IE comunicaciones

Revista Iberoamericana de

Informática Educativa

ADIE

Revista Iberoamericana de
Informática Educativa

IE comunicaciones

Revista Oficial de la
Asociación para el Desarrollo de la Informática Educativa
ADIE

IE Comunicaciones es una publicación periódica editada y distribuida por la Asociación para el Desarrollo de la Informática Educativa ADIE. No se solidariza, necesariamente, con la opinión expresada por los autores de los artículos.

Edita: Asociación para el Desarrollo de la Informática Educativa, ADIE
ISSN: 1699-4574

Número 37, Enero – Junio 2023

Sumario

Artículos

ADIE

Fines

- Fomentar la Informática Educativa.
- Promover la formación de las personas en las Nuevas Tecnologías Educativas.
- Intercambiar trabajos, ideas y experiencias.
- Evaluar la calidad pedagógica de los productos existentes

Actividades

- Cursos, Seminarios, Mesas Redondas, Conferencias, Talleres, etc.
- Congresos periódicos de carácter nacional e internacional.
- Evaluación y Biblioteca de Software Educativo.
- Base de Datos de investigaciones, estudios sistemáticos.

Revista

IE comunicaciones
Revista Iberoamericana de Informática Educativa
Editora: Diana Pérez (diana.perez@urjc.es)

Miembros

Socios Honorarios
Socios Institucionales
Socios Individuales
Alumnos

Socios Institucionales

Universidad Nacional de Educación a Distancia
Universidad Autónoma de Madrid
Facultad de Informática (Universidad de Málaga)
Universidad Politécnica de Madrid
Dpto. De Informática y Automática, Fac. de Ciencias (Universidad Complutense de Madrid)
Universidad de Castilla-La Mancha

Web

Adie.es

Envío de contribuciones

diana.perez@urjc.es

Número 37, Enero-Junio 2023, pp 1-13
Artículos

Computational Thinking, Visual Impairment and Melodic

Rui Costa[1], Cristiana Araújo[2] , Alvaro Costa Neto[3], Angélica Cunha[4], Pedro Rangel Henriques[5]

[1, 2, 4, 5]ALGORITMI Research Centre/LASI, DI-University of Minho, Portugal
[3]Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
Barretos, Brasil

[1]rui.diogo.costa@hotmail.com, [2]decristianaaraujo@hotmail.com, [3]alvaro@ifsp.edu.br,
[4]angelicasc22@hotmail.com, [5]pedrorangelhenriques@gmail.com

Resumen: La creciente presencia de dispositivos informáticos ha transformado las habilidades de razonamiento lógico de deseables a indispensables. Aunque siempre ha sido crucial poseer la capacidad de pensar lógicamente y descomponer problemas en partes manejables, el uso constante y el rápido desarrollo de dispositivos digitales han incrementado enormemente su valor. El Pensamiento Computacional (PC) ha surgido como un conjunto de habilidades mentales para hacer frente a los nuevos desafíos planteados por las computadoras y sus aplicaciones. Se considera ahora un nuevo dominio de conocimiento que debe ser desarrollado en las escuelas, junto con disciplinas clásicas como las Matemáticas y los Idiomas. Dada esta creciente importancia, resulta lógico considerar injusta y desalentadora la falta de apoyo inclusivo en la formación en PC, especialmente para las personas con necesidades especiales. *Melodic* fue desarrollado con el propósito de combinar música, sonidos y lógica en una plataforma de juego virtual y física, con el objetivo de capacitar a personas con y sin discapacidad visual en el ámbito del Pensamiento Computacional. Esta plataforma se compone de diversas piezas de madera que permiten a los jugadores programar una secuencia melódica utilizando operaciones similares a las instrucciones de programación de computadoras. Además, se complementa con una aplicación móvil que enriquece la experiencia de juego, desafiando a los participantes a construir diferentes melodías. La realización de cada desafío requiere la aplicación de diversas habilidades de PC, lo que permite a los jugadores entrenar su capacidad de razonamiento lógico y resolución de problemas. En este artículo se presenta el diseño, desarrollo, evolución y validación de Melodic, así como una introducción a *PathIt*, un proyecto complementario que extiende los esfuerzos inclusivos de Melodic hacia nuevos paradigmas. Ambas plataformas educativas de juegos tienen como objetivo democratizar y respaldar la formación en PC de personas con y sin discapacidad visual a través de características físicas y auditivas, con la esperanza de preparar mejor a personas de todas las edades para la presencia impactante e irreversible de los dispositivos informáticos.

Palabras clave: Pensamiento Computacional, Educación de Estudiantes con Discapacidad Visual, Aprendizaje Inclusivo, Educación Especial, Evaluación del Pensamiento Computacional

Abstract: The ever growing presence of computational devices shifted logical reasoning abilities from desirable to required. While it is factual that the abilities to think logically and break down problems into manageable pieces have always been important, the constant use and rapid development of digital devices have increased their values tenfold. Computational Thinking (CT) emerged as a set of mental skills to cope with the novel challenges that computers and their uses have raised. It is being considered a new domain of knowledge to be constructed in schools, along with Mathematics, Languages and other classic disciplines. Given this growing importance, it is only logical to consider unfair and disheartening the absence of inclusive support in CT training, especially to those with special needs. *Melodic* was developed to combine music, sounds and logic in a mix of virtual and physical game platform, aimed at training visual and visually impaired people in Computational Thinking. It is composed of several wooden pieces that allow players to program a melodic sequence using operations similar to computer programming instructions. A mobile app completes the gaming experience, challenging trainees with the construction of different melodies. Several CT skills must be applied in order to complete each challenge and in doing so, players train their mental abilities in logical reasoning and problem solving. This paper presents Melodic's design, development, evolution and validation, as well as an introduction to *PathIt*, a sibling project that carries on Melodic's inclusive efforts to new paradigms. Both educational gaming platforms aim to democratize and support CT training of visual and visually impaired people through physical and audio features, in the hopes to better prepare people of all ages to the impactful and irreversible presence of computational devices.

Keywords: Computational Thinking, Visually Impaired Students Education, Inclusive Learning, Special Education, Assessment Computational Thinking.

1. Introduction

Learning and growing as a capable human being requires overcoming several challenges through the development of important skills. Many factors influence this process, including access to teaching and learning resources, experimentation, social inclusion *etc.* While not as cemented in cultural, historical and social knowledge as more classic disciplines such as Mathematics and Literature, Computational Thinking (CT) has been gathering attention in recent years by its important role both in Information Technology development and in general problem-solving education.

In a general sense, Computational Thinking may be summarized as a problem solving method, based on typical computer programming related skills. Wing (2006) defined not only the term, but also specified the six initial cornerstone features that support it:

● *Decomposition:* breaking a problem or system down into different and easier to solve chunks using a "divide and conquer" strategy;
● *Pattern recognition:* finding similarities between problems that have been successfully solved to reuse successful strategies;
● *Abstraction:* filtering information by eliminating pointless or redundant elements, in order to simplify a problem and concentrate on what is important;
● *Algorithmic thinking:* developing a set of clear, unambiguous rules or instructions that solve a problem or accomplish a goal;
● *Logical reasoning:* analyzing and understanding the problem in hands to deduct as much new information based on existing knowledge in a clear and organized way;
● *Evaluation:* verifying if the proposed solution is effective and efficient to some specified requirement.

Wing also emphasized that Computational Thinking must be taught from early ages in order to improve its efficacy. Nonetheless, adequate educational material, tools and training programs are paramount to success in this endeavor. Also fundamental is a better preparation of educators to face these challenges, paving the way for students to reason according to CT.

There are several resources to aid in training CT, either common digital games—such as Lightbot [1] and Code.org [2]—supported on paper—like FlorestaBook (Martinelli, Zanki, Cordenonsi, & Bernardi, 2021) and ALGO+RITMO (Silva, Moura, Paula, & Jesus, 2019)—or robot-based devices—such as Robot KIBO (Sullivan, Bers, & Mihm, 2017) and Robot DOC (Silva, Fonseca, Costa, & Martins, 2021). However, there is one pervasive problem with all of these resources: they are inherently visual. This fact undermines their accessibility to visually impaired people.

That is not to say that the current scenario for CT training of visually impaired people is empty. A few proposals for fulfilling this lack have been published and tested, such as Story Blocks (Koushik, Guinness, & Kane, 2019), TACTOPI (Abreu, Pires, & Guerreiro, 2020), ACCembly (Rocha, Pires, Neto, Nicolau, & Guerreiro, 2021), Project Torino (Morrison *et al.*, 2020) and the works of Sabuncuoglu (2020). While these are undoubtedly important contributions, it is clear that the landscape of non-visually centered educational resources is still lacking and could be improved with more ideas, projects and general contributions. This fact led to the creation of Melodic, a system consisting of both software and hardware

[1] Accessible at: https://lightbot.com/

[2] Accessible at: https://studio.code.org/courses/

components that combine music, audio and tactile feedback to engage and train visually impaired people in Computational Thinking.

This article presents both a general overview of the challenges concerning the education of visually impaired people, and Melodic proposal to aid in overcoming them. It is organized into six sections: Section 2 contains a bibliographic review about challenges and limitations that visually impaired people face when learning; Section 3 presents Melodic and its components; Section 4 reports on the test of Melodic effectiveness through the physical implementation of an established CT evaluation technique; Section 5 discusses recent efforts being carried in a new system named PathIt, designed to complement Melodic work; finally, Section 6 concludes the paper with the closing arguments, while also presenting future projects.

2. Visually Impaired People

Visual Impairment (VI) is a significant sensory disability that affects millions of individuals globally. It encompasses a spectrum of VI, ranging from low vision, characterized by partial visual functionality and challenges in perceiving visual details, to legal blindness or total blindness, representing severe vision loss or the complete absence of visual perception (World Health Organization, 2022).

The experience of individuals with VI markedly differs from that of sighted individuals, particularly in terms of their perception and interaction with the world. Unlike sighted individuals who heavily rely on vision from an early age for learning and exploring the environment, individuals with VI learn and navigate their surroundings primarily through their other senses, including hearing, touch, taste, and smell.

However, it is important to recognize that individuals with visual impairment are capable of learning and acquiring knowledge just as effectively as their sighted counterparts (Pogrund, Fazzi, & American Foundation for the Blind, 2002). They possess the same intellectual abilities and potential for academic achievement. Unfortunately, the learning environment and educational materials are often designed with a heavy reliance on visual cues, which can create barriers and hinder the educational journey of students with VI.

Teaching individuals with VI requires a customized approach that addresses their distinct learning requirements. To create effective learning resources, several factors must be considered to cultivate meaningful engagement and facilitate successful learning outcomes for visually impaired learners.

One crucial aspect is the use of tactile artifacts (Alotaibi, Al-Khalifa, & AlSaeed, 2020) and materials. Texture plays a significant role in providing sensory information and can enhance the learning experience for individuals with visual impairments. By incorporating various textures, such as rough, smooth, or bumpy surfaces, tactile artifacts can convey different concepts or represent objects. For example, raised symbols or Braille can be utilized to provide tactile representations of letters, numbers, or mathematical symbols. These tactile cues allow learners to explore and interpret information through touch, enabling them to access content that is typically conveyed visually.

In addition to tactile cues, auditory cues and guidance are vital for individuals with visual impairments (Taylor & Sternberg, 1989). Auditory cues can provide essential information, feedback, and instructions. For instance, in a classroom setting, teachers can use verbal descriptions to supplement visual demonstrations or gestures. Providing clear and concise auditory cues helps students with VI understand the context, follow instructions, and engage in activities effectively. Moreover, audio resources, such as recorded lectures, audiobooks, or digital voice assistants, can further support learning by providing auditory access to educational materials.

Assistive technology plays a crucial role in facilitating the learning process for individuals with visual impairments (Hallahan & Kauffman, 2006). Advancements in technology have led to the development of various tools and devices that offer accessibility features. For example, screen readers and screen magnifiers enable visually impaired learners to access digital content by converting text into speech or enlarging text and graphical elements on electronic devices. Additionally, refreshable Braille displays allow individuals to read digital text through touch-based Braille output. These assistive technologies empower visually impaired learners by providing them with independent access to educational resources, digital platforms, and online learning materials.

Designing inclusive learning resources that prioritize non-visual sensory modalities and embrace a multi-sensory approach is essential for providing equal learning opportunities and enhancing the educational experience for individuals with visual impairments. Embracing inclusivity in education not only benefits individuals with VI but enriches the educational journey for all learners, fostering a more inclusive and holistic learning environment.

3. The Melodic System

Melodic is a multi-platform system[3] that aims to support the complex tasks of teaching and learning Computational Thinking, specially targeted at aiding visually impaired and musically inclined people (Costa, Araújo, & Henriques, 2021). It is based on a predefined set of physical and tactile blocks that represent musical operations (Figure 1) and a mobile App that identifies the spatial distribution of these blocks to compose equivalent melodies.

3.1. The Blocks

The blocks that represent musical operations are integral to the Melodic System. Figure 1 shows all currently implemented blocks, ranging from playback and loop operations (1), instrument selection (2), numbering and counting (3), musical notes (4) (beginning with *Dó* and ending with *Si*}), and velocity (5). These blocks incorporate two different roles: instructions used by users to program the melody—akin to mnemonics in assembly or keywords and tokens in programming languages—and a translation mechanism, converting from a pure physical representation into an actual melody, communicating to the Melodic App.

Figure 1. Melodic blocks used to compose musical sequences.

Being a system focused on training visually impaired students, Melodic blocks are constructed in a specific way to facilitate their handling. Several characteristics are important and help users to identify each kind of block:

● *Material:* wood was chosen as standard material for its natural texture, ease of shaping, low cost, and high durability and availability. Although wood is the standard choice, any other kind of easy-to-shape and affordable material could also be used, such as clay as an example. Another possibility would be to 3D print the blocks using different filaments, which would cost more, but generate higher quality blocks;
● *Texture:* block types (numbers, notes and special operations) are carved with different textures to be promptly identifiable by touch;
● *Color:* two contrasting tones of wood (natural or dyed) are applied in the construction to aid identification by low-vision users;
● *Shape:* blocks are formed of 3D wooden icons (shapes) on top of a standard circular base, providing instant feedback as to which specific operation a certain block represents.

The translation from a block's physical entity to a logical representation in the Melodic App is achieved through QR Codes glued to the bottom of each piece, such as shown in Figure 2. The Melodic App asks the user to scan these codes in order to identify each block, who in turn, must take care to scan only one QR Code

[3] Accessible at: https://epl.di.uminho.pt/~gepl/Melodic/

at a time to avoid misidentifying blocks.

Figure 2. Top and bottom of a block, showing the QR Code used to identify each block.

Inclusion was a centerpiece to the design of the Melodic System, both in accessibility and affordability. The design of the physical blocks and the use of low-cost, highly available materials and simple building techniques support this goal in such a way that even without modern fabrication techniques, such as 3D printing, people should be able to construct the blocks by hand and use the system to support Computational Thinking education.

3.2. The App

The mobile App is the digital counterpart to the implementation of the Melodic System. Its intended use is to scan a distribution of blocks, in an order dictated by the user, and playback either the resulting melody or a message explaining that some error has occurred.

To minimize incompatibilities between different mobile devices' brands and models, Melodic App was constructed using the React Native framework[4], which uses JavaScript code to build apps for different platforms.

3.3. Use and Examples

The use of the Melodic System may be described in three main steps:

1. A sequence of blocks is distributed on a surface in a logical manner to form a melody. There is no obligation as to which kind of relative positioning is used (vertical, horizontal, tree-like etc.), but to avoid errors in the next step, a common rule for distribution may be established by the user;
2. Each block's QR Code in the sequence is scanned through the mobile camera (Figure 3), taking care to avoid showing the App more than one QR Code at a time. To this intent, it is useful to attach the QR Codes to the blocks' bottoms and turn them over one at a time;
3. The App recognizes the sequence and plays back the resulting melody (or the error message).

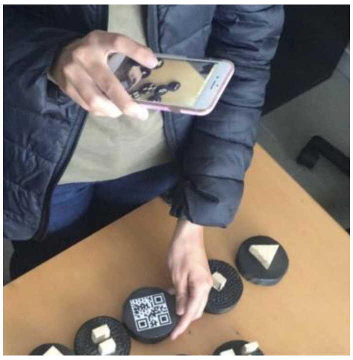


Figure 3. Blocks scanning process that allows the App to identify the sequence.

In a teacher-guided situation, exercises might be conducted using either pre-recorded melodies or by constructing the required melodies live using the blocks and the App. In the later case, the teacher should require students to only listen to the melody and not show them the sequences.

A simple melody that could be used as an introductory example or exercise is shown at the top of Figure 4. This sequence, scanned from left to right, indicates that a new melody is starting (the *Play* block), then a *Dó* note is played, followed by the *Mi* note. The sequence finishes the melody with the *Stop* block. As long as the user (student or teacher) scans the blocks in the right order (*Play*, *Dó*, *Mi*, *Stop*), the melody should be correct.

Complex melodies may Computational Thinking concepts such as change of state (by changing the instruments or the speed) and repetition (through the *Loop* or *X times* blocks). The bottom image in Figure 4 shows a complex sequence that changes the state of the melody via the *Fast* block, defines the instrument to be played with the *Piano* block and applies repetition using the *Three times* block before the *Mi* note.

[4] Accessible at: https://reactnative.dev

Figure 4. Melody sequences ranging from simple to complex.

3.4. Formal Melodic Language Definition

The fact that a melody is logically created using concepts such as changes of state and repetitions indicates that the Melodic sequences obey an intrinsic language that can be formally defined as such:

Terminals: START, STOP, REPEAT, EREP, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, SPEED1, SPEED2, SPEED3, DO, RE, MI, FA, SOL, LA, SI, GUIT, PIANO, FLUTE;

Productions:
- **Prog:** START Inst+ STOP;
- **Inst:** Note|NoteRep|Speed|Instrument|Cycle;
- **NoteRep:** Note Num;
- **Cycle:** REPEAT Num CycleInst+ EREP;
- **CycleInst:** Note|NoteRep|Speed|Instrument;
- **Note:** DO|RE|MI|FA|SOL|LA|SI;
- **Num:** 1|2|3|4|5|6|7|8|9|10;
- **Speed:** SPEED1|SPEED2|SPEED3;
- **Instrument:** GUIT|PIANO|FLUTE;

This formal definition supports the main objective of teaching and learning Computational Thinking. It proves that a melody correctly constructed through Melodic blocks is an actual algorithmic implementation, akin to the source code for an application written in a computer programming language[5]. For more detailed information on the Melodic System, see (Costa *et al.*, 2021;Costa, 2022).

4. Computational Thinking Assessment

Melodic contribution to teaching and learning Computational Thinking was assessed through a direct, progressive and extensive approach:

1. Users participated in a diagnostic session in order to assess their Computational Thinking skills before Melodic was introduced (Subsection 4.4);
2. All of Melodic components were explained and a series of exercises applied aiming to develop and train their Computational Thinking skills. These exercises ranged from the construction of a very simple, one note sequence, to more complex, iterative and sophisticated melodies;
3. Finally, a sequence of Computational Thinking tests was applied in an evaluation session to observe the users' progress.

The tests mentioned in item 3 were carefully selected in order to obtain an overview of how Computational Thinking development occurred as a result of Melodic use.

A structured strategy was defined in order to identify Melodic contribution to Computational Thinking development. Instead of using Melodic itself as an evaluation tool, which could bias the success rate by repetition and familiarity with the system, the assessment was based on simple logical exercises, designed specifically for this purpose (Zapata-Cáceres, Martín-Barroso, & Román-González, 2020). These exercises were proposed to understand users' reasoning and algorithmic design abilities.

Two different kind of exercises were applied:

1. Simple and direct exercises to test pattern matching and algorithmic structures recognition, such as the one shown at the top of Figure 5. This kind of exercises was proposed to evaluate a user's capacity to correlate the reasoning behind the drawing sequence—a line drawn on the left side of the image—and the directions that should be taken in order to draw it—the arrows in columns A, B and C;
2. Complex algorithmic reasoning exercises, where a valid path had to be constructed in order to reach a

[5] That is not to say that Melodic formal definition generates a language that is computationally equivalent to a lambda-complete programming language.

destination. This kind of exercise, as shown at the bottom of Figure 5 demands users to construct the path from beginning to end positions, without a pre-established guide. The only requirements are that the path should start at the source—the chick—follow only connected cells, avoid obstacles—the cat—go through required cells—the brown chick—and stop at the destination—the hen.

The complexity of each exercise was defined by the presence of elements in the grid, such as obstacles and required cells, and the available options of answers, that could demand recognition of algorithmic concepts such as repetitions and blocks of operations. An example of the latter is shown in option A at the bottom of Figure 5, where the first two arrows are grouped inside a rectangle—indicating that they form a block of operations—that should be repeated—represented by the preceding "2x" label.

While these exercises were used as a basis for testing users' Computational Thinking development after practicing with Melodic, they were not applied in their original form. Since Melodic is a system aimed at training Computational Thinking to visually impaired people, a series of adaptations had to be made in order to reach its intended audience.

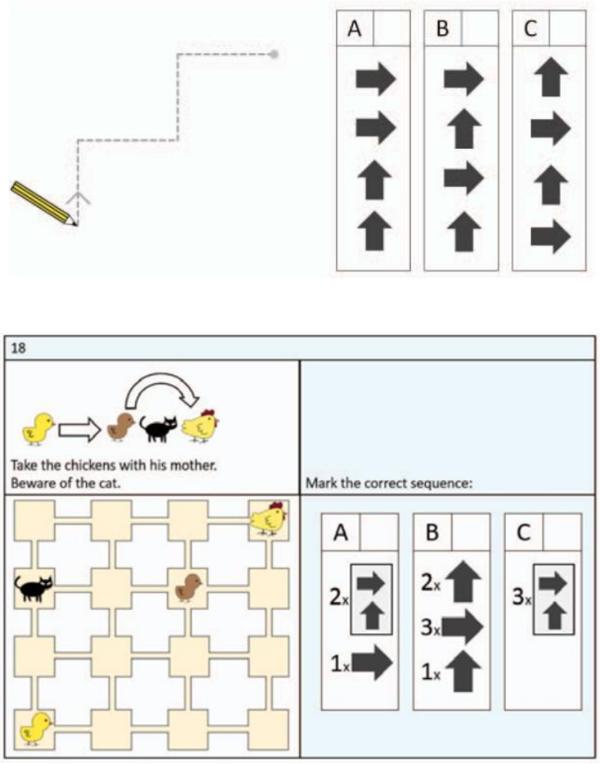


Figure 5. A simple exercise (top) to identify which of the proposed sequences (A, B or C) would draw the line on the left and a complex exercise (bottom) to identify a valid path between a source and a destination, while avoiding obstacles (the cat) and attending to required points (the brown chick) (from Zapata-Cáceres *et al.* (2020)).

4.1. Adapting the Exercises to Visually Impaired Users

Just as it was done with the blocks, the grid and the answers had to be physically represented through the use of tactile materials, such as wood or clay. For the same reasons stated in Subsection 3.1, wood was chosen as the default material for the construction. Its texture, versatility and availability made it the optimal choice to implement the grid, the paths and the elements in the cells. Figure 6 shows the engraved wooden grid (board) and the pieces used to represent both the problems and the answers of the exercises. As can be seen in the image, there are chiseled channels that represent the connections between cells, which in turn are indicated by square recesses. There are also wooden pegs (dowels) to construct the path and four wooden pieces (shown in Figure 7): circles for the source, triangles for the destination, squares to define obstacles, and crosses to establish required cells.

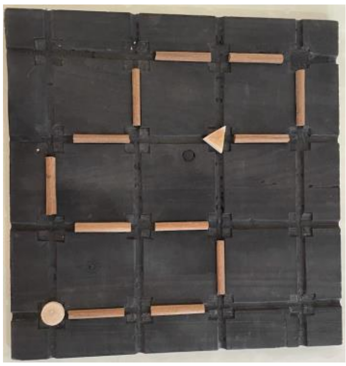


Figure 6. Example of a simple exercise implementation using the physical grid and its parts.

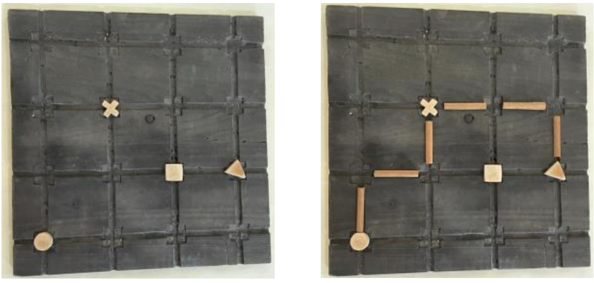


Figure 7. Complex exercise implementation using all possible pieces. This represents the most complex level of exercise. Initial state on the left and a valid answer to the right.

All parts were stained (or had their wood species chosen by their natural colors) to present visual contrast, aiming to aid users with low vision, while the shapes and engravings (channels and recesses) were created to support people with total vision loss. Using these physical implementations—the board, the pegs and the pieces—both kinds of exercises from Section 4 were adapted and applied.

4.2. Step 1: Simple Exercises

Exercises of the simple kind were implemented with some adaptations. Presenting multiple choices of answers as shown in columns A, B and C in Figure 5 would be inaccessible to blind users, so the final path was constructed and promptly presented such as shown in Figure 6. In order to complete this exercise, users had to completely describe the path in an algorithmic manner, such as: two times right, up, two times left etc.

Users were evaluated by both time to completion and conciseness of the answer, including the use of algorithmic structures and operation blocks (groupings). Three levels of answers were defined:

● *Basic*, for answers that missed all possible groupings of steps and algorithmic structures, being “Left, left, up, right, up, right” an example;
● *Proficient*, where single steps were grouped and repetitions were used with single inputs, such as “Two times left”;
● *Expert*, for answers that grouped multiple steps and used them as inputs to other structures, effectively combining both concepts, as in “Two times (up, right)”.

4.3. Step 2: Complex Exercises

For complex exercises, users were required to construct the path from a set source, to a defined destination, while also taking into consideration obstacles and required cells. Only pieces were initially placed on the board for identification. After examination, the users constructed their answers using the pegs to represent the path.

Four levels of complexity were defined for this kind of exercise:

● *Simple*, in which only the source (circle) and destination (triangle) pieces were used and users were expected to just create the shortest path between both;
● *Avoidance*, where besides the source and destination, some cells contained obstacles (square). Users were expected to create the shortest path that avoided the obstacles;
● *Requirement*, in which instead of obstacles to avoid, required cells (cross) were added. For this level of complexity, users were expect to create the shortest path between source and destination that passed through the required cells;
● *Complete*, in which all possible pieces were used to construct the most complex kind of exercise. An example of this level of exercise is shown in Figure 7.

Similarly to the simple exercise kind, the expected answers should use the lowest possible number of operations (pegs) and apply algorithmic structures, while also taking the shortest time possible to be constructed, counted from the beginning of the users' analysis to the completion of the answer.

4.4. Assessment Methodology

The complete methodology was composed of a diagnostic session using the wooden grid, followed by Computational Thinking training with Melodic, and ending with evaluation exercises, once again using the wooden grid.

An incremental approach to complexity was adopted and five exercises of each kind were created, from simpler to more complex, both in diagnostic and evaluation sessions. Care had to be taken to keep complexity levels equivalent between diagnostic exercises and their evaluation equivalents, so that Melodic contribution to Computational Thinking development could be effectively measured. In order to achieve this, evaluation exercises represented rotations or mirroring of their diagnostic equivalents, such as shown in Figure 8 in which the grid was rotated twice counter-clockwise.

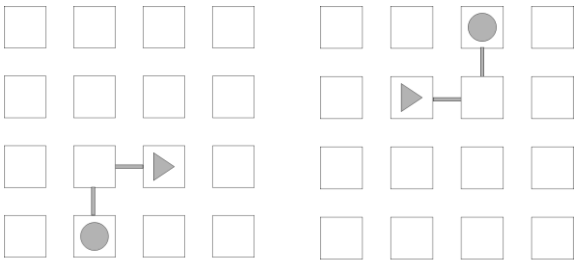


Figure 8. Rotation of the diagnostic exercise (left) to create the evaluation equivalent (right).

Users' evolution was carefully identified between the diagnostic and evaluation sessions in order to compile meaningful results for Melodic's contribution to Computational Thinking development.

4.5. Results

Two users participated in Melodic assessment, a 15 years old student and a university staff member around 40 years old, both with complete vision loss. Melodic contribution to Computational Thinking development was evaluated under three perspectives: execution time, answer complexity and number of instructions.

Execution Time

Time was measured from the beginning to the end of the exercises resolution for each user. The execution times taken to solve the simple and complex exercises are shown in Figure 9. In general, it is possible to observe that time taken to finish an exercise decreased after Melodic use (difference from the diagnostic times in blue and evaluation times in red), with just a few inversions. The first user presented worse times in the evaluation session in three simple and one complex exercises, although on average there was a 7% improvement (time reduction) for the simple exercises and 42.6% for the complex.

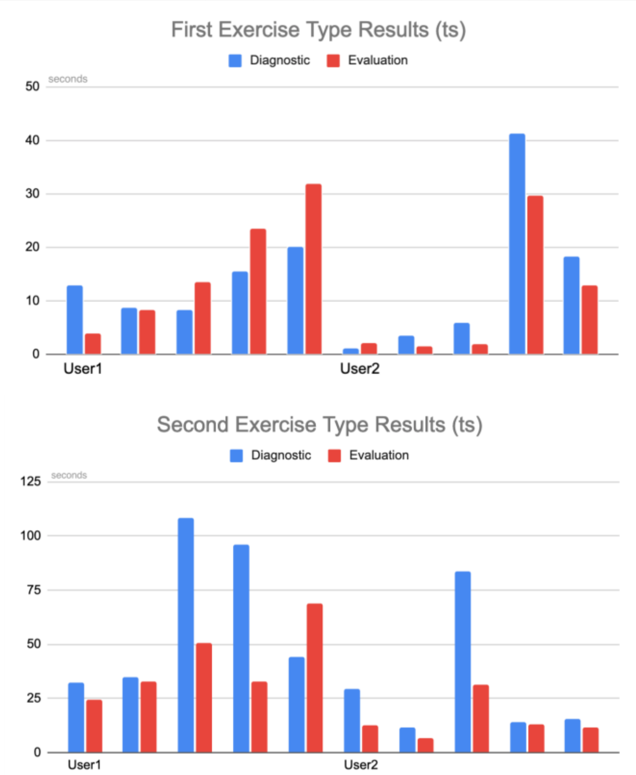


Figure 9. Execution times for all ten exercises.

Answer Complexity

Answer complexity was evaluated by asking both users to describe their reasoning to construct the exercise solutions on the board. The goal was to qualify and quantify the logic reasoning followed to build the answer. The explanation was given for the ten exercises—five simple and five complex—and the application of more sophisticated algorithmic concepts was identified. The self-described resolutions were ranked in levels of complexity, as stated in Subsection 4.2. Improvements were detected in 15% of the answers for the complex exercises. Simple exercises, given their lower challenge level, ranked equally in both diagnostic and evaluation sessions.

Number of Instructions

The number of instructions involved in each exercise solution was counted when users self-reported their reasoning for the answers, using the same strategy applied to the measurement of answer complexity. Overall, a decrease of 36\% in the number of instructions was observed between the diagnostic and the evaluations sessions. This result was only possible through the correct application of operation blocks and repetition. Both facts evidence an increase in Computational Thinking skills.

5. Technical and Conceptual Opportunities for Improvement: PathIt

The Melodic App was meticulously crafted utilizing Expo, a robust framework and platform designed for the development, construction, and deployment of universal React applications. This remarkable tool enables developers to bypass the arduous process of publishing the app on the App Store or Play Store, granting users unparalleled accessibility. By simply scanning a QR Code using the Expo GO app installed on their mobile device, users can effortlessly engage with the app's captivating features.

Unfortunately, a challenge arose after the initial release of the Melodic App when the Expo SDK utilized at that time became deprecated, resulting in the app becoming inaccessible to its user base. In response to this predicament, significant efforts were undertaken to address the matter promptly and ensure the app's continued accessibility to the public.

Therefore, the Melodic App has been recently re-released, incorporating the latest Expo SDK version, enabling users to once again enjoy its features. In a synchronized effort, the Melodic website, https://epl.di.uminho.pt/~gepl/Melodic/, has also undergone a comprehensive overhaul, incorporating the new QR Code and providing detailed instructions, ensuring a seamless and hassle-free user experience. Expanding on the achievements of the Melodic app, we are currently immersed in a captivating new venture called *PathIt*.

PathIt is an engaging and accessible game designed for both visually impaired individuals and sighted players. It offers a unique and immersive experience that combines physical game components with auditory ones from a mobile app . The game revolves around the challenge of creating a path between two positions on a game board (see leftmost image of Figure 10) while navigating through various restrictions and using sequence blocks (see the other two images of Figure 10) .

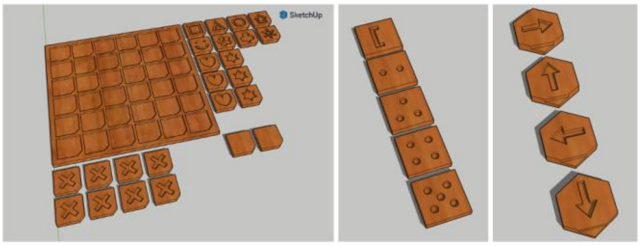


Figure 10. PathIt physical components.

The gameplay flow is as follows: players engage in a step-by-step construction process to build the game board, following voice instructions and signaling success through a shake movement. Subsequently, each challenge is presented and players utilize physical sequence blocks equipped with Near Field Communication (NFC) tags to input the desired path sequence into the mobile app. The app offers feedback and guides players through the challenges, which encompass diverse board configurations and progressively heightened difficulty levels.

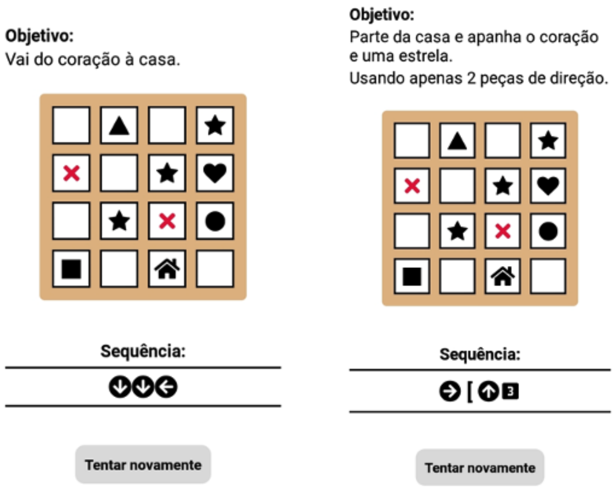


Figure 11. PathIt app challenges.

Figure 11 presents two challenges from the PathIt app to exemplify different levels of complexity and restrictions. The left image of Figure 11 depicts a straightforward challenge where the goal is to navigate from the heart symbol to the house symbol, with only two prohibited spots on the board.

On the right side of the same Figure 11, the image showcases a more intricate challenge that incorporates additional constraints. The task is to start from the home symbol, collect one heart and one star, and accomplish the objective using a maximum of two direction blocks. This challenge encourages the use of loop structures, fostering the application of Computational Thinking concepts.

PathIt promotes the development of Computational Thinking skills through its gameplay features. Users enhance their decomposition skills by breaking down the challenge into smaller steps when establishing a path. They practice abstraction by focusing on key points and disregarding irrelevant information. Algorithmic thinking is fostered through the use of repetition blocks to construct a path within imposed restrictions. Pattern recognition is honed as users recognize and reuse patterns between levels. Logical reasoning is encouraged when making decisions aligned with the main goal and constraints. Lastly, users engage in evaluation by assessing their decisions to optimize efficiency in achieving their position.

PathIt not only promotes the development of Computational Thinking skills but also strives to provide an inclusive gaming experience for all players. By engaging in the game's challenges and mechanics, users can enhance their Computational Thinking abilities while enjoying an accessible and inclusive environment.

6. Conclusion

Visually impaired people should have the same learning opportunities as everyone else. For this, it is important to design and build learning resources that are inclusive, that is, that can be used by people with or without visual impairments in their learning process.

Melodic is a learning resource that aims to allow people with or without visual impairments to train their Computational Thinking (CT) skills. Melodic is made up of blocks with symbols that have special meaning (musical notes, musical instruments, repetition, ...), and a mobile application that reads the sequence of blocks built by the learner. After reading the sequence with the app, the sequence is converted into sounds that play a melody (the origin of the LR name).

To assess the efficiency of Melodic, a three-step evaluation approach was created: diagnostic assessment of users' CT skills; training of CT using the Melodic system; and a final evaluation to measure the participants' progress. We only got two participants (one young and one adult) to test the Melodic system following the approach summarized above. The evaluation of Melodic CT training capabilities was measured through three parameters: time spent to finish the exercise; response complexity, considering the answer given to the exercise; and number of instructions of the resolution.

Regarding the execution time, we concluded that in general the exercise completion time decreased after Melodic training. The complexity of the responses was assessed by asking users to describe their reasoning for constructing the responses on the board. In the complex exercises, improvements were detected for 15% of the answers. For simple exercises improvements were not detected. The number of instructions was calculated when users self-reported their reasoning for the responses. It was concluded that in general there was a 36% decrease in the number of instructions between the diagnostic and final evaluation sessions.

However the main contribution of this paper is not the assessment of Melodic, which we recognize to be in a preliminary stage requiring its application to a larger set of participants. Instead, the main contribution is the definition of a *validation strategy for learning resources* aimed at training CT. This is, in our opinion, the focus of the work here reported, to enable us to assess properly the effectiveness of the tools we are developing to prepare students to solve problems using a computer. We do not state that the strategy proposed is free of flaws. We recognize that the evaluation techniques and tools applied to assess Melodic effectiveness might have biased the results. One feasible way to minimize this effect would be to apply a controlled study, measuring the relative evolution of a group that uses Melodic when compared to a group that does not. While not a trivial endeavor, this kind of assessment would further strengthen our current strategy. However the feedback from *Irís Inclusiva* and the people who participated was positive, as they considered that Melodic has potential, it is easy to use, and that it helps to train various CT skills as well as music.

After testing Melodic and improving it, we decided to design and build yet another inclusive resource to train CT—PathIt. PathIt is a game composed of a physical board with symbols, a mobile application to guide the challenges, and physical pieces to solve the challenges. PathIt was designed and conceived following the requirements suggested by the associations: *Irís Inclusiva* and *ACAPO*.
Both Melodic and PathIt are inclusive resources in the sense defined above and we plan to use it with children without visual problems.

As future work, we intend to finalize the implementation of the PathIt project. We consider it necessary and urgent to design and carry out more experiments (with training plans and evaluation tests) to complete the evaluation of Melodic effectiveness and evaluate PathIt. In addition, we consider it important to build usability and feedback surveys for both resources. In order to extend Melodic potential to train CT skills, we intend to create conditional and concurrency blocks to play different pieces of music in parallel and train traditional conditional algorithmic control structures. Finally, we intend to implement a computer vision module in the App to identify each block by its top, improving usability. As the blocks have different shapes and contrasting colors, the user would not need to turn the block over to scan, completely avoiding the use of QR Codes and facilitating the use of Melodic.

Acknowledgments

This work has been supported by *FCT – Fundação para a Ciência e Tecnologia* within the R&D Units Project Scope: UIDB/00319/2020

References

Abreu, L., Pires, A. C., & Guerreiro, T. (2020). TACTOPI: A playful approach to promote computational thinking for visually impaired children. *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. Presented at the 22nd International ACM SIGACCESS Conference on Computers and Accessibility, Virtual Event, Greece.

Alotaibi, H., S. Al-Khalifa, H., & AlSaeed, D. (2020). Teaching Programming to Students with Vision Impairment: Impact of Tactile Teaching Strategies on Student's Achievements and Perceptions. *Sustainability*, *12*(13), 5320.

Costa, R. (2022). *Melodic, using music to train visually impaired kids in computational thinking* (Mac Thesis). Universidade do Minho.

Costa, R., Araújo, C., & Henriques, P. R. (2021). Melodic - teaching computational thinking to visually impaired kids. *Proceedings of the Second International Computer Programming Education Conference (ICPEC 2021)*, *91*, 4:1–4:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Hallahan, D., & Kauffman, J. (2006). *Exceptional Learners: Introduction to Special Education* (10th ed.). Boston: Pearson/Allyn and Bacon.

Koushik, V., Guinness, D., & Kane, S. K. (2019). StoryBlocks: A tangible programming game to create accessible audio stories. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–12. Glasgow: Association for Computing Machinery.

Martinelli, A., Zanki Cordenonsi, Andre, & Bernardi, G. (2021). Floresta Book – Jogo de tabuleiro para a introdução do pensamento computacional nas séries iniciais do ensino fundamental. *Temática*, *17*, 206–222.

Morrison, C., Villar, N., Thieme, A., Zahra Ashktorab, Taysom, E., Salandin, O., Zhang, H. (2020). Torino: A tangible programming language inclusive of children with visual disabilities. *Human–Computer Interaction*, *35*(3), 191–239.

Pogrund, R. L., Fazzi, D. L., & American Foundation for the Blind. (2002). Early Focus: Working with Young Blind and Visually Impaired Children and Their Families. American Foundation for the Blind.

Rocha, F., Pires, A. C., Neto, I., Nicolau, H., & Guerreiro, T. (2021). Accembly at home: Accessible spatial programming for children with visual impairments and their families. *Interaction Design and Children*, 100–111. Athens, Greece: Association for Computing Machinery.

Sabuncuoglu, A. (2020). Tangible music programming blocks for visually impaired children. *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction}*, 423–429. Sydney: Association for Computing Machinery.

Silva, R., Fonseca, B., Costa, C., & Martins, F. (2021). Fostering computational thinking skills: A didactic proposal for elementary school grades. *Education Sciences*, *11*(9).

Silva, V., Moura, H., Paula, S., & Jesus, Â. (2019). ALGO+RITMO: Uma proposta desplugada com a música para auxiliar no desenvolvimento do pensamento computacional. *Anais do XXV Workshop de Informática na Escola*, 404–413. Brasília: SBC.

Sullivan, A. A., Bers, M. U., & Mihm, C. (2017). Imagining, playing, and coding with KIBO: using robotics to foster computational thinking in young children. *International Conference on Computational Thinking Education (CTE2017), 110*.

Taylor, R. L., & Sternberg, L. (1989). Students with Visual Impairments. In *Exceptional Children: Integrating Research and Teaching* (pp. 290–315). Springer.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(March 2006), 33–35.
World Health Organization. (2022). ICD-11 - Mortality and Morbidity Statistics. Retrieved from https://icd.who.int/browse11/l-m/en

Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020). Computational thinking test for beginners: Design and content validation. *Proceedings of the 2020 IEEE Global Engineering Education Conference (EDUCON)*, 1905–1914.

MindMaths: learning mathematics in the early years through computational thinking and robotics

Cristina Gomes[1], Helena Gomes[2], Maria Figueiredo[1], António Lucas[3], Luís Menezes[1]

[1] CI&DEI School of Education Polytechnic Institute of Viseu
[2] CIDMA School of Education Polytechnic Institute of Viseu
[3] School of Education Polytechnic Institute of Viseu
{mcagomes, hgomes, mfigueiredo, amjlucas, menezes}@esev.ipv.pt

Resumen: Las matemáticas en la educación en la escuela primaria siguen siendo un gran desafío. Los conceptos y habilidades matemáticas a menudo se identifican, tanto por estudiantes como por maestros, como aquellos con las mayores dificultades de aprendizaje. Explorar la robótica educativa puede asumir una estrategia facilitadora y de poder para promover el aprendizaje del conocimiento matemático (conceptos, procedimientos y métodos). El desarrollo del pensamiento formal, el razonamiento matemático, el pensamiento computacional y las habilidades de resolución de problemas pueden estar mediados por escenarios de aprendizaje tangible, ensamblar contextos significativos donde los alumnos pueden manejar y programar robots para aprender matemáticas. Por lo tanto, es importante trabajar en estos enfoques con futuros maestros de su capacitación inicial. Esta es la idea central detrás del desarrollo del proyecto transnacional MindMaths: desarrollar un conjunto de recursos, un curso curricular y recursos educativos, en el contexto de la capacitación inicial de los maestros, para liberar la ansiedad matemática con el uso de robots en el aprendizaje de las matemáticas.
Palabras clave: Educación Matemática, Robótica, Pensamiento Computacional.

Abstract: Mathematics in primary school education remains a great challenge. Mathematical concepts and skills are often identified, both by students and teachers, as those with the greatest learning difficulties. Exploring educational robotics can assume a facilitating and empowering strategy to promote the learning of mathematical knowledge (concepts, procedures and methods). The development of formal thinking, mathematical reasoning, computational thinking and problem-solving skills can be mediated by tangible learning scenarios, assembling meaningful contexts where pupils can handle and program robots to learn mathematics. Therefore, it is important to work on these approaches with future teachers from their initial training. This is the central idea behind the development of the MindMaths transnational project: To develop a set of resources, a curricular course and educational resources, in the context of initial teacher training, to release math anxiety with the use of robots in math learning.
Key words: Mathematics Education, Robotics, Computational Thinking.

1. Introduction

MindMaths is an EU-funded project, started in 2020, with partners from Portugal, Italy, Turkey and Latvia, aiming to cope with math anxiety and problems with mathematics learning in primary education, using educational robotics. Throughout Flipped Learning Practices its focus is on initial primary teacher training, designing learning activities and resources to fulfill this challenge of preparing the future teachers to use robotics to release math anxiety. In the case of school mathematics, anxiety as an emotion that is characterized by a set of reactions to a certain thing or context is common and can have several causes, highlighting, the difficulties experienced in their learning and also the social conceptions about the discipline, which associate it with some difficult and not pleasant. Currently in Portugal, the development of Computational Thinking (CT) is one of the transversal skills in the curriculum documents of the first years, which frames the pertinence of the relationship between two areas and makes the reflection on the potential of Robotics and the CT even more emergent for learn mathematics in the early years.

This paper presents an overview of the MindMaths Project and the main ideas underlying its development in the Portuguese context. It frames the more relevant learning issues in primary math education, presents the most significant initiative of programing and robotics in the Portuguese scene and discuss the potential that Computational Thinking and robotics offer for learning meaningful mathematics in the early years. It also presents a pilot study in a polytechnic school with the participation of 20 students, future teachers, discussing the main ideas of its development and results.

1.1. Defining a framework

1.1.1. Computational Thinking and robotics in Portugal primary school contexts.

The importance of coding and robotics across the K-12 curriculum has been strongly recognized in the basic education context. Robotics can improve motivation and children engagement, boost collaboration and foster Computational Thinking. Computational Thinking is a powerful and gathering idea to enable students to develop skills considered essential for the full education of citizens of the XXI century. However, contours, understandings and appropriation of this concept are not consensual, making it important and crucial to find a rationale to foster these significant learnings and skills in basic education. The seminal work of Wing (2017) introduces the idea of thinking like a computer scientist, focusing on a way of thinking rather than on digital skills and abilities, "Computational Thinking is the thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer—human or machine—can effectively carry out" (Wing, 2017). This way of thinking encompasses skills such as abstraction, algorithmic thinking, automation, decomposition, debugging, and generalization and can be learned in "unplugged" scenarios, without technology, focusing on problem formulation, problem representation and communication and problem solving (Bocconi, 2016). Kafi (2016) extends the idea of Computational Thinking to computational participation, as coding is no longer a solitary, tool-based activity and Computational Thinking and programming are social creative practices. Children valued in their learning process the creation of something real and tangible that can be shared with others. Participation highlights the importance of the communities of practice that has become a key for learning to code. Scratch community is a paradigm of this participation and collaboration, where children share and remix their ideas and projects (Resnick, 2019).

Historically children have played with physical objects to learn a variety of skills. Tangible interfaces may be of significant benefit to education by enabling children to play with actual physical objects augmented with computing power (O'Malley, n.d.). Educational robotics gives children the opportunity to relate tangible concepts with programming and Computational Thinking. In this way, educational robotics is driven in Papert's constructionism approach Papert, 1980), engaging children in making their own meaningful constructions, "objects-to-think-with", to accomplish better achievements and significant learning (Papert, 1993).

Align with other international politics and initiatives, Portuguese educational policies recognize the importance of developing Computational Thinking from an early age. As said before, curriculum guidelines for the 1st cycle of the Portuguese primary education system assumes Information and Communication Technology (ICT) as a cross-cutting area, challenging a significant based-curriculum approach to integrate ICT, fostering meaningful learning in the several curriculum areas. This movement goes beyond the narrow idea of preparing more well-equipped programmers and engineers, as it is a broader approach to empower children problem solving, creativity, communication or abstract thinking skills.

In 2015 the ministry of education launched an initiative entitled "Introduction to Programming in the 1st cycle of basic education", aiming to develop Computational Thinking, digital literacy and transversal skills (IniProg, 2020). Children should develop decomposing and solving problems skills, apply algorithmic thinking, create animated stories and build games using computer programs. In a methodological perspective the activities should draw upon different curriculum areas and students should work in groups to build their projects. Creativity and diversity should be encouraged, as well as sharing projects with others. (Ramos, 2016).

The initiative "Probótica", programing and robotics in basic education (ME, 2022), followed this first two year project, which ended up involving more than seventy thousand students. In a similar way it favours skills and competences across the curriculum based on learning scenarios that can make learning more active and meaningful. Robotics provides a deeper learning of technology, fostering students to "learn by doing", in a tactile way. Gathering technology, programming languages and tangible objects students can explore problems and visualize results. Thus the initiative aims to contribute to the development of skills associated with computing, increase students' digital literacy levels and promote transversal skills to the curriculum. In short, portuguese governmental initiatives to introduce programming and robotics in primary schools are grounded and sustained in the development of Computational Thinking in a multidisciplinary and based curricula approach.

1.1.2. Robotics and Mathematical Education.

There are several common processes between the development of Computational Thinking and the learning of mathematics, mainly at the level of problem solving, reasoning (thinking at multiple levels of abstraction) and the representation of ideas.

More systematically, elements such as (abstractions and pattern generalizations; information processing; symbol systems and representations; algorithmic notions of flow of control; structured problem decomposition; iterative, recursive, and parallel thinking; conditional logic; efficiency and performance constraints; debugging and systematic error detection) are widely accepted as comprising CT and form the basis of curricula that aim to support its learning as well as assess its development (Grover, 2013).

Mathematics is an area that, in a very natural way, can create numerous and significant opportunities for the development of algorithmic thinking at any level of education and, in particular, at the early years level. In an almost equivalent way, by the very nature of Mathematics, algorithmic thinking is a fundamental process for those who study, apply or do Mathematics. Several researchers claims that children engaging with programming robots have opportunity to explore spatial concepts, problem solving, measurement, geometry, and engage with meta- cognitive processes (Clements, 1993), (Yelland, 1994).

In 1993, R. Hembree and H. Marsh (1993) published several studies that highlight the role of problem solving in the development of mathematical thinking in younger children. In general, problem solving involves steps such as i) understanding the problem; ii) devising a plan to solve it; iii) executing that plan; iv) reflecting on the work. There are several classifications for problem typologies. The best known include one-step problems, two- or more-step problems, process problems, application problems and puzzle-type problems (Bers, 2018). Process problems, which use mechanised or standardised processes, and application problems, which include the collection of real-life data, involving one or more solving strategies will be those that resort to thinking mechanisms closest to those used in the creation, use and representation of algorithms.

In problem solving strategies, we find ideas very close to the ideas of algorithm creation such as pattern discovery, logical deduction or reduction to a simpler problem. For example, games as mathematical tasks require ways of solving problems, with the use and analysis of different strategies, defined objectives and often involve the use of algorithms with structures of various natures such as structures of selection or repetition during their resolution. Also the work with algorithms of arithmetic operations or the discovery of patterns (geometric, numerical, ...) approaches the logic of the use/reuse and evaluation of algorithms of various natures.

In fact, Computational Thinking represents a form of analytical thinking very close to the mathematical thinking form, particularly at the level of problem solving. Thus, programming, Computational Thinking and algorithmic thinking provides a significant opportunity to engage in logical, abstract and sequential thinking, problem solving and the creative design process (Quigley, 2019).

Tasks that involve Computational Thinking and robotics offer significant learning opportunities because they retrieve many of the actions that children perform and see performed in their daily lives, while facilitating the integration of more abstract mathematical thinking into their world and providing the mobilisation and representation of logical reasoning, simpler or more complex, fundamental to solving challenging problem situations, using various strategies, properties or mathematical relationships allowing the development of solid and creative thinking. In this sense, working on the basis of programming and robotics problems enables the deepening of fundamental knowledge and skills for learning meaningful mathematics through multidisciplinary and authentic learning contexts at the early years level.

As mentioned, in Portugal, new mathematic programs are already in practice and Computational Thinking is consider a transversal skill to develop since the first school years (ME, 2022). In this way, coding and Computational Thinking are being explicitly integrated into the curricular guidelines for the teaching and learning of Mathematics.

2. The MindMaths Project

MindMaths is a two year Erasmus+ project coordinated by Koacaeli University – Turkey and with Polytechnic Institute of Viseu – Portugal, Scuola di Robotica – Italy, Latvijas University–Latvia, Canakkale Onsekiz mart University – Turkey and Educloud Egitim Organizasyon Teknoloji Ticaret Ld – Turkey, as partners (MindMaths, 2022). It intends to develop, in the context of the initial primary teacher training, learning scenarios that explore new approaches to mathematics teaching, tackling the resistance and anxiety that some children show regarding math contents. MindMaths covers primary education level because it is the most appropriate period for recognition and early intervention of children with Math Anxiety. Thus, the aim of this international project is to develop skills for future teachers, who will teach in the early years, in solving possible problems related to mathematical anxiety on the part of their students. Computational Thinking, coding and robotics framed a collaborative development of a teacher training module about teaching/learning mathematics along with the development of a video library to be implemented in all partner countries.

The main intellectual outputs of this project can be retrieved in the project website ((MindMaths, 2022).

The modular course curriculum was designed in a Flipped Learning approach and integrates 5 modules, in a total of 28 face-to-face. Through this modules the students, future primary school teachers, has the opportunity to learn about Math Anxiety, the use of robotics in Math education, to discuss and reflect about alternative methodological approaches, such as flipped learning and blended learning, and to explore, plan, implement and reflect about the use of robotics in mathematics education in primary schools. The video library share videos introducing each learning activity supporting the flipped learn approach. All these resources were tested in a pilot study in each of the partner countries.

3. MindMaths: the portuguese pilot study

In Portugal, the MindMaths course was attended by 20 students of the 3rd year (6th and last semester) of the Basic Education bachelor course in a polytechnic school of education. The average age of students is 22.4 years and 90% are female. Throughout the Basic Education course, students attended some subjects corresponding to a training component in the teaching area, within the scope of Mathematics: Fundamentals of Mathematics, Geometry I (Euclidean Geometry), Numbers and Operations, Algebra, Mathematical Modeling, Geometry II (Coordinate Geometry), Statistics and Probabilities and Manipulating Materials in the teaching and learning of Mathematics. In addition, a discipline of Information and Communication Technology is included in the syllabus of the course. None of the students participated in any previous training on educational robotics.

The course explored teaching methodological approaches to support active and significant learning. Flipped and blended learning, debates, collaborative work and a focus on hands on activities, supporting the students to experience and reflect about the use of robotics in mathematics primary education. In addition to the face-to-face activities, in the polytechnic school of education, students went in small groups to a primary school to promote extracurricular activities with a group of 10 children. This dynamic was crucial to allow students to plan teaching learning scenarios, implement the activities with children and assess the real potential of robotics to teach mathematics.

To develop the activities they used Doc robots with children 6-8 years old, and Mbot robots with children 9-10 years old.

Some of the learning activities are proposed in the MindMaths curriculum. But the students also add some new proposals to be explored with the children. For instance, they proposed a supermarket scenario, fig. 1, to challenge the robot to go shopping the ingredients to make a chocolate cake.

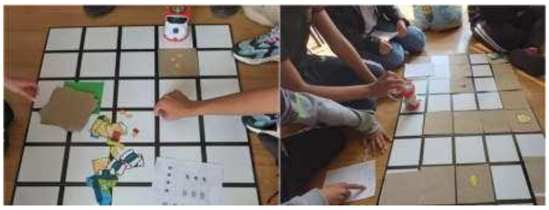


Figure 1. Children playing with the Doc Robot in the supermarket scenario

To accomplish this task, children developed spatial skills, numeracy, measures and estimation.
Other tasks were developed using the Mbot robot, programmed in block language, as shown in the figure 2.

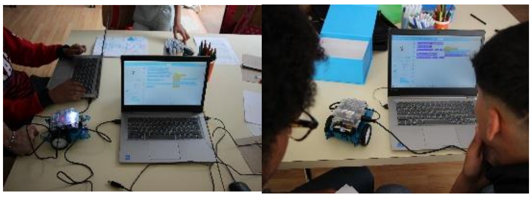


Figure 2. Children playing with the Mbot Robot in different tasks

The 20 students answered to a pre-test and pos-test, both with the goal of understanding their perspectives on the use of educational robotics to learn Mathematics, the knowledge they have in the area and the confidence to use this resource in the subject classes in the 1st cycle of basic education.
A quantitative analysis of the data was performed with determination and interpretation of absolute and relative frequencies, measures of central tendency and dispersion (through the coefficient of variation - CV) to assess the representativeness of the mean. We considered the CV as a measure that evaluates effectively the representativeness of the mean as a central tendency measure, combining the relationship between this measure and the standard deviation. In general, the reference value used for the CV is 50%.
The Wilcoxon test was used to assess the significance of the evolution of the results from the pre to the post-test, in our case with a significance of 5%. In the Wilcoxon statistic we calculate the p-value that is a probability that measures the evidence against the null hypothesis (ie against the hypothesis that t
here were no significant changes from pre to post test). A smaller p- value provides stronger evidence against the null hypothesis). With a significance of 5%, if p-value ≤ 0.05, the difference between the medians is significantly different (and so we reject the null hypothesis).
The tests were divided into 4 sub-dimensions for analysis: Benefits (10 questions), Attitudes (6 questions), Negative perspective (6 questions) and Self-confidence/confidence (8 questions). Each of the test questions had response options within a discrete scale of 1 to 5 where 1 represents strongly disagree, 2 disagree, 3 neither agree nor disagree, 4 agree and 5 strongly agree. Table 1 presents the data related to the means and respective CV calculated for each question of the 4 dimensions, in the pre and post-test.

		Pre-test		**Post-test**	
		Av.	**CV**	**Av.**	**CV**
BENEFITS Sub-Dimension	1. I think that using Robotic Application increases academic success in mathematics lessons.	3,8	15%	4,3	17%
	2. I think that using Robotic Application increases the performance of the teacher in mathematics lessons.	3,8	20%	4,2	18%
	3. I think that using Robotic Application enables students to have fun in math lessons.	4,2	13%	4,6	16%
	4. I think that using Robotic Application enables learning by doing and experiencing in mathematics lessons.	4,3	11%	4,6	16%
	5. I think that using Robotic Application provides a learner-centered teaching approach in mathematics lessons.	3,8	15%	4,2	22%
	6. I think that using Robotic Application is effective in motivating students who have negative attitudes towards mathematics in mathematics lessons.	4,0	16%	4,3	23%
	7. I think using Robotic Application makes math problems more meaningful for students.	4,0	19%	4,2	17%

	8. I think that using Robotic Application in Mathematics lessons will improve students' thinking skills.	4,1	15%	4,2	19%
	9. I think that the topics in Mathematics can be made more concrete by using Robotic Application.	3,6	16%	4,3	19%
	10. I think that using Robotic Application in Mathematics lessons will motivate students to solve complex problems.	3,7	15%	4,2	20%
ATTITUDES Sub-Dimension	1. I am willing to use Robotic Application in Mathematics lessons.	3,8	13%	4,3	19%
	2. I like/I think I will like to use Robotic Application in Mathematics class.	3,8	16%	4,2	20%
	3. I would like to do new research on the use of Robotic Application in Mathematics class	3,7	26%	4,0	22%
	4. I would like/like to develop different Robotic Application designs to use in the Mathematics lesson.	4,1	17%	4,3	19%
	5. I like to learn new information about the use of Robotic Application in Mathematics class	4,2	18%	4,3	19%
	6. The processing of mathematics lessons with Robotic Application applications attracts my attention.	4,1	15%	4,2	19%
NEGATIVE PERSPECTIVE Sub-Dimension	1. I think that methods such as Robotic Application in teaching mathematics are a waste of time.	1,7	52%	1,4	67%
	2. I am not so sure that methods such as Robotic Application can be useful in teaching mathematics.	2,0	47%	1,6	70%
	3. I think that methods such as Robotic Application are overemphasized in mathematics teaching.	1,8	43%	2,1	52%
	4. I think that using Robotic Application in Mathematics lessons removes the clarity about what is learned.	1,8	40%	1,8	63%
	5. I think that using Robotic Application in Mathematics lessons confuses students.	1,9	38%	1,5	63%
	6. I think that using Robotic Application in Mathematics lessons is too costly.	2,8	32%	2,4	43%
SELF-CONFIDENCE/CONFIDENCE Sub-Dimension	1. I think that I am sufficient in creating different scenarios forRobotic Application applications in mathematics teaching.	3,3	22%	3,8	22%
	2. I think/I can use Robotic Application applications effectively in mathematics lessons.	3,4	20%	4,0	21%
	3. I don't think I will have any difficulties while programming robots in robotics applications in math class.	2,9	27%	3,8	20%
	4. I think that using robotic applications in mathematics lessons can force me.	4,1	13%	4,1	24%
	5. Robotic App is easy for me to learn to usein Math Lessons.	2,9	30%	3,8	21%
	6. I think that I have improved myself on Robotic Application in the field of mathematics teaching.	3,8	19%	4,2	20%
	7. I have enough knowledge of Robotic	2,3	40%	3,6	18%
	Application to use in Mathematics Lessons.				
	8. If I try, I can program a robot to usein Mathematics lessons.	2,9	29%	4,0	18%

Table 1. Means and coefficient of variation – pre-test and post-test

In the pre-test, from the data presented, we highlight the fact that 75% of the questions related to the benefits, attitudes and self-confidence/confidence dimensions had averages above (approximately) 4 on the considered scale (which corresponds to the recognition of agree and strongly agree), all representative given that the CV is less than 50%. These facts may reveal that students, even before the development of workshop, recognize the educational potential of robotics to learn mathematics in the 1st Cycle of Basic Education. The post-test r answers accentuate this trend in recognizing the educational potential of robotics, with a more significant increase of 33% corresponding to confidence in programming robots to teach mathematics (question 3 of the self-confidence/confidence dimension) or 20% in what it concerns the recognition that mathematics is more concrete by working with robots (question 9 of the benefits dimension).

Regarding the Negative perspective dimension, students respond in a more diverse way. In these cases, the means of agreement are the least representative among all dimensions because they are associated with higher CV. In any case, the respondents' opinions improve the idea of the importance of robotics for learning mathematics, considering that its use in mathematics classes is not a waste of time because it is useful and makes abstract concepts clearer. Still on this dimension, students recognize that using Robotic Application in Mathematics lessons is too costly (highest and most representative average agreement in this group of questions).
The questions with the lowest frequency of agreement in pre-test were, in the first dimension, question 9 (47%) and in dimensions two and four, questions three and seven, respectively (37% and 89%). Regarding the third dimension, there are high levels of disagreement with statements, particularly about the first, third, fourth and fifth (100% of responses that represent disagree and strongly disagree). In the post-test, the non-agreement with the ideas associated with the negative perspective dimension (all with more than 89% of frequency) stands out with high frequency. In the pre-test, the answers with the highest levels of agreement are associated with questions three and four of the first dimension, five of the second dimension and four of the fourth dimension (95%, 100%, 89% and 89%, respectively). In the post-test, the highest frequencies of fours and fives (which correspond to agreement or total agreement) were attributed to questions one, three, four and five of the first dimension (95%), one, four, five and six of the second dimension (89%) and six and eight of the last dimension (84%).
The analysis of the data to compare the students' responses from the pre to post test, using the paired Wilcoxon test (with a significance of 5%), confirms the strong tendency for students to recognize: [p-value equal to 0.0270] the importance of using Robotic Application in mathematics lessons because it increases the performance of the teacher (question two of dimension one); [p-value equal to 0.0095] that educational robotics makes mathematics more concrete (question nine of dimension one); [p-value equal to 0.0015] that they are sufficient in creating different scenarios for Robotic Application applications in mathematics teaching (question one of the fourth dimension); [p-value equal to 0.0032] that Robotic App is easy to learn to use in Math Lessons (question five of the fourth dimension).
The following representation, fig. 3, allows comparing the means of answers in each dimension, in two tests.

COMPARISON OF AVERAGES BY DIMENSION

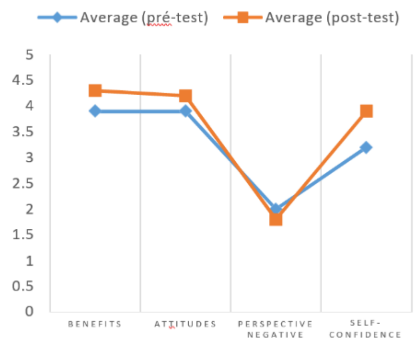


Figure 3. Pre-test and post-test averages by dimension

From the information it's possible to conclude that the averages of agreement with the statements are always higher in the post-test, regarding the benefits, attitudes and self-confidence dimensions. As expected, the opposite is true for the data on the negative perspective dimension.
It is also shows that the levels of agreement with the statements are higher in the questions associated with the first two dimensions for any test: benefits and attitude (for any dimension the averages are representative, since that the CV is always less than 3% in the case of the first, second and fourth dimensions and less than 20% in the perspective negative dimension).

4. Conclusion

Computational Thinking, coding and robotics are strongly recognized as crucial since early education. However, its representations and appropriations are not consensual, as some educational practices persists in teaching coding by coding without exploring the real opportunities of Computational Thinking to scaffold children to formulate, express and represent a problem and propose solutions.
Reactions associated to anxiety in school mathematics can have several causes, highlighting, the difficulties experienced in children learning, and also the social

Número 37, Enero-Junio 2023, pp 14-22
Artículos

conceptions about the discipline, which associate it with something not pleasant. Working with technology on real and significant contexts about meaningful problems for children enables them to learn mathematics better and to feel more confident, reducing anxiety in learning the subject. Robots and tangible objects can assume an important role to connect concrete to abstraction thinking, a crucial process in mathematics learning.

MindMaths project explored, in real initial teacher education settings, an approach to foster the use of educational robotics to teach mathematics. Results about the impact on the students, future primary teachers, encourages to address this challenge both in initial training and in in-service training.

In this sense, it is important to rethink how to prepare future teachers to promote authentic and significant teaching/learning mathematics activities, based on Computational Thinking and robotics, promoting better children engagement and learning with math topics. This is the main goal of the MindMaths Project.

5. Acknowledgment

This work was supported by the European Comission through Erasmus+ funding as “MINDMATHS: Flipped Learning Practices to Release Maths Anxiety with the Use of Robotics”, 2020-1-TR01-KA203-092209. Furthermore, we would like to thank the Centre for Studies in Education and Innovation (CI&DEI) and the Polytechnic of Viseu for their support.

References

Wing, J. (2017). ‘Computational Thinking’s Influence on Research and Education for All’, Italian Journal of Educational Technology, vol. 1, no. 1, doi: 10.17471/2499-4324/922.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A. and Engelhardt, K. (2016). Developing computational thinking in compulsory education: implications for policy and practice. LU: Publications Office. Accessed: May 16, 2021. [Online]. Available: https://data.europa.eu/doi/10.2791/792158

Kafai, Y. B. (2016). From computational thinking to computational participation in K-12 education. Communications of the ACM, 2016, 59(8), 26–27. https://doi.org/10.1145/2955114

Resnick, M. (2019). The Next Generation of Scratch Teaches More Than Coding. EdSurge. Accessed: June 14, 2021. [Online]. Available: https://www.edsurge.com/news/2019-01-03-mitch-resnick-the-next- generation-of-scratch-teaches-more-than-coding.

O’Malley, C., Fraser, D. S., & Facer, K. (n.d.) Literature Review in Learning with Tangible Technologies REPORT 12: FUTURELAB SERIES.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas. New York: Basic Books.

Papert, S. (1993). The Children's Machine: Rethinking School in the Age of the Computer, HarperCollins, pp 241

IniProg. Projeto Iniciação à Programação no 1.º Ciclo do Ensino Básico. (2017). retrieved in 14 de outubro de 2020. Available at https://www.erte.dge.mec.pt/iniciacao-programacao-no-1o-ciclo-do- ensino-basico.

Ramos, J. L., Espadeiro, R. G. (2016). Iniciação à Programação no 1.º ciclo do Ensino Básico. Estudos de avaliação. DGE – Direção Geral de Educação.

Ministério da Educação. Programação e Robótica no Ensino Básico (2022). retrieved in 19 de junho de 2022. Available at https://erte.dge.mec.pt/programacao-e-robotica-no-ensino-basico-0

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. Educational Researcher, 42(1), 38–43. https://doi.org/10.3102/0013189X12463051

Clements, D.H., & Meredith, J.S. (1993). Research on Logo: Effects and efficacy. Journal of Computing in Childhood Education, 4, 263-290.

Yelland, N. J. (1994). The strategies and interactions of young children in LOGO tasks. Journal of Computer Assisted Learning.10, 33-49.

Número 37, Enero-Junio 2023, pp 14-22

Hembree, R., & Marsh, H. (1993). Problem solving in early childhood: Building foundations. In R. J. Jensen (Ed.), Research ideas for the classroom: Early childhood mathematics,. (pp. 58–77). Reston, VA: NCTM.

Bers, M. U. (2018). Coding as a Playground. London and New York: Routledge Press.

Quigley, C. F., & Herro, D. (2019). An educator´s guide to STEAM. Engaging students using real-world problems. New York: Teachers College Press. 153 pp.

Ministério da Educação. (2022). Aprendizagens Essenciais de Matemática. retrieved in 19 de junho de 2022. Available at https://www.dge.mec.pt/noticias/aprendizagens-essenciais-de- matematica

MindMaths. (2021). retrieved in 19 de junho de 2022. Available at http://www.mindmaths.org/

Propuesta de evaluación del impacto emocional de la programación en Educación Infantil

Cristina Sáez Fernández, Diana Pérez Marín

Escuela Técnica Superior de Ingeniería Informática, Universidad Rey Juan Carlos
Avda. Tulipán s/n, Móstoles, Madrid, España
diana.perez@urjc.es

Resumen: La relación entre programación y estado emocional ha atraído un gran interés en la literatura en Educación Primaria, siendo necesario su estudio en Educación Infantil al haberse introducido también la programación con robots como Cubetto. La principal diferencia en Educación Infantil es que la investigación debe ser adaptada al menor grado de madurez de estos alumnos que aún es posible que no sepan leer o escribir. Por este motivo, en este trabajo, se presenta un análisis vivencial y emocional de la experiencia de programación con Cubetto de 77 niños de 3, 4 y 5 años a través de la interpretación de sus dibujos. Los resultados muestran cómo vivieron la experiencia los niños y también una primera propuesta de evaluación de la relación entre programación y su impacto emocional en Educación Infantil.
Palabras clave: Educación Infantil, Cubetto, Programación, Emociones.

Abstract: The relationship between programming and emotional state has attracted great interest in the literature in Primary Education. It would be interesting to study that relationship in Early Childhood Education as programming with robots such as Cubetto has also been explored. The main difference in Early Childhood Education is that the research must be adapted to the lesser degree of maturity of these students who still may not know how to read or write. For this reason, in this work, an experiential and emotional analysis of the programming experience with Cubetto of 77 children of 3, 4 and 5 years of age is presented through the interpretation of their drawings. The results show how the children lived the experience and also a first evaluation proposal of the relationship between programming and its emotional impact in Early Childhood Education.
Keywords: Early Childhood Education, Cubetto, Programming, Emotions.

1. Introducción

La enseñanza de la programación en edades tempranas ha atraído un gran interés en las últimas décadas (CSTA, 2012; Jacobsen, 2014; Hromkovič et al. 2019). En particular, en Educación Infantil se ha investigado el uso de robots como Cubetto (Primo Toys, 2018; Sáez-Fernández et al. 2018; Bueno Fenero, 2022).

Cubetto es un pequeño robot de madera en forma de cubo que se mueve según los niños le programan con unas fichas de colores en un mando también de madera. Los movimientos posibles son adelante, atrás, izquierda y derecha. El mando tiene incorporado un botón para comenzar la ejecución de movimientos, que solo funcionará correctamente si se han insertado en el sitio correcto.

De esta forma los niños pueden corregir la secuencia del programa de movimientos ellos mismos según los resultados que ven de forma inmediata, si el robot gira para el lado que no es, deberán cambiar la ficha de giro hacia el otro lado, o si no avanza tantos pasos como quisieran deberán añadir más fichas de movimiento al tablero. Esto se basa en la filosofía pedagógica de María Montessori (1986).

Otro tema de interés en las últimas décadas ha sido la relación entre programación y emociones. En Morales-Urrutia (2021) se hace una revisión de la integración de emociones para la enseñanza de la programación en Educación Primaria, proponiendo el uso de compañeros de aprendizaje con soporte emocional para alcanzar el estado óptimo de aprendizaje. En general, el uso de robots y tecnología se ha visto que puede ayudar en la enseñanza de conceptos complejos y abstractos como pueden ser la programación o las matemáticas en Educación Primaria (Konahli et al. 2021).

En el caso de enseñar programación en Educación Infantil se ha destacado la importancia de usar el juego como vehículo de motivación para atraer el interés y la innata curiosidad de los niños entre 1 y 6 años (Sáez Fernández et al. 2018; Bueno Fenero, 2022; Bers et al. 2022). Sin embargo, la investigación de la enseñanza de la enseñanza de la programación en Educación Infantil aún está en una etapa inicial (González-González, 2019) que necesita de profundizar en los aspectos ya estudiados cómo los kits robóticos que se usan y qué competencias desarrollan o escalas de evaluación (Santos Miranda-Pinto & Osorio, 2019), y seguir avanzando en nuevas líneas de investigación por explorar cómo la relación entre programación y emociones también en Educación Infantil.

En esta línea de explorar la relación entre programación y emociones en Educación Infantil se propone en este trabajo el uso del dibujo como herramienta (Anning & Ring, 2004). Dar sentido al dibujo de los niños cuando su edad no permite recoger información de las formas más tradicionales como cuestionarios o entrevistas puede ser la forma de poder identificar sus emociones a través de los colores, formas, tamaños y distribución de elementos que usan en el papel (Alomía & Samaniego, 2013).

En este artículo, se propone una evaluación de las emociones de los niños de Educación Infantil a través del análisis del dibujo. Para poder validar este tipo evaluación se ha realizado una intervención en el aula con 77 estudiantes de 3 a 6 años que programaron durante una semana con Cubetto, y a los que se les pidió que dibujaran antes y después de usar Cubetto cómo se sentían usando lápiz, colores y folios en blanco.

El análisis de los dibujos se centró en el uso del color, de líneas rectas o curvas, y el estudio de los elementos dibujados. Los resultados arrojan información útil para la relación entre programación y emociones en Educación Infantil que valida el uso del dibujo como herramienta de evaluación del impacto emocional de la programación en etapas tan tempranas.

El artículo está organizado en cuatro apartados: la Sección 2 repasa el trabajo relacionado en la enseñanza de la programación y su relación con las emociones; la Sección 3 describe la propuesta de intervención en el aula de Infantil de enseñanza de la programación y el análisis vivencial y emocional usando dibujos; y, finalmente la Sección 4 cierra el artículo con las principales conclusiones obtenidas.

2. Trabajo relacionado

2.1. Enseñanza de la programación

Enseñar programación en Educación Infantil es la nueva alfabetización de este siglo (Bers, 2019). Países de todo el mundo han introducido esta enseñanza en su curriculum (Siu & Lam, 2005). Debido al nivel de maduración y edad de los niños en Educación Infantil se suele enseñar con robots como los mostrados en la Figura 1.

Figura 1. Ejemplo de Figura

Los robots son tangibles, dan una respuesta inmediata y las instrucciones suelen ser sencillas. Normalmente un programa en Educación Infantil es una secuencia de movimientos. Los niños pulsan botones en el robot que se mueve según la secuencia de movimientos introducida sobre algún tapete que les resulte familiar con colores, animales o siguiendo líneas o figuras geométricas.

Especial atención se quiere dar en esta revisión al robot Cubetto (Primo Toys, 2018). A diferencia de los robots que tienen botones para pulsar directamente sobre ellos, en este caso el robot es un cubo de madera que se mueve según las fichas que se introducen en un mando también de madera como se muestra en la Figura 2.

Figura 2. Cubetto con su mando y fichas

Cubetto se basa en la pedagogía Montessori (1986) según la cual los niños aprenden explorando por ensayo y error, sin necesidad de que el profesor les esté corrigiendo y sin pantallas. Las fichas solo encajan en una determinada posición en el tablero que además ilumina un led en azul indicando que se ha insertado correctamente. El tablero tiene un botón azul para ejecutar el programa sobre Cubetto. De esta forma, si el niño inserta 2 fichas verdes podrá ver como Cubetto avanza 2 posiciones en el tapete que suele estar cuadriculado. El objetivo podría ser que al avanzar 2 posiciones llegase a un árbol, a un determinado color o incluso se ha combinado el uso de Cubetto con un lápiz para que en lugar de llegar a metas, pudiese pintar sobre un tapete en blanco entre otras experiencias de uso (Jiménez Rodríguez et al. 2022).

2.2. Interpretación de emociones en dibujos

Salguero (2011) destaca la importancia del desarrollo de la inteligencia emocional en Educación Infantil trabajando entre otros elementos la expresión plástica. A través de los dibujos los niños expresan lo que no pueden hablar o escribir debido a su temprana edad. La interpretación de los dibujos infantiles incluye el estudio de los garabatos que pueden hacer entre los 2 y los 4 años como muestra la Figura 3 y el nivel presquemático o creación consciente de la forma desde los 4 a los 7 años como se muestra en la Figura 4 (Lowenfield y Brittain, 1972).

Figura 3. Dibujo en forma de garabato (2-4 años)

Figura 4. Dibujo presquemático (4-7 años)

Marín (1988) identificó algunos principios básicos que rigen la organización general de los dibujos, entre los que destaca el uso de líneas, el tamaño, el aislamiento de cada parte del conjunto, y el uso del color. Así para cada indicador se le asocia un significado, como por ejemplo, en el caso del color:

- amarillo suele estar asociado a confianza
- azul a estados de calma
- gris está asociado a intentar evitar obligaciones
- marrón puede indicar necesidad de relajarse
- negro suele ser elegido para protestar
- rojo expresa seguridad, energía y vitalidad
- verde está relacionado con la autoestima
- morado con la sensibilidad y deseo de afecto.

El dibujo en Educación Infantil se puede usar como un medio de detección de enfermedades y trastornos como retrasos a nivel cognitivo o motriz, daltonismo, TDAH, TEA o problemas en la adquisición del lenguaje según Zufiaur (2014). No se ha encontrado en la literatura relación con la detección del impacto de la enseñanza de la programación, a diferencia de lo que ocurre en el caso de Educación Primaria donde ya hay estudios en los que se relaciona el estado anímico con la enseñanza de la programación (Morales-Urrutia, 2021).

3. Propuesta de intervención en el aula

3.1. Objetivo

El objetivo es realizar una propuesta de evaluación del impacto emocional de la programación en niños de Educación Infantil. Esto se ve motivado porque en Educación Primaria se ha detectado que alcanzar un buen estado anímico ha permitido un aprendizaje significativo de los conceptos de programación, y se quiere poder detectar qué impacto emocional tiene la programación en Educación Infantil por si tuviese también algún impacto en su aprendizaje.

3.2. Muestra

La muestra la componen 77 niños cursando Educación Infantil en un colegio privado de España según esta distribución:

- 25 niños de 3 años
- 25 niños de 4 años
- 27 niños de 5 años

Se solicitó inicialmente permiso tanto a la Dirección del Centro como a los padres para poder recoger los dibujos anónimos de los niños y poder entrar en el aula durante una semana para enseñarles a programar secuencias básicas de movimiento con Cubetto. Todos los niños estaban agrupados en su aula según su edad.

3.3. Procedimiento

Varios investigadores se distribuyeron en las aulas para proceder de la siguiente forma:

1) El primer día de clase, antes de enseñarles Cubetto, se les proporciona un folio en blanco, pinturas y lápiz a cada niño de forma individual para que dibujen cómo se sienten antes de conocer a este robot amigo para aprender a programar que es cómo se les presenta brevemente.

2) En cada sesión durante una semana (una sesión por día) se les presenta una posibilidad de uso de Cubetto, para moverse hacia adelante, izquierda, derecha, e iniciarles en la programación con este robot Montessori. El profesor siempre acompaña a los niños para evitar daños en el robot y garantizar que pueden intentar todos los compañeros de la clase probar a programar alguna secuencia básica según lo que se les ha enseñado.

3) En la última sesión, al final del todo, se les vuelve a pedir a todos los niños que dibujen cómo se han sentido en la experiencia de uso de Cubetto, entregándoles el mismo material que el primer día de clase.

3.4. Resultados

Para todas las edades y siguiendo lo revisado en la literatura se miden cuatro parámetros en los dibujos de antes de la experiencia de programación con Cubetto (pre) y después de la experiencia (Post):

- Uso de colores (Uso. Color)
- Si se trazan líneas rectas (L. Rectas)
- Si se aprecian emociones (Emociones)
- Si dibujan el robot (Dib. Robot)

En el caso de los niños de 3 años destaca cómo aumentan el uso del color, líneas rectas después de programar con Cubetto como se muestra en las Figuras 5 y 6. Sin embargo, no se registran grandes cambios en sus emociones como se muestra en la Figura 7 ni en cómo dibujan al robot como se muestra en la Figura 8.

Las Figuras 9 y 10 son ejemplos de dibujos de antes y después de usar Cubetto de niños de 3 años. Como se puede apreciar siguen dibujándolo con brazos y piernas, cuerpo cuadriculado y cara con ojos, nariz y boca. En algunos casos lo dibujan solo (Figura 9), y en otros casos acompañado (Figura 10), pero en todo caso si antes de usar Cubetto ya lo dibujaban solo después lo dibujan solo, y si antes de usar Cubetto lo dibujaban en compañía después de usar Cubetto lo siguen dibujando acompañado.

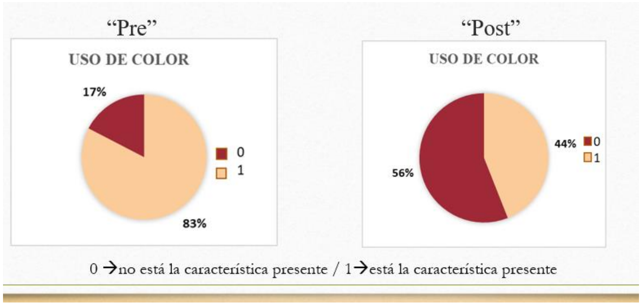


Figura 5. Uso del color (niños de 3 años)

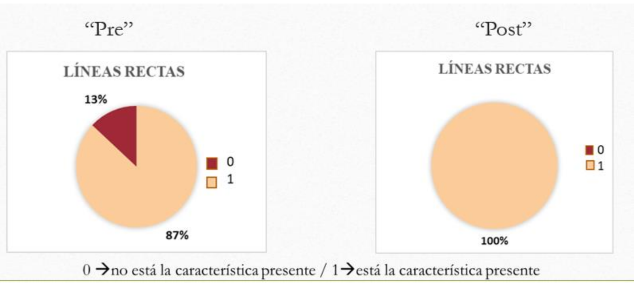


Figura 6. Uso de líneas rectas (niños de 3 años)

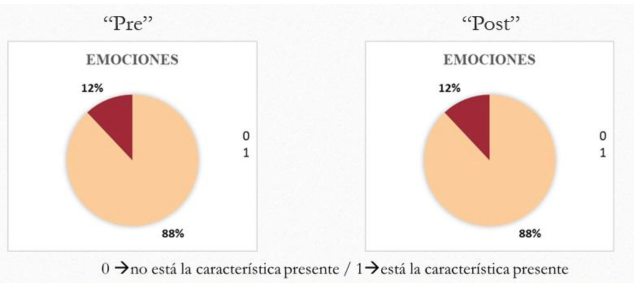


Figura 7. Emociones (niños de 3 años)

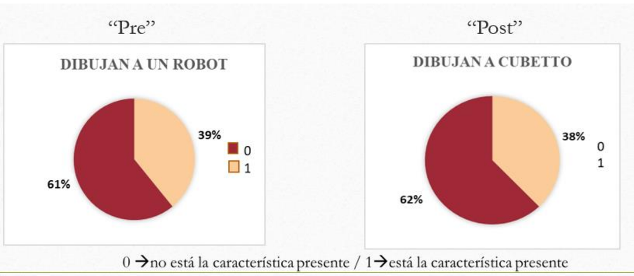


Figura 8. Dibujo del robot (niños de 3 años)

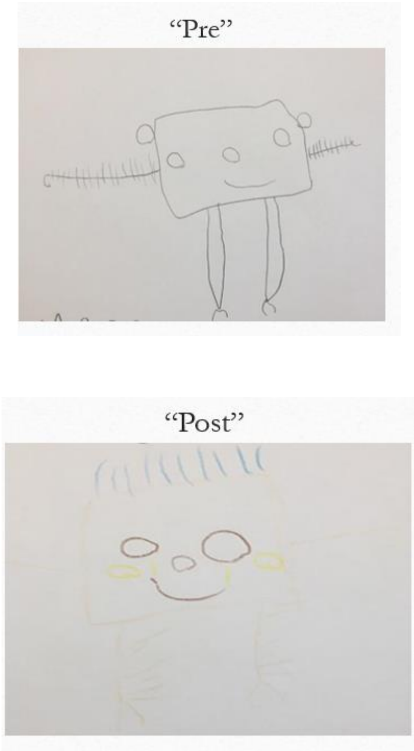


Figura 9. Ejemplo de Cubetto dibujado solo (3 años)

Figura 10. Ejemplo de Cubetto dibujado acompañado (niño de 3 años)

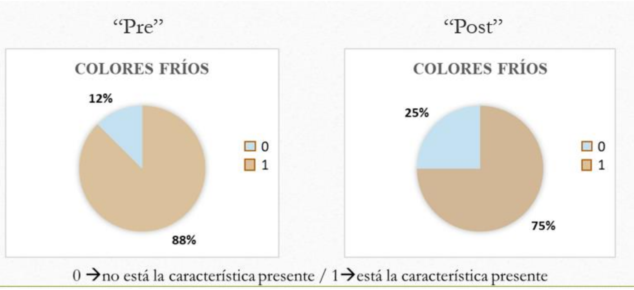


Figura 11. Uso de colores fríos (niños de 4 años)

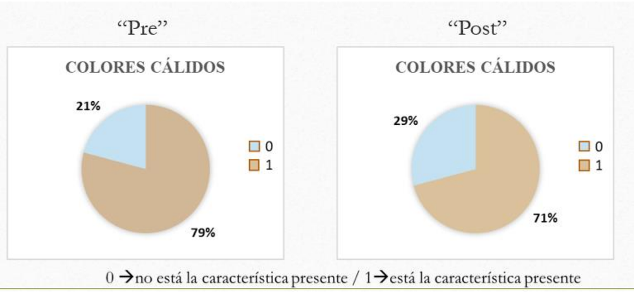


Figura 12. Uso de colores cálidos (niños de 4 años)

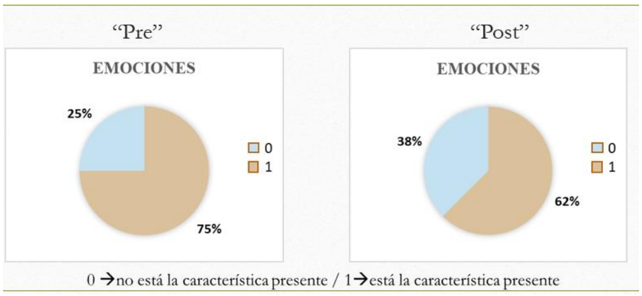


Figura 13. Emociones (niños de 4 años)

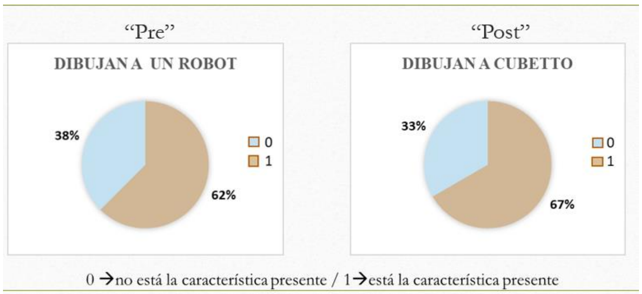


Figura 14. Dibujo del robot (niños de 4 años)

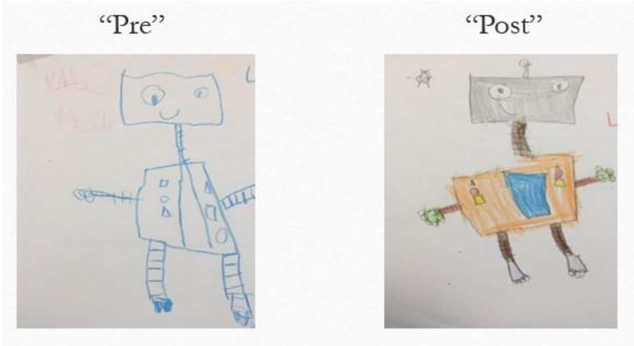


Figura 15. Dibujo de Cubetto solo (niño de 4 años)

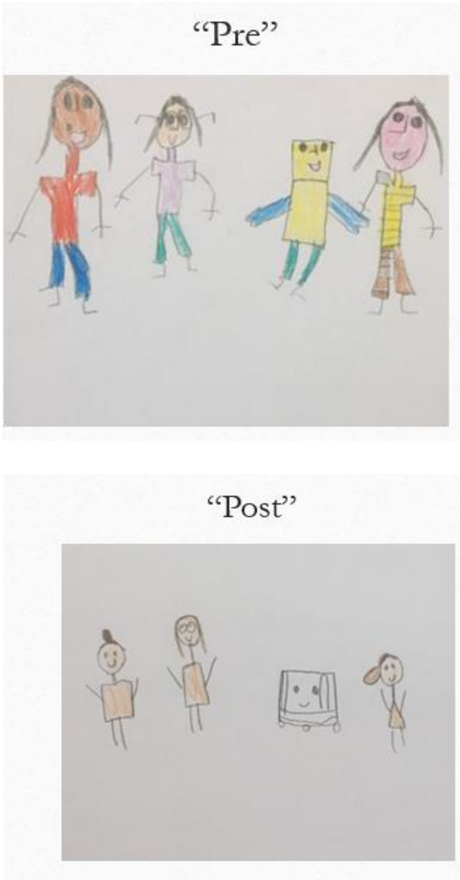


Figura 16. Dibujo de Cubetto acompañado (niño de 4 años)

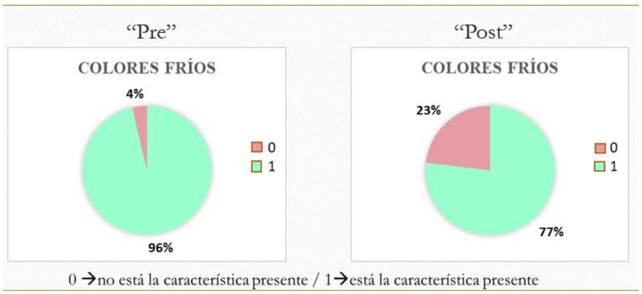


Figura 17. Uso de colores fríos (niños de 5 años)

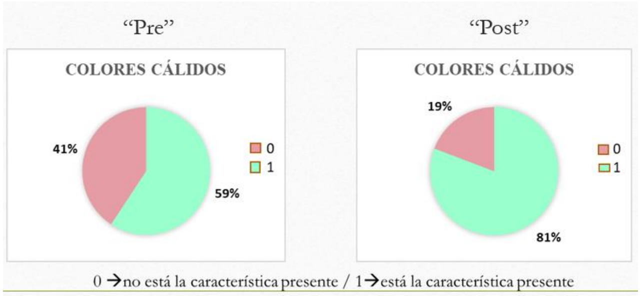


Figura 18. Uso de colores cálidos (niños de 5 años)

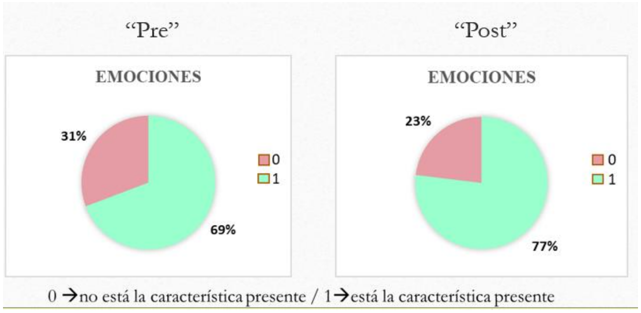


Figura 19. Emociones (niños de 5 años)

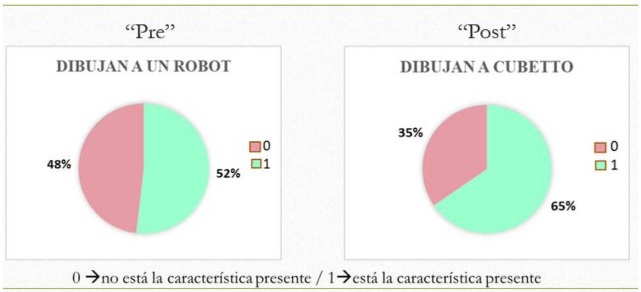


Figura 20. Dibujo del robot (niños de 5 años)

Figura 21. Ejemplo 1 de Cubetto (niño de 5 años)

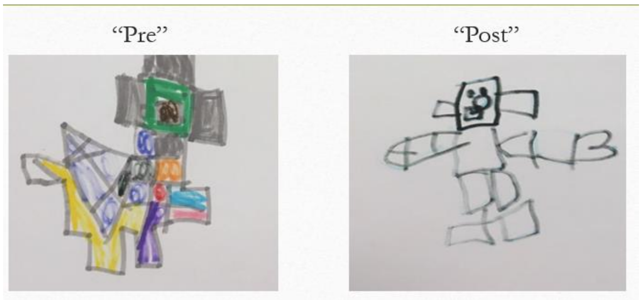


Figura 22. Ejemplo 2 de Cubetto (niño de 5 años)

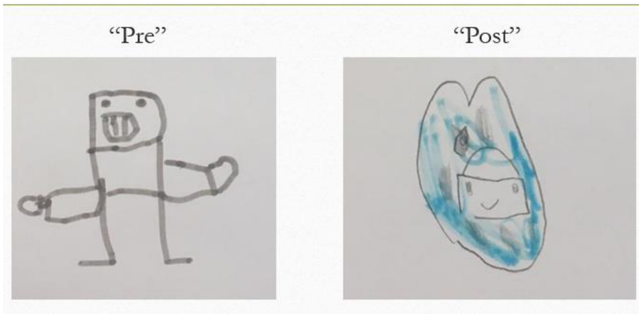


Figura 23. Ejemplo 3 de Cubetto (niño de 5 años)

En el caso de los niños de 4 y 5 años al tener ya un mayor nivel madurativo se analizan más en profundidad si los colores qué usan son fríos o cálidos, además de analizar las emociones y cómo dibujan a Cubetto en las Figuras de la 11 a la 23.

En particular, como se puede observar en las Figuras 11 y 12 en el caso de los colores fríos no hay diferencias significativas en el caso de los niños de 4 años. Sin embargo, en el caso de los niños de 5 años destaca cómo reducen significativamente el uso de colores fríos del 96% al 77%, aumentando el uso de colores cálidos del 59% al 81%.

Respecto a las emociones, como se puede observar en las Figuras 13 y 19 al contrario que en el caso de los niños de 3 años en los que se podía apreciar un aumento del 74% al 88% expresando emociones en sus dibujos, parece que los niños de 4 años reducen ligeramente su nivel de expresividad emocional en los dibujos del 75% al 62%. Es el único caso en el que se reduce la expresión de emociones, puesto que también como en 3 años, en el caso de los niños de 5 años se observa un ligero aumento del 69% al 77%.

Respecto a cómo dibujan a Cubetto los niños de 4 años también lo dibujan en ocasiones solo y en otras ocasiones acompañado, siendo lo habitual que si antes de conocer a Cubetto lo dibujaban solo, después de interactuar con él sigan dibujándolo solo como en el caso de la Figura 15, y si estaba acompañado sigan dibujándolo acompañado como en la Figura 16.

Además, en el caso de los niños de 4 y 5 años, se quiso distinguir si los niños eran capaces de dibujar un robot más parecido a Cubetto una vez terminada la experiencia de programación. Sin embargo, ni en el caso de los niños de 4 años ni en el caso de los niños de 5 años se observa que dibujen el robot más parecido a Cubetto después de haber trabajado con él, siguen dibujando un robot similar al que habían dibujado inicialmente como se muestra en las Figuras 15, 21 y 22. Solo algunos niños dibujaron un robot más parecido a Cubetto, tanto en el caso de 4 años como muestra la Figura 16 como en el caso de 5 años como muestra la Figura 23.

Por último, se presenta en las Tablas 1-3 un resumen de los porcentajes obtenidos para los criterios analizados en el caso de los niños de 3, 4 y 5 años.

Tabla 1. Porcentajes antes y después para 3 años

	Colores	Trazan líneas rectas	Emociones	Dibujan a un robot/ Cubetto
Cumple característica **antes** de conocer a Cubetto	44%	87%	74%	39%
Cumple característica **después** de conocer a Cubetto	83%	100%	88%	38%

Tabla 2. Porcentajes antes y después para 4 años

	Colores fríos	Colores cálidos	Emociones	Dibujan a un robot/ Cubetto
Cumple característica **antes** de conocer a Cubetto	88%	79%	75%	62%
Cumple característica **después** de conocer a Cubetto	75%	71%	62%	67%

Tabla 3. Porcentajes antes y después para 5 años

	Colores fríos	Colores cálidos	Emociones	Dibujan a un robot/ Cubetto
Cumple característica **antes** de conocer a Cubetto	96%	59%	69%	52%
Cumple característica **después** de conocer a Cubetto	77%	81%	77%	65%

4. Conclusiones

Es necesario investigar la relación entre programación y emociones también en Educación Infantil con herramientas adaptadas a la edad y nivel madurativo de los niños.

Se valida el uso del dibujo como herramienta para realizar una análisis vivencial y emocional de la experiencia de programación con Cubetto de niños de 3 a 5 años que aún es posible que no sepan leer o escribir con una experiencia en la que a 77 niños se les pidió dibujar antes y después de programar con Cubetto qué sentían en relación a este robot pudiendo expresar libremente sus emociones con lápiz, colores y un folio en blanco.

El análisis de los dibujos se centró en el uso del color, líneas rectas o curvas, y el estudio de los elementos dibujados, revelando que los niños de 4 y 5 años emplearon colores más cálidos después de interactuar con Cubetto y por regla general, independientemente de la edad no dibujaron al robot más parecido a Cubetto después de terminar la experiencia de programación manteniendo su dibujo inicial.

Es posible que si la duración de la experiencia hubiera podido ser superior a la semana estos resultados fueran distintos. Por lo tanto, se quiere destacar la necesidad de seguir investigando en esta línea con más estudios y niños de distintas edades y culturas para contrastar estas primeras conclusiones.

Agradecimientos

A todos los niños que participaron en la experiencia y al proyecto P2018/TCS-4307 que además ha sido cofinanciado por los fondos estructurales FSE y FEDER.

Referencias

Alomía, M. A., y Samaniego, V. D. (2013). *Estudio del Significado Emocional del Color y Formas Básicas, en Niños de Cuarto Año de Educación Básica de la Ciudad de Riobamba*, Bachelor's thesis.

Anning, A. y Ring, K. (2004). *Dar sentido a los dibujos de los niños.* McGraw-Hill Education (Reino Unido).

Bers, M. U. (2019). Coding as another language: a pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education*, *6*(4), 499-528.

Bers, M.U., Strawhacvker, A., & Sullivan, A. (2022). The state of the field of computational thinking in Early Childhood Education. OECD Education Working Papers, no. 274. https://dx.doi.org/10.1787/3354387a-en

Bueno Fenero, A. (2022). La robótica y el pensamiento computacional mediados por el Aprendizaje Basado en Juegos en la etapa de Educación Infantil, Trabajo Fin de Máster, Universitat Oberta de Catalunya.

Computer Science Teachers Association (CSTA). (2012). Computer Science K–8 Building a Strong Foundation http://csta.acm.org/Curriculum/sub/CurrFiles/CS_K8_Building_a_Foundation.pdf

González-González, C. S. (2019). Estado del arte en la enseñanza del pensamiento computacional y la programación en la etapa infantil. Education in the Knowledge Society (EKS), 20, 15. https://doi.org/10.14201/eks2019_20_a17

Hromkovič, J., Komm, D., Lacher, R., Staub, J. (2019). Teaching with LOGO Philosophy. *Encyclopedia of Education and Information Technologi*es, A. Tatnall (ed.), Springer Nature, https://doi.org/10.1007/978-3-319-60013-0_76-1

Jacobsen, H. (2014). Five-years-old learn coding in schools to prepare for future labour market. *EurActiv.com* - EU News & policy debates, across languages

Jiménez Rodríguez, Y.D., Cueto Barragán, V., Jiménez Vásquez, L.V. (2022). Programando al robot Cubetto: una estrategia para el desarrollo del pensamiento lógico y creativo en la educación inicial. Tesis Doctoral. Universidad Magdalena de Colombia.

Konakli, T. (Ed.) (2021). Curriculum – Releasing Maths Anxiety with the Use of Robotics. Higher Education Course Curriculum for Primary School Prospective Teachers, Kocaeli University. http://www.mindmaths.org/wp-content/uploads/2021/01/CURRICULUM-MINDMATHS_COMPILED_LAST-VERSION.pdf

Lowenfeld, V. y Brittain, W. (1972). La capacidad creadora. Buenos Aires. Ediciones Kapelusz.
Marín, V. R. (1988). El dibujo infantil. Tendencias y problemas en la investigación sobre la investigación plástica de los escolares. Arte, individuo y sociedad. 1, 5, 30. Recuperado de: http://dialnet.uniroja.es/servlet/articulo?codigo=898182.

Montessori, M. (1986). La mente absorbente del niño. Editorial Diana.

Morales-Urrutia, E. (2021). MEDIE_LECOE: propuesta de metodología para la integración de emociones en compañeros de aprendizaje para la enseñanza de la programación en educación primaria. Tesis Doctoral, Universidad Rey Juan Carlos.

Sáez-Fernández, C., Viera-López, G., Pérez-Marín, D. (2018). Propuesta metodológica de la enseñanza de la programación en Educación Infantil con Cubetto, IE Comunicaciones, 1-8.

Salguero, M. J. C. (2011). Importancia de la inteligencia emocional como contribución al desarrollo integral de los niños/as de educación infantil. Pedagogía magna, (11), 178-188.

Santos Miranda-Pinto, M., Orosio, A.J.M. (2019). Aprender a programar en Educación Infantil . *Pixel-Bit. Revista de medios y educación*, no. 55, 133-156. https://doi.org/10.12795/pixelbit.2019.i55.08

Siu, K. W. M., & Lam, M. S. (2005). Early childhood technology education: A sociocultural perspective. *Early Childhood Education Journal*, 32(6), 353–358.

Zufiaur, O. (2014). Potencialidades del dibujo infantil en las aulas de Educación Infantil (Bachelor's thesis).

LARVA, a multiagent system to collect learning analytics and provide immediate feedback to students during laboratory assignments

Luis Castillo
Dpt. Computer Science and A.I
University of Granada
Spain
L.Castillo@decsai.ugr

Francisca Sánchez-Carrascosa
Acceptare Psicología
Spain

Resumen: Las analíticas del aprendizaje pueden tener muchos objetivos, pero entre los más importantes se encuentra lograr el éxito de los estudiantes. Este trabajo presenta una experiencia en un laboratorio remoto 24x7, conectado a un LMS estándar como Moodle, cuya principal intención es promover dos de los ingredientes clave para este éxito: una alta motivación y un alto esfuerzo. En este sentido, esta experiencia proporciona a los estudiantes un feedback inmediato y multimodal sobre las actividades que están realizando con el fin de lograr una mejor autorregulación de su esfuerzo.
Palabras clave: Analíticas del aprendizaje, instrucción personalizada, Telegram chat-bots.

Abstract: Learning analytics may have many objectives, but among the most important ones is achieving student success. This paper presents an experience in a 24x7 remote laboratory, attached to a standard LMS such as Moodle, whose main intention is to promote two of the key ingredients for this success: a high motivation and a high effort. In that sense, this experience provides students with immediate and multimodal feedback on the activities they are performing im order to achieve a better self-regulation of their effort.
Key words: Learning analytics, personalized instruction, Telegram chat-bots.

1. Introduction

After a rather long period of pandemic has strike, from the ground up, the way that college courses are taught, switching from virtual to presence classrooms very frequently and within very short periods of time, many institutions believe that this is an age of opportunity to rethink and reshape the vision and mission of Universities to get closer to that desired excellence. And one of the reiterative issues that arise is that of equity of students, aiming not that all students are the same, but that all students must have the same opportunities to succeed in any college course s/he is engaged in when given enough time, attention and instruction to do so (Ormrod, 2013). And in this vision relies a good part of the teaching excellence that this institutions must pursue (Grajek, 2022). The design of college course must, therefore, be such that allows the greatest number of students to succeed in the course by themselves, with their own effort, taking into account their inherent diversity (Ifenthaler and Yau, 2020). Hence, the design of college courses might be seen like tracing a long term strategy to reach the success, and this cannot be based on a black box with an starting point and an end point with nothing in between. Neither it is defining the path as a set of landmarks, or learning objectives, for which the students pass through with little or none feedback on their trajectory along this path. According to this, there have been many initiatives in the past trying to provide the methodology and scaffolding needed to guide the students to success, monitor their evolution as close as possible, taking into account their different pace of working, and provide them with as much feedback as possible in order to empower their own self regulation. One of these most successful and well founded 1methodology is known as P SI, which stands for Personalized System of Instruction (Fox, 2004; Keller, 1968).

It is based on a system that rewards the advance of the students rather than penalize them for their errors.
P SI promotes student self-regulation, to advance at their own pace, a set of increasingly difficult teaching units which must be sequentially mastered, giving immediate feedback to the student about their progress and having a kind of individual tutoring or proctors to help students to advance. P SI stands for flexibility, that is, deliver education anywhere, anytime at their own pace. Unfortunately, though initial experiences with P SI where shown to be very successful, it gradually felt into disgrace as the requirements of self pace did not fit into regular academic calendars, and the cost of immediate, any time feedback and proctoring became unaffordable.

This paper presents LARVA, a multiagent ecosystem which uses state of the art, free technological tools to overcome some of the main drawbacks of PSI. LARVA stands for Learning Analytics Recollection and Visualization Agents[1], and it is a 24x7 remote laboratory, implemented as a multiagent ecosystem, which uses uses state of the art, free technological tools to monitor the progress of students while they learn to implement multiagent systems coded in Java. LARVA gives them immediate, real-time feedback about their success or failure in achieving the goals of their Java programs through different means, being the following two, the most important ones. Firstly, direct data returned by LARVA system functions to their Java programs, to keep students programs informed of successes and failures, And, most importantly, they also receive textual, friendly readable feedback through a Telegram chat-bot, who is a core part of LARVA, that allows students to interact and request further details and support for the next steps. In order to do that, LARVA is based on a set of milestones, defined by the teacher, that should be accomplished by students' programs to succeed in the course. Then, students execute their programs within the remote laboratory and LARVA automatically monitors the evolution of the students and detects the achievement, or not, of every milestone required by the teacher. LARVA obtains its data from a well known LMS as Moodle (Rice and William, 2006) and sets up a 24x7 virtual laboratory to automatically monitor the evolution of the students lab work. It is based on Java, Telegram chat-bots and GMail gateways to provide a multimodal intensive and immediate feedback to students, who end up to be very engaged and motivated (two critical behaviors for success (Tempelaar et al., 2021)) and hence, as the results show and the surveys underline, highly productive.

2. Related work

When all students, not only the mainstream ones, are given enough time and feedback (Ormrod, 2013), they do their best to overcome a course and then, a large majority of them succeed in this endeavor (Ifenthaler and Yau, 2020). And in this vision relies a good part of the teaching excellence that universities must pursue (Grajek, 2022) and it is also one of the grand challenges for learning analytics to become the enabling discipline for these goals (Ihantola et al., 2015) . With this aim, this work relies on a successful and well founded methodology which is known as P SI, Personalized System of Instruction (Fox, 2004, Keller, 1968). It is based on a system that rewards the advance of the students rather than penalize them for their errors to keep them as much engaged and motivated as possible, which are two critical behaviors for success (Tempelaar et al., 2021). PSI promotes student self-regulation, to advance at their own pace, a set of increasingly difficult teaching units which must be sequentially mastered. It suggests both giving the student immediate feedback about their progress and having a sort of individual tutoring or proctors to advice the students on how to advance. P SI foundations lie at the core of many other milestone-based approaches like (Stanton and Lindsay, 2011) in a mechatronic college course, which increases students autonomy and self management to reach more or less difficult goals according to their own skills or in (Keengwe, 2022) where it is shown that it stimulates critical thinking of students when faced to the challenge. Even in (Falkner et al., 2014) a very similar approach to LARVA focuses on achieving an automated assessment of students based on the set of milestones reached. On the other hand, the use of chat-bots to support the progress of students is an area which is receiving increasingly more attention (Hien et al., 2018, Setiaji and Paputungan, 2018) as their capabilities to support students as intelligent assistants is also increasing.

[1]*All source code in Java may be downloaded from the following repositories*
https://github.com/Anatoli-Grishenko/es.ugr.larva.core
https://github.com/Anatoli-Grishenko/LARVA

Unfortunately, though initial experiences with PSI were be very successful, it gradually started to decline as the requirements of self pace did not fit into regular academic calendars, and the cost of immediate, any time feedback and proctoring became unaffordable. To this end, LARVA is going to show that these flaws might be overcome with free, state of the art technologies that allow a continuous 24x7 monitoring of students and an immediate supportive feedback through the Telegram chat-bot.

3. Setting the scenario

This section describes how the target course has been structured as a set of increasingly difficult, sequential milestones, and how it has been implemented as a remote laboratory (Castillo, 2016) named LARVA. Structuring the course as a set of milestones is not something new, and it has been implemented that way during the past years, from 2015, with different feedback strategies. Before 2018, the progress in the achievement of milestones was only visible to the teacher, just for assessment purposes. Later (2018 and 2019) a web site was used to inform students, near real-time, about their progress (Castillo Vidal and Sanchez Carrascosa, 2020). And, finally, in 2002 and 2021, an interactive Telegram chat-bot has been offered to students (the web site no longer exists) to allow students to interact with it and gather useful information and clues on their progress.

3.1. Course structure

In the very core of LARVA, every course is composed by a set of participants and a set of assignments, which can be imported from the data in Moodle LMS (Rice and William, 2006). Students are arranged into groups, randomly generated by the teacher, between 3 and 6 members each group. Every assignment is composed of one or more problem instances and every problem is described by a set of milestones that the student must reach. In our case, there are 9 problems in increasing level of difficulty and each problem is composed of file milestones that the students must accomplish one after the other. It is commonly the case that the skills developed to solve a problem are not enough, qualitatively or quantitatively speaking, to solve the next problem, therefore, moving onto the next problem pushes the skills of the students a little forward, generating new milestones to be achieved. Each of these milestones are fully detectable by the server while the session is open and this provides the basement of all the remaining features. As a remark, students are allowed to repeat the same problem as many times as necessary, until the quality of the solution meet their expectations.

It is very important noting that this structure of course has been active during all those years, but only until 2018, and just for the eyes of the teacher, strictly for evaluation purposes. In 2018, with the goal of increasing the feedback given to students, this structure was made visible through a very simple web (Castillo, 2018), which show their progress in the achievement of every milestone of every problem, nearly in real time. Later, in 2019 this web was enhanced with more features to allow students to interact with the web and analyze their progress. And finally, from 2020 on, the web was replaced with the Telegram bot, with a fully interactive dialogue with students and, more importantly, able to push, in real time, the achievement of milestones directly into students cellular phones. So students do not have to go to a web to check their progress explicitly. Instead, the Telegram bot reaches them immediately and interact with them.

3.2. Privacy and security

In order to provide the necessary security (only students officially registered in the course can access LARVA) and privacy (students keep a public record of progress but also a private one) any access to LARVA is based on the use of a private passport for each student. This passport is a RSA encoded ASCII string which encrypts the identification of the students in the Moodle LMS. These passports are delivered to students shortly before the beginning of the course and it is also required to connect from their Java programs in order to authenticate these programs as belonging to those students.

3.3. Remote lab architecture

LARVA's basic infrastructure was initially set up in (Castillo, 2016) where the virtual server was built for the presence course Agent Based Development in the fourth year of the Grade on Computer Science. It is provided with a variety of problem instances so that the teacher can choose among different implementations year by year, and, although the main ideas still remain, it has been further enhanced with a rich ecosystem of specialized agents who provide specialized capabilities that are depicted in Figure 1 and that are described in this section.

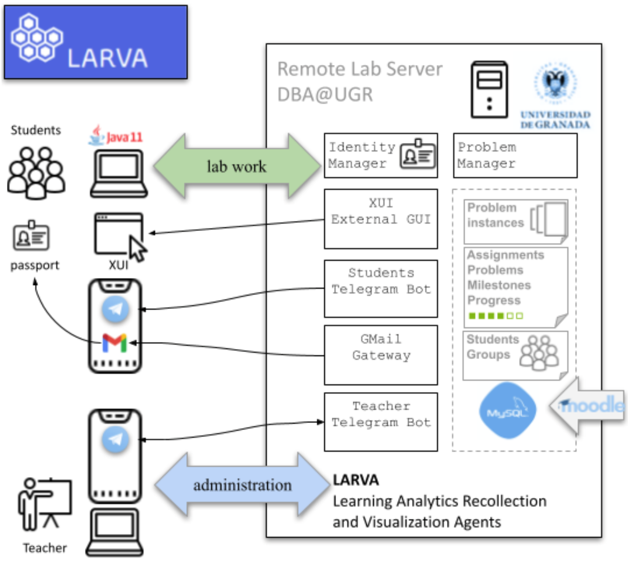
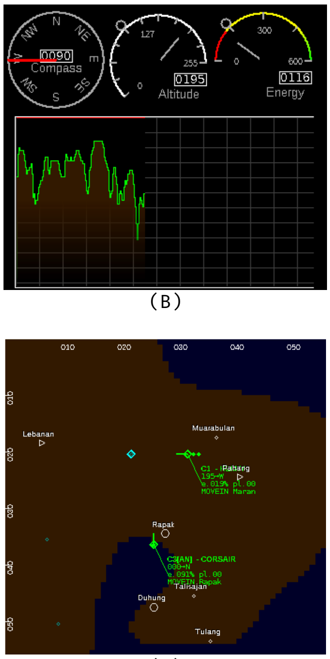


Figure 1. LARVA's architecture. The server provides a 24x7 remote laboratory for students to develop theirlab work, providing immediate feedback at multiple, but complementary, levels.

Basically LARVA is a server running an instance of Fedora Linux, to which students remotely connect in every session of the laboratory classroom from any laptop running either Windows, Mac or Linux. On top of this, LARVA is a running instance of a Jade multiagent ecosystem (Bellifemine et al., 2007) whose agents develop specific functions. The first remarkable agent is the "Identity Manager", who is LARVA's gate keeper, in charge of checking that any access to LARVA is an authorized and authenticated one. Second, there is a "Problem Manager", the agent in charge of opening problems for authenticated students and monitor their evolution. On the other hand, students connect to the server, open a problem and try to solve it, receiving direct information from the server about their results in every problem (see Figure 2).

```
2022-01-06 11:32:33:295 Next: moveW
2022-01-06 11:32:33:295 trying to execute RIGHT
2022-01-06 11:32:33:295 Executing RIGHT
2022-01-06 11:32:33:295 Updating dashboard
2022-01-06 11:32:33:296 Sending ACLM|PFM|INFORM|
SND|3E0C93|RCV|S.M. Cafedow|CNT|Request execute
RIGHT|RPT|CNV|SESSION::NzgjpHkv|
IRT|SESSION::NzgjpHkv|2022-01-06 11:32:33:298
Received ACLM|PFM|INFORM|SND|S.M. Cafedow|
RCV|3E0C93||CNT|Inform execution of action RIGHT:
Success|RPT|CNV|SESSION::NzgjpHkv|
2022-01-06 11:32:33:299 OK
2022-01-06 11:32:33:299 Status: SOLVEPROBLEM
```

(A)



(B)

(C)

Figure 2. (A) Textual, timestamped output of a student session as it is received by their own Java programs. This information is rendered by the XUI agent into visually understandable assets like the next ones. (B) the state of the compass, the geographical altitude and the remaining energy or the vertical evolution of the trajectory. (C) a map showing the situation of agents in the 3D-world and the interactions with other groups of students in the same work session. Each agent is displayed as a green label showing its position, its course or additional data about its current activity.

This year, the problems assigned to the students are related to autonomous vehicles, such as Google car or Tesla, which move through a three-dimensional, unknown world that agents must navigate by using a set of sensors in order to find their own routes to reach their designated goals.

The server records every interaction with the students and keep a record R of all these interactions (Table 1). There are nearly 90000 tuples stored which corresponds to more than 30500 work sessions and 77 groups of students registered during all those seven years (see Table 2)[2]

Group	Year	Time	Session	Problem	Milestone
GA	Y2017	08/11/2017 10:50:14	Achernar6	P9	20
GA	Y2017	08/11/2017 10:50:15	Achernar6	P9	40
GA	Y2017	08/11/2017 10:50:16	Achernar6	P9	60
GA	Y2017	08/11/2017 10:50:24	Achernar6	P9	80
GA	Y2017	08/11/2017 11:14:39	Achernar7	P9	20
GA	Y2017	08/11/2017 11:14:40	Achernar7	P9	40
GA	Y2017	08/11/2017 11:14:39	Achernar7	P9	60
GA	Y2017	08/11/2017 11:14:41	Achernar7	P9	80
GA	Y2017	08/11/2017 11:14:52	Achernar7	P9	100

Table 1. A sequence of the activity record R of LARVA which shows a group, named GA opening two work sessions of the same problem (P9) which consists of five milestones (20,40,60,80,100, being 100 the full solution to the problem). The first session (at 10:50:14, a little later) ended failing to solve P9 after 10 seconds, but the next one at 11:14:39 does complete the problem P9 by reaching the last milestone

Year	Y2015	Y2016	Y2017	Y2018	Y2019	Y2020	Y2021
N. Days	33	24	30	18	28	16	39
n. Records	12088	12525	9088	5705	14475	21188	11961
N. Sessions	4489	4538	3661	2811	5156	3904	6113

Table 2. Size of record R

3.3. Multimodal feedback

Providing immediate and detailed feedback to students is a key issue within the P SI approach (Keller,1968, Fox, 2004) to validate their progress and reached milestones, inform of the errors committed or about the next steps to fix mistakes or to reach new milestones. This feedback is the basis for the self-regulation of students and must be provided to each student at their own pace or work, at any time of the day, any day of the week, as they need it. LARVA provides several specialized agents to provide as much feedback as possible to participants by different channels.

Direct feedback

Java programs run by students must make use of the passport to communicate with the server, therefore, only those programs are reported back with the direct outcomes of their execution (see Figure 2) so students have a first level at which they receive information about their progress. However, the truth is that, since this feedback is from software to software, its content is strictly limited to the reception of plain strings of ASCII characters which contains agents's sensors and the outcomes of their actions.

External user interface

Since the programmatic feedback obtained by students is very informative, but it is just textual information, LARVA provides an enhanced eXternal User Interface (XUI) which is a personalized agent for every student, able to provide them with added value graphical visualization, more easily interpreted by students about the state of their programs, and the track produced by them in the server (Figure 2). Usually a crash of these programs is both reported programmatically in Java but also visually through the XUI (and explained in the Telegram chatbot), so students may have a better understanding of their failures and successes and, even, have the possibility of comparing their benchmarks with other students or other groups.

Dialogued feedback

Telegram chatbot ”DBA DROID” provides explainable references to the advancement and progress of students. This information is complementary to that provided by either direct feedback or the external graphical interface. It highlights the main outcomes of the students programs and explain them in a friendly language: the milestones already achieved, problems solved, errors committed and cumulative grades obtained in a friendly way (see Figure 3) or even administrative tasks carried out by the teacher. Students programs run

[2]*All the datasets and scripts in R-Studio can be downloaded from the public repo https://github.com/Anatoli-Grishenko/ProcessMiningDBA2015-2020*

very quickly and there might be a lot of information on-screen. This telegram bot, wraps up the most useful information on each episode and sends it to the students cellular phones for a complementary and interactive analysis of the episode. Even more, LARVA's database links all milestones to the corresponding teacher slides on how to reach those milestones and steps, and how to address errors committed, so the Telegram bot may also provide this support when students miss a certain milestone (Figure 3).

Weekly and administrative reports

In addition to this server feedback, students are also reported weekly of the whole progress of the class. This report includes details about both the effort of each group of students, like the number of sessions open, and performance, like the number of milestones reached. Therefore, it provides a good mechanism of comparison and cross check of the progress of every group of students.

3. Results

All this effort to structure the course as a set of milestones, automatically detect their achievement, and report back the students about the outcomes of their programs, as soon as possible, has been developed with a simple goal: allow students to progress at their own pace but receiving as much feedback as possible, in real time, to guide them towards success. This section shows an analysis of the metrics which are extracted from record R, all of them related to each group of students.

Sessions Total number of work sessions of each group g recorded in R

Perseverance It is the number of failed work sessions needed to solve a given problem for the first time. It is relative to the number of sessions of each group and averaged between all problems.

Grade Final grade obtained by each group

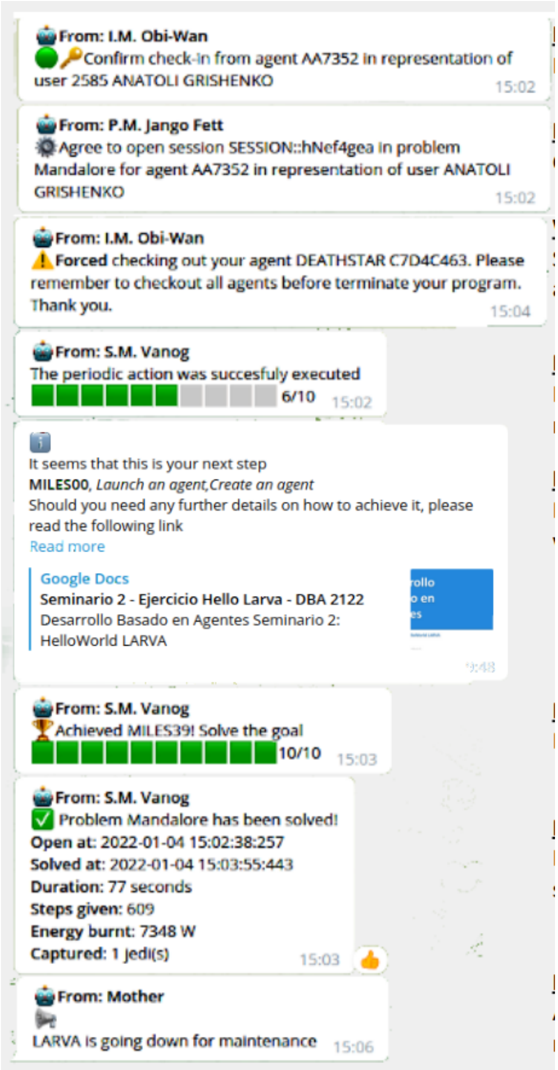


Figure 3. Friendly feedback provided by the Telegram bot during a work session, informing of their progress, errors, privacy issues and maintenance tasks

Weighted Problems Solved The set of problems solved is a poor indicator, but since it is also a time series, that is, the time at which every problem has been solved, the Weighted Problems Solves is the average of the times of solving of each problem. The closer to 0 is this metric, the higher is the number of problems and the earlier they are solved.

Closing.Time The first time that all the problems have been solved, that is, the assignment might be delivered to the teacher, unless students wanted to make any further improvement before the end of the laboratory.

Secs.Day The total duration of working sessions of each group normalized to the duration of the laboratory.

Reaction.Time It is the time elapsed between the first time a problem is open and the first time the same problem is solved, normalized to the duration of the laboratory

Improving.Time It is the duration of all working sessions on a problem, after it was solved, looking for a better solution to the problem.

4.1 Students dedication

Firstly, Figure 4 shows a summary of the sessions opened by students. At the top there are the number of sessions opened in the server and their distribution along the days of the week (left) and hours of the day (right). It may be seen that students are very engaged, connecting to the virtual lab 24 hours a day, 7 days a week, with a quite high number of active working session every year. In the second row, there is the analysis of the two first metrics, Sessions and Perseverance. It may be seen that they are fairly normal metrics, given the number of groups studied. Although Shapiro-Test (S) gives a low pvalue ($\alpha = 0.05$), Kolmogorov-Smirnov (KS) give a higher value, so both seem quite acceptable.

The third row in the figure, shows the distribution of quartiles of each metric along the studied years. ANOVA (A) and Kruskall-Wallis (KW) tests give the same result about these distributions, also supported by a further TukeyHSD test (THSD): there is no evidence that these distributions might be different among all years studied, that is, the number of sessions and the fail ratio might be considered equivalent every year with some differences due to randomness. The last row (green) shows a comparative between the period previous to larva (2015-2017) and the period after LARVA (2018-2021). The first comparative (T.Test) shows that the number of sessions is slightly lower with LARVA than without it and it is statistically significant. However, the perseverance might be considered the same. That is, student carry out almost the same effort every year, with a similar number of sessions and an equivalent fail ratio. Let's go deeper to speculate about why this could have happened.

4.2 Students performance

Previous section just show a quite strong commitment and a quite relevant amount of working sessions open by students, but let us focus now on the effectiveness of this effort and the performance of the students under different perspectives. Figure 5 shows very interesting and robust results about how LARVA might have influenced the performance of students by looking at the most important key performance indicators. The weighted problems solved presents some discrepancies over the years, ANOVA a TukeyHSD give independence over the years but Kruskall-Wallis does not. However the differentiation between LARVA and before is statistically very significant (T.Test) and give advantage to LARVA, that is, students with LARVA solve more problems and much earlier that the ones without LARVA. On the other hand, grades slightly vary over these years, and a further analysis (in green) shows that the grades during LARVA are slightly higher , though the evidence is not statistically significant, by very little.

Now let us look at the secondary metrics, as shown in Figure 6. The first column shows that during LARVA, students show a higher frequency of connections to the server, with a significant higher time/day. Closing time, in the second column, is very similar every year, with a slight, but significant, difference: during LARVA students seem to finish a bit earlier. If we look now to the Reaction time (third column), although there seems to be a clear advantage in favor of LARVA to the naked eye, clearly with shorter reaction times in the first two quartiles, but there is not enough evidence and this might be due to randomness. And finally, a very interesting metric is that of improving time, the time devoted by students to improve their solutions. Its clearly visible in the distribution over the years but there is also a significant supporting evidence to claim that students with LARVA devote more time to improve their results.

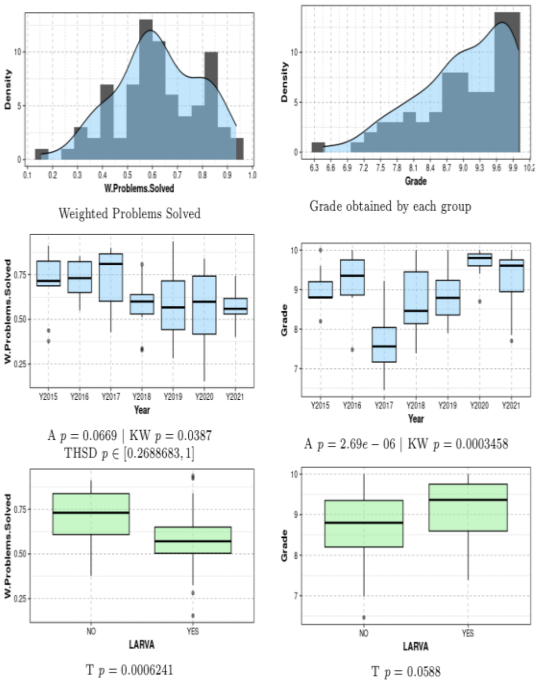
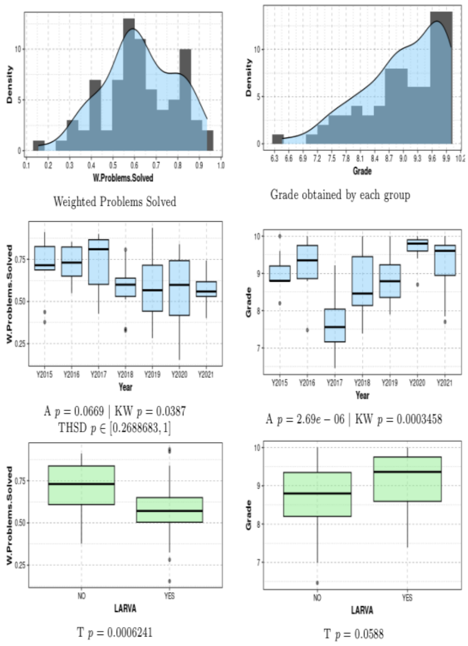


Figure 4 : Distribution of the number of sessions open in the record R. These tests show that the effort is practically equivalent across the years, with some non significant variations due to randomness



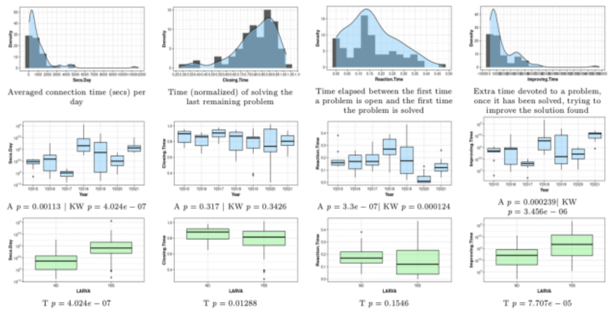


Figure 6. Secondary metrics studied along the years

4.3 Students survey

Previous analysis has shown that the effort of students has been really important, with many different trials on the same problem, students end up being very productive and that more groups have struggled to reach a higher performance. This might have been enough to justify the cost of building a system such as LARVA, but in addition, students (only last year) were required to fulfill a Likert style survey which has produced the following results (Table 3), which could not be better aligned with the goals of this paper. The first interesting thing is that 89% of students do not think it is a difficult course. It seems that the huge amount of time devoted does not constitute a burden for them. They agree that the assessment method is fair, so the general feeling points to a satisfactory learning experience. Another 100% think that structuring the course over a set of consecutive milestones has been useful for their learning experience. 100% of students believe that the feedback provided by the Telegram bot is useful and another 100% recognize that the feedback provided by DBA Droid has produced a change in their behavior at some point in the course, much stronger than the influence of the weekly report, although they contain similar information.

Continuing with the Telegram bot, 82% of students think that they would have not meet all the milestones without the help of the bot, 100% of students recognize that being informed of the achievement of milestones has kept their motivation high and that their effort is being recognized in the assessment (even 93% think it has been funny!).

4.4 Limitations of this study

One of the main concerns regarding the results presented in this paper might be that the environment in which it has been carried out is very specific, in the case of Java programming language and the use of a multi agent system server, what could make its adaptation to other environments a little harder.

However authors would like to add that an extension of this study is being carried out in other general programming courses like introduction to C++ or advanced C++. In these environments, using an Integrated Development Environment like NetBeans3 allows a relatively easy integration of a Java monitoring agent with the following capabilities:

- I can be easily integrated into NetBeans make scripts to parse the result of the compilation and execution of students programs, accounting for errors committed, every time students compile their code
- At each call of make scripts, it can also parse count lines of code, number of classes, number of methods, cross references along students code files as a measure of their progress
- In our case, students work is based on a Test Driven Development methodology and use our own adaptation of Google Test 4 able to cover not only unit test, but also integration tests 5, all of them provided by the teachers beforehand. Hence, it gives a very deep information about the integrity and quality of the source code of students and the milestones achieved.
- The monitoring agent can, at any time, connect to the server, which would be exactly the same, to register all milestones achieved and to report the result to students with the same Telegram bot, which would be the same too.

Authors expect to report these results in a shortcoming paper.

5. Conclusions

PSI teaching methodology has been an efficient teaching system based on a general principle of human behavior (Skinner, 1965): the sooner we get the consequences of our behavior, the better this behavior is reinforced either in positive or negative ways. In the context of higher education, giving students immediate feedback about their progress and the outcomes of their actions keeps them alert and busy and reinforces their will to improve (Keller, 1968).

This study simply confirms this postulate. It shows a deep analysis of the recorded behavior of students during their laboratory assignments within a virtual lab during seven years. Their activity has been strongly monitored and recorded and it is divided in two periods, three first years without continuous feedback of their progress and four more years with LARVA, an infrastructure based on an increasing feedback, both with a static web page and, more recently, with a Telegram chatbot that provides immediate feedback and with whom students may dialogue 24 hours a day 7 days a week.

	strong disagree				strong agree
It has been a difficult course	-	8	17	2	1
The assessment method is fair	-	-	-	3	25
Structuring the course as a set of milestones has helped me	-	-	-	1	27
Teachers' weekly reports produced a change in my behavior	-	1	4	10	13
DBA Droid produced a change in my behavior	-	-	-	1	27
Select the most useful features of DBA Droid					
Query my progress	-	-	-	-	28
Feedback on milestones	-	-	-	-	28
The achievement of milestones motivated me to work further	-	-	-	-	28
My effort is being recognized	-	-	-	-	28

Table 3: Part of the survey fulfilled by the students

The use of LARVA has been critical to maintain students motivation and performance very high and as a pleasant experience for them. Indeed, despite the enormous amount of time devoted to the course, the huge number of sessions open in the server at any time, any day of the week, students believe that ”this has not been difficult and the assessment method is fair”. In addition to this, the Telegram bot has been very important, ”critical” under many students opinions, to succeed, and this should not be surprising, since our students today were born as digital students and for them, text messaging applications like Telegram, are part of their daily social relationships with their pairs.

It is always difficult to prove that there could be a cause-effect relationship between an intervention of the teacher and the outcomes of students since there may be important fluctuations year by year, and many hidden variables. However, this study has provided strong evidence that the use of LARVA has improved the learning curve of students (Weighted Problems Solved) in the following way: during the LARVA period, students devote more time every day to their assignments (Secs.Day) and do not stop after the problem is solved. Instead, they also devote more time trying to improve the solution found (Improving.Time) and this has an obvious immediate consequence: they end up solving more problems earlier (WProblems.Solved and Closing.Time). Although not proved by evidence, but clear to the naked eye, during LARVA, the median and top quartiles of their grades are higher (Grade) and their reaction time improves (Reaction.Time). And all this with a clear motivational and satisfaction status of students shown in the survey.

References

(Bellifemine et al., 2007) Bellifemine, F. L., Caire, G., and Greenwood, D. (2007). Developing multi-agent systems with JADE, volume 7. John Wiley & Sons.

(Castillo, 2016) Castillo, L. (2016). A virtual laboratory for multiagent systems: Joining efficacy, learning analytics and student satisfaction. In IEEE, editor, International Symposium on Computers in Education (SIIE), pages 1–6.

(Castillo, 2018) Castillo, L. (2018). Resultados preliminares más significativos tras cuatro años de aplicación de la metodología scrum en las prácticas de laboratorio. ReVisión, 11(1):5.

(Castillo Vidal and Sanchez Carrascosa, 2020) Castillo Vidal, L. and Sanchez Carrascosa, F. (2020). Análisis cualitativo y cuantitativo del impacto de una herramienta de monitorización del aprendizaje en tiempo real. In Actas de las Jornadas sobre Enseñanza Universitaria de la Informática.

(Falkner et al., 2014) Falkner, N., Vivian, R., Piper, D., and Falkner, K. (2014). Increasing the effectiveness of automated assessment by increasing marking granularity and feedback units. In Proceedings of the 45th ACM technical symposium on Computer science education, pages 9–14.

(Fox, 2004) Fox, E. J. (2004). The personalized system of instruction: A flexible and effective approach to mastery learning. In Evidence-based educational methods, pages 201–221. Elsevier.

(Grajek, 2022) Grajek, S. (2022). Top 10 it issues 2022: The higher education we deserve. Technical report, Brandeis University.

(Hien et al., 2018) Hien, H. T., Cuong, P.-N., Nam, L. N. H., Nhung, H. L. T. K., and Thang, L. D. (2018). Intelligent assistants in higher-education environments: the fit-ebot, a chatbot for administrative and learning support. In Proceedings of the ninth international symposium on information and communication technology, pages 69–76.

(Ifenthaler and Yau, 2020) Ifenthaler, D. and Yau, J. Y.-K. (2020). Utilising learning analytics to support study success in higher education: a systematic review. Educational Technology Research and Development, 68(4):1961–1990.

(Ihantola et al., 2015) Ihantola, P., Vihavainen, A., Ahadi, A., Butler, M., Börstler, J., Edwards, S. H., Isohanni, E., Korhonen, A., Petersen, A., Rivers, K., et al. (2015). Educational data mining and learning analytics in programming: Literature review and case studies. Proceedings of the 2015 ITiCSE on Working Group Reports, pages 41–63.

(Keengwe, 2022) Keengwe, J. (2022). Handbook of Research on Digital-Based Assessment and Innovative Practices in Education. IGI Global.

(Keller, 1968) Keller, F. S. (1968). Good-bye, teacher... Journal of applied behavior analysis, 1(1):79.

(Ormrod, 2013) Ormrod, J. E. (2013). Educational psychology: Pearson new international edition: Developing learners. Pearson Higher Ed.

(Rice and William, 2006) Rice, W. and William, H. (2006). Moodle. Packt publishing Birmingham.

(Setiaji and Paputungan, 2018) Setiaji, H. and Paputungan, I. V. (2018). Design of telegram bots for campus information sharing. In IOP Conference Series: Materials Science and Engineering, volume 325, page 012005. IOP Publishing.

(Skinner, 1965) Skinner, B. F. (1965). Science and human behavior. Number 92904. Simon and Schuster.

(Stanton and Lindsay, 2011) Stanton, K. and Lindsay, E. (2011). A preliminary motivational evaluation of milestone-based laboratory assessment. In 2011 Frontiers in Education Conference (FIE), pages T3G–1–T3G–6.

(Tempelaar et al., 2021) Tempelaar, D., Rienties, B., and Nguyen, Q. (2021). Enabling precision education by learning analytics applying trace, survey and assessment data. In 2021 International Conference on Advanced Learning Technologies (ICALT), pages 355–359. IEEE