Education and Information Technologies (2022) 27:1237–1260
https://doi.org/10.1007/s10639-021-10636-7

Patrony: A mobile application for pattern recognition learning

María Lucia Barrón-Estrada[1] · Ramón Zatarain-Cabada[1] · Jorge Abraham Romero-Polo[1] · Julieta Noguez Monroy[2]

Received: 10 March 2021 / Accepted: 16 June 2021 / Published online: 8 July 2021

Abstract
Pattern recognition is an important skill of Computational Thinking and is one of the most important competences for solving a problem that involves finding similarities or patterns in small problems to solve more complex ones. In this work, we present the mobile application software Patrony. The main contribution of this work is to promote the learning of Computational Thinking, especially pattern recognition, in specific sectors of education in Mexico through the simple use of a software application. To evaluate the effectiveness of the mobile application, tests were carried out in two elementary schools with a total of 43 students, which were divided into 2 groups: a control group and an experimental group. The results of the tests showed that the learning gain (M = 6.50 in postest compared to M = 4.94 on pretest) of the students who used our mobile application produces a significant difference with respect to students who learned using a traditional method of classroom teaching. The results also infer that computational thinking applications can be used as effective learning tools within some important Mathematics topics in public and private schools in Mexico.

Keywords Computational thinking · Learning environments · Pattern recognition · Mobile learning

1 Introduction

Computational Thinking (CT) is a mental process for formulating a problem and its solution (Wing, 2006, 2017). It allows the development of competences like building algorithms and detecting and using (recognizing) patterns that are applied to any discipline and that are feasible to build with or without computers.

Ramón Zatarain-Cabada
ramon.zc@culiacan.tecnm.mx

Extended author information available on the last page of the article

In Mexico, CT is not included as a specific subject in the curriculum, and its incorporation as a national educational strategy would make it possible to raise the academic level of students at different levels, providing more opportunities for development (Anonymous, 2020). However, many teachers at different educational levels have approached the teaching of CT through computer programming, arguing that it is the easiest and most appropriate way to do so. In addition, it has been shown that CT is widely used in areas such as robotics, mathematics, biology, computer science and language (Noh & Lee, 2020, Hsu et al., 2018).

Currently, even though the education system in Mexico does not specifically include CT in its basic education curriculum, some CT-related skills that students should acquire (e.g. modularity, algorithms, and abstraction) are addressed in different subjects and areas of study through activities of various types. An example of the above is in 4th grade Math where the student must find the missing figure in a sequence that presents two geometric figures that are reproduced to form more complex figures. Other pattern exercises are with whole numbers where students must find missing terms in the sequences as well as describe how they found the terms.

To this end, teachers use the traditional method of teaching with group explanations and students solve a limited number of exercises provided in the textbooks; the main problem with this method is that there is no personalized follow-up on student learning. On the other hand, it is recognized that a process of great importance in the evolution of the human brain is the process of pattern recognition. That process is produced when information is received from the environment, stored in the short-term memory, and an automatic process of identification of specific contents of the long-term memory is activated in order to predict and wait for what is coming (Pi et al., 2008). This skill focuses on finding similarities or patterns in small problems that have been broken down to solve more complex problems. CT promotes the development of this skill and considers it essential in problem solving (Hsu et al., 2018). Nowadays, an example where pattern recognition is applied is in the political context during an election process, where it is possible to analyse the polarity of comments on social networks, in order to predict trends in voting during an electoral process (big problem). In this case, certain amounts of words (small problems) determine the positive or negative polarity for a candidate. Another example is in the context of public transportation, where it is possible to analyse information on vehicular traffic, the frequency of truck routes, holidays, and the time of day (small problems) to predict the time it will take to reach a destination (big problem).

Similarly, there is no doubt that mobile devices such as smart phones and tablets have become embedded in the daily life of society and are inherent to the new generations. This current lifestyle must be exploited in every possible way. If there is a daily relationship with mobile devices, adapting education on them directly or indirectly would give rise to a generation with higher educational achievements. In relation to the above, it has been demonstrated that learning in science and mathematics subjects or courses at primary and secondary level is improved by means of mobile devices (Crompton et al., 2017). In Mexico, the National Institute of Statistics, Geography, and Informatics (INEGI, 2018) reported that, in 2018, 73.5% of the population aged six or more uses cell phones in some way; that is, at least seven out of every ten people have a smartphone.

In relation to the above and with the aim of addressing and proposing a solution to that problem, this paper presents a mobile application that implements an educational game to promote and motivate the development of computational thinking in children aged 8 to 10, through the learning and exercising of pattern recognition. Our application was designed to be used by children from 8 to 10 years old because in Mexico the official educational program of Math (https://educacionbasica.sep.gob.mx/), for 4th, 5th, and 6th grade, all contain the theme of patterns and sequences, and students enrolled in these grades are in the range of 8 to 10 years old.

The hypothesis raised at the beginning of this project was that those students using the mobile application will more effectively learn and develop the ability to recognize patterns for problem solving in comparison to those students using traditional learning methods.

The document is structured as follows: Section *Theoretical Framework* refers to the important concepts of computational thinking and pattern recognition, under which this work was proposed; section *Related works* describes the work related to the topic of computational thinking and pattern recognition; section *Patrony: An Educational Application for Pattern Recognition* presents the proposal of solution through the Patrony system; *Methodology* contains the description of the methodology; section *Experiments and Analysis of Results* carries out the analysis of results, which are discussed in section *Discussions* and, finally, section *Conclusions and future work* shows the conclusions and future work.

2 Theoretical framework

This section mentions some relevant concepts related to the research work presented here.

2.1 Computational Thinking (CT)

CT emerges as a concept that involves thinking processes in the formulation of problems and representation of their solutions, so that such solutions can be effectively executed by an information processing agent (human, computer, or combinations of humans and computers) (Wing, 2006).

According to several authors, pattern recognition is an important skill of Computational Thinking that involves finding similarities or patterns in small problems to solve more complex problems (Wing, 2006, Grover & Pea, 2013). In (Hsu et al., 2018) a very detailed classification of computational thinking is presented through 19 steps, including abstraction, algorithm design, decomposition, and pattern generalization, among others.

A major motivation for introducing CT practices into science and mathematics classrooms is the rapid changes and advances in these disciplines as they are practiced in the professional world. In the last 20 years, almost all fields related to science and mathematics have seen the growth of a computational counterpart. Examples include bioinformatics, computational statistics, chemometrics, and neuroinformatics (Weintrop et al. 2016).

Each country has integrated CT into the curriculum in a different way. In some cases, it is integrated across subject areas, particularly in elementary school, while in others it is part of a separate computer science subject usually taught in high school. In addition, these two approaches are often combined. Some countries consider CT and related concepts to be part of the digital competence curriculum. This is also the case in the European Digital Competence Framework for Citizens.

The Computer Science Teachers Association (CSTA) states that "the study of CT enables all students to better conceptualize, analyse, and solve complex problems by selecting and applying appropriate strategies and tools, both virtually and in the real world" (CSTA, 2011). Although considerable effort has been made to advance the understanding of CT, there are still challenges to be addressed, particularly in terms of bringing computational thinking to schools.

2.2 Pattern Recognition (PR)

The purpose of PR is to extract information related to physical or abstract objects that allows establishing properties from among sets of such objects (Fu, 1982). From the point of view of psychology and cognitive neuroscience, PR is described as a cognitive process that combines information from a stimulus with information retrieved from memory (Eysenck & Keane, 2005). It occurs when information is received from the environment and entered in the short-term memory, resulting in the automatic activation of specific content in the long-term memory.

The PR process involves matching information received with information already stored in the brain and making the connection between memories and perceived information as a pattern recognition step called identification. Pattern recognition requires repetition of the experience. The main type of memory involved with PR is semantic memory (Shugen, 2002). PR allows us to read words, understand language, recognize friends, and even appreciate music.

Within pattern recognition, seriation is the ability to organize elements in a logical order along with a quantitative dimension, such as length, weight, and age (Kidd et al., 2012). Seriation means understanding that objects that can be arranged along a dimension and, to do this effectively, the child must be able to answer the question "What comes next?" These seriation skills also help to develop problem-solving skills, which are useful for recognizing and completing modelling tasks.

This paper focuses on the development of a tool that applies seriation-based pattern recognition.

2.3 Related works

The development of computational thinking has been addressed by different non-profit organizations dedicated to expanding access to computer science in schools.

Their vision is that every student, in every school, can learn computer science in the same way that they learn biology, chemistry, or algebra (Code.org, 2019; Disney, 2019). Also, projects such as Scratch help young people learn to think creatively, reason systematically, and work collaboratively by providing tools for programming interactive stories, games, and animations online (MIT Media Lab, 2019), and applications such as Duolingo offer tools for learning languages through the use of a computer platform where the student learns while playing (Duolingo, 2019).

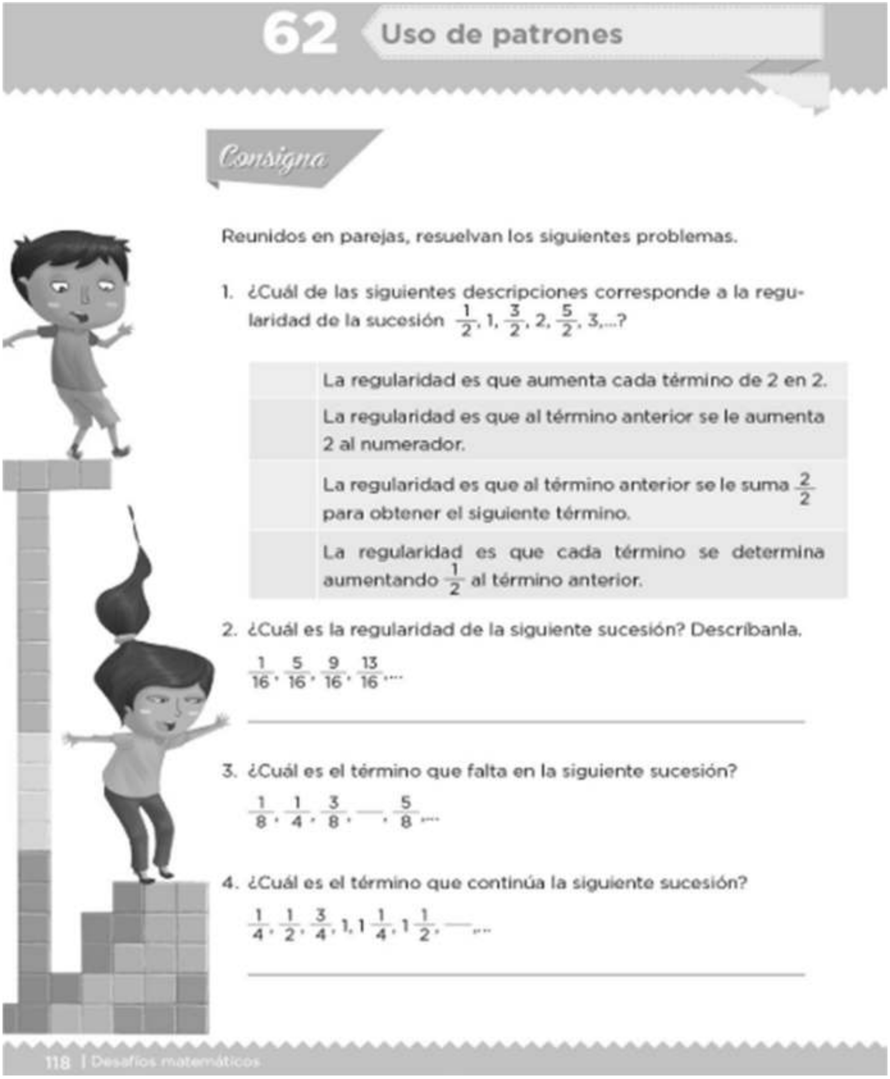
On the other hand, there is a lot of research work in this field of computational thinking.

In Calderon et al. (2015), The authors present an example methodology to include computational thinking in early childhood education. Likewise, the development of computational thinking is addressed through pattern recognition in the first years of education (between preschool and the first two years of basic school education) in the United Kingdom. In this work, the authors mentioned that the proliferation of tablets, such as the iPad, has made it so many children already have previous experience of informal learning through mobile devices once they enter formal education. The foregoing gives rise to a pedagogical change in which mobile devices are used for the development of skills within the mentioned educational period. In Grover & Pea (2013), the authors review recent research on computational thinking. Among the results, they mention that the defenders of computational thinking accept that, although it shares elements with mathematical, engineering, and even design thinking, it also extends each of these abilities in a unique way. It is also highlighted that countries such as Russia, South Africa, New Zealand, and Australia have already included computational thinking in the K-12 curriculum. The work concludes by highlighting that there is still much to do to develop a clearer theoretical and practical understanding of computational skills in children. In Hsu et al. (2018), the authors carry out a review of the literature available in academic journals on computational thinking, from 2006 to 2017. Among the results of this research, they found that the number of articles on computational thinking has increased substantially in recent years. The study reveals, through statistical analysis, that computational thinking activities were primarily introduced in courses in program design, computer science, biology, and robot design. The work concludes that so far, a series of computational thinking activities have been integrated into various subjects in a life-based way. Another finding from the study was that most of the research focused on project-oriented learning, problem-oriented learning, collaborative learning, and game-based learning. In Roscoe et al. (2014), the authors discuss three approaches to teaching computational thinking to high school students. The first approach is a technology camp called Printcraft in which the use of Minecraft to design a specific building or group of buildings (such as a school) is combined with the use of 3D printers to physically build the elaborate design. Another approach analysed was the AppInventor workshop, a graphical environment for mobile application development. In this workshop, students create a rudimentary application following the instructor's guidance. As a result, students gain an appreciation for the complexity of applications and understand how software interacts with the hardware of a device. The third approach is about a robotics club using Arduino and Lego Mindstorms microcontrollers. The work concludes by mentioning that the analysed approaches teach theoretical concepts to promote the appreciation of technology in the everyday world and clarifies that the evidence on learning gain of those attending the three mentioned approaches is empirical. Yadav et al. (2016) discuss the constructs of computational thinking (algorithms, abstraction, and automation) and how they relate to K-12 learning, and present resources and examples that can help educators integrate computational thinking easily and effectively in their classrooms. They mention that elementary school students can learn about algorithms by dividing a simple daily task, such as brushing teeth, into a sequence of steps; abstraction can be taught through data analysis and representation. On the other hand, automation requires the use of computational tools. The authors mention that options available on the Internet, such as NetLogo, can be used to review computational thinking concepts through simulation. In Yadav et al. (2014) the authors report the results of an experiment that evaluated the impact of computational thinking modules on prospective teachers in a required course in educational psychology at a large midwestern university. The course introduces prospective K-12 teachers to the basics of classroom management, learning, motivation, and assessment. Three hundred and fifty-seven teachers in training participated in this study, which was divided into a control group and an experimental group. The control group had a total of 200 students enrolled in the introductory educational psychology course during the fall of 2011. The control group received typical course content, which included lectures on higher-level cognitive processes, such as problem solving, problem transference, and critical thinking. The experimental group, on the other hand, received the computational thinking module. As a conclusion of the study, it was found that a higher percentage of the experimental group (99% versus 93% of the control group) considered that computational thinking could be integrated into the classes. Finally, it was concluded that the experimental group achieved a better understanding of computational thinking in general compared to the control group.

Candy Crush Saga (www.king.com/es/game/candycrush), is a puzzle game developed by King.com Ltd. originally released in 2012 as a Facebook application. Today the game is also available on iOS, Android, Microsoft Store, and King.com, and has millions of players around the world. Candy Crush Saga has a theme about baking and the main mechanics consists of joining three or more candies of the same type within a board by exchanging their positions. The board is different in each level, with different types of candy and may contain static or behavioural obstacles. Combinations of more than three candies produce special candies depending on the pattern formed. The application has gamification aspects, such as points, difficulty level, progression levels, and stars. In pattern recognition and seriation, (Touw et al. 2019) investigated the effects of a computerized serial completion test which was implemented using a dynamic test approach and its relationship with school performance. The results proved that teaching with graduated prompts is efficient in improving the probability that children can solve series completion problems with precision.

On the other hand, LOTUM GmbH developed a game titled 4 Pictures 1 Word (GmbH, 2019), that requires the user to determine which word is being used for reference on four pictures that are pressed into the interface. The main mechanism available to the user is to respond by touching the letters at the bottom of the screen.

The user needs to recognize the patterns in the pictures provided by the application to find the word that relates them. These illustrations may or may not be elements with which the user is familiar, so their performance is affected by their general knowledge.



62 Uso de patrones

Consigna

Reunidos en parejas, resuelvan los siguientes problemas.

1. ¿Cuál de las siguientes descripciones corresponde a la regularidad de la sucesión $\frac{1}{2}$, 1, $\frac{3}{2}$, 2, $\frac{5}{2}$, 3,...?

	La regularidad es que aumenta cada término de 2 en 2.
	La regularidad es que al término anterior se le aumenta 2 al numerador.
	La regularidad es que al término anterior se le suma $\frac{2}{2}$ para obtener el siguiente término.
	La regularidad es que cada término se determina aumentando $\frac{1}{2}$ al término anterior.

2. ¿Cuál es la regularidad de la siguiente sucesión? Descríbanla.

$\frac{1}{16}, \frac{5}{16}, \frac{9}{16}, \frac{13}{16}, ...$

____

3. ¿Cuál es el término que falta en la siguiente sucesión?

$\frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \_\_\_, \frac{5}{8}, ...$

4. ¿Cuál es el término que continúa la siguiente sucesión?

$\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1\frac{1}{4}, 1\frac{1}{2}, \_\_\_, ...$

____

118 | Desafíos matemáticos

Fig. 1 Numerical pattern exercises with fractions in 5th grade book

3 Patrony: An educational application for pattern recognition

The Patrony software application was designed and implemented based onthe official basic education curriculum in Mexico as a reference, specifically fourth, fifth, and sixth grade for elementary schools. Figure 1 presents four exercises on numerical patterns found in the book (Spanish Language) for 5th grade students (Secretaría de Educación Pública, 2017). For example, question 1 is “Which of the following descriptions corresponds to the regularity of the sequence 1/2, 1, 3/2, 2, 5/2, 3, ...?”

3.1 Educational application design

The exercises found in the textbooks and a series of interviews with teachers helped us to define the features of the software application, which should allow the user to exercise pattern recognition in a dynamic and fun way, and to solve problems related to sequences of elements. The application would be developed so that, the student could easily navigate it from a mobile device (smartphone or tablet). In addition, the mobile device should capture information on the student’s performance and generate actions such as presenting exercises with greater or lesser degree of difficulty, adapting the contents to the learning pace of each student. The application should be dynamic, generating multiple exercises with different levels of difficulty, and contain some elements of gamification to motivate the students during the learning process.

The student interacts with the screen of the mobile device to create their profile and carry out the activities included in Patrony. Likewise, Patrony needs the activity templates that will be generated for the user and, on the other hand, it must store the student’s data and the results of the activities carried out by him/her. This interaction is shown in the context diagram in Fig. 2.

For the development of the system, the React Native framework (https://reactnative.dev/) was considered in the architecture design. The complete system was divided into three packages: Navigation, Containers, and Components. Additionally, the modules: React Navigation and AsyncStorage, were considered. The system architecture diagram is presented in Fig. 3. The React Navigation module contains components dedicated to creating and supporting browsers. In the Navigation package, we find the components that have the task of managing the navigation between the screens (containers) of the application. The AsyncStorage module is in charge of managing the persistence of the information. The Data component of the Containers package makes use of it to store and query the application data. This package also includes the components used to build the main displays. They are the ones that make up the routes of the browsers making the application explorable. The Components package contains all those components that are not browsers or containers and add functionality to other components, such as the audio manager, the image manager, timer, activities, elements, and exercises.

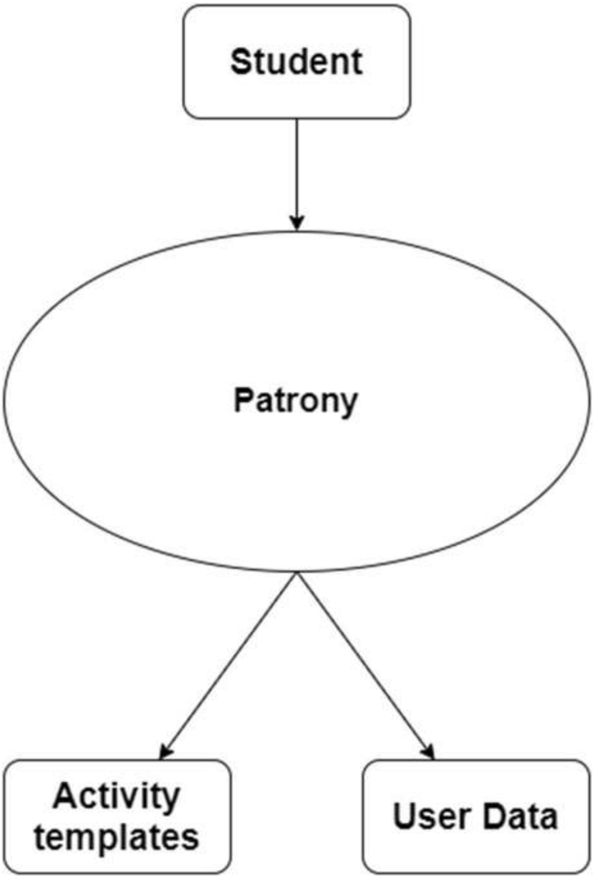


Fig. 2 Context Diagram of Patrony

Table 1 describes the functional requirements that Patrony addressed and covered. The functional requirements represent the behavior of the system in the face of possible user inputs, as well as the outputs it generates towards it.

Table 2 shows the quality requirements that Patrony meets. These requirements serve as a condition for evaluating the acquiescence of the project by validating the acceptability of an attribute or characteristic for the quality of the software.

The main use cases are shown in Fig. 4. The student must be able to select a user profile. Subsequently, the student must select the activity to be executed to finally execute it.

3.2 Implementation of the educational application

Software development models have evolved in recent years to allow the generation of software systems and applications in a short time and to improve them through an iterative and incremental process. Traditionally, software development involves various stages starting with the definition of customer requirements, until obtaining

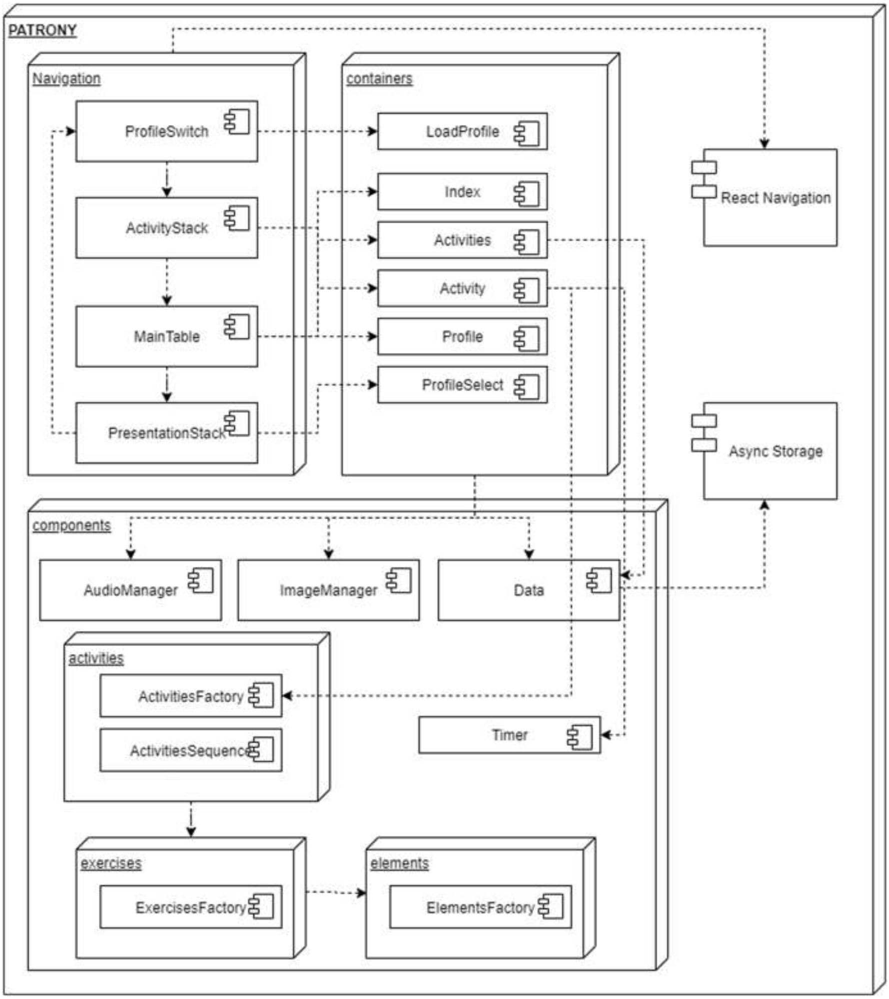


Fig. 3 Software architecture for Patrony

a software product that meets these requirements. In general, all software development methodologies define stages that involve activities such as: requirements analysis, design, implementation or coding, testing and debugging, ending with obtaining a version of the software system.

For the development of the Patrony application, two development models were combined: the prototyping model and the iterative improvement model (Jalote, 2008). The combination of these development models is presented in Fig. 5. The prototyping model (left part of the Fig. 5) was used to develop a basic prototype of the application defined to perform pattern recognition activities. In this model, proofs of concept were carried out defining the graphical user interfaces as well as the interaction activities between the user and the system. The

Table 1 Functional requirements for Patrony

Id	Description
FR-01	The system must contain exercises that require recognition of patterns.
FR-02	The system must organize exercises within activities. The activities will be the levels (stages or phases) of the game that the user must overcome.
FR-03	The system must implement various gamification mechanisms (for example, handling a scoring mechanism when answering the exercises).
FR-04	The system must count the wrong answers of the activities.
FR-05	The system must include a stopwatch to know the completion time of the activities.
FR-06	The system must store the number of wrong answers, the completion time, and the score for each activity.
FR-07	The system must create user profiles to store their information (name, age, gender, and activity statistics) individually.
FR-08	The system must allow changing the current user for an existing one or a new one.
FR-09	The system must have a fuzzy logic module to determine the degree of difficulty of the next exercise.
FR-10	The system must have an affective module to determine the emotional state of the student.
FR-11	The system must contain different levels of difficulty of the activities.

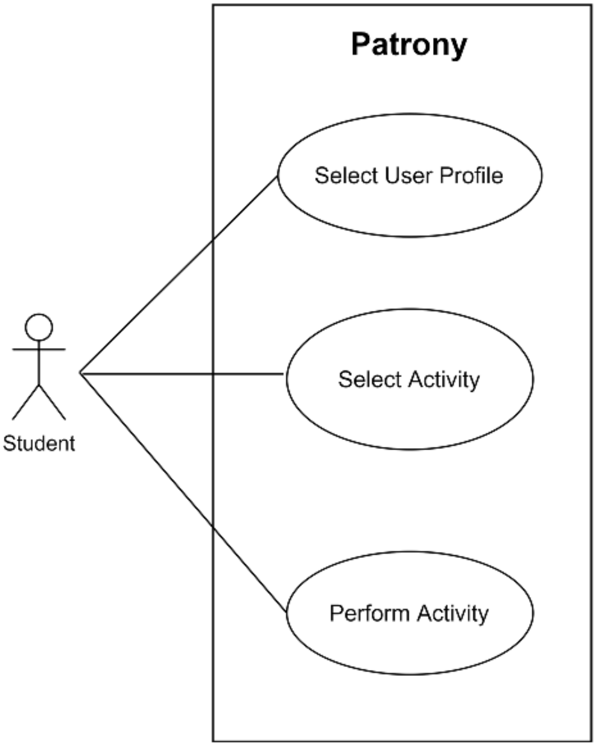
Table 2 Quality requirements for Patrony

Id	Attribute	Description
QR-01	Performance	The system should not take more than two seconds to perform each function.
QR-02	User Satisfaction	The system must include background music and feedback to user actions.
QR-03	Portability	The system must work on both Android and iOS devices.
QR-04	Usability	The system should include a graphical user interface that is easy to use.

prototype produced in this model contained simple interfaces and static activities, and was used as the basis for a system development with the iterative improvement model. The product of the prototyping model was used as the basis for the iterative improvement model (right part of the Fig. 5). The iterative improvement model repeats the development stages producing versions of the software system that improve in each iteration. For the development of the final version of the system, it was necessary to carry out two iterations in order to produce incremental versions of the system. In the first iteration, a system was obtained that used the graphical user interfaces through which we established the main user interaction with the system in addition to the activity generation mechanism. In the second iteration, the final application was developed, which included gamification elements, user profiles, and activities.

The application was built with the React Native framework, with its native handling of the application state, in conjunction with React Navigation 2.X for navigation between screens and AsyncStorage for the storage of user data.

Fig. 4 Patrony User Cases



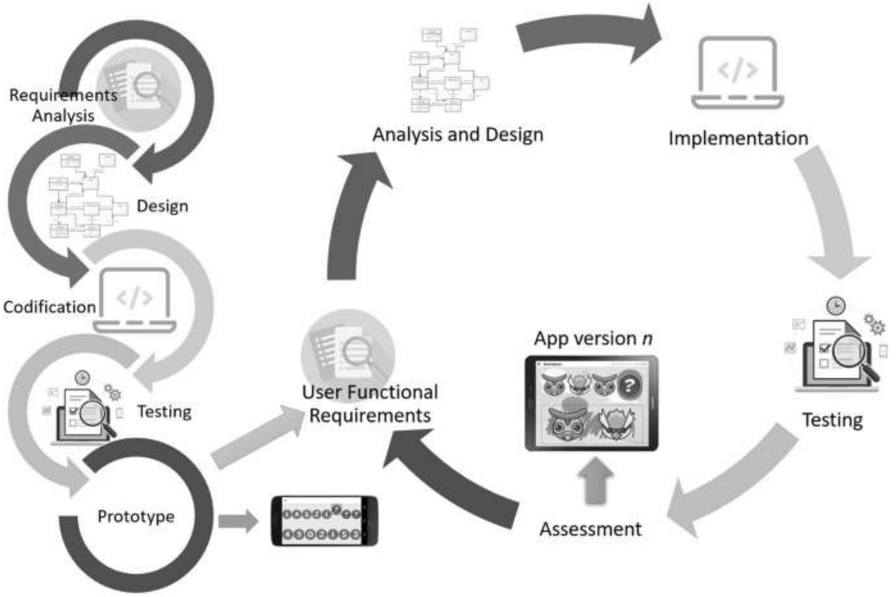


Fig. 5 Combination of development models for Patrony

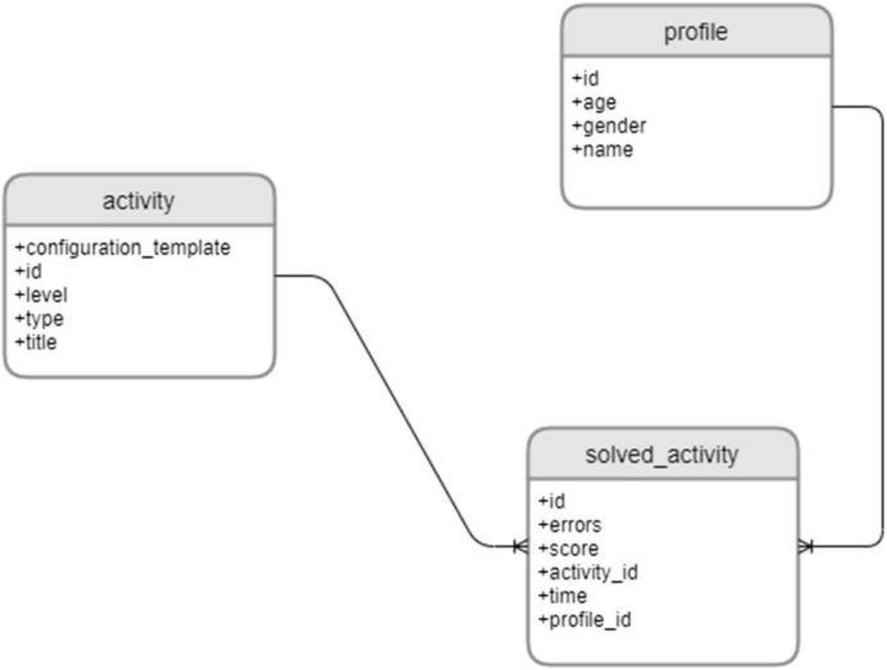


Fig. 6 Data Model Diagram

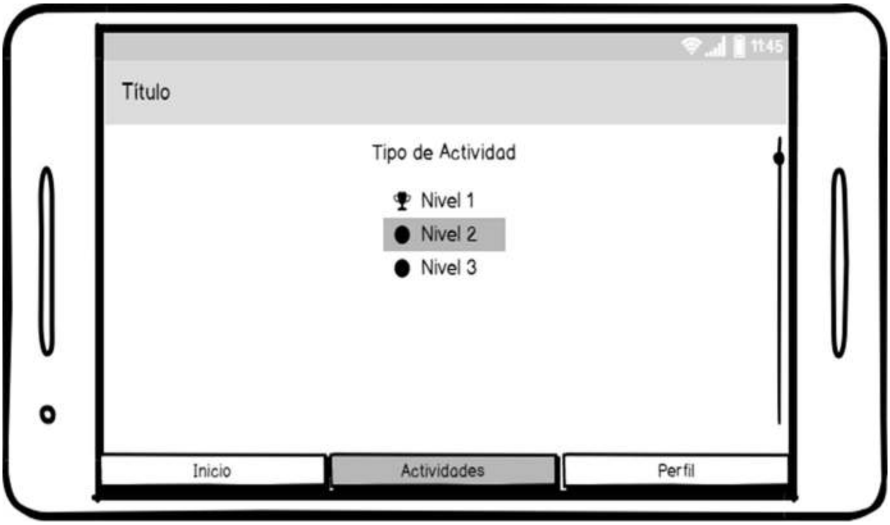


Fig. 7 Mockup of the interface with the main sections

For data persistence, React Native Async Storage was used, which is an asynchronous storage system of the key-value type, without encryption, persistent and global for React Native. Internally it uses RocksDB or SQLite to store the information. Its simplicity of use, the handling of little information and the absence of sensitive information were the main reasons for the use of this component included in the React Native libraries (https://reactnative.dev/). In Fig. 6, we can see the diagram of the data model used.

Figure 7 shows the mockup (design sketch) of the three tabs of the main interfaces of the application with the tab Activities (in Spanish) selected where the list of available activities is shown.

Figure 8 presents the general sketch designed for the activities. There are elements such as: the menu button, the name of the activity, the level of difficulty, a stopwatch and two other gamification elements: the user's score and a progression bar for the number of exercises in the activity. These items are grouped on the top bar. The rest of the screen corresponds to the activity to be carried out.

3.3 Application testing and debugging

The application was tested and debugged on a Moto G5 Plus device and then small adjustments were made to the graphical interface to match the features of the Samsung Galaxy S2 tablet, which would be used in the experiments with students. Figure 9 shows the application (Patrony) deployed on a tablet Samsung Galaxy S2 for an exercise in the basic level of learning.

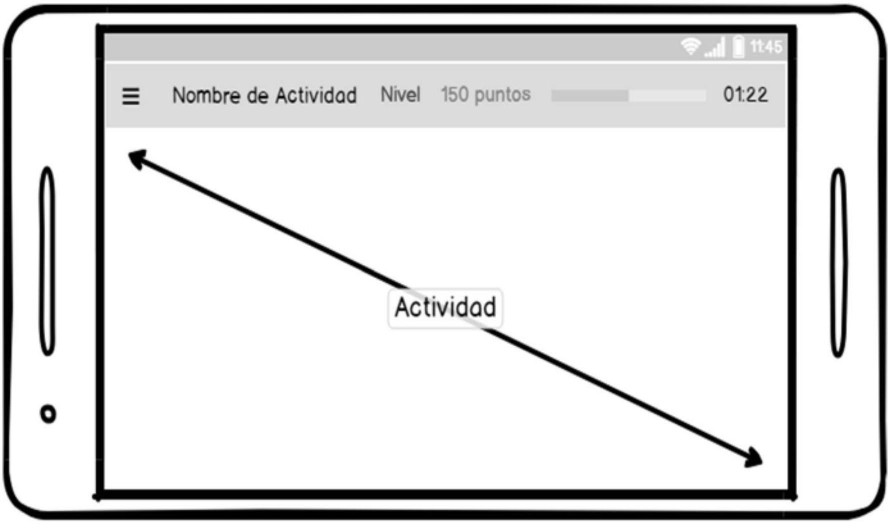


Fig. 8 Mockup of a general activity

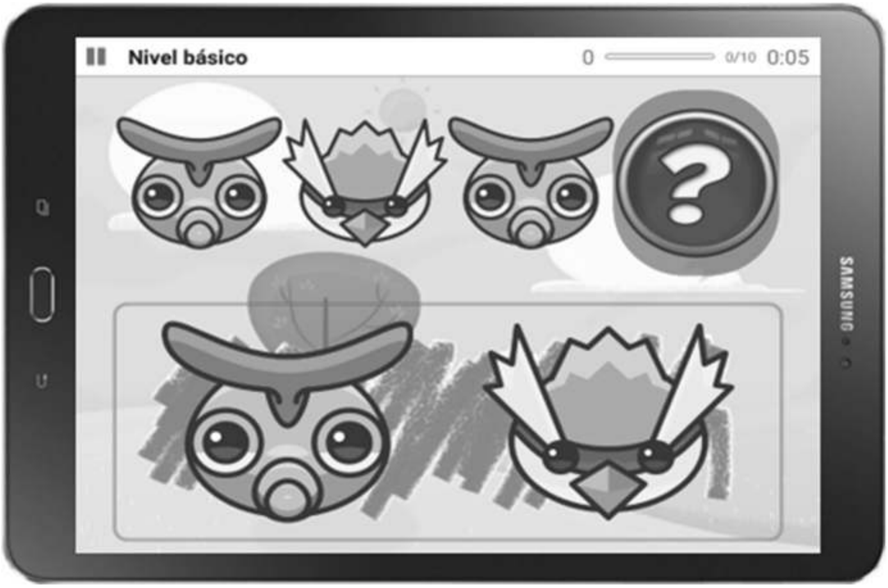


Fig. 9 The Patrony application running on a tablet Galaxy Tab S2

4 Evaluation of software quality

We apply the QASAR methodology for the development of the software system considering mainly the non-functional requirements of Usability, Performance, Portability, and Satisfaction (see Table 2).

Additionally, to evaluate the quality of the software in Patrony, we decided to apply McCall's quality model (McCall et al., 1977). This model classifies all software requirements into 11 software quality factors: Product operation factors which are: Correctness, Reliability, Efficiency, Integrity, and Usability; Product revision factors which are: Maintainability, Flexibility, and Testability; and, finally, Product transition factors which are: Portability, Reusability, and Interoperability. In addition, each quality factor has several quality criteria, giving a total of 33 quality criteria. In the McCall model, the evaluation metric can be binary, giving each quality criterion a value of two (yes / no) according to whether it meets that criterion.

Regarding the quality factor "Correctness", the three quality criteria (Traceability, Completeness, and Consistency) are fulfilled. The quality factor "Reliability" has three quality criteria (Consistency, Accuracy, and Error Tolerance), of which only the first one is met. Efficiency has two criteria that are met: Execution efficiency and Storage efficiency. In Quality factor Integrity, both criterias, Access control and Audit Control, are met. In Usability, the two criterias, Operability and Trainning Communicativeness, are met. In Quality factor Maintainability, only criteria Instrumentation is not met. In testability, criteria Instrumentation again is not met. Factor Flexibility has one criterion that is not met: Expandability. In Factor Portability, the three Quality criterias are met. In Reusability, only criteria Generality is not met. Finally, Quality factor

Interoperability has two Quality criterias that are not met. Of 33 Quality criterias, only 8 are not met.

5 Methodology

Two different learning methods to support the learning activity (pattern recognition) were compared: a gamma mobile application for pattern recognition (Patrony) which classifies the exercises by degree of difficulty and contains multimedia elements such as sounds, image, and scene variations as well as game elements such as activity progress, score achieved and trophies; a traditional teaching method presenting the sequence of exercises for pattern recognition contained in the official student book provided by the Mexican Ministry of Public Education (https://www.gob.mx/sep). Both learning methods were designed with the same educational content, as well as carrying the same workflow according to the answers of the evaluation elements.

Two types of schools were considered to carry out the intervention or assessment: public and private schools. It should be noted that both types of schools are regulated and follow the same curriculum and use the same book in the topic numerical pattern recognition in the subject of mathematics.

In this assessment, the goal was to study whether the type of learning method (Patrony application versus traditional teaching method) and the type of school (private versus public schools) significantly influence the learning outcomes of the students.

5.1 Participants

In May 2019, a quasi-experimental study was carried out, where two groups were selected with a total of 43 primary school students (aged 10–12 years), from two schools; 14 from a private school and 29 from a public school in the city of Culiacán, Sinaloa, Mexico, (see Table 3). The head teachers of each group and the parents of the students received a document, which describes the purpose of the research and the right of the students to withdraw from participation at any time. The informed consent of each participant's parents was obtained. After the evaluation phase, 5 students were dropped from the evaluation analysis due to lack of data, leaving a total of 38 students.

Table 3 Data from participating schools in the study

School name	School type	Initial Number of students	Final Participants in Experiment
Primaria Simón Bolivar	Public	29	24
Instituto Los Pinos	Private	14	14
TOTAL		43	38

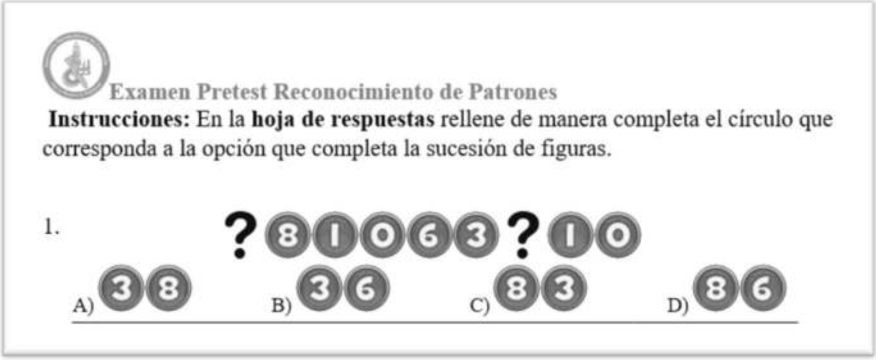
5.2 Sample equivalence

Once the pretests were carried out, the values of asymmetry (skewness) and kurtosis were obtained, with corresponding values for the public school (Sk = 0.196, K = −0.493) and corresponding values for the private school (Sk = 0.461, K = −0.369). Both criteria range from −1 to 1, so it can be established that the data are normal; this determined that parametric tests were used to evaluate the rest of the analyses.

A Student t-test of independent samples was applied to the pretest data and with it the comparison of students' prior knowledge between public and private school students. The result revealed that there were no statistically significant differences between public (M = 4.62, SD = 1.837) and private (M = 5.15, SD = 1.82) school students, with t (38) = 0.685, *p* value = .498. Therefore, public, and private school students had similar prior knowledge in the subject of numerical patterns, prior to the intervention.

5.3 Instrumentation measures

To evaluate the effectiveness of the two learning methods, a pretest and postest evaluation were completed and analyzed. Both tests consisted of 10 multiple choice questions, each correct answer with a value of one. Based on information from official textbooks, the researchers of this work designed both tests, and elementary school teachers who participated in the study assessed the validity of the content. An example (in Spanish language) of a question contained in these tests is shown in Fig. 10. The question gives the following instruction: "on the answer sheet, fill in the circle that corresponds to the option that completes the sequence of figures".



Examen Pretest Reconocimiento de Patrones

Instrucciones: En la **hoja de respuestas** rellene de manera completa el círculo que corresponda a la opción que completa la sucesión de figuras.

1. ? 8 1 0 6 3 ? 1 0

A) 3 8 B) 3 6 C) 8 3 D) 8 6

Fig. 10 Example of a pretest question

5.4 Procedure

One week before the intervention, students received two one-hour instruction sessions. During these sessions, students studied concepts and performed exercises related to the activity. The session was given by their respective teachers following the same work plan. During the last 20 min of the second session, students completed the pretest exam.

In the following week, students were randomly divided into two groups: control and experimental group. Later, each student of the experimental group received a tablet with the application (software) Patrony installed and a tutorial with general instructions to continue. This activity lasted 10 min. The control group received printed material with exercises related to pattern recognition. The learning activity (Intervention) for both groups lasted 50 min. During this time, the students received technical and procedural assistance from the research team. After completing the learning activity, students from both groups completed the postest questionnaire. The maximum time given to complete the questionnaire was 20 min. Figure 11 shows the steps with their intervention times.

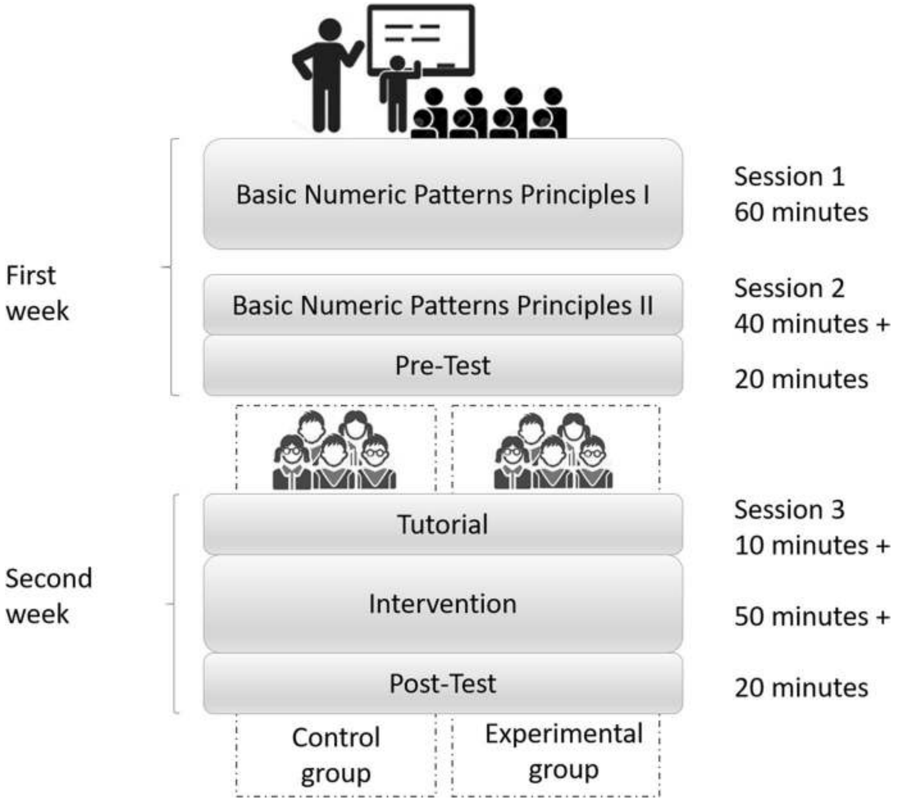


Fig. 11 Steps and times of the intervention

6 Experiments and analysis of results

A T-Student test for related samples was applied to compare the effects that the type of experiment (control, experimental), and the type of school (public, private) have on the score obtained in numerical patterns in both the pretest and the postest. As we mentioned before (subsection Sample Equivalence), the results in the pretest evaluation (T (38) = 0.685, *p* value = 0.498) indicated that there is no significant difference in the evaluation results of students in the control group and students in the experimental group in either public or private schools.

Furthermore, the results of the pretest (M = 5.15 SD = 1.808) and postest (M = 6.21, SD = 2.120), in a student t-test for paired samples, showed that there are significant differences in student performance (T = -3.246, *p* value = 0.002, SD = 1.999).

A Student t-test for paired samples was also performed comparing the control and experimental groups during the pretest and the postest. In the pretest evaluation, students in the control group scored (M = 5.35, SD = 1.69) and students in the experimental group scored (M = 4.94, SD = 1.95). The results in the postest evaluation indicated that the improvement of learning in the control group was lower (M = 5.95, SD = 1.93) than the experimental group (M = 6.50, SD = 2.33). Figure 12 clearly illustrates this.

In another Student t-test applied to the data collected from the public school, it was also shown that there was a significant difference (T (24) = −2.219, *p* value =0.037, SD = 2.116) between the control group and the experimental group in favour of the latter. In this case, the mean performance scores of students from the two groups in the pretest (M = 4.62, SD = 1.83) were lower compared to the performance scores of students from the groups in the postest (M 5.58, SD = 2.08). In the pretest the control group obtained (M = 5.07, SD = 1.93) and the experimental group

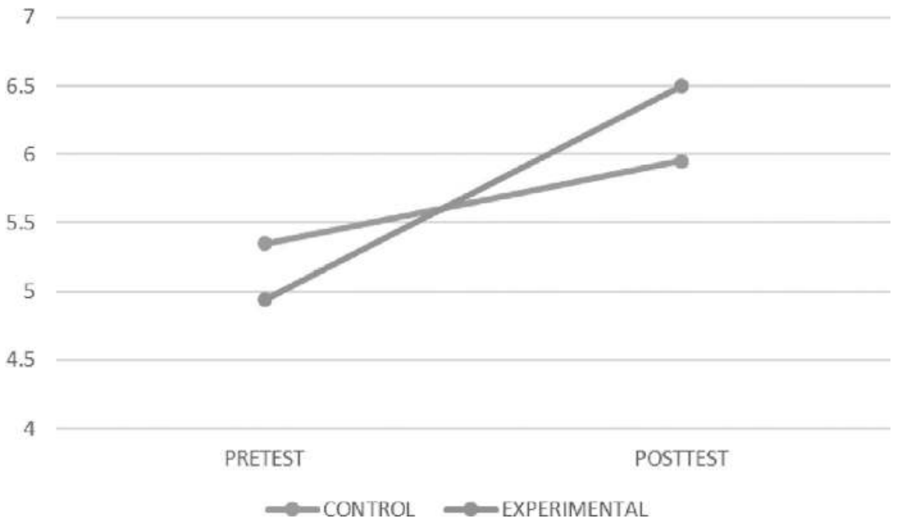


Fig. 12 Pretest and Postest results of both groups

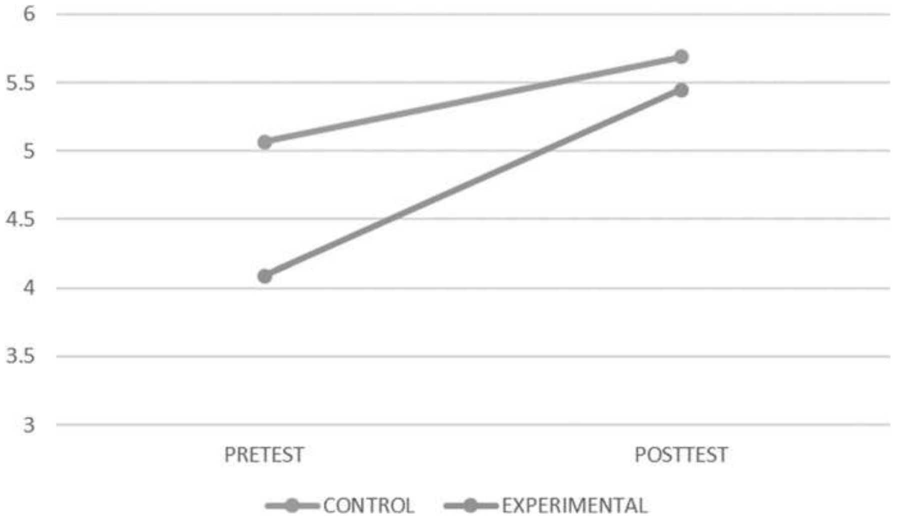


Fig. 13 Pretest and postest results in public school

obtained (M = 4.09, SD = 1.64). In the postest the control group obtained (M = 5.69, SD = 1.93) while the experimental group obtained (M = 5.45, SD = 2.33). Figure 13 shows that the tendency of the experimental group is to grow faster than the control group.

In the Student t-test applied to the data collected from the private school, the main effect of test time was also significant (T (14) = −2460, *p* value = 0,029, SD = 1847). As expected, the mean performance scores of students from both groups in the postest (M = 7.28, SD = 1.77) were higher compared to the performance scores of

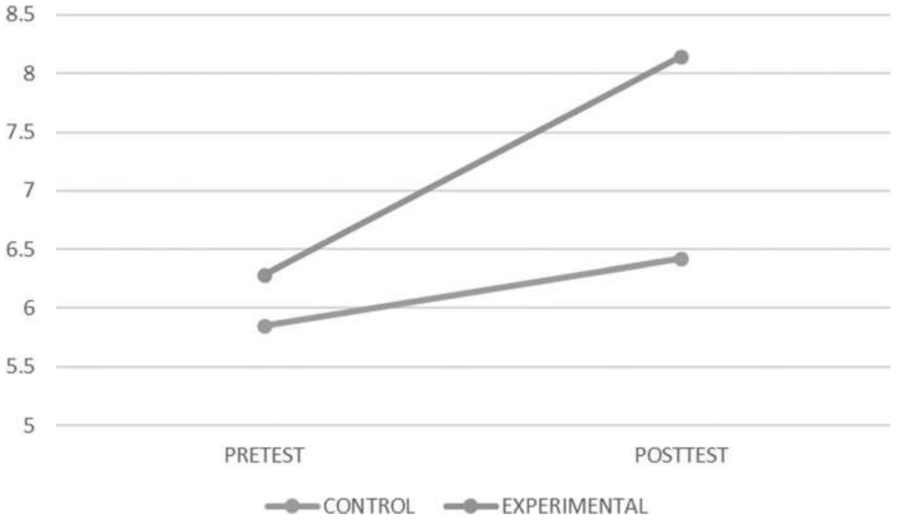


Fig. 14 Private school pretest and postest results

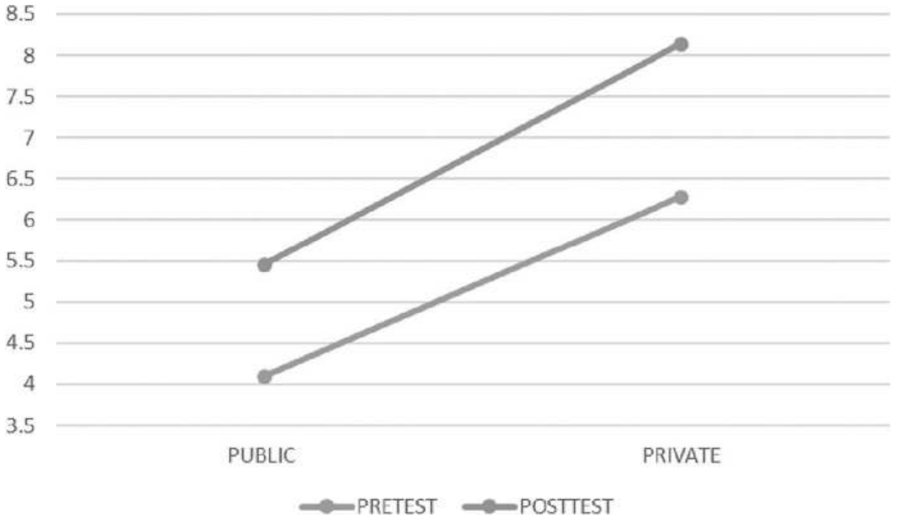


Fig. 15 Results of the experimental group between public and private schools

students in the pretest (M = 6.07, SD = 1.38). The student t-test for independent samples reflects that in the pretest the students in the control group obtained (M = 5.85, SD = 1.07) and the students in the experimental group obtained (M = 6.28, SD = 1.70). However, in the postest the results reflect that the students of the control group obtained (M = 6.42, SD = 1.99) and the experimental group obtained (M = 8.14, SD = 1.07). Figure 14 also shows that the tendency of the experimental group is to grow faster than the control group.

When comparing public schools with private schools, in the groups that used the Patrony tool (experimental groups), the results of the student t-test showed us that there is a significant difference in favour of the private school (T (18) = −3.50, *p* value =0.003, SD = 1.885). In the pretest, private school students scored (M = 6.28, SD = 1.70) and public-school students scored (M = 4.09, SD = 1.64). In the post-est, private school students scored (M = 8.14, SD = 1.07) and public-school students scored (M = 5.45, SD = 2.33). It can be established that in both schools there is a favourable trend in learning gains when using the learning tool. However, there is a small trend in favour of private schools as can be observed in Fig. 15. This trend is significant (p value =0.003) as explained above.

7 Discussion and conclusions

In this study, we investigated the effectiveness of student learning in pattern recognition. Regarding the effectiveness of pattern recognition learning with the Patrony tool compared to the traditional learning-based activity, after statistical analysis of pretest and postest scores, results of the pretest in the control group (M = 5.35 SD = 1.69) and experimental (Patrony) group (M = 4.94, SD = 1.95), and the postest in the control group (M = 5.95, SD = 1.93) and the experimental group (M = 6.50, SD = 2.33) showed that students who used the Patrony computational thinking application performed significantly better compared to those who used the traditional learning-based application. Also, in a student t-test for paired samples, we showed that there are significant differences in student performance (T (24) = −2.219, *p* value =0.037, SD = 2.116). That is, the hypothesis raised at the beginning of the research was proven to be true. Student using application Patrony have a significant learning gain in comparison to student learning with the traditional approach.

Our findings are consistent with the results of other studies in which computational thinking contributed to improved learning outcomes compared to other teaching methods (Van der Linde et al., 2018; Conde et al., 2017; Moreno-León et al., 2015). These results were similar in public and private schools. On the other hand, more studies are needed to identify which specific characteristics within pattern recognition influence student learning.

It is necessary to point out some limitations of this study. First, the study involved the practice and improvement of basic pattern recognition skills in a short time; it is likely that a long-term retention assessment will provide more information on the effectiveness of learning activities based on the use of computational thinking. Second, the study was conducted in schools in only one city, so more tests will be conducted in schools in other locations.

Based on the results presented in this study, it can be concluded that the application with a computational thinking approach was more effective. The learning gain in students of the control group in the pretest and posttest was from 5.35 to 5.95. The learning gain in students using Patrony was from 4.94 to 6.50. Then, the learning gain in the experimental group was higher (0.6 versus 1.56) with a significant difference (p value =0.037). The results also suggest that computational thinking applications can be used as effective learning tools within Mathematics courses in public and private schools in Mexico.

Future work will evaluate whether the app is attractive and effective to children in a different age range. The game was initially designed to implement exercises like those presented in the official 4th and 5th grade math book in Mexico. However, in a second version, other graphic elements (images) were added to create the sequences and make the game more attractive. On the other hand, we plan to carry out a long-term retention evaluation of the effectiveness of Patrony once the Covid-19 pandemic ends and the students return to in-person classes.

We also plan to expand the sample by conducting more tests in schools in other locations. Finally, a motivation study will be included to compare the degree of motivation found in students who used Patrony against students who used the traditional method of teaching in basic education schools in Mexico.

Acknowledgements We appreciate the support provided by teachers and principals of Escuela Primaria Simón Bolivar and Instituto Los Pinos from Culiacán, México. The work described in this paper was fully supported by a scholarship from CONACYT (Consejo Nacional de Ciencia y Tecnología) in México.

Declarations

Research involving human participants and/or animals Approval was obtained from the ethics committee of the Instituto Tecnológico de Culiacán. The procedures used in this study adhere to the tenets of the Declaration of Helsinki.

Informed consent Informed consent was obtained from all individual participants included in the study.

Disclosure of potential conflicts of interest The authors have no relevant financial or non-financial interests to disclose. The authors have no conflicts of interest to declare that are relevant to the content of this article. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interests in any material discussed in this article.

References

Anonymous (2020). Details omitted for double-blind reviewing.

Calderon, A. C., Crick, T., & Tryfona, C. (2015). Developing computational thinking through pattern recognition in early years education. In Proceeding of the 2015 British HCI Conference on - British HCI '15, 259–260.

Code.org. (2019). Resource document. https://code.org/international/about.

Conde, M. Á., Fernández-Llamas, C., Rodríguez-Sedano, F. J., Guerrero-Higueras, Á. M., Matellán-Olivera, V., & García-Peñalvo, F. J. (2017). Promoting Computational Thinking in K-12 students by applying unplugged methods and robotics. In Proceeding of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality (p. 7). ACM.

Crompton, H., Burke, D., & Gregory, K. H. (2017). The use of mobile learning in PK-12 education: A systematic review. *Computers & Education, 110*, 51–63.

CSTA. (2011). Operational Definition of Computational Thinking for K-12 Education. https://cdn.iste.org/www-root/ct-documents/computational-thinking-operational-definition-flyer.pdf.

Disney. (2019). Hour of Code. https://partners.disney.com/hour-of-code.

Duolingo. (2019). Duolingo for Schools. https://schools.duolingo.com/.

Eysenck, M. W., & Keane, M. T. (2005). *Cognitive psychology: A student's handbook*. Taylor & Francis.

Fu, K. S. (1982). *Application on pattern recognition*. CRC Press Inc..

GmbH, L. (2019). Our Apps. https://lotum.com/en/our-apps/.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43.

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296–310.

INEGI. (2018). Encuesta Nacional sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares (ENDUTIH) 2018. Available online at https://www.inegi.org.mx/programas/dutih/2018/.

Jalote, P. (2008). *A concise introduction to software engineering*. Springer Science & Business Media.

Kidd, J. K., Curby, T. W., Boyer, C., Gadzichowski, K., Gallington, D. A., Machado, J. A., & Pasnak, R. (2012). Benefits of an intervention focused on oddity and seriation. *Early Education and Development, 23*(6), 900–918.

McCall, J., Richards, P. and Walters, G. (1977). Factors in Software Quality. NTIS AD-A049–014, 015, 055. Available online at https://apps.dtic.mil/dtic/tr/fulltext/u2/a049015.pdf.

MIT Media Lab. (2019). Scratch. https://scratch.mit.edu.

Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia,* (46), 1–23.

Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Education Tech Research Dev, 68*, 463–484. https://doi.org/10.1007/s11423-019-09708-w.

Pi Y., Liao W., Liu M., & Lu J. (2008). Theory of cognitive pattern recognition. In *Pattern Recognition Techniques, Technology and Applications*, editors Peng-Yeng Yin, ISBN 978–953-7619-24-4, pp. 626, November 2008, I-Tech, Vienna, Austria.

Roscoe, J. F., Fearn, S., & Posey, E. (2014). Teaching Computational Thinking by Playing Games and Building Robots. 2014 International Conference on Interactive Technologies and Games, 9–12. https://doi.org/10.1109/iTAG.2014.15.

Secretaría de Educación Pública (2017) *Desafíos matemáticos.* Libro para el alumno. Quinto grado. Cuarta reimpresión, 2017 (ciclo escolar 2018–2019).

Shugen, W. (2002). Framework of pattern recognition model based on the cognitive psychology. *Geospatial Information Science, 5*(2), 74–78.

Touw, K. W. J., Vogelaar, B., Bakker, M., et al. (2019). Using electronic technology in the dynamic testing of young primary school children: Predicting school achievement. *Education Tech Research Dev, 67*, 443–465. https://doi.org/10.1007/s11423-019-09655-6.

van der Linde, D., van der Aar, N. & Voogt, J. (2018). Best of The Netherlands: How children use computational thinking skills when they solve a problem using the Ozobot. In T. Bastiaens, J. Van Braak, M. Brown, L. Cantoni, M. Castro, R. Christensen, G. Davidson-Shivers, K. DePryck, M. Ebner, M. Fominykh, C. Fulford, S. Hatzipanagos, G. Knezek, K. Kreijns, G. Marks, E. Sointu, E. Korsgaard Sorensen, J. Viteli, J. Voogt, P. Weber, E. Weippl & O. Zawacki-Richter (Eds.), Proceedings of EdMedia: World Conference on Educational Media and Technology (pp. 2151–2157). Amsterdam, Netherlands: Association for the Advancement of Computing in Education (AACE). Retrieved May 3, 2021 from https://www.learntechlib.org/primary/p/184224/.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology., 25*, 127–147.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology, 25*(2), 7–14. https://doi.org/10.17471/2499-4324/922.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education, 14*(1), 1–16.

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends, 60*(6).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

María Lucia Barrón-Estrada[1] · Ramón Zatarain-Cabada[1] · Jorge Abraham Romero-Polo[1] · Julieta Noguez Monroy[2]

María Lucia Barrón-Estrada
lucia.be@culiacan.tecnm.mx

Jorge Abraham Romero-Polo
jorge_romero@itculiacan.edu.mx

Julieta Noguez Monroy
jnoguez@tec.mx

[1] Tecnológico Nacional de México, Instituto Tecnlógico de Culiacán, Juan de Dios Bátiz 310 Pte.,col. Guadalupe, C.P. 80220, Culiacán, Sinaloa, Mexico

[2] Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), Calle Puente 222, Coapa, Arboledas del Sur, Tlalpan, 14380 Ciudad de México, CDMX, Mexico