successful. Children between 4 and 6 years found TITIBOTS easy to use. They manipulated the tool to infer the meaning and the use of the icons and commands. In addition, children were always interested, happy, and attentive while using the PAT, but most importantly, they had fun. It was evident for us that this PAT allows the children to play and to program robots in order to solve specific tasks.

The evaluation of TITIBOTS is also considered successful, because it was possible that children learn basic programming concepts, such as sequential problem solving. They were able to verbalize their thinking when were asked. We believe that the use of concrete and physical exercises (without the tool) with the children facilitates the use of the developed environment. The children that use the tool acquired a first approach to the basic process for solving a given problem, following the common programming steps: planning, implementation (writing the program), and testing (sending the commands to the robot). The designed teaching-learning

activities were successful, caught the attention of children, and achieved the expected objectives for each one.

Studies show that the use of robotics in the teaching-learning process of children has accomplished that children learn specific curricular content in STEM areas. In our study, we have not tested any of this, but it will be held as part of the future work of this research.

Through TITIBOTS, children develop soft and technical skills that are necessary nowadays. Moreover, we think that modifying the environment and designing it appropriately we can create strategies for the children to collaborate in solving a given problem. This is part of our further work: a collaborative version of TITIBOTS.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

This work was supported by ECCI-UCR (Escuela de Ciencias de la Computación e Informática) and by CITIC-UCR (Centro de Investigaciones en Tecnologías de la Información y Comunicación), Grant No. 834-B3-260. Thanks are due to the FOD (Fundación Omar Dengo) for helping us in the validation and evaluation of TITIBOTS.

## References

- [1] G. S. Tjaden, "Measuring the information age business," *Technology Analysis and Strategic Management*, vol. 8, no. 3, pp. 233–246, 1996.
- [2] M. Prensky, "Digital natives, digital immigrants Part 1," *Horizon*, vol. 9, no. 5, pp. 1–6, 2001.
- [3] M. Binkley, O. Erstad, J. Herman et al., "Defining twenty-first century skills," in *Assessment and Teaching of 21st Century Skills*, pp. 17–66, Springer, Berlin, Germany, 2012.
- [4] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, Inc., New York, NY, USA, 1980.
- [5] I. M. Verner and D. J. Ahlgren, "Conceptualising educational approaches in introductory robotics," *International Journal of Electrical Engineering Education*, vol. 41, no. 3, pp. 183–201, 2004.
- [6] C.-T. Hsin, M.-C. Li, and C.-C. Tsai, "The influence of young children's use of technology on their learning: a review," *Educational Technology and Society*, vol. 17, no. 4, pp. 85–99, 2014.
- [7] M. Resnick, *Learn to Code, Code to Learn*, EdSurge, 2013.
- [8] M. Binkley, O. Erstad, J. Herman, S. Raizen, M. Ripley, and M. Rumble, "Defining 21st century skills," 2010.
- [9] M. U. Bers, *Designing Digital Experiences for Positive Youth Development*, Oxford University Press, Oxford, UK, 1st edition, 2012.
- [10] M. U. Bers, *Blocks to Robots: Learning with Technology in the Early Childhood Classroom*, Teachers College Press, 2008.
- [11] M. Koorsse, C. Cilliers, and A. Calitz, "Programming assistance tools to support the learning of IT programming in South African secondary schools," *Computers and Education*, vol. 82, pp. 162–178, 2015.
- [12] M. Buchenau and J. F. Suri, "Experience prototyping," in *Proceedings of the Conference on Designing Interactive Systems*:

- Processes, Practices, Methods, and Techniques (DIS '00)*, pp. 424–433, New York, NY, USA, August 2000.
- [13] R. Mack and J. Nielsen, “Usability inspection methods,” *ACM SIGCHI Bulletin*, vol. 25, no. 1, pp. 28–33, 1993.
- [14] M. Resnick, F. Martin, R. Sargent, and B. Silverman, “Programmable bricks: toys to think with,” *IBM Systems Journal*, vol. 35, no. 3-4, pp. 443–452, 1996.
- [15] S. Einhorn, “MicroWorlds, computational thinking, and 21st century learning,” LCSI White Paper, 2011.
- [16] K. Mayerová, “Pilot activities: LEGO WeDo at primary school,” in *Proceedings of 3rd International Workshop Teaching Robotics, Teaching with Robotics: Integrating Robotics in School Curriculum*, pp. 32–39, Riva del Garda, Italy, 2012.
- [17] S. H. Kim and J. W. Jeon, “Programming LEGO Mindstorms NXT with visual programming,” in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '07)*, pp. 2468–2472, October 2007.
- [18] D. Nam and T. Lee, “The effect of robot programming education by pico cricket on creative problem-solving skills,” in *Proceedings of the 19th International Conference on Computers in Education (ICCE '11)*, pp. 1–9, Chiang Mai, Thailand, December 2011.
- [19] G. Tomoyose, *Minibloq, el Lenguaje de Programación Argentino Para Robots que Llega a Todo el Mundo*, LA NACION, 2014.
- [20] A. Yera-Gil, *Iniciación a la programación visual e interactiva y desarrollo de robótica educativa con Scratch y Enchanting*, Universidad Pública de Navarra, Pamplona, Spain, 2010.
- [21] S. Romero, I. Angulo, I. Ruiz, and J. M. Angulo, “Competencias y habilidades con el robot ‘MOWAY’,” in *TAAE: Tecnologías Aplicadas a la Enseñanza de la Electrónica*, p. 8, Universidad de Deusto, Bilbao, Spain, 2008.
- [22] M. Rollins, *Beginning LEGO MINDSTORMS EV3*, Apress, 2014.
- [23] V. Gupta, S. Gupta, and M. Greaves, “Play-i,” in *Madrona Venture Group and Charles River Ventures*, 2013.
- [24] F. Yacob, M. Loglio, D. Di Cuia, V. Leonardi, L. Rabago, and J. Valman, “PRIMO,” *Solid Labs*, 2013.
- [25] A. Sullivan, M. Elkin, and M. U. Bers, “KIBO robot demo: engaging young children in programming and engineering,” in *Proceedings of the 14th International Conference on Interaction Design and Children (IDC '15)*, pp. 418–421, Medford, Ore, USA, June 2015.
- [26] K. C. Louden and K. A. Lambert, *Programming Languages: Principles and Practices*, Cengage Learning, Boston, Mass, USA, 3rd edition, 2011.
- [27] International Organization for Standardization, “ISO 13407:1999 Human-centred design processes for interactive systems,” 1999.
- [28] S. Greenberg and R. Bohnet, “GroupSketch: a multi-user sketchpad for geographically-distributed small groups,” in *Proceedings of the Graphics Interface*, pp. 207–215, Calgary, Canada, June 1991.
- [29] J. Nielsen, *Usability Engineering*, Morgan Kaufmann, 1993.
- [30] A. Granić and M. Ćuković, “Usability testing and expert inspections complemented by educational evaluation: a case study of an e-learning platform,” *Educational Technology and Society*, vol. 14, no. 2, pp. 107–123, 2011.
- [31] R. E. Eberts, *User Interface Design*, Prentice Hall College Div, Englewood Cliffs, NJ, USA, 1994.
- [32] J. Nielsen, *Usability Engineering*, Elsevier, 1994.
- [33] M. Bekker, W. Barendregt, S. Crombeen, and M. Biesheuvel, “Evaluating usability and challenge during initial and extended use of children’s computer games,” in *People and Computers XVIII—Design for Life: Proceedings of HCI 2004*, pp. 331–345, Springer, London, UK, 2005.