A Case Study on Learning visual programing with TutoApp for Composition of Tutorials: An approach for Learning by Teaching

Maximiliano Paredes-Velasco, Isaac Lozano-Osorio, Diana Pérez-Marín and Liliana Patricia Santacruz-Valencia

***Abstract*—Teaching programming is a topic that has generated a high level of interest among researchers in recent decades. In particular, multiple approaches to teaching visual programming have been explored, from the use of tools such as Scratch, robots, unplugged programming or activities for the development of computational thinking. Despite the wide range of resources used, students generally tend to perform poorly academically and perceive learning visual programming as a complex and demotivating task.**

In this article, the TutoApp system is proposed together with a new methodology based on "Learning by Teaching", where students create tutorials in their mobile devices to explain programming concepts to their peers. The hypothesis of this paper is that the use of this tool improves learning outcomes and the level of student satisfaction. An experiment with a pre-post-test design has been carried out with 57 university students in an introductory programming course, 30 belonging to a control group (did not use TutoApp) and 27 belonging to the experimental group (used TutoApp). The findings indicate that the creation of tutorials with TutoApp significantly improved students' academic performance over those who did not use it, specifically in learning the loops and conditional control structures. However, it was observed that anxiety increased in all students while learning visual programming.

The results of this study open the door to the validation of the use of systems and methodologies for creating tutorials for teaching visual programming to university students.

***Index Terms*: emotions, learning by teaching, visual programming, written composition, mobile devices.**

I. Introduction

The teaching of visual programming has become more widespread in recent decades thanks to the use of multimedia block-based programming languages such as Scratch [1] and [2], robots [3] and [4] and unplugged approaches [5] - [7] at all levels of education from early childhood [8] and [9] to university education [10] and [11]. According to existing research, the benefits of learning to program are multiple. Some benefits are the development of logical thinking [12] and "computational thinking" [13]. However, despite the multiple approaches, teaching programming remains a complex issue [9] and [14].

Numerous challenges complicate the learning of programming in university introductory courses [15]. Aspects such as heterogeneity in the level of knowledge or the discouragement and demotivation of most students, who perceive programming as a complex cognitive task, generate difficulties in learning of programming in university courses [16].

Learning programming is a complicated task. It involves understanding of theoretical concepts, the practical use of language semantics, coding, and logical reasoning for problem solving [17] and [18]. Furthermore, the use of visual tools in learning programming may distract students' attention [19], affecting the emotional state of students in aspects such as anxiety or the sense of achievement [20].

Scratch has become one of the most popular block-based programming tools in recent years. Scratch allows users to create animations, stories, and interactive games by dragging blocks of instructions. Studies indicate that this playfulness can motivate students [21]. However, according to Melguizo et al. [22], more scientific studies are needed to measure impact of its use in teaching programing.

It is also important to consider the influence of emotions in teaching in general and, in particular, in the teaching of visual programming. Pekrun et al. [23] identified emotions as a key element in learning and proposed a taxonomy model to analyze the emotions present in the academic context together with a specific self-report assessment instrument (Academic Emotions Questionnaire [AEQ]) [24]. Some authors suggest that emotions play a critical role in learning and performance in mathematics and science [21] and [23]. It has been proved that when an activity is enjoyable, more attention is paid, and more cognitive resources are used to increase learning and performance [25]. Therefore, research on the relationship between visual learning programming and emotions constitutes a subject of interest [25].

This article presents TutoApp, a new mobile app which, together with the TutoLearning methodology, proposes the creation of tutorials by Undergraduate Education Students to learn visual programming.

This paragraph of the first footnote will contain the date on which you submitted your paper for review.

The authors acknowledge support from the Government of the Region of Madrid and "Fondos Estructurales" of the European Union with grant references S2018/TCS-4307. (*Corresponding author: M. Paredes-Velasco*).

M. Paredes-Velasco is with the Universidad Rey Juan Carlos, 28933 Madrid, Spain (e-mail: maximiliano.paredes@urjc.es).

I. Lozano-Osorio is with the Universidad Rey Juan Carlos, 28933 Madrid, Spain (e-mail: isaac.lozano@urjc.es).

D. Pérez-Marín is with the Universidad Rey Juan Carlos, 28933 Madrid, Spain (e-mail: diana.perez@urjc.es).

L. P. Santacruz-Valencia is with Universidad Rey Juan Carlos, 28933 Madrid, Spain (e-mail: liliana.santacruz@urjc.es).

Once a tutorial is created by a student, it is reviewed by the teacher and shared with the rest of his/her peers, developing the "Learning by teaching" paradigm [26] - [28]. The objective of this paper is to measure the impact of this methodology on learning outcomes and students' emotions (including anxiety) by conducting an experiment in a control and experimental setting. The control group was taught programming without using TutoApp and the test group was taught programming following the TutoLearning methodology and using TutoApp. The students' learning outcomes and emotional levels were measured at the end of the course to find out the impact of learning programming through the creation of tutorials at the university level.

The article is organized into eight sections: Section II presents a review of the different topics addressed by the article; Section III focuses on the proposal for teaching visual programming with the tutorial creation system; Section IV describes the experiment carried out; Section V describes the research methodology; Section VI provides the results obtained; Section VII discusses the results and relates them to the current literature; and finally, Section VIII contains the main conclusions and lines of future work.

II. Related work

A. Visual programming

The literature [29] - [34] reports various approaches to teaching programming, involving the use of visual software, design tools, and robotics, driven by paradigms such as Problem-Based Learning (PBL) and cognitive learning. Such approaches can be characterized into six groups [29]**:**

1. Lectures and labs: the operant conditioning theory is used, whereby the student learns as a result of positive or negative reinforcement, motivated by the attitude and mood of the teacher.
2. Software visualization: abstract ideas described in source code are mapped into visual representations that make it easier for the observer to understand how a system works [35]. Within this approach, categories are distinguished, such as: Programs view, Algorithm animation, Visual programming, Demonstrations programming, and Computational visualization [36]**.**
3. Robots: through this approach, students can propose hypotheses and test them with immediate feedback. It also encourages collaboration, promotes leadership, and encourages good design, planning, and autonomous learning.
4. PBL: this approach refers to real-life problems, encouraging collaborative work, creativity, responsibility, and autonomous learning. It also promotes the development of transversal competences such as oral expression, writing and critical thinking [29].
5. Cognitive apprenticeship: it is a collaborative teaching model focused on knowledge construction. It combines various teaching strategies, in which students build up their own knowledge with the help of a specialist, until they become independent learners [29] and [32] - [34].
6. Miscellaneous approach: this is the result of a combination of the above-mentioned approaches, in order to be adapted to specific learning contexts [29].

Visual programming refers to approaches and methods that use two-dimensional graphical elements, through which students without programming skills can create, extend, and customize software applications [37]. Visual programming languages are described by constructors and programming rules that are represented visually and, according to recent studies [30] and [38], are classified as follows: *(i)* Block-based languages (e.g. Scratch, App Inventor, Puzzle, etc.), which allow the student to drag blocks from a predefined list of commands and drop them into the work area, in order to assemble them and build a program. In addition, such languages prevent syntactic errors and reduce the mental workload of the learner, allowing them to focus more on concepts than on implementation details; *(ii)* Icon-based languages (e.g., Kodu or Limnor), which use simple and complex graphical symbols and icons to represent objects or actions; *(iii)* Form-based languages (e.g., forms/3), which allow setting up a user interface by dragging and dropping visual components representing services into a form to subsequently build a program; and *(iv)* Diagram-based visual programming languages (e.g., LabView or Tersus), also known as diagrammatic or data flow languages, which connect graphical objects by means of arrows, lines, and arcs representing relations.

Regardless of the approach used for teaching programming, the potential enhancement of programming skills referred to in the related work is closely linked to emotions, which are considered to be a critical factor in relation to 21st century skills such as communication, collaboration, critical thinking and creativity, all of which are highly relevant in the field of problem solving [39]. Therefore, in Section B, the relation between teaching and emotions is addressed.

B. Teaching and emotions

The influence of emotions on the achievement of educational outcomes, at all academic levels, has been a prominent field of research in recent years [40] and [41]. These kinds of emotions are known as achievement emotions [42]. Any type of test under examination conditions can be considered as an achievement activity, while instance grades and scores represent the achievement outcomes.

Nowadays, the most used framework to study achievement emotions is Pekrun's control-value theory model [25] and [43].

Pekrun's model [23] classifies emotions into three dimensions: *(i)* object focus (related to the success and outcome of activities), *(ii)* valence (pleasant or unpleasant); and *(iii)* activation (agitation or excitement). With respect to the degree of agitation or stimulation (activation dimension) that the subject may experience, Pekrun's model classifies emotions as follows:

- Activation emotions: these are emotions that produce a high degree of agitation such as fear, anxiety, anger, etc.
- Deactivation emotions: are those that produce low

agitation such as depression, calmness, boredom, etc.

In turn, considering the valence dimension, emotions in the academic context can be classified as positive (pleasant feeling) or negative (unpleasant or uncomfortable feeling).

Therefore, the Pekrun's model proposes the following emotions taxonomy [19]:

- Positive emotions with activation: enjoyment, hope, and pride.
- Negative emotions with activation: anger, anxiety, and shame.
- Negative emotions with deactivation: hopelessness and boredom.

This taxonomy is expanded within Perkun's extended model of integration for achievement goals, achievement emotions, and academic performance (see Figure 1).

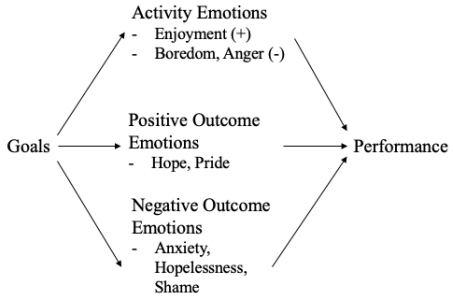


Fig. 1. Perkun's extended model of integration proposed by [88]

On the other hand, outcome emotions can be prospective and anticipatory (hope for success, anxiety for failure) or retrospective (pride or shame).

In addition, effort and intrinsic motivation based on interest is promoted, which can benefit learning and performance. In contrast, for emotions resulting from pride, the focus is on the achievement of outcomes. For example, a successful experience can divert the student's attention away from the learning process, promoting extrinsic motivation [39] (e.g., motivation to win an academic award).

In the particular case of teaching programming, a recent study [20] stresses the importance of students possessing both problem solving skills and cognitive skills (analysis, synthesis, and evaluation) in order to understand a program, and also points out that there is a relation between cognitive skills and anxiety [44] and [45], which affects student performance in learning programming.

Finally, in the context of education, it is worth noting the growing interest in exploring the potential of visual programming in various contexts [39] and [41], as well as the design of tools to support this type of programming and the study of the impact of the use of tutorials on improving learning capacity through these tools [30]. Therefore, in Section C, the implications of creating tutorials for learning programming are discussed.

C. Creation of tutorials for learning to program

The didactic approach of "Learning by Teaching" is based on the key idea that the best way to learn something new is to be able to teach it [26], [28] and [46]. Research with students who prepared their own learning material showed that their academic performance improved compared to the performance of students who did not prepare the material to be taught [47]. This improvement is also found at the emotional level [48], both in face-to-face and remote environments [49]. Some applications of this didactic approach are: assessing student-generated content [50], addressing the design of educational computer games [51], analyzing video production in content-area classrooms [52], analyzing media creation in tertiary science education [53] and, students' generation of written content by explaining the topics studied to their own peers, in the form of reports or written compositions.

Typically, written composition techniques have been used in educational contexts to develop writing competence, for example, in language learning strategies (LLSs) [54]. However, according to the relevant literature in the context of learning programming, there are no studies that have explored the possibilities of this approach. The following is a review of the most closely related papers in the current literature.

In general, writing as a pedagogical resource is rarely used in technical subjects such as engineering or science. Mon et al. [55] conducted a study of how teachers apply write-to-learn (WTL) activities in their classes in different STEM university studies. The results indicated that more than a third of the participating teachers (13/33) believe that the pedagogical resource of writing falls outside the scope of their STEM classes and do not use it in their classes or use it very little. These teachers perceived writing as only related to their knowledge and understanding, although five of them recognized that writing was a valuable tool to promote understanding. The same study found that only 27% of teachers (9/33) found writing tasks very useful in their classes, although they acknowledged that they did not apply it with sufficient time commitment in the classroom.

Some works have applied the use of written composition in different ways in learning of different computer science concepts. Vasilchenko et al [56] carried out an experience in which students with a master's degree in software and Information Technology had to interview an expert and report on the interview by creating a short video and a summary essay of the interview. Other authors have applied reflective writing as a teaching resource. Epp et al. [57] carried out an experience with a Human Computer Interaction (HCI) course where students were required to perform reflective writing practice and peer reviews, providing feedback to their classmates. These authors found that the quality of the writing did not differ between different HCI topics and that the most common feedback was to provide a straightforward solution. On the contrary, Leinonen et al. [58] applies a collaborative approach and combines interdisciplinary learning with writing tasks by pairing English as a Foreign Language students with CS students where they had to jointly write the Game Design Document (GDD) in which the former students composed the narrative and the latter designed a prototype. These authors found that interdisciplinary approaches to writing and problem solving allowed students to connect with knowledge and

develop skills in effective communication and collaborative work.

In learning programming, the application of composition techniques is even scarcer, although there is work that points to the importance of learning the syntax of programming languages with writing practice [59]. Some authors propose to apply pedagogies called "syntax-first pedagogy", where learning the syntax of a programming language is separated from problem solving [60] and focuses especially on writing. Some studies have found that the application of syntax-first pedagogy provides benefits in the learning outcomes of programming languages [60], especially in novice and inexperienced programming students [61]. Therefore, it seems that the application of writing techniques in programming learning could improve the results. However, this point is still unclear. The relation between the ability to write in the natural language and coding in the Java and Python languages was analyzed, finding differences in aspects such as writing speed or error correction, but the impact on learning outcomes was not analyzed [62]. Curl et al [63] conducted an experiment in which groups of 5 students were formed, where one of them assumed the role of programmer, developing a program in Pascal, and the other four had to review it and write a report describing different aspects of the program. Curl et al [63] stopped short of studying the impact on learning outcomes but did study students' perceptions of the writing assignment and found that 60% of students thought they could have learned more by doing the program instead of the report. On the other hand, Vasilchenko et al [56] found that the application of writing-workshop models in introductory coding with Scratch in K-12 offered certain advantages for children to practice certain programming concepts such as "parallel execution" or "event-handling".

However, the processes of coding and writing are closely related. Hassenfeld et al [64] reported the actions in one task (coding) and in the other (writing), which is summarized in Table I, and it can be seen that the construction process in both tasks follows a very similar sequence of actions.

TABLE I
RELATION BETWEEN WRITING AND CODING PROPOSED BY [56]

Coding composition	Written Composition
Planning	Prewriting
Creating	Drafting
Testing/evaluating	Evaluating
Debugging: Mechanics	Editing
Debugging: Stylistic	Revising

Therefore, the joint performance of written composition and programming tasks could have a positive impact on learning. Hassenfeld et al. [64] combined composition tasks with Scratch coding in K-8, with the goal of improving children's writing skills through programming and observed improvements in writing proficiency. However, there are no studies that have explored the inverse relation: whether performing written composition tasks improves coding learning. This establishes the relevance of researching new teaching approaches, guides and tools that promote the use of writing as an activity to facilitate the learning of computer science [57].

III. TUTOAPP AND THE TUTOLEARNING MODEL

A. *TutoApp*

TutoApp is an application made in Android Studio in the Java programming language (available on Github[1]), whose system is based on a service architecture. Figure 2 shows a comparison of the TutoApp architecture versus common architectures for application development.

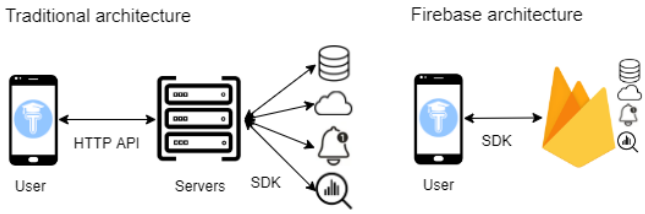


Fig. 2. Traditional vs Firebase system architecture

In both architectures, a student connects to the application server which provides different services such as databases, cloud, notifications, or analytics. TutoApp is based on the Firebase architecture. Firebase is a platform for the development of web and mobile applications from Google with native libraries. The TutoApp architecture allows, through a connection to the Firebase service, to have services without the need to implement them in an own server, which provides greater scalability and security in the applications.

The software architecture of TutoApp is based on two main blocks (see Figure 3). The first one corresponds to the Firebase API block that implements a connection to the Firebase service to use services such as student registration, storage in databases connected in real time, statistics system, notification management for both students and teachers, automatic email sending, and file/image storage. The second is the backend block, which takes over the management of the application's functionalities such as displaying, creating, and performing tutorials, as well as providing the ranking of the most voted tutorials, the individual profile, and statistical records for teachers.

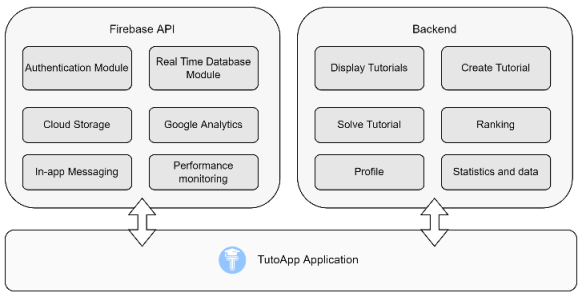


Fig. 3. TutoApp architecture

[1] https://github.com/isaaclo97/TutoApp

The structure of the TutoApp user interface is represented in Figure 4, by the navigation diagram between its main screens. The application can be used by multiple students. For this purpose, a registration, login, and password recovery system is available. If a student or teacher has a registered account, once logged into the application, the Tutorials *main screen* is displayed from which the following screens are accessed (Figure 4):

- *Create tutorial*: Allows for adding the description, images, and title of the tutorial being created. Before proceeding to the screen to create the quiz (see Figure 5.a), TutoApp has a preview feature to visualize how the tutorial will look like. Subsequently, users access the *Create quiz* screen, adding at least one multiple-choice question (between 1 and 4 possible options). Once completed, it is sent for review (*Send to review Tutorial* screen).
- *View tutorial*: From the main screen any published tutorial can be seen. When one of them is accessed, the description of the tutorial (see Figure 5.b) and its contents are displayed for reading. After finishing the reading, the next screen shows the tutorial quiz (see Figure 5.c), which will evaluate whether the student who has read the tutorial has understood it (*Take quiz* screen). Finally, the student will be taken to another screen where he/she can rate the quality of the tutorial created by his/her partner (*Rate tutorial* screen).
- *Help*: Displays information about the use of the application.
- *My profile*: In addition to basic settings such as changing *passwords*, profile picture, or student name, this screen shows all tutorials that the student has completed.
- *Ranking*: Shows the ranking of published tutorials based on student ratings.

A user, with teacher permissions, can access to the following screens:

- *Manage tutorials*: Allows deleting tutorials once the learning tasks have been completed.
- *Download reports*: This screen provides different options for downloading reports on the use of the application, such as tutorial visits, ratings, student connections, etc.
- *Administration permissions*: Manages the creation and permissions of students (teacher role).

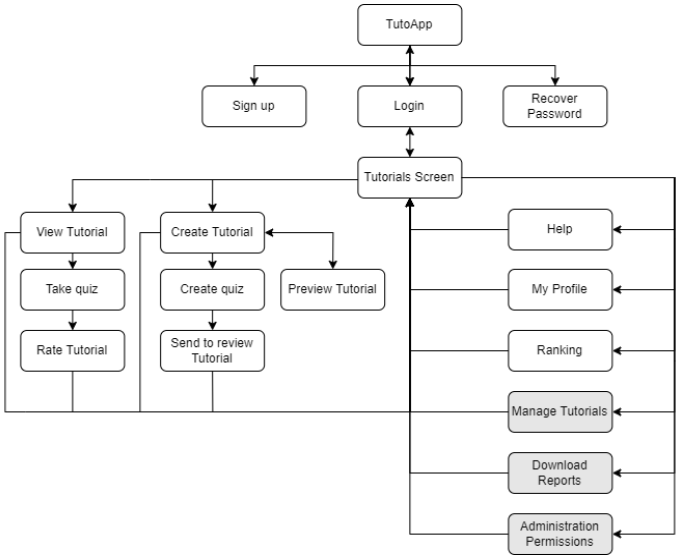
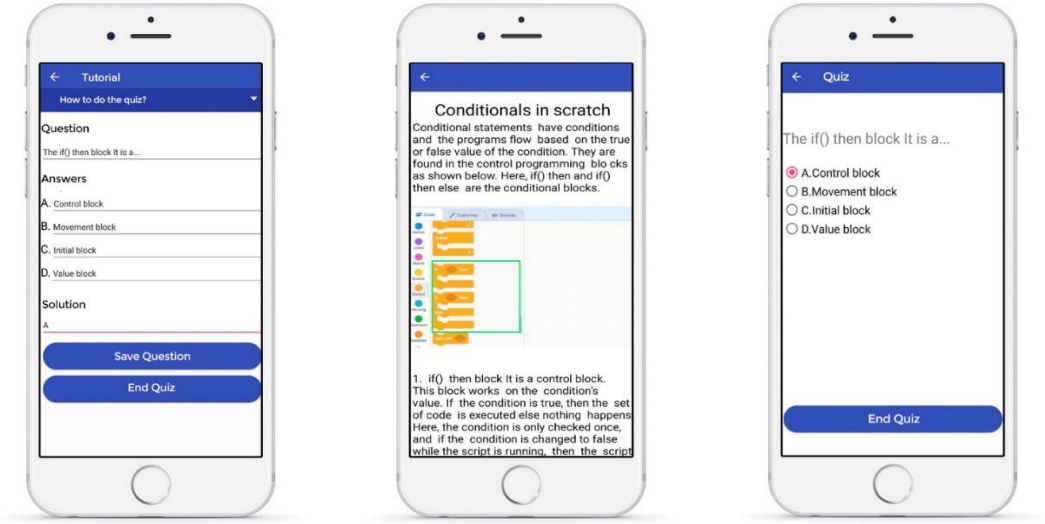


Fig. 4. TutoApp UI navigation diagram



a) Tutorial quiz creation b) Tutorial view page c) Tutorial quiz

Fig. 5. TutoApp application

B. Description of the TutoLearning educational model

TutoLearning proposes the use of tutorial writing/reading tasks as a teaching resource. The educational model is based on the didactic approach application of the "Learning by

Teaching" [26] - [28] and [46], combined with the WTL approach [47], both introduced in Section II.C.

Writing a tutorial as an explanatory document favor the learning of a competence or concept presented in class. In our proposal, the tutorials incorporate a knowledge quiz. Figure 6 illustrates in brief the general structure of TutoLearning. The implementation of this model requires the interaction between two types of roles, a teacher role (represented by a T at the beginning of each activity) and a student role (corresponding to those beginning with S).

Fig. 6. Diagram of TutoLearning

The purpose of TutoLearning is to be able to acquire and transmit a new concept. In this way, the students will create a tutorial about the concept including a quiz to evaluate future readers. This tutorial will be reviewed by the teacher, and it will be reviewed and evaluated by the readers providing a rating.

TutoLearning is based on 6 steps, 2 of them to be performed by the teacher, and 4 by the students:

1. The first step consists of conveying to the students the knowledge of the concept on which the different tutorials will be carried out. Presenting the concept can be done in multiple ways (presentations, videos, lectures, discussions, etc.). To this end, it is proposed that the teacher conduct a training session on the concept so that once the students have understood it, they are able to transmit it.
2. Once the concept has been presented, the next step in TutoLearning is to write a tutorial explaining the concept. When students are required to explain a concept, it forces them to devote time to it, which strengthens their understanding of the concept.
3. When the student has completed the tutorial, the teacher reviews it, observing if the student has really acquired the knowledge of the concept. In case an error is found, the teacher gives feedback to the student, helping him/her to understand the concept.
4. Student will correct the errors pointed out by the teacher, and then the tutorial will be published to make it available to the rest of the classmates.
5. To complete the experience, students are asked to review and complete the tutorial quiz of their peers.
6. Finally, students rate the quality of the tutorial.

The TutoLearning model adapts the Learning by Teaching model as follows:

- The first step in both cases is the explanation of the concept by the teacher, the expert, or a source of knowledge.
- The second step of the Learning by Teaching model is the explanation of the concept among peer students covering steps 4 and 5 of the TutoLearning model.
- The third step of the Learning by Teaching model is the supervision of the students' teaching of the concept covering steps 3 and 6 of the TutoLearning Model.

TutoLearning also adapts the compositional process proposed by Hassenfeld and Bers [56] related to the WTL approach, which determines the relation between coding and written composition. Focusing on written composition of the model (see Table I, "Written composition" column), TutoLearning applies Prewriting and Drafting in the step 2, where student must draft a tutorial. Subsequently, Evaluating is developed in the step 3 of TutoLearning since the teacher reviews the draft created by the student. Finally, Editing and Revising are carried out by the student in step 4 of TutoLearning.

IV. Experiment

A. Objectives

The main objective of the experiment is to investigate whether the use of a new methodology and tutorial system to teach visual programming to university students can improve their academic performance as well as to investigate the impact on the students' emotions according to the Pekrun's model [19] described in Section II.B, quantified using the AEQ questionnaire described in Section IV.D. To meet this objective, three quantitative research questions are formulated from which several hypotheses to be validated are derived:

RQ1: Does the pedagogical method of creating tutorials using TutoApp for learning visual programming improve learning outcomes?

H1. The method of creating tutorials with TutoApp improves the outcome of learning visual programming concepts compared to the traditional method.

RQ2: Does the pedagogical method of creating tutorials using TutoApp have a positive impact on student emotions?

H2: Student's emotions improve when the teaching method of creating tutorials with TutoApp is applied compared to the traditional method.

RQ3: Does the student profile (age and gender) influence learning outcomes and emotions?

H3: The student's personal profile (age and gender) influences his/her visual programming learning outcomes and emotions.

B. Sample and context

The experiment took place during the 2020/2021 academic year with 57 students of the Computer Science and Digital Competence course in Early Childhood Education and Primary Education Degrees at a university in Madrid (Spain). This course is an introduction to visual programming with

Scratch with the aim of training future computer science teachers in secondary education. The students were between 18 and 20 years old (first year) and the percentage of women was 56% and men 44%. None of the students had ever used a tutorial system for university learning in any domain before; nor had they previously studied visual programming.

No student received financial compensation or a score increase to motivate them to participate in the experiment. Participation was voluntary and anonymous, although students were asked for a code to identify them in the pre- and post-test.

C. Experimental material

The teachers of the course had the entire syllabus and exercises to teach visual programming structured in the classic order of starting with basic programming concepts such as sequence (instructions sequential execution), memory (definition and use of variables), and input/output (read data from keyboard and print through screen), and then delving into conditionals and loops (execution flow control) [65].

Students had access to the University's platform where they could download the worksheets and the topics in document format for review. The exercises did not have the solution published until the date of submission, to prevent students from seeing the solution before solving them themselves.

D. Instruments

Since the objective is to measure whether students improved their academic performance and how their emotions were affected, the following two instruments were used:

1. A digital questionnaire to measure the level of knowledge at the beginning of the experiment (pre-test) and at the end of the experiment (post-test) of the programming concepts taught in the subject according to the experimental material described in section IV.C.
2. The validated AEQ questionnaire [66] to measure achievement emotions and investigate to what extent the methodology and use of the tutorial system have affected their emotions.

Both questionnaires also asked about the student's profile in order to investigate whether there were differences in relation to age and gender in both learning outcomes and emotions.

The questionnaire to measure the level of knowledge consists of three questions, one for each key concept of the syllabus:

1. Create a program with Scratch instructions to say “Hello” (*input/output concept*).
2. Create a program for the computer to indicate the greater of two numbers requested from the student (*concept of conditional and memory*).
3. Create a program to make the computer count from one to five (*loop and memory concept*).

Each question was valued with a maximum of 1 point if the exercise was solved correctly. If no answer was given or the solution provided was totally incorrect, then it was 0 points. The heading also contemplated the following cases:

- Score of 0.25 if any block was correct.
- Score of 0.50 if all blocks were correct but the structure was incorrect.
- Score of 0.75 if all the blocks used were correct and, although the structure was correct, there were bugs that prevented the program from executing. For example, syntax errors.

The AEQ questionnaire was used to measure students’ emotions because it is a validated instrument [60] and has been previously used in educational context [67] and [68]. It should be noted that the AEQ questionnaire allows the measurement of emotions in the academic setting according to Pekrun's model [23]. Therefore, the AEQ instrument allows measuring and classifying emotions as follows: *(i)* Positive emotions with activation (enjoyment, hope, and pride); *(ii)* Negative emotions with activation (anger, anxiety, and shame); and *(iii)* Negative emotions with deactivation (hopelessness and boredom).

It is important to note that the AEQ questionnaire consists of three sections: before the study, during the study, and after the study. Each section includes a number of items that express emotions, and the student is asked to rate the degree to which he/she agrees on a scale ranging from 1 (strongly disagree) to 5 (strongly agree). The AEQ questionnaire was administered in two phases, i.e., before and after the intervention, using two sections of the instrument: "before studying" (consisting of items 81 to 95 on the scale) and "after studying" (consisting of items 120 to 155) [66].

V. METHODOLOGY

Figure 7 illustrates the pre-post-test procedure of 4 weeks that was followed during the experiment for the control group and Figure 8 illustrates the procedure followed for the test group.

Initially, all students were asked to complete the knowledge pre-test during the last week of April and the “before studying” questions on the AEQ questionnaire. Each student was asked to create a unique code in order to be able to associate these pre-test results with the post-test without the need to know their personal data and to be able to proceed anonymously. The questionnaire was completed digitally during the class.

Subsequently, as it was not possible to randomly distribute the students, since the classes were held according to the on-site and distance learning shifts imposed by the chancellor office of the University, they were divided into groups respecting the shifts, but randomly between shifts: one shift was assigned to the control group and another shift to the experimental group. Both groups worked the same amount of time, with the methodology being the main difference.

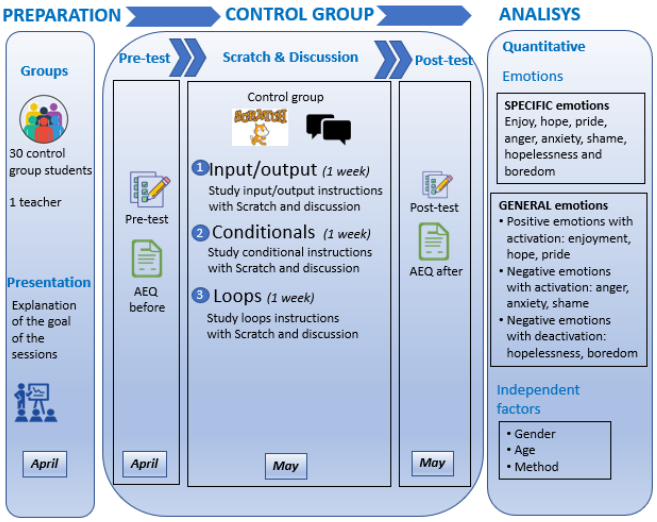


Fig. 7. Diagram of the experiment procedure for the control group

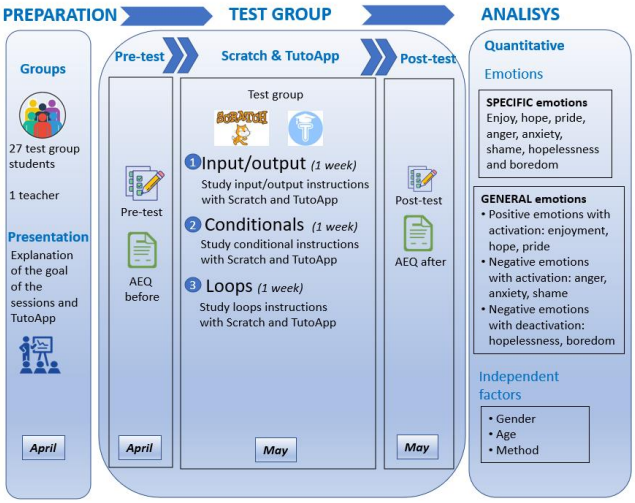


Fig. 8. Diagram of the experiment procedure for the test group

The methodology used in the control group was the traditional lecture to teach the concepts with Scratch exercises to practice and a discussion. The methodology applied in the experimental group was the TutoLearning approach, based on the use of tutorials with TutoApp and Scratch.

The first week of May, the study of the programming concepts began for both groups. Each session lasted 2 hours and was face-to-face. Students in the control and test groups listened to their teacher explaining the programming concept for about half an hour. Subsequently, the control group discussed in small groups (2-3 students) what the teacher had explained, and the test group created the tutorial using TutoApp for about an hour. Finally, all students had to complete a set of exercises using Scratch for about half an hour.

Each week had 2 face-to-face sessions of 2 hours each. The first week was devoted to the concept of input/output, the second to the conditional concept, and the third was dedicated to the concept of loops.

The last week of May, all students were again asked to complete the post-test digitally and were also asked to complete the "after studying" questions of the AEQ questionnaire.

VI. Results

In order to validate the hypotheses raised, a descriptive and inferential analysis of the learning and emotion data collected was carried out, together with a correlation study. Hypothesis tests were performed with a confidence level of 95%.

A. Variables

Based on the measuring instruments used, and in order to analyze the data collected with them, a set of dependent variables have been defined (which refers to the outcomes or effects that the researchers are interested in). For most of the emotions collected in the AEQ questionnaire, two types of variables have been defined, one to measure the emotion before starting the experience ("before studying" section of the AEQ) and the other to measure it after the experience ("after studying" section of the AEQ). These variables have been designated with the first letter of the related emotion, by adding one of these two possible suffixes as an indicator of the moment at which it is measured: "_B" (Before) and "_A" (After). For emotions measured in a single moment, hope, pride, and boredom, a single name has been defined while maintaining the corresponding suffix of the moment at which it is measured. Considering the classification of emotions from the point of view of neuroeducation made by Pekrun's model [23], six more variables have been defined with their respective suffixes "_B" and "_A": EMO_POS_B/_A, EMO-A_NEG_B/_A and EMO-D_NEG_B/_A. These variables contain the averages of different emotions grouped by the nature of the feeling and agitation experienced by the student: positive emotions, negative emotions of activation and negative emotions of deactivation (see Table II). Finally, to measure learning outcomes, several variables have been defined to measure the level of knowledge globally and specifically in different concepts of visual programming with Scratch: input/output operations, conditional control flow, and loop control flow (see Table II). These variables begin with the prefix KN (abbreviation of Knowledge).

On the other hand, we consider as independent variables factors that may affect the student's emotional state and knowledge acquisition. Obviously, the list of these factors can be very long, so we have focused on the main factors: personal profile and teaching methodology used, considering therefore the student's gender, age, and the use of the methodology of creating tutorials with TutoApp or the traditional methodology used in class for teaching visual programming. Consequently, three independent variables have been defined: GENDER, AGE, and METHOD.

TABLE II
DEPENDENT VARIABLES

Emotions	Variables Before	Variables After
Enjoyment	ENJOY_B	ENJOY_A
Hope	HOPE_B	–
Pride	–	PRIDE_A
Anger	ANGER_B	ANGER_A
Anxiety	ANXIETY_B	ANXIETY_A
Shame	SHAME_B	SHAME_A
Hopelessness	HOPEL_B	HOPEL_A
Boredom	–	BOREDOM_A
Positive	EMO_POS_B	EMO_POS_A
Negative of activation	EMO-A_NEG_B	EMO-A_NEG_A
Negative of deactivation	EMO-D_NEG_B	EMO-D_NEG_A
Knowledge		
I/O operations	KN_IO_B	KN_IO_A
Conditional blocks	KN_CONDI_B	KN_CONDI_A
Loops	KN_LOOP_B	KN_LOPP_A
Global knowledge	KN_B	KN_A

B. Learning Outcomes

The study of knowledge acquired during the experience is presented here. First, the normal distribution of the knowledge variables was studied using the *Shapiro-Wilk* test [68], concluding that none of them followed a normal distribution. Second, a hypothesis test of equal means of the knowledge scores of the two groups was performed. For this purpose, the *Mann-Whitney* test for independent samples was applied. These results are shown in Table III, where the descriptive statistics of the knowledge variables can be seen, and the significant differences found between the control group and the experimental group are marked in bold. Similarly, the differences between the knowledge variables were analyzed, but this time the factor variable was gender and age, applying *Kruskal-Wallis* for independent samples jointly to the two groups.

No differences were found with respect to gender, but significant differences were found in the KN_CONDI_A variable with respect to age (*p-value*=0.09<0.01). To determine which age is different, the participants' age, 18 to 30 years, was distributed into five ranges and compared pairwise by applying the *Kruskal-Wallis's* test with *Bonferroni* correction.

The results indicated that in the KN_CONDI_A variable, significant differences were found between students aged 23-24 years with respect to students aged 19-20 and 21-22 years, the mean knowledge of the former ($M_{23\text{-}24}$=0.67) being higher than that of the latter ($M_{19\text{-}20}$=0.12 and *M21-22 = 0.00*, respectively).

The same analysis of the KN_CONDI_A variable was repeated, this time differentiating between the treatment received in the experimental group and in the control group. In the experimental group no differences were found, however, in the control group significant differences were found, such that students aged 23-24 obtained significantly higher scores with respect to the rest of the students in all other age ranges ($M_{23\text{-}24}$=1.00 vs. $M_{18\text{-}22}$= $M_{25\text{-}30}$=0.00).

TABLE III
OUTCOMES LEARNING BEFORE AND AFTER EXPERIENCE

	Control group (N=30)		Experimental group (N=27)		EG vs. CG (Mann-Whitney)
Variable	Mean	SD	Mean	SD	Sig. Test
KN_IO_B	0.03	0.18	0.00	0.00	0.34
KN_IO_A	0.35	0.46	0.54	0.48	0.14
KN_CONDI_B	0.00	0.00	0.00	0.00	1.00
KN_CONDI_A	0.03	0.18	0.31	0.41	**0.00**
KN_LOOP_B	0.00	0.00	0.00	0.00	1.00
KN_LOOP_A	0.09	0.27	0.31	0.42	**0.02**
KN_B	0.01	0.06	0.00	0.00	0.34
KN_A	0.16	0.22	0.39	0.38	**0.02**

Finally, we tested whether the knowledge acquired after the experience was significant with respect to the beginning of the experience for each of the groups, applying the *Wilcoxon* test [69] for related samples. Table IV shows the results of this test, marking in bold the significant differences found and separated by treatment group.

TABLE IV
KNOWLEDGE BEFORE AND AFTER EXPERIENCE

Pair	Control Group	Experimental Group
KN_IO_B - KN_IO_A	**0.00**	**0.00**
KN_CONDI_B - KN_CONDI_A	0.32	**0.00**
KN_LOOP_B - KN_LOOP_A	0.07	**0.00**
KN_B - KN_A	**0.00**	**0.00**

C. Analysis of emotions

A hypothesis test of equality of means between the emotional variables was performed, first determining which variables followed a normal distribution by applying the *Shapiro-Wilk* test. Consequently, the parametric *t-Student* test [70] with *Levene's* test for equality of variance was applied to the variables ANGER_B, ENJOYMENT_A, EMO_POS_A, and EMO-A_NEG_B, and the nonparametric *Mann-Whitney* test [71] for independent samples to the rest of the variables. Table V shows the results of these tests, where we can see the mean, standard deviation, and *p-value* of the hypothesis tests, finding no significant differences in any of the variables.

TABLE V
DESCRIPTIVE STATISTIC OF EMOTIONS (MANN-WHITNEY AND T-STUDENT)

	Control group (N=30)		Experimental group (N=27)		EG vs. CG
Variable	Mean	SD	Mean	SD	Sig. Test
ANGER_B	2.10	0.67	2.10	0.70	0.99*
ANXIETY_B	1.99	0.76	2.05	0.77	0.64
ENJOY_B	3.50	0.99	3.85	0.66	0.13
HOPE_B	3.91	0.69	3.75	0.56	0.26
HOPEL_B	2.24	1.04	2.23	0.97	0.88
SHAME_B	2.30	1.18	2.44	1.31	0.70
EMO_POS_B	3.71	0.77	3.80	0.51	0.72
EMO-A_NEG_B	2.13	0.67	2.20	0.74	0.72*
EMO-D_NEG_B	1.93	0.71	2.06	0.84	0.64
ANGER_A	1.93	0.98	1.74	1.10	0.31
ANXIETY_A	2.87	0.90	2.96	1.06	0.70
ENJOY_A	3.51	0.68	3.65	0.82	0.48*
HOPEL_A	1.93	0.82	1.95	0.65	0.52
PRIDE_A	4.05	0.85	3.98	0.71	0.55
SHAME_A	1.94	0.77	1.99	0.95	0.89
BOREDOM_A	1.67	0.65	1.91	0.89	0.39
EMO_POS_A	3.78	0.70	3.82	0.72	0.84*
EMO-A_NEG_A	2.25	0.65	2.23	0.81	0.58
EMO-D_NEG_A	1.93	0.82	1.95	0.65	0.52

*t-Student

In addition, in order to test whether there were significant differences with respect to age and gender, an ANOVA test was applied for normally distributed samples and a *Mann-Whitney* test for non-normally distributed samples, with the AGE and GENDER variables as factor variables. The results did not indicate significant differences.

Finally, in order to analyze whether there have been variations in the students' emotions after the experience, we compared the variables of emotions measured before and after the experience grouped by the factor of the treatment of the teaching methodology received. For this purpose, the *Wilcoxon* related samples test and the *t-Student* test were applied for non-normal and normal samples, respectively. Table VI shows the *p-values* of these tests, marking in bold the significant results (variable ANXIETY_A).

TABLE VI
EMOTIONS BEFORE AND AFTER EXPERIENCE (WILCOXON AND T-STUDENT)

Pair	Control Group	Experiment Group
ANGER_A - ANGER_B	0.23	0.08
ANXIETY_A - ANXIETY_B	**0.00**	**0.00**
ENJOY_A - ENJOY_B	0.97	0.30
HOPEL_A - HOPEL_B	0.16	0.12
SHAME_A - SHAME_B	0.34	0.18
EMO_POS_A-EMO_POS_B	0.77*	0.43*
EMO-A_NEG_A – EMO-A_NEG_B	0.60*	0.19*

C. Correlation study

A correlation analysis was carried out to check if there is a relation between the knowledge acquired by the students and the emotions experienced during the experiment. For this purpose, *Spearman's* test was applied [72]. Table VII shows in bold the correlations found for global knowledge (variable KN_A) with other emotion variables after the experience differentiated by the control and experimental groups.

TABLE VII
LEARNING CORRELATIONAL ANALYSIS

KN_A	EMO_POS_A	PRIDE_A	ENJOY_A	AGE
Experimental group	-0.03	-0.05	-0.01	-0.07
Control group	**-0.49**	**-0.39**	**-0.37**	**0.39**

The relation between specific emotions may be an important factor in better understanding the impact of the teaching methodology on the student's emotional state. Therefore, a second correlation analysis was performed comparing the relation between the variables prior to the experience and how they changed after the experience. Table VIII shows the most relevant correlations found, marking them in bold. The results indicate that there is a strong correlation between enjoyment and pride after the end of the experience in the experimental group, while in the control group, this correlation is lower.

TABLE VIII
SPECIFIC EMOTIONS CORRELATIONAL ANALYSIS

ENJOYMENT_A/_B	PRIDE_A	HOPEL(_A, _B)	SHAME(_A,_B)
Experimental group	**0.73***	(**-0.43**, 0.12)	(**-0.50***, 0.20)
Control group	**0.55***	(-0.28, -0.30)	(-0.09, -0.08)

* Correlation significative at 0.01

VII. DISCUSSION

A. The TutoLearning tutorial composition method with TutoApp enhances visual programming learning.

The results indicate that after the experience, the students who followed the tutorial creation method with TutoApp obtained better scores than those who were taught with the traditional method, with a significant difference (*M*=0.39 vs. *M*=0.16, see Table III). The former acquired 143.75% more knowledge than the latter. In conclusion, the *H1* hypothesis is accepted*: "The method of creating tutorials with TutoApp improves the outcome of learning visual programming concepts compared to the traditional method".* Although both groups have learned after the experience, it should be noted that the students who created TutoApp tutorials improved significantly in all the programming concepts studied throughout the experience. However, those who were taught with the traditional method only improved significantly in one of the concepts studied (see Table IV).

This improvement is confirmed by comparing the level of knowledge acquired by the two groups at the end of the experience for each of the concepts evaluated. As can be seen, the difference in the knowledge of conditional structures and loops of students who used tutorial creation with TutoApp was significantly higher than those who did not use it (Table III).

Ultimately, an interesting finding of this research is that the creation of tutorials with TutoApp significantly improves the understanding of control structures in block-based programming languages.

Some works point out benefits of using WTL tasks to help in the understanding of STEM concepts [55], such as in mathematics [73], or in biology where it is appreciated that students reach a better conceptual and structural understanding

of the concepts learned [74]. However, as far as the authors are aware, there are no studies in the field of learning programming that have contrasted these benefits. Curl et al. [63] applied writing tasks with CS1 students, and the quality of the reports written by the students has been slightly reviewed, but the quality of the programs and the impact on learning programming have not been analyzed. Edwards et al. [62], analyzing the impact of this type of task on the development of coding skills, such as the speed of coding in CS1, has been measured without analyzing the learning efficiency. Other works have developed experiences with high school and K8 level students by combining written composition tasks with Scratch programming tasks (the same language used in our experience) [64] and Google's CS First [54]. However, they have only explored the improvements that were achieved in writing proficiency with the use of coding tasks. This is an objective approach contrary to our experience, without analyzing the possible benefits in learning coding.

In summary, the authors consider that the evidence found at this point of the study presented constitutes a very relevant contribution to the scientific community since there are no previous studies and it is a novel finding.

B. Student emotions do not vary with the use of the teaching method of creating tutorials with TutoApp.

In relation to the emotions experienced by the students, the results show no significant differences between the students who created tutorials with TutoApp and those who followed the traditional methodology. It should be remembered that specific variables have been defined to measure certain emotions such as boredom or enjoyment, positive and negative, and activating and deactivating emotions, with a total of 19 emotional variables. However, no significant differences were found in any of them between the two groups of students. Therefore, Hypothesis H2 is rejected: *Student's emotions improve when the teaching method of creating tutorials with TutoApp is applied compared to the traditional method.*

However, although no significant differences in the behavior of emotions between the two groups, some interesting results related to emotions were found. Anxiety has increased in both groups of students, regardless of the methodology applied. The students who followed the teaching method of creating tutorials with TutoApp started with a level of anxiety of M=2.05 and at the end of the experience had a level of M=2.96, while the students who followed the traditional method had values of M=1.99 and M=2.87, showing significant differences for both groups (Table VI).

The authors cannot explain the reason for this increase in anxiety. Since it has occurred in both groups with the same intensity (there are no significant differences either at the beginning or at the end, see Table VI), it could be thought that it is not due to the teaching method used or to the use of TutoApp. The authors believe that it could be related to the difficulty of the tasks as perceived by the students and by the intrinsic difficulty of learning programming concepts in novice students. Paredes-Velasco et al. [75], identify that the feeling of anxiety during the algorithm learning task is related to the higher levels of Bloom's taxonomy [76]. This could explain the increase in anxiety in the participants of our experience, since the tasks performed, consisting of the development and creation of programs, constitute tasks typical of the higher levels of Bloom's taxonomy. This perception of difficulty may also be increased by the writing task itself, as students value certain difficulties in writing reports and papers related to the code developed by them [63]. Finally, another factor that can influence the increase in anxiety is the interaction with mobile devices experienced by students using TutoApp [77]. This increase in anxiety found in our study is consistent with the work described by Myyry et al. [37], where both students who worked with hybrid block programming with the Blockly tool, as well as those who programmed with the textual language Python, experienced a growth in anxiety during the creation of the programs. However, there are other works that claim to have found a decrease in student anxiety in introductory programming courses with the use of Scratch, both in university [36] and secondary education [19]. Therefore, the impact of anxiety in the context of learning visual programming is still unclear and further research is needed.

It has also been found that the sense of enjoyment during the experience is correlated with the pride perceived at the end of the activity. Although this correlation occurs in both groups, it should be noted that it is a strong correlation in students who created tutorials with TutoApp (0.73, see Table VIII) while in those who followed the traditional methodology, it is a medium correlation (0.55, see Table VIII). These results are in line with and extend the work of [19], which indicates that students experienced playfulness and enjoyment when using block-based programming environments such as Scratch. Finally, a correlation has been found between knowledge and positive emotions. While in students who created tutorials with TutoApp no correlation between these two factors has been found, in students who followed the traditional classroom method, the knowledge acquired was inversely related to some positive emotions (pride and enjoyment, see Table VII). Currently, there is a debate about the effect of negative and positive emotions on learning [78]. The results of our study seem to indicate that positive emotions do not have a positive correlation on learning outcomes, in contrast to works such as [79], in which positive correlations have been found between these two variables in learning programming [80]. However, it should be pointed out that the correlations found in our work are weak and should therefore be treated with caution, as they are not decisive and additional studies are needed to investigate this aspect in greater depth.

C. Student age influences learning visual programming.

The age and gender of the students is a factor that could influence emotions and learning outcomes. In relation to emotional state, the results of our study indicate that there are no significant differences between students of different

genders or ages in either of the two participating groups. In relation to the acquisition of knowledge in learning visual programming overall, no significant differences were found with respect to age or gender. Therefore, hypothesis H3 must be rejected: *The student's personal profile (age and gender) influences his/her visual programming learning outcomes and emotions.* This statement is in contradiction with previous studies indicating that engineering students often organize group work according to gender stereotypes [81] and that the distribution of task types in programming (writing code, design, writing reports and documentation, etc.) is gender biased among students [82]. The discrepancy of these previous works with respect to our results may be due to the fact that the former has focused on measuring the impact of participants' gender on details related to task organization and development, while our research measures the impact on more concrete aspects such as emotions and learning outcome.

However, in our research, by analyzing in detail the learning outcome, it has been found that older students (23 and 24 years old) have learned more in some visual programming concepts than younger ones (18-22). Specifically, in the concepts of conditional control structures (see Section V.A), even a positive, albeit weak, correlation between age and overall knowledge acquired after the experience was observed in one of the groups of students (see Table VII), particularly the group that followed a traditional methodology. This is in line with related research in this area, which has found that older students obtained higher average knowledge than younger ones [83] and interacted more positively [84]. Another research found that students in higher level of studies show greater academic stress than those in the first years of study [85]. This could explain the improved learning outcomes in the older students in our experience, since, as confirmed above, they experienced an increase in anxiety during the performance of the tasks, coping better with the pressure resulting from this situation.

D. Limitations of the study

In relation to the validity of the design of this study, the following issues have been considered: the treatment and control groups were formed with different individuals; the level of knowledge on the subject, as well as the emotional state did not show significant differences between the two groups at the beginning of the experience (see Tables III and VI); and the contents and tasks were the same, varying only the methodology applied and the use of TutoApp.

However, we can identify some risks or limitations in the experiment. First, the groups were not formed randomly, but rather by students enrolled in different shifts, so there is no certainty that the sample is representative of the population.

Second, the use of mobile phones in the experimental group, the use of TutoApp, could represent certain difficulties that influenced the student's enthusiasm or disenchantment.

Finally, in relation to the statistical study, a descriptive analysis was performed and then an inferential study in which errors of type 1 (incorrect rejection of a null hypothesis) and type 2 (failure to reject a false null hypothesis) could occur [86].

VIII. CONCLUSIONS

This article presents a new application for mobile devices for learning visual programming called TutoApp which, together with the TutoLearning methodology, proposes the creation of tutorials by students as a learning by teaching method. Through TutoApp, students create, with their own smartphones, short documents explaining to their classmates the programming concepts seen in class. In this way, students are involved in learning, not only to understand the concepts but also to explain them to their peers.

To validate the impact of the proposal on learning outcomes and students' emotions during learning, a pre-post control-test group design experience was conducted with a course of university students of introduction to programming with Scratch. In this experience, a group of students used TutoApp and Scratch with the TutoLearning methodology consisting in creating tutorials, while another group only used Scratch with a more traditional methodology. A specific scale for measuring the acquisition of knowledge about basic programming concepts (input/output, loops, and conditionals) was used, as well as the validated AEQ scale for measuring emotions.

The results show that the students who used TutoApp for the creation of tutorials understood visual programming concepts better than those who did not. This improvement was generalized across all concepts, but specifically, a significant improvement was found in the learning of loops and conditional control structures. In addition, it was found that students' anxiety increased during the experience, regardless of the teaching method and the use or not of TutoApp, and that the feeling of enjoyment during task completion is correlated with the feeling of pride at the end of the task.

As future work, the authors propose to conduct new studies investigating the reasons why anxiety grows in programming tasks, analyzing its relation with students' perceived difficulty in writing reports and papers related to the programs they develop themselves [63] and [87], with the intrinsic difficulty of learning programming concepts in novice students.

REFERENCES

[1] M. Resnick et al., “Scratch: programming for all” *Commun. ACM*, vol. 52, no. 11, pp. 60–67, 2009, doi: 10.1145/1592761.1592779.

[2] J. Maloney, M. Resnick, N. Rusk, B. Silverman and E. Eastmond, “The scratch programming language and environment” *ACM Transactions on Computing Education (TOCE), 2010*, vol. 10, pp. 1–15, doi: 10.1145/1868358.1868363.

[3] N. C. Zygouris, A. Striftou, A. N. Dadaliaris, G. I. Stamoulis, A. C. Xenakis, and D. Vavougios, “The use of LEGO mindstorms in elementary schools” in *Proc IEEE Glob. Eng. Educ. Conf.*, Athens, Greece, 2017, pp. 514–516. doi: 10.1109/educon.2017.7942895.

[4] L. Major, T. Kyriacou and O. P. Brereton, “Systematic literature review: teaching novices programming using robots” *IET software, 2012*, vol. 6, pp. 502–513, doi: 10.1049/iet-sen.2011.0125.

[5] C. Brackmann, D. Barone, A. Casali, R. Boucinha, and S. Muñoz-

Hernandez, "Computational thinking: Panorama of the Americas" in *Proc. Inter. Symp. Comput. Educ.*, Salamanca, Spain, 2016, pp. 1–6. doi: 10.1109/siie.2016.7751839.

[6] A. Battal, G. Afacan Adanır and Y. Gülbahar, "Computer Science Unplugged: A Systematic Literature Review" *Journal of Educational Technology Systems, 2021,* vol. 50, pp. 24–47, doi: 10.1177/00472395211018801.

[7] G. Aranda and J. P. Ferguson, "Unplugged Programming: The future of teaching computational thinking?" *Pedagogika, 2018,* vol. 68, doi: 10.14712/23362189.2018.859.

[8] M. U. Bers, C. González-González, and M. B. Armas–Torres, "Coding as a playground: Promoting positive learning experiences in childhood classrooms" *Comput. Educ. 2019,* vol. 138, pp. 130–145, doi: 10.1016/j.compedu.2019.04.013.

[9] D. Pérez-Marín, R. Hijón-Neira and C. Pizarro, "Coding in early years education: which factors influence the skills of sequencing and plotting a route, and to what extent?" *International Journal of Early Years Education, 2022,* pp. 1–17, doi: 10.1080/09669760.2022.2037076.

[10] Q. Sun, J. Wu, and K. Liu, "Toward understanding Students' learning performance in an object-oriented programming course: The perspective of program quality" *IEEE Access*, vol. 8, pp. 37505–37517, 2020, doi: 10.1109/access.2020.2973470.

[11] Kanika, S. Chakraverty and P. Chakraborty, "Tools and techniques for teaching computer programming: A review" *Journal of Educational Technology Systems, 2020,* vol. 49, pp. 170–198, doi: 10.1177/0047239520926971.

[12] C. S. González-González, M. D. Guzmán-Franco, and A. Infante-Moro, "Tangible technologies for childhood education: A systematic review" *Sustainability*, vol. 11, no. 10, p. 2910, 2019, doi: 10.3390/su11102910.

[13] J. M. Wing, "Computational thinking" *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006, doi: 10.1145/1118178.1118215.

[14] J. Hromkovič, D. Komm, R. Lacher, and J. Staub, "Teaching with LOGO philosophy" *Encycl. Educ. Inf. Technol.*, 2019, doi: 10.1007/978-3-030-10576-1_76.

[15] E. Kaila, M.-J. Laakso, and E. Kurvinen, "Teaching future teachers to code — Programming and computational thinking for teacher students" in *Proc. 41st Int. Conv. Inf. Commun. Tech., Electron. Microelectron.*, Opatija, Croatia, 2018, pp. 0677–0682. doi: 10.23919/MIPRO.2018.8400127.

[16] O. Aktunc, "A teaching methodology for introductory programming courses using Alice" *Int. J. Mod. Eng. Res.*, vol. 3, no. 1, pp. 350–353, 2013.

[17] J. Moons and C. D. Backer, "The design and pilot evaluation of an interactive learning environment for introductory programming influenced by cognitive load theory and constructivism" *Comput. Educ.*, vol. 60, no. 1, pp. 368–384, Jan. 2013, doi: 10.1016/j.compedu.2012.08.009.

[18] G. M. M. Bashir and A. S. M. L. Hoque, "An effective learning and teaching model for programming languages" *J. Comput. Educ.*, vol. 3, no. 4, pp. 413–437, Jul. 2016, doi: 10.1007/s40692-016-0073-2

[19] M. M. Rahman and R. Paudel, "Preliminary experience and learning outcomes by infusing interactive and active learning to teach an introductory programming course in Python" in *Proc. Int. Conf. Front. Educ. Comput. Sci. Comput. Eng.*, Athens, Greece, 2018, pp. 51–57.

[20] F. Demir, "The effect of different usage of the educational programming language in programming education on the programming anxiety and achievement" *Educ. Inf. Technol.*, Oct. 2021, doi: 10.1007/s10639-021-10750-6.

[21] J. Thompson and G. Childers, "The impact of learning to code on elementary students' writing skills" *Comput. Educ.*, vol. 175, p. 104336, Dec. 2021, doi: 10.1016/j.compedu.2021.104336.

[22] R. Conde Melguizo, M. Vega Barbas, C. García Vázquez, and others, "Analizando el auge de Scratch para la enseñanza de la programación: revisión del conocimiento científico publicado en España" *Tarbiya: revista de investigación e innovación educativa*, 2020, doi: 10.15366/tarbiya2020.48.001.

[23] R. Pekrun, A. C. Frenzel, T. Goetz, and R. P. Perry, "The control-value theory of achievement emotions: An integrative approach to emotions in education" in *Emotion in Education*, 1st ed., Cambridge, USA: Academic Press, 2007, pp. 13–36.

[24] R. Pekrun, T. Goetz, W. Titz, and R. P. Perry, "Academic emotions in students' self-regulated learning and achievement: A program of qualitative and quantitative research" *Educ. Psychol.*, vol. 37, no. 2, pp. 91–105, 2002, doi: 10.4324/9781410608628-4.

[25] R. Pekrun, "The control-value theory of achievement emotions: Assumptions, corollaries, and implications for educational research and practice" *Educ. Psychol. Rev.*, vol. 18, no. 4, pp. 315–341, 2006, doi: 10.1007/s10648-006-9029-9.

[26] K. Leelawong and G. Biswas, "Designing learning by teaching agents: The Betty's Brain system" *Int. J. Artif. Intell. Educ.*, vol. 18, no. 3, pp. 181–208, 2008.

[27] J. Hamer, Q. Cutts, J. Jackova, A. Luxton-Reilly, R. McCartney, H. Purchase, C. Riedesel, M. Saeli, K. Sanders and J. Sheard, "Contributing student pedagogy," *ACM SIGCSE Bulletin, 2008,* vol. 40, pp. 194–212, doi: 10.1145/1473195.1473242.

[28] J. Ribosa and D. Duran, "Do students learn what they teach when generating teaching materials for others? A meta-analysis through the lens of learning by teaching," *Educational Research Review, 2022,* vol. 37, pp. 100475, doi: 10.1016/j.edurev.2022.100475.

[29] S. C. Santos, P. A. Tedesco, M. Borba, and M. Brito, "Innovative approaches in teaching programming: A systematic literature review" in *Proc. 12th Int. Conf. Compt. Support. Educ.*, virtual event 2020, pp. 205–214. doi: 10.5220/0009190502050214.

[30] M. A. Kuhail, S. Farooq, R. Hammad, and M. Bahja, "Characterizing visual programming approaches for end-user developers: A systematic review" *IEEE Access*, vol. 9, pp. 14181–14202, 2021, doi: 10.1109/access.2021.3051043.

[31] M. Noone and A. Mooney, "Visual and textual programming languages: a systematic review of the literature" *Journal of Computers in Education, 2018,* vol. 5, pp. 149–174, doi: 10.1007/s40692-018-0101-5.

[32] V. Kuleto, M. P. Ilić, N. P. Šević, M. Ranković, D. Stojaković and M. Dobrilović, "Factors Affecting the Efficiency of Teaching Process in Higher Education in the Republic of Serbia during COVID-19" *Sustainability, 2021,* vol. 13, pp. 12935, doi: 10.3390/su132312935.

[33] C. Malliarakis, M. Satratzemi and S. Xinogalos, "Optimization of server performance in the CMX educational MMORPG for Computer Programming," *Computer Science and Information Systems,* vol. 11, pp. 1537–1553, 2014, doi: 10.2298/csis131220078m.

[34] M. Meccawy, Z. Meccawy and A. Alsobhi, "Teaching and Learning in Survival Mode: Students and Faculty Perceptions of Distance Education during the COVID-19 Lockdown" *Sustainability, 2021,* vol. 13, pp. 8053, doi: 10.3390/su13148053.

[35] D. Ayrapetov and S. Graham, "Program Visualisation: Cognitive Issues and Technological Implementations," 2002.

[36] E. Costelloe, "Teaching programming the state of the art" The Center for Research in IT in Education. Dept. of Computer Science Education. Dublin: Trinity College. Retrieved from http://www. scss. tcd. ie/disciplines/information_systems/crite/crite_web/publications/sources/programmingv1. pdf, 2004.

[37] B. Jost, M. Ketterl, R. Budde, and T. Leimbach, "Graphical programming environments for educational robots: Open roberta-yet another one?" in *Proc. IEEE Int. Symp. Multimed.*, Taichung, Taiwan, 2014, pp. 381–386, doi: 10.1109/ism.2014.24.

[38] M. S. Fayed, M. Al-Qurishi, A. Alamri, M. A. Hossain, and A. A. Al-Daraiseh, "PWCT: a novel general-purpose visual programming language in support of pervasive application development" *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 3, pp. 164–177, Aug. 2020, doi: 10.1007/s42486-020-00038-y.

[39] J. Camacho-Morles, G. R. Slemp, R. Pekrun, K. Loderer, H. Hou, and L. G. Oades, "Activity achievement emotions and academic performance: A meta-analysis" *Educ. Psychol. Rev.*, vol. 33, no. 3, pp. 1051–1095, 2021, doi: 10.1007/s10648-020-09585-3.

[40] L. Myyry, T. Karaharju-Suvanto, M. Vesalainen, A.-M. Virtala, M. Raekallio, O. Salminen, K. Vuorensola and A. Nevgi, "Experienced

academics\textquotesingle emotions related to assessment" *Assessment & Evaluation in Higher Education*, *2019*, vol. 45, pp. 1–13, doi: 10.1080/02602938.2019.1601158.

[41] E. Karagiannopoulou, A. Desatnik, C. Rentzios and G. Ntritsos, "The exploration of a `model' for understanding the contribution of emotion regulation to students learning. The role of academic emotions and sense of coherence", 2022, *Current Psychology*, doi: 10.1007/s12144-022-03722-7.

[42] R. Pekrun, S. Lichtenfeld, H. W. Marsh, K. Murayama, and T. Goetz, "Achievement emotions and academic performance: Longitudinal models of reciprocal effects," *Child Development*, vol. 88, no. 5, pp. 1653–1670, 2017, doi: 10.1111/cdev.12704.

[43] R. Pekrun and R. P. Perry, "Control-value theory of achievement emotions" de *International handbook of emotions in education*, Routledge, 2014, pp. 130–151.

[44] T.-C. Hsu and G.-J. Hwang, "Interaction of visual interface and academic levels with young students' anxiety, playfulness, and enjoyment in programming for robot control" *Universal Access in the Information Society*, pp. 1–13, 2021, doi: 10.1007/s10209-021-00821-3.

[45] A. Unal and F. B. Topu, "Effects of teaching a computer programming language via hybrid interface on anxiety, cognitive load level and achievement of high school students," *Educ. Inf. Technol.*, vol. 26, no. 5, pp. 5291–5309, 2021, doi: 10.1007/s10639-021-10536-w.

[46] J. Holmes, "Designing agents to support learning by explaining," *Comput. Educ.*, vol. 48, no. 4, pp. 523–547, 2007, doi: 10.1016/j.compedu.2005.02.007.

[47] L. Fiorella and R. E. Mayer, "Role of expectations and explanations in learning by teaching," *Contemp. Educ. Psychol.*, vol. 39, no. 2, pp. 75–85, 2014, doi: 10.1016/j.cedpsych.2014.01.001.

[48] S. Park and C. Kim, "Boosting learning-by-teaching in virtual tutoring," *Comput. Educ.*, vol. 82, pp. 129–140, 2015, doi: 10.1016/j.compedu.2014.11.006.

[49] S. Y. Okita, S. Turkay, M. Kim, and Y. Murai, "Learning by teaching with virtual peers and the effects of technological design choices on learning," *Comput. Educ.*, vol. 63, pp. 176–196, 2013, doi: 10.1016/j.compedu.2012.12.005.

[50] K. Gray, C. Thompson, J. Sheard, R. Clerehan and M. Hamilton, "Students as Web 2.0 authors: Implications for assessment design and conduct" *Australasian Journal of Educational Technology, 2010*, vol. 26, doi: 10.14742/ajet.1105.

[51] K. Hava and H. Cakir, "A systematic review of literature on students as educational computer game designers" *EdMedia+ Innovate Learning, 2017*, p. 407–419, http://www.learntechlib.org/primary/p/178530/:.

[52] C. Snelson, "Video production in content-area pedagogy: a scoping study of the research literature" *Learning, Media and Technology, 2018*, vol. 43, pp. 294–306, doi: 10.1080/17439884.2018.1504788.

[53] J. Reyna and P. Meier, "Learner-Generated Digital Media (LGDM) as an Assessment Tool in Tertiary Science Education: A Review of Literature" *IAFOR Journal of Education, 2018*, vol. 6, pp. 93–109, doi: 10.22492/ije.6.3.06.

[54] M. G. McMullen, "Using language learning strategies to improve the writing skills of Saudi EFL students: Will it really work?" *Syst.*, vol. 37, no. 3, pp. 418–433, Sep. 2009, doi: 10.1016/j.system.2009.05.001.

[55] A. Moon, A. R. Gere, and G. V. Shultz, "Writing in the STEM classroom: Faculty conceptions of writing and its role in the undergraduate classroom," *Sci. Educ.*, vol. 102, no. 5, pp. 1007–1028, Jun. 2018, doi: 10.1002/sce.21454.

[56] A. Vasilchenko, A. Cajander, and M. Daniels, "Students as Prosumers: Learning from Peer-Produced Materials in a Computing Science Course" In *Proc. IEEE Front. Educ. Conf.*, Uppsala, Sweden, 2020, pp. 1-9, doi: 10.1109/fie44824.2020.9274042.

[57] C. D. Epp, G. Akcayir, and K. Phirangee, "Think twice: exploring the effect of reflective practices with peer review on reflective writing and writing quality in computer-science education" *Reflective Practice*, vol. 20, no. 4, pp. 533–547, 2019, doi: 10.1080/14623943.2019.1642189.

[58] R. Lansiquot, A. Satyanarayana, and C. Cabo, "Using Interdisciplinary Game-based Learning to Develop Problem Solving and Writing Skills" in *Proc. 121st ASEE Annu. Conf. Expo.*, Indianapolis, IN, USA, 2014, pp. 24.1334.1 - 24.1334.19, doi: 10.18260/1-2–23267.

[59] A. Leinonen, H. Nygren, N. Pirttinen, A. Hellas, and J. Leinonen, "Exploring the Applicability of Simple Syntax Writing Practice for Learning Programming" in *Proc. 50th ACM Tech. Symp. Comput. Sci. Educ.*, Minneapolis, MN, USA, 2019, pp. 84–90 doi: 10.1145/3287324.3287378.

[60] J. Edwards, J. Ditton, D. Trninic, H. Swanson, S. Sullivan, and C. Mano, "Syntax Exercises in CS1" *in Proc. ACM Conf. Int. Comput. Educ. Res.*, Lugano and Virtual Event, Switzerland, 2020, pp. 216–226, doi: 10.1145/3372782.3406259.

[61] A. Ly, J. Edwards, M. Liut, and A. Petersen, "Revisiting Syntax Exercises in CS1" in *Proc. 22st Annu. Conf. Inf. Technol. Educ.*, SnowBird, UT, USA, 2021, pp. 9–14, doi: 10.1145/3450329.3476855.

[62] J. Edwards, J. Leinonen, C. Birthare, A. Zavgorodniaia y A. Hellas, "Programming Versus Natural Language" in *Proc. ACM Conf. Int. Comput. Educ. Res.*, Lugano and virtual event, Switzerland, 2020, pp. 204–215.

[63] L. A. Curl, "Writing about programming in CS1" *ACM SIGCSE Bull.*, vol. 24, no. 4, pp. 7–10, 1992, doi: 10.1145/141837.141840.

[64] Z. R. Hassenfeld and M. U. Bers, "Debugging the Writing Process: Lessons From a Comparison of Students' Coding and Writing Practices" *Read. Teach.*, vol. 73, no. 6, pp. 735–746, 2020, doi: 10.1002/trtr.1885.

[65] O. Meerbaum-Salant, M. Armoni, and M. (Moti) Ben-Ari, "Learning computer science concepts with Scratch" *Compt. Sci. Educ.*, vol. 23, no. 3, pp. 239–264, 2013, doi: 10.1080/08993408.2013.832022.

[66] R. Pekrun, T. Goetz, A. C. Frenzel, P. Barchfeld, and R. P. Perry, "Measuring emotions in students' learning and performance: The Achievement Emotions Questionnaire (AEQ)" *Contemp. Educ. Psychol.*, vol. 36, no. 1, pp. 36–48, 2011, doi: 10.1016/j.cedpsych.2010.10.002.

[67] J. M. Harley, S. P. Lajoie, T. Tressel and A. Jarrell, "Fostering positive emotions and history knowledge with location-based augmented reality and tour-guide prompts" *Learning and Instruction,2020*, vol. 70, pp. 101163, doi: 10.1016/j.learninstruc.2018.09.001.

[68] N. M. Razali, Y. B. Wah, and others, "Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests" *Journal of Statistical Modeling and Analytics*, vol. 2, no. 1, pp. 21–33, 2011.

[69] E. A. Gehan, "A generalized Wilcoxon test for comparing arbitrarily singly-censored samples" *Biometrika*, vol. 52, no. 1–2, pp. 203–224, 1965, doi: 10.2307/2333825.

[70] R. A. Sánchez Turcios, "t-Student: Usos y abusos" *Revista mexicana de cardiología*, vol. 26, no. 1, pp. 59–61, 2015.

[71] N. Nachar and others, "The Mann-Whitney U: A test for assessing whether two independent samples come from the same distribution" *Tutorials in Quantitative Methods for Psychology*, vol. 4, no. 1, pp. 13–20, 2008, doi: 10.20982/tqmp.04.1.p013.

[72] L. F. Restrepo and J. González, "De Pearson a Spearman" *Revista Colombiana de Ciencias Pecuarias*, vol. 20, no. 2, pp. 183–192, 2007.

[73] P. Connolly and T. Vilardi, *Writing to learn mathematics and science*, New York, USA: Teachers' College Press, 1989.

[74] L. Mason and P. Boscolo, "Writing and conceptual change. What changes?" *Instr. Sci.*, vol. 28, no. 3, pp. 199–226, 2000, doi: 10.1023/a:1003854216687.

[75] M. Paredes-Velasco, M. G. Rios, and J. Á. Velázquez-Iturbide, "Analysis of the Emotions Experienced by Learning Greedy Algorithms with Augmented Reality" in *Proc 22th Int. Symp. Comput. Educ.*, Malaga, Spain, 2020, pp. 87-92.

[76] D. R. Krathwohl, "A revision of Bloom's taxonomy: An overview" *Theory into practice*, vol. 41, no. 4, pp. 212–218, 2002, doi: 10.1207/s15430421tip4104_2.

[77] M. B. Ibáñez, Á. Di Serio, D. Villarán, and C. D. Kloos, "Experimenting with electromagnetism using augmented reality: Impact on flow student experience and educational effectiveness"

Comput. Educ., vol. 71, pp. 1–13, 2014, doi: 10.1016/j.compedu.2013.09.004.

[78] D. Finch, M. Peacock, D. Lazdowski, and M. Hwang, "Managing emotions: A case study exploring the relationship between experiential learning, emotions, and student performance" *Int. J. Manag. Educ.*, vol. 13, no. 1, pp. 23–36, 2015, doi: 10.1016/j.ijme.2014.12.001.

[79] E. R. Um, H. Song, and J. Plass, "The effect of positive emotions on multimedia learning" in *Proc. EdMedia+ Innovate Learning*, 2007, pp. 4176–4185.

[80] H. Zhu, X. Zhang, X. Wang, Y. Chen, and B. Zeng, "A Case Study of Learning Action and Emotion from a Perspective of Learning Analytics" in *Proc. 17th Int. Conf. Comput. Sci. Eng.*, Chengdu, China. 2014, pp. 420-424, doi: 10.1109/cse.2014.105.

[81] L. A. Meadows and D. Sekaquaptewa, "The influence of gender stereotypes on role adoption in student teams" in *Proc. ASEE Annu. Conf. Expo.*, Atlanta, Georgia, 2013, pp. 23.1217.2–23.1217.23.

[82] L. A. Stein, D. Aragon, D. Moreno, and J. Goodman, "Evidence for the persistent effects of an intervention to mitigate gender-stereotypical task allocation within student engineering teams" in *Proc. IEEE Front. Educ. Conf.*, Madrid, Spain, 2014, pp. 1–9. doi: 10.1109/fie.2014.7044435.

[83] S. Gyawali, "Knowledge, Attitude and Practice of Self-Medication Among Basic Science Undergraduate Medical Students in a Medical School in Western Nepal" Journal Of Clinical And Diagnostic Research, 2015, doi: 10.7860/jcdr/2015/16553.6988.

[84] M. A. Tichon and E. Seat, "In their own words: the experience of older undergraduate students placed on engineering project teams with traditional first-year students" in *Proc. 32nd Annu. Front. Educ.*, Boston, MA, USA, 2002, pp. S3C-S3C.

[85] O. N. Aihie and B. I. Ohanaka, "Perceived academic stress among undergraduate students in a Nigerian University" *Journal of Educational and Social Research, 2019*, vol. 9, pp. 56–56, doi: 10.2478/jesr-2019-0013.

[86] J. Lazar, J. H. Feng, and H. Hochheiser, *Research methods in human-computer interaction*, 2nd ed. Burlington, USA: Morgan Kaufmann Publ., 2017.

[87] Z. Atiq and M. C. Loui, "A Qualitative Study of Emotions Experienced by First-year Engineering Students during Programming Tasks" *ACM Transactions on Computing Education, 2022*, vol. 22, pp. 1–26, 9, doi: 10.1145/3507696.

[88] R. Pekrun, A. J. Elliot and M. A. Maier, "Achievement goals and achievement emotions: Testing a model of their joint relations with academic performance" *Journal of Educational Psychology, 2009*, vol. 101, pp. 115–135, doi: 10.1037/a0013383.

APPENDIX EMOTIONS QUESTIONNAIRE

Note: items taken from AEQ questionnaire.

PRETEST BEFORE THE EXPERIENCE

The questionnaire assesses your emotional state before studying. The answers are treated anonymously and have no impact on the grades obtained in the subject. As the experience is totally anonymous, you must enter a code (which only you will know). The code will be formed by the last three digits of your ID + letter of your ID + the last three digits of your phone number. For example, if my ID finishes in 123R and my mobile finishes in 456, my code is 123R456.

The scale consists of 15 items that you must value from 1 (strongly disagree) to 5 (strongly agree).

1. I look forward to studying (1-strongly disagree..5-strongly agree)
2. I get so nervous that I don't even want to begin to study.
3. I feel confident that I will be able to master the material.
4. Because I get so upset over the amount of material, I don't even want to begin studying.
5. When I have to study I start to feel queasy.
6. When I look at the books I still have to read, I get anxious.
7. Because I'm bored I have no desire to learn.
8. I have an optimistic view toward studying.
9. I feel ashamed about my constant procrastination.
10. I get angry when I have to study.
11. My lack of confidence makes me exhausted before I even start.
12. I'm annoyed that I have to study so much.
13. I would rather put off this boring work till tomorrow.
14. I feel optimistic that I will make good progress at studying.
15. I feel hopeless when I think about studying.

POSTTEST AFTHER THE EXPERIENCE

The questionnaire assesses your emotional state after studying. The answers are treated anonymously and have no impact on the grades obtained in the subject. As the experience is totally anonymous, you must enter a code (which only you will know). The code will be formed by the last three digits of your ID + letter of your ID + the last three digits of your phone number. For example, if my ID finishes in 123R and my mobile finishes in 456, my code is 123R456.

The scale consists of 36 items that you must value from 1 (you totally disagree) to 5 (you totally agree).

1. I turn red when I don't know the answer to a question relating to the course material.
2. I get angry while studying.
3. When I solve a difficult problem in my studying, my heart beats with pride.
4. I'm resigned to the fact that I don't have the capacity to master this material.
5. I enjoy the challenge of learning the material.
6. The subject scares me since I don't fully understand it.
7. While studying I seem to drift off because it's so boring.
8. I feel ashamed.
9. I get annoyed about having to study.
10. Because I want to be proud of my accomplishments, I am very motivated.
11. I feel helpless.
12. I enjoy dealing with the course material.
13. Worry about not completing the material makes me sweat.
14. Studying for my courses bores me.
15. I feel embarrassed about not being able to fully explain the material to others.
16. When I excel at my work, I swell with pride.
17. I get physically excited when my studies are going well.
18. Studying is dull and monotonous.
19. I feel ashamed when I realize that I lack ability.
20. I enjoy acquiring new knowledge.
21. The material is so boring that I find myself daydreaming.
22. I worry whether I have properly understood the material.
23. Because I have had so much troubles with the course material, I avoid discussing it.
24. After extended studying, I'm so angry that I get tense.
25. I'm proud of myself.
26. After studying I'm resigned to the fact that I haven't got the ability.
27. I am so happy about the progress I made that I am motivated to continue studying.
28. When I can't keep up with my studies it makes me fearful.
29. My memory gaps embarrass me.
30. I'm discouraged about the fact that I'll never learn the material.
31. Reflecting on my progress in coursework makes me happy.
32. I don't want anybody to know when I haven't been able to understand something.
33. I think I can be proud of my accomplishments at studying.
34. I feel resigned.
35. Certain subjects are so enjoyable that I am motivated to do extra readings about them.
36. I worry because my abilities are not sufficient for my program of studies.

M. Paredes-Velasco received the Ph.D. degree in computer science from the Universidad Castilla – La Mancha, Spain, in 2006. He is currently an

Associate Professor of Computer Science at the Universidad Rey Juan Carlos and a member of the Laboratory of Information Technology in Education. His research interests include different computer supported learning fields (collaborative learning, active learning and mobile learning) and human-computer interaction.

I. Lozano-Osorio is a PhD student at Universidad Rey Juan Carlos, Madrid, where he is part of the Group for Research in Algorithms For Optimization (GRAFO). His research interests focus on the applicability of Artificial Intelligence (AI) techniques to solve social network problems, facility location problems and information technology in education.

D. Pérez-Marín received the Ph.D. degree in computer science and telecommunication from European, in 2007. She worked as a Lecturer and a Researcher with the Universidad Autónoma de Madrid for ten years. She has been a Lecturer and a Researcher with the Universidad Rey Juan Carlos, Madrid, Spain, for ten years, where she is currently an Associate Professor in the Computer Science Department. She is a member of the Laboratory of Information Technologies in Education (LITE). Her research interests include human–computer interaction, computer assisted education, and computer science education. She has published more than 100 articles in national and international journals and conferences. She was the recipient of several awards.

L. P. Santacruz-Valencia received the Ph.D. degree in Telecommunications. She is currently Associate Professor in the Department of Computer Science, at the Universidad Rey Juan Carlos. Her main areas of research focus on the use of technologies such as augmented reality for the development of applications aimed to improve teaching and learning processes.