

Mobile Learning Tools to Support in Teaching Programming Logic and Design: A Systematic Literature Review

Regina Célia COELHO, Matheus F. P. MARQUES, Tiago de OLIVEIRA,

*Department of Computer Science, Federal University of São Paulo
e-mail: rccoelho@unifesp.br, matheus.perrenoud@unifesp.br, tiago.oliveira@unifesp.br*

Received: October 2022

Abstract. Learning programming logic remains an obstacle for students from different academic fields. Considered one of the essential disciplines in the field of Science and Technology, it is vital to investigate the new tools or techniques used in the teaching and learning of Programming Language. This work presents a systematic literature review (SLR) on approaches using Mobile Learning methodology and the process of learning programming in introductory courses, including mobile applications and their evaluation and validation. We consulted three digital libraries, considering articles published from 2011 to 2022 related to Mobile Learning and Programming Learning. As a result, we found twelve mobile tools for learning or teaching programming logic. Most are free and used in universities. In addition, these tools positively affect the learning process, engagement, motivation, and retention, providing a better understanding, and improving content transmission.

Keywords: mobile learning, algorithms, programming logic, systematic literature review.

1. Introduction

Programming languages and programming logic disciplines are a great challenge for teachers and students for several reasons. Beginners are limited to superficial knowledge. They might learn all syntax and semantics of individual commands but cannot combine them to create a program (Bosse and Gerosa, 2017). Still, students need help classifying some categories to explain the difficulties in learning programming logic, such as syntax errors, variables, and programming language.

As Yadin and Aharon (2011) reported, besides beginners' problems in programming logic, programming activity or logical textual construction is broader than just the construction of programs. Developing skills for understanding and analyzing problems is also necessary but needs further exploration.

As expected, we should pay attention to technical and motivational aspects. When we talk about programming logic learning disciplines, motivational aspects have a significant impact (Shi and White, 2013). To address some problems described previously and the complexity of teaching-learning processes, new methodologies are applied to facilitate content teaching and transmission.

Mobile devices, including tablets and smartphones, are the most prevalent digital technology worldwide. Their increasing daily use among young and adults has become

an attractive and easy-to-access technology for users (Heflin *et al.*, 2017). Specifically, there is an increase in mobile apps focusing on learning and teaching programming, considering their teaching opportunities and difficulties. This upsurge is due to research works and their availability in virtual stores (Google Play, Apple Store, and others), as shown by (Amro and Romli, 2019). Thus, there is a need to provide the best learning environments and methodologies for beginners in programming logic. One of the most recommended educational methods is mobile learning, also called m-learning (Hwang and Tsai, 2011).

Isidora and colleagues (2015) point out that expected performance or innovative personal skills significantly influenced students' behavior while using m-learning, improving learning productivity and performance, concluding that m-learning is positive.

An example of a positive factor is the acceptance of SMS as an m-learning tool to distribute the content of distance educational activities. Klein and others (2015) present an experience in which SMS was used for reminders of work dates or exams, content review, and highlighting essential story points, tips, and motivational messages. The results showed increased student engagement and great method acceptance.

As Lindsay (2015) pointed out, diverse pedagogical approaches need to be investigated to understand the difference between teaching with and without mobile learning technology.

This work aims to evaluate and demonstrate some tools cited in the literature (published articles), the impact of m-learning, and the most accepted and used tools.

This systematic review begins with a study of the background, presenting the related works in section 2. Section 3 presents the research process, its implementation in section 4, and the data process in section 5. Later, in section 6, we present some discussions, followed by the threats to validity in section 7, and the conclusions and suggestions in section 8.

2. Background and related work

As Medeiros and others (2016) mention, there are many issues to cover regarding the problems of learning programming logic in a systematic literature review. They define problem-solving as one of the skills needed for programming, which can be translated as understanding the context of a problem, along with specific basic mathematical skills. Moreover, they stated that many students have difficulties in calculus and need help understanding some of the basic mathematical concepts expected in higher education. One main problem is abstract concepts due to the lack of accessible or everyday examples. Because of all this, our SLR tries to bring other perspectives about learning or understanding concepts. Macrides and colleagues (2021) conducted an SLR presenting the advantages of teaching programming at an early age (3-8 years old), the effects of educational approaches and tools, and the lack of publications or studies in early childhood education. According to them, some tools, such as games, screen-based platforms, and the web, were better accepted, indicating the possibility of introducing programming early.

When teachers use appropriate technologies or strategies, teaching programming can promote creativity, communication, collaboration, and content creation.

Criollo and partners focus on the teacher's role in m-learning use. They cite that most innovation initiatives, including the design and conception of mobile applications, ignore the teacher's perspective, i. e., whether he/she is trained to use mobile devices in the classroom (Criollo *et al.*, 2021). Still, they show that it is essential to incorporate mobile technology at the beginning of the learning curriculum.

Evrin (2014) presents a study on the tutor's perspective, including studies focused on students' mobile learning and whether there was an increased trend in integrating mobile learning and contexts in teacher education. She shows that if educators understand the potential of mobile learning in education, their role in integrating mobile devices becomes essential to assist students' learning needs across several disciplines. In her research in virtual libraries, several articles on mobile learning in various topics (e.g., physics, mathematics, robotics) showed the growth of tool usage.

Borges and others (2018) reveal an increasing interest in programming teaching projects, using gamification and tools (such as Scratch) to address new methodologies to improve motivation and results. They evidenced that some tools are not used in the institutions where they were created, even if the teachers of the other institutions recognize their benefits.

New tools have been proposed, developed, and used in different contexts. Although higher education is still the focus of many approaches, the interest in presenting programming to children and teenagers has grown, stimulating new methodologies that can also be applied to young people and adults with the appropriate adjustments.

In a systematic review, Perera and colleagues (2021) were concerned about which programming language is best suited for novice learners. They used the Technology Acceptance Model (TAM) to validate the findings. The model applied included two variables: 'Perceived Ease of Use' (PEOU) and 'Perceived Usefulness' (PU). These variables evaluate the execution, visibility, liveness, grammar, interfaces, and resource availability. The study discovered that a) visual-based tools are more suitable for learning, b) a simplified syntax is better for understanding, and c) a beginner-friendly user guide is vital for effective learning.

Several strategies are related to teaching how to program (Sobral, 2021). We can sort the strategies into the following categories: flipped classroom - students get contact with new subjects before the class; pair programming - two students work together on a code or problem; active learning - the instructor coaches the student who is in the center of the learning process, showing that the subjects are essential to students' success, continuation, or dropout in undergraduate courses.

Furthermore, Correa and others (2021) overview the context-aware analysis of m-learning and u-learning processes, a broader definition of learning anywhere and anytime, accessing content via any mobile device. With the continuous progression of technologies for mobile devices, more innovative computational techniques, and deep learning in virtual learning environments, the new generation of systems can adapt content and didactic approaches based on students' characteristics and learning styles.

Salleh and others (2013) map these programming teaching tools and their issues. The results point out that students favored visualization programming tools and that game tools were the most interesting to teach. The main concern was identifying and rectifying errors, which frustrated and discouraged students. She also cites the

importance of clear and precise instruction for students' understanding. However, the studies presented pointed out that technology use does not replace knowledge and practice in mathematical logic and logical reasoning. Thus, the work presented here aims to complement or bring new perspectives on the teaching-learning processes, how mobile learning tools are accepted, and whether they can help to understand and logically construct codes and design.

3. Systematic Review Planning

This SLR aims to verify which mobile learning tools are used to teach Programming Logic, their learning process, and their difficulties to understand the efficiency of those tools and Mobile Learning acceptance.

Therefore, we used as review protocol the **PICOC** (Population, Intervention, Comparison, Outcome, Context) criteria, as suggested by Wohlin and partners (2007):

Population: Articles concerning mobile learning in teaching programming logic, m-learning process, and learning programming process on introductory courses.

Intervention: Collect evidence of the mobile teaching tools, methods, and methodologies of teaching programming logic.

Comparison: Collect information on mobile learning tools, methods, and evidence.

Outcomes: Collect pertinent information from the tools, evaluations, results, advantages, and disadvantages to reach a scenario of possible acceptance, together with the different methodologies used. Evidence that the tools help to learn programming logic.

Context: Usage in universities and educational institutions.

3.1. Research Questions

This literature review aims to answer the following research questions:

- **RSQ01:** What are the mobile tools for teaching Programming Logic?
- **RSQ01.01:** What are the technologies of these tools?
- **RSQ01.02:** What is the teaching degree (grade/year), and is it still available online?
- **RSQ01.03:** Is the tool paid, free, or partially free?
- **RSQ02:** Which methodologies are used by these tools to support programming logic learning?
- **RSQ02.01:** What is the feedback type on these tools?
- **RSQ03:** How do these tools validate the learning process, and what are the techniques to evaluate the learning effect?
- **RSQ03.01:** What are the number of participants and the duration of each validation process?
- **RSQ04:** What are the listed advantages and disadvantages of these tools?
- **RSQ04.01:** What is the result or learning outcome?

From these questions, we conducted a study to define the terms used and the strategy to find the papers in the databases.

3.2. Identification of Studies

This research was restricted to electronic databases. We used Scopus¹, IEEE², and Science Direct³ because they offered relevant publications about the subject and allowed us to easily retrieve the full text of the papers. The research considered the period between 2011 and 2022 and only papers in English.

The terms used in the research refer to learning or teaching programming, logic content on basic or introductory courses, mobile tools, or mobile learning tools in various knowledge areas.

We used the search string in three combinations to fit the best research objective for any data source selected. After tests, we used different string combinations, according to a previous analysis of papers and the quantities arising from the searches, to optimize the search in the journals and obtain more consistent results.

The search strings used were: (1) Scopus ("algorithms and data structures tools" OR "algorithms teaching tools" OR "programming logic teaching tools" OR "programming teaching tools" OR "algorithms and data structures teaching tools" OR "mobile tools" OR "mobile teaching tools") AND ("introductory programming courses" OR "introductory courses" OR "Novice" OR "m-learn" OR "mobile learning"); (2) IEEE "(((algorithms OR teaching) OR (mobile tools OR programming logic) OR (computational logic OR applications)) AND ((m-learning OR m-learn) OR (programming courses OR Introductory courses) OR (novice OR introductory)))"; and (3) Science Direct "((m-learning OR m-learn OR mobile) AND (Introductory courses OR novice) AND (logic or algorithm))".

¹ <http://scopus.com>

² <http://ieeexplore.ieee.org>

³ <http://sciencedirect.com>

3.3. Studies Selection and Evaluation

Based on the titles and abstracts, together with the strings used on the digital sources (according to the year interval used), we analyzed all papers using the following inclusion and exclusion criteria:

Inclusion criteria:

- **IC01:** The study describes mobile learning tools for programming logic or content.
- **IC02:** The study discusses approaches to mobile learning for teaching.
- **IC03:** The study discusses the usage of mobile learning advantages or disadvantages for teaching.
- **IC04:** Papers published from 2011 until 2022.

Exclusion criteria:

- **EC01:** The study is not related to a mobile learning tool for learning or teaching.
- **EC02:** The study does not bring an approach to mobile learning.
- **EC03:** The study does not discuss the usage and results.

At this stage, we also used some quality assessment questions:

- **EQ01:** Do the studies show or cite mobile learning tools?
- **EQ02:** Does the article cover mobile application software for teaching?
- **EQ03:** Does the article present the methodology used?
- **EQ04:** Do the studies present how the tool is validated?
- **EQ05:** Does the article show or cite problems in using these tools?

We used the Parsifal tool as a model and to monitor the entire SLR protocol. We used Excel to create metrics and calculations for the entire planning and data collection phase.

After these steps:

1. The researcher searched the selected data sources, applying the string according to the search engine.
2. We selected papers according to inclusion and exclusion criteria. This verification was done by reading the works' titles, abstracts, and introductions.
3. We added the papers to an Excel worksheet for a second analysis and data extraction.

3.4. Data Extraction Strategy

After selecting papers, according to the criteria of inclusion and exclusion, basic information was collected for references, such as the paper title, author's name, keywords, and year of publication. Furthermore, to answer the RSQ01, RSQ01.01, and RSQ01.02, which characterize mobile tools for learning context on programming logic, the following features were extracted:

- **Tools Used (Tool Name):** The tool used to teach programming logic content or programming basics.
- **Application Type (Native/Hybrid/Web):** indicates the tool's application and use platform.
- **Still Online:** indicates if the tool is still working and available.
- **Teaching Degree (Grade):** indicates which grade the tools are used or applied.
- **Type of Use:** indicates if the tool is free (F), paid (P), or partially free (PF) - free content with exclusive paid features.

In order to answer RSQ02 and RSQ02.01, which determine the methodologies these tools use in teaching programming logic, we extracted the following information:

- **Methodology Type (self-regulated learning/assist learning):** indicates the methodology to monitor and use the tools.
- **Feedback Type (automatic/assisted):** points out the type of feedback used by the tools.

To answer RSQ03 and RSQ03.01, which determine how these tools validated learning, we extracted the following information:

- **Instructional Techniques (investigation/observation/game):** points out the techniques used and determines how activities are performed to achieve the objectives.
- **Explored content (type of content addressed):** indicates what contents were explored and used.
- **Activity Duration (time or period):** the average work application time or

- period.
- **Evaluation Method (qualitative/quantitative/experimental/mixed):** to identify the statistical methods used to evaluate the results.
 - **Number of People (number of students/teachers):** indicate the number of individuals participating, including students only, tutors, educators, and others.
- Finally, to answer RSQ04 and RSQ04.01, which intended to cite advantages or disadvantages and outcomes, we analyzed the following information:
- **Research Focus:** the aim of the paper, the target, or the proposal to be observed or validated.
 - **Positive Outcomes (advantages):** identifies the positive outcomes observed or reported by students through surveys.
 - **Problematic Outcomes (disadvantages):** identifies the problematic effects observed or reported by students; for instance, slow applications, few examples, or crash issues.
 - **Learning Outcomes (effects):** indicates or identifies learning effects observed or reported by students; for example, satisfaction or increased motivation.

4. Systematic Review Implementation

The research strings in the digital databases sources occurred in May 2022, including papers published between 2011-2022.

The databases indicated 1803 articles, 382 from Scopus, 616 from IEEE, and 805 from Science Direct. After analyzing the title, abstract, and inclusion and exclusion criteria, we excluded 1740 articles and analyzed 63. Figure 1 summarizes the selection process and the number of articles included and excluded by the criteria.

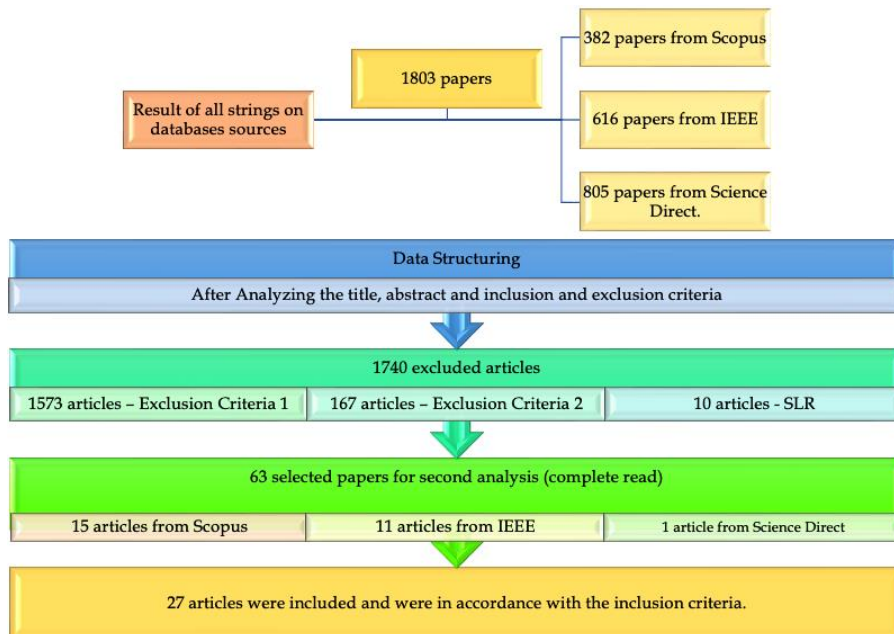


Fig. 1. Article selection process

4.1. Results Obtained from Review Implementation

As shown in Fig. 1, 63 articles remained for a second analysis. We also applied the inclusion and exclusion criteria and thoroughly read these works. Table 1 below shows the summarized result of the process, with the number of excluded articles by exclusion criteria. We considered articles that only partially met the three inclusion criteria.

Table 1
A second analysis of 63 papers

Exclusion Criteria		Quantity
EC-01	The study is not related to a mobile learning tool for learning or teaching.	21
EC-02	The study does not bring an approach to mobile learning.	5
EC-03	The study does not discuss the usage and results.	0
Total		26

Table 2 shows the remaining 27 articles, their identification, references, data source, and a brief description.

Table 2
Selected papers

ID	Reference	Source	Abstract
SP01	(Klein <i>et al.</i> , 2015)	IEEE Explore	This article presents the results of an experience using SMS (Short Message Service) to support teaching and learning in undergraduate courses.
SP02	(Alepis and Troussas, 2017)	Scopus	M-learning Programming Platform: Evaluation on Elementary Schools.
SP03	(Amro and Romli, 2019)	IEEE Explore	This study investigates the extent of the available m-learning applications via Google Play and Apple Stores using various programming techniques.
SP04	(Burke, 2021)	Scopus	The research aims to describe the m-learning experiences of high-school students in mathematics and sciences and to determine if distinctive pedagogical dimensions impact students' perceived learning.
SP05	(Charles <i>et al.</i> , 2019)	IEEE Explore	This paper addresses the under-researched connection between mobile learning technologies incorporated in an Introduction to Computer Engineering course and student retention.
SP06	(Condé <i>et al.</i> , 2013)	Scopus	This paper describes a service-based approach to implementing a mobile Personal Learning Environment.

SP07	(Elfeky and Masadeh, 2016)	Scopus	The study aimed to examine the effect of Mobile Learning, a kind of E-learning that uses mobile devices, on the development of academic achievement.
SP08	(Fagan, 2019)	Scopus	This study aimed to explore students' acceptance of mobile learning in a higher education setting.
SP09	(Forkides, 2018)	Scopus	The study presented the results of a project that used tablets and a ready-made application for teaching introductory programming concepts to young primary school students.
SP10	(Gezgin, 2019)	Scopus	This article investigates the effect of mobile learning support on students' academic success in a database management system (DBMS).
SP11	(Gikas, 2013)	Scopus	This paper presents findings on students' perceptions of learning with mobile computing devices and the roles of social media.
SP12	(Martínez <i>et al.</i> , 2018)	IEEE Explore	This paper presents the structure and experience of PLSOOP (Platform for Learning Support and Object-Oriented Programming).
SP13	(Lindsay, 2015)	IEEE Explore	This study finds that the primary use of mobile technology is to enhance learning with task activities and information access. However, a typical use is also innovative content production.
SP14	(Isidora <i>et al.</i> , 2015)	IEEE Explore	This paper presents research results from applying new technologies in higher education, particularly emphasizing M-learning as a modern innovative approach.
SP15	(Sergio and German, 2020)	IEEE Explore	This paper presents a workshop on mobile application programming using MIT App Inventor with first-year students to address the lack of basic programming skills.
SP16	(Omar, 2021)	Scopus	The current study aimed to know the factors that affect university college students' acceptance and use of Mobile learning (ML).
SP17	(Martin <i>et al.</i> , 2013)	Scopus	This paper documents the outcomes of a study that focused on identifying what motivates students to use mobile devices for learning.
SP18	(Matthew <i>et al.</i> , 2020)	Scopus	This paper focuses on learning and teaching psychological theories and emerging mobile learning theories.
SP19	(Cavus, 2020)	IEEE Explore	The study's primary purpose was to identify whether the developed m-learning system is acceptable as a mobile learning environment.
SP20	(Ng and Wong, 2020)	Scopus	This study aims to explore the practices of university students in China regarding mobile learning using a questionnaire.

SP21	(Ortiz et al., 2015)	IEEE Explore	This paper presents an experience based on m-learning in higher education.
SP22	(García and Rosa, 2016)	Scopus	This paper presents the results of an experimental approach that evaluated the interaction of a group of Colombian students with a Web solution within the context of Mobile Robotics to learn programming and algorithmics.
SP23	(Rodríguez <i>et al.</i> , 2020)	Science Direct	This paper presents a Context-Aware Mobile Learning System (CAMLs) usability assessment.
SP24	(Malik, 2020)	IEEE Explore	This study developed and introduced a 'PROBSOL' application in three different learning systems (E-learning, M-learning, and Game-based learning).
SP25	(Sakibayev <i>et al.</i> , 2019)	Scopus	This paper aims to test the hypothesis that introducing mobile technology to college-level database courses for future IT specialists has a positive academic impact on students.
SP26	(Burke et al., 2022)	Scopus	This paper investigates m-learning experiences and determines which dimensions impact the students' learning.
SP27	(Kao and Ruan, 2022)	Science Direct	This study uses a programming learning system based on Augmented Reality to help students improve their programming concepts.

Table 3 lists nine papers in the literature related to the mobile learning tool or mobile software in teaching programming logic, followed by their identification, references, data source, and a brief description.

Table 3
Selected papers with Mobile Learning Tool

ID	Reference	Source	Abstract
S01	(Ortiz et al., 2015)	IEEE Explore	This paper presents an experience based on m-learning in higher education.
S02	(Martínez <i>et al.</i> , 2018)	IEEE Explore	This paper presents the structure and experience of using PAEPOO - PLSOOP (Platform for Learning Support and Object-Oriented Programming).
S03	(Amro And Romli, 2019)	IEEE Explore	This study investigates the availability of m-learning applications via Google Play and Apple Stores, which use various techniques to support effective programming learning.

S04	(Malik, 2020)	IEEE Explore	This paper presents a 'PROBSOL' application in three learning systems (E-learning, M-learning, and Game-based learning) to enhance problem-solving skills.
S05	(Cavus, 2020)	IEEE Explore	The study's primary purpose was to identify whether the developed m-learning system is acceptable.
S06	(Sergio and German, 2020)	IEEE Explore	This paper presents a workshop on mobile application programming using MIT App Inventor offered to first-year students to address the lack of basic programming skills.
S07	(Alepis <i>et al.</i> , 2017)	Scopus	The work presents m-AFOL, a mobile learning platform for elementary school students to teach basic programming principles.
S08	(Feijóo and Rosa, 2016)	Scopus	This paper presents the results of an experimental approach that evaluated the interaction of a group of Colombian students with a Web solution within the context of Mobile Robotics to learn programming and algorithmics.
S09	(Kao and Ruan, 2022)	Science Direct	This study uses a programming learning system based on Augmented Reality to help students improve their programming concepts.

5. Arrangement Process

After finding the results of the systematic review, we classified the data to analyze and help to answer the research questions.

Table 4 followed the criteria to answer the predefined research questions, **RSQ01**, **RSQ01.01**, **RSQ01.02**, **RSQ02**, and **RSQ02.01**, which intended to characterize the mobile tools for learning context on Programming Logic and their methodologies. The twelve mobile learning tools found are in this table.

Table 4
Data collected to answer RSQ01 and RSQ02 and their secondary questions.

ID	Tool Name	Application Type	Still Online?	Grade	Type of Use	Methodologies
S01	Eduoct	Web Application	N	University	F	Assist-Learning
S02	PLSOOP	Web Application	N	University	F	Self-Regulated
*S03.01	Sololearn	Multi-platform	Y	University	PF	Self-Regulated
*S03.02	Programmi ng Hub	Multi-platform	Y	University	PF	Self-Regulated

*S03.03	Mimo	Multi-platform	Y	University	PF	Self-Regulated
S04	PROBSOL	Multi-platform	Y	University	F	Self-Regulated
**S05	MoblrN	Multi-platform	Y	University	F	Self-Regulated
S06	App Inventor	Multi-platform	Y	University	F	Assist-Learning
S07	m-AFOL	Android OS.	N	Elementary	F	Self-Regulated
*S08.01	RoBlock	Multi-platform	N	High School	F	Self-Regulated
*S08.02	Scratch	Multi-platform	Y	High School	F	Self-Regulated
S09	AR-based learning	Multi-platform	-	Elementary	-	Assist-Learning

*The rows for S03 and S08 articles refer to papers that cited more than one mobile learning tool.

**The S05 article cites MoblrN, which has a similar tool to the new m-learning digital environment for teaching and learning. This tool can have computational-related content populated previously by any professor or tutor, so we included this tool here.

Butler and Morgan (2007) show two principal types of feedback: automatic and assisted. The first is auto-generated feedback, and the second is teachers' feedback, which can help students get unstuck and correct their errors. However, automatic feedback about usage, exercise errors, and results can also be beneficial, reducing teacher effort. Every tool listed in this work use automatic feedback.

The defined criteria to answer the validation types and duration of each process for the tools found (as questioned in **RSQ03** and **RSQ03.01**) are in Table 5.

Table 5
Criteria to answer the RSQ03 and its secondary questions

ID	Instructional Techniques	Covered Content	Evaluation Method	Activity duration	Number of people
S01	Investigation	Programming Logic	Mixed	4 Months	163
S02	Investigation	Oriented Object Algorithms Basic	Quantitative	6 Months	174
S03.01	Investigation	Programming Logic	Mixed	Undefined	Undefined

S03.02	Investigation	Programming Logic	Mixed	Undefined	Undefined
S03.03	Investigation	Programming Logic	Mixed	Undefined	Undefined
S04	Observation Game	Programming Logic	Quantitative	Undefined	Undefined
S05	Observation	Any Content	Qualitative	Undefined	74
S06	Observation	Programming Logic	Experimental	10 Hours	164
S07	Investigation	Programming Logic	Qualitative	Undefined	40
S08.01	Investigation	Programming Logic	Mixed	Undefined	46
S08.02	Investigation	Programming Logic	Mixed	Undefined	46
S09	Observation	Programming Logic	Mixed	4 Weeks	50

Finally, Table 6 shows the defined criteria to understand the tools' advantages, disadvantages, and final effects (used to answer the questions of RSQ04 and RSQ04.01).

Table 6
Criteria used to answer the RSQ04 and secondary questions.

ID	Research Focus	Advantages	Disadvantages	Result Outcome
S01	Evaluate acceptance of the m-learning tool	Useful tool Help to learn	Technical Problems	Engagement
S02	Evaluate acceptance of the m-learn tool	Easy to use	Undefined	Productivity Interactivity
S03.01	Investigate acceptance of three tools from online stores	Useful tool	Undefined	Productivity Interactivity
S03.02	Investigate acceptance of three tools from online stores	Useful tool	Inefficient	Productivity Interactivity
S03.03	Investigate acceptance of three tools from online stores	Useful tool	Undefined	Engagement Creativity

S04	Enhance problem-solving skills through m-learn	Easy to use User friendly	Few examples	Engagement
S05	Evaluate new m-learning system acceptance and usability	Useful tool User friendly	Undefined	Motivation Interactivity
S06	Workshop to tackle the lack of basic programming skills	Useful tool Help to learn	Undefined	Motivation Creativity
S07	Enhance basic programming principles via m-learn	User Friendly Useful tool	Interface Issues	Engagement
S08.01	Evaluate tool acceptance	User Friendly	Undefined	Engagement
S08.02	Evaluate tool acceptance	User Friendly	Undefined	Engagement
S09	Evaluate new m-learning system acceptance and usability	Useful tool Easy to use	Undefined	Motivation Interactivity

6. Discussion

RQ01 - What are the mobile tools for teaching Programming Logic?

We identified 8 of 12 papers from the systematic review results that use mobile learning apps to teach computational content, such as programming basics or algorithms. One article of these eight uses a brand-new mobile learning environment that allows the inclusion of computational content.

The tools found in this research, followed by the selected paper with mobile learning tools:

- (S01) Eduoct;
- (S02) PLSOOP;
- (S03.01) Sololearn;
- (S03.02) Programming Hub;
- (S03.03) Mimo;
- (S04) PROBSOL;
- (S05) MoblrN;
- (S06) App Inventor;
- (S07) m-AFOL;
- (S08.01) RoBlock;
- (S08.02) Scratch;
- (S09) AR-based programming learning system.

Next, we briefly present these tools, showing their technologies and functionalities.

(S01) Eduoct: mobile learning tool created by an educational experience in Spain, using the C language to implement algorithms. It was created as a Web Application for most mobile devices. The exercises were organized with increasing difficulty, showing an arrangement of disorganized blocks to be ordered. It presents a simple interface and controls within a small local database to record the exercises and users' results (Ortiz *et al.*, 2015).

(S02) PLSOOP: object-oriented programming platform for learning support used in academic fields. A web application with an android app version has three layers: Definer, CodeFactory, and ErrorFinder. The first layer is a guide to declaring terms in the java languages, such as variables, classes, and objects. The second layer is a java code generator in which, based on the question, the code is created piece by piece. Finally, the last layer has code examples with errors for the student to find and solve (Martínez *et al.*, 2018).

(S03.01, S03.02 and S03.03) Sololearn, Programming Hub, and Mimo: these three tools are present on the same paper. SoloLearn is a programming app that uses several free code learning contents, such as Python, Java, Kotlin, C++, C, C#, and PHP. It has free courses, a Web Application, and a Mobile application. Programming Hub and Mimo offer similar programming language contents, such as Java, HTML, and Python. However, Programming Hub has more programming topics for learning coding concepts than Mimo. Both apps allow users to write and run actual code. All these apps have a clean layout with drag-and-drop, fill the text, and use code blocks (Amro and Romli, 2019).

(S04) PROBSOL: an introductory programming course that is a multi-based application used to enhance problem-solving. With a downloadable Android version, the app is based on pseudocode and covers most of the initial topics on programming. The web layout was developed with ASP.net, and the use is based on clicks and ordering rows of code. However, it also has an area to write the code and a button to run the test. It is based on different questions ranked by difficulty (Malik, 2020).

(S05) MoblrN: a mobile learning system that offers an m-learning environment with access control, user profile, learning materials, assignments, self-test, announcements, performance, and report generation. The application was built using PHP, Java, and MySQL (Cavus, 2020).

(S06) App Inventor: a multi-platform visual programming environment with blocks-based coding programs driven by examples and mini-courses. It introduced basic programming concepts as a self-regulated tool or assisted learning (Sergio and German, 2020).

(S07) m-AFOL: a mobile learning platform named "Mobile-Affective Object-Oriented Logo Language," an evolution of past "Logo" programming language. It is only available for Android, requiring an Internet connection. It uses a user-friendly interface designed to create effective interaction among children. It also has illustrated tutorials

and exercises. The app prompts the students to write simple commands using lines and basic shapes (Alepis and Troussas, 2017).

(S08.01 and S08.02) Roblock & Scratch: both tools are on the same paper. Roblock is a Web Application solution designed for teaching programming and algorithms. It uses the concept of teaching through programming a virtual robot with multiple-choice problems, stipulated time to answer, and usage of code blocks with tutorials and videos. Scratch is an MIT Web Application used to learn algorithms within an event-oriented paradigm, learn to program by game design on a blank canvas, and syntax-free Visual Blocks Programming. There is a Scratch Jr version for Android (Garcia and Rosa, 2016).

(S09) AR-based programming learning system: an augmented reality system for programming learning that uses Unity 3D game platform, Maya 3D and Vuforia AR developer portal. It covers memory-oriented programming concepts, such as functions and algorithms, sequences, selections, repetition structures, and logic-oriented content. The student uses an iPad to interact with the system (Kao and Ruan, 2022).

The results after data collection and analysis demonstrate that the twelve mobile tools are distributed as follows: 2 are web applications (S01 and S02); 9 are multi-platform such as web applications, mobile or desktop (S03.01, S03.02, S03.03, S04, S05, S06, S08.01, S08.02, and S09); and one of these tools is designed only for Android operating system (S07). Here we see that the vast majority aim to reach as many devices as possible, to attract and maintain the public.

We also point out that more than half of mobile tools are on university usage: 8 in total, 2 in high school, and 2 in elementary school, showing the lack of applications or studies in other educational levels, together with the lack of portability to other languages.

Eight of these mobile apps are free to use (S01, S02, S03, S04, S05, S06, and S07), 3 of them are partially free (S03.01, S03.02, and S03.03), and there is no information for one of them (S09). Demonstrating the experimental academic development and usage, the majority (7 tools) are still online (S03.01, S03.02, S03.03, S04, S05, S06, and S08.02), four tools are no longer online (S01, S02, S07, and S08.01), and we have no information about one (S09).

All twelve mobile tools (S01, S02, S03.01, S03.02, S03.03, S04, S05, S06, S07, S08.01, S08.02, and S09) focus on algorithm learning, some using the language C and others java algorithms.

Finally, the main characteristic observed in all these mobile tools is that they have a visual type of content, such as blocks to organize or write together with a compiler that reads the user input and sends the result immediately. The negative aspects are the lack of tools to use offline mode. Furthermore, not all have a pseudocode programming language to start learning to program. Some use a market/production programming language at the beginning of programming concepts, hindering learning by forcing the use of more complex syntax.

RQ02 - Which methodologies are used by these tools to support learning programming logic?

Zimmerman (2008, p.167) refers to self-regulated learning (SRL) as "the self-directive processes and self-beliefs that enable learners to transform their mental abilities, such as verbal aptitude, into an academic performance skill derived from the learner's advantageous motivational feelings and beliefs as well as metacognitive strategies".

Sha and colleagues (2012) believe that mobile learning effectiveness depends on students' ability to correctly determine what, when, where, and how to learn, their mood and motivation.

- **The self-regulated learning methodology:** human behavior results from inner orientation linked to emotions and behavior. The student mediates engagement and learning, which also depends on external stimuli. Motivation and lack of interest are the most common problems. The technological resource can foster the desire and need for its use. However, it will be left aside without stimulus after users lose interest or fulfil an obligation, as pointed out by Sha and colleagues (2012).

- **The assist-learning methodology:** an individual learning process with support materials and guidance from a tutor or teacher. In this case, it is common to have events and/or new activities, applications, and programs updated to new questionnaires and assessments. (Emran *et al.*, 2021).

We noticed that nine tools use the self-regulated methodology (S02, S03.01, S03.02, S03.03, S04, S05, S07, S08.01, and S08.02). This methodology is common in m-learning tools because, after its completion, the contents of visual, audio and video stimuli keep the user connected and interested, as reminded by Emran and colleagues (2021).

There are only three tools with the Assist-Learning methodology (S01, S06, and S09). The S06 tool is also used as an SRL methodology. The paper S06 presented an experiment (workshop) with assisted learning, in which the tools are environments to post content. They also allow communication and data updates (Ortiz *et al.*, 2015).

Noticing that all twelve tools have automatic feedback, Pinheiro and colleagues (2021) emphasize that feedback is essential in teaching-learning, helping to identify gaps and substantial learning progress (Butler and Winne, 1995).

As expected, the tools bring agility and assertiveness, facilitating the tutor's work with automatic corrections and notification of submissions and grades. It is important to remember the quality of the feedback and how users receive it. Feedback plays a crucial role, as participants have no face-to-face interaction (Ypsilandis, 2002).

All twelve tools have some visual or textual feedback, from users' hits and misses, questionnaires, and general use. Some offer resolution tips and the chance for new attempts. The feedback has a "programmed content" with no recommendation or assistance in the execution, thus only checking the success and error of the activities. Not all have a tutorial or extra content to help users understand them.

RQ03 - How do these tools validate the learning process?

Fig 2 presents the evaluation methods of the tools listed on this SLR.

Evaluation Methods - Results

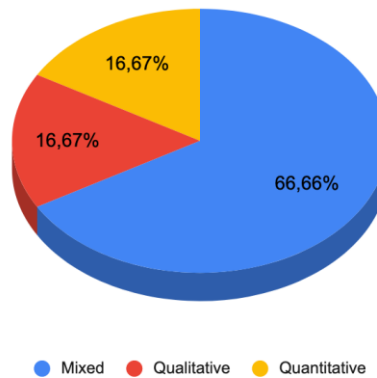


Fig. 2. Learning effects distribution

We first need to conceptualize three methods to understand the answers better: qualitative, quantitative, and experimental.

Qualitative: is not concerned with numerical representatively but a deeper understanding of a given problem. The objective is to represent or reproduce the information to understand the various dimensions of the analyzed problem (Flanagan, 2013).

Quantitative: generally has as many data samples as possible to represent the population. The results constitute the general view of the entire population (Martin and Bridgmon, 2012).

Fig. 2 represents the research method used to evaluate the results. We can see that the qualitative method was used on two articles (S05 and S07) and the quantitative one on two others (S02 and S04), followed by mixed evaluation methods. Papers with more than one research method (S01, S03.01, S03.02, S03.03, S06, S08.01, S08.02, and S09) were the most common (8 papers).

As we can see, mixed evaluation methods are preferred in the papers, merging the data collected from these tools with personal feelings and understanding when facing programming learning and practice.

Accepting these tools increases engagement and motivation, which correlate to a dropout decrease.

Methods that contain good practices are defined as those with applied instructional techniques using strategies that enable the student to acquire knowledge effectively (Benjamin *et al.*, 2014). We categorized the instructional techniques found as observation or investigation.

Nine resulting papers proposed that students seek content to acquire knowledge (investigation instructional technique) (S01, S02, S03.01, S03.02, S03.03, S04, S07, S08.01, and S08.02). Three papers followed an observation technique in which students had to observe content to learn (S05, S06, and S09).

RQ04 - What are the listed advantages and disadvantages of these tools?

To evaluate the environments of the mobile tools, we observed that the mobile learning applications have multiple positive effects, observed by teachers or students in different cases or scenarios. Fig. 3 represents the words and expressions used in the papers to indicate advantages (each paper could have more than one selected word).

The words used were: 1) useful tool, considered helpful for learning, applying, and assimilating content, with eight papers (S01, S03.01, S03.02, S03.03, S05, S06, S07, and S09); 2) helping to learn/ help in learning (S01 and S06) as cited by users (2 articles); 3) easy to use (S02, S04, and S09) with three articles and it applies to the tool itself; 4) user-friendly, clean interface and enjoyable to use (S04, S05, S07, S08.01, and S08.02) appeared on five papers. These advantages were raised in 18 papers, as seen in figure 3 below:

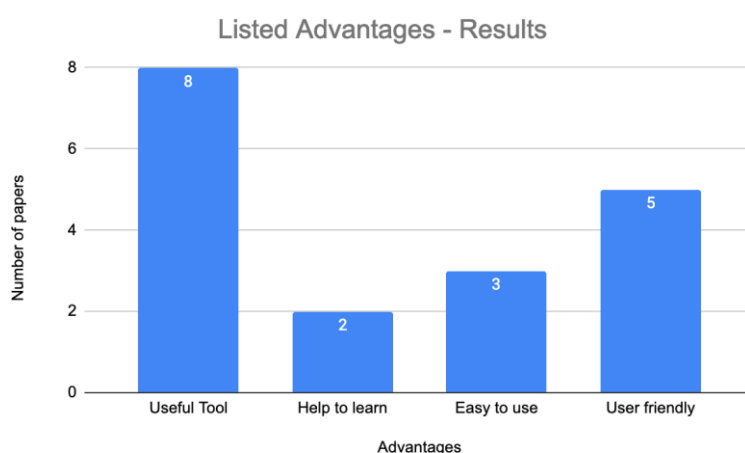


Fig. 3. List of Advantages by papers

We could also see that mobile learning applications have multiple adverse effects, such as disadvantages and other problems observed by teachers or students in different cases or scenarios.

Fig. 4 represents the words and expressions collected for disadvantages or problems (one paper can have more than one selected word). The words used are: 1) technical problems, which corresponds to mobile tool technical failure or problem with one paper (S01); 2) undefined (S02, S03.01, S03.02, S03.03, S05, S06, S08.01, S08.02, and S09) when the paper does not explicitly cite the problem or disadvantage (nine papers); 3) few examples (S04) when the application does not have a significant number of statements or activities (one paper); and, finally, 4) interface issues (S07), when the application can crash, close or does not show well all objects on screen (one paper). We can observe a lack of negative feedback or documentation of possible problems, as most articles in this review have a positive bias.

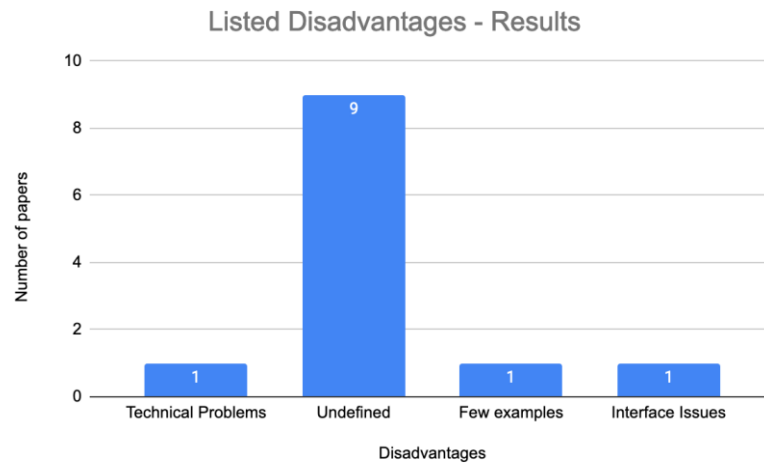


Fig. 4. List of Disadvantages by papers

We also collected the possible actual outcomes of each proposed mobile learning tool, separated into five keywords.

Fig. 5 represents the words collected for outcomes based on analysis and the researcher's understanding of every conclusion and final result. One paper can receive more than one word. The selected words were: 1) motivation (S05, S06, and S09), based on the feedback after usage of each tool or environment (three papers); 2) creativity (S03.03 and S06), the possible perception of increased user's creativity during or after using the tool (two papers); 3) productivity (S03.01, S03.02, and S03.03), perception of more significant productivity increase (three papers); 4) interactivity (S02, S03.01, S03.02, S05, and S09), student- tool and student-teacher integrations (five papers); and, finally, 5) engagement (S01, S03.03, S04, S07, S08.01, and S08.02), possible increase in satisfaction, more significant effort in using the tool, either for learning or by their own free will, that is, more participative people, with six papers. Carlson and others (2019) cite that mobile learning can increase motivation and positively affect student confidence.

This study has nineteen occurrences based on the analysis and results demonstrated in the selected papers.

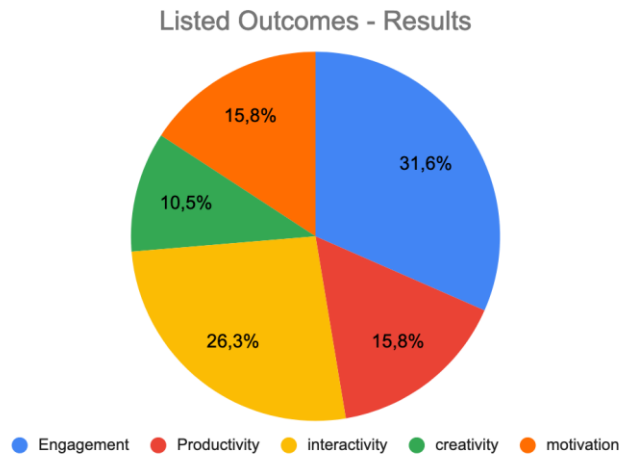


Fig. 5. Real outcomes representativeness

7. Threats to Validity

In order to carry out the Systematic Literature Review, we analyzed all data and followed the previously established protocol. Thus, the bias in results is reduced compared to traditional review methods.

However, though we followed all the guidelines, some threats can affect the research results. So, we chose high-quality digital libraries for this work: Science Direct, Scopus, and IEEE. However, other relevant articles might have been published in other sources. On the other hand, including more libraries would significantly increase the number of articles, affecting the quality and assertiveness of the analysis. One possibility would be to continue this review by adding new libraries, following the protocol in this research. Even though we conducted several tests with the search strings for each digital library, it is challenging to measure its best performance until reaching the final string.

Therefore, the search string used must be chosen very carefully. Several discarded papers did not address relevant content to our research objective, though some presented words in the string. At the same time, relevant articles may have only some words in the string, so they were not considered.

However, generating new strings to capture more relevant articles to the topic can also increase the number of irrelevant articles.

Another critical issue is the trend of publishing positive research results rather than negative ones, even when they can help identify or demonstrate problems with the current methods. The inclusion and selection criteria and the quality assessment can also pose a significant threat due to subjective decisions of classification and choice.

The validity of an SLR depends on the quality and clearness of the information presented in the articles. Therefore, the researcher's experience is essential to solve any possible doubts.

Finally, the data extraction phase requires the researcher to understand various information about the different methodologies applied in the articles and to standardize the collected information.

8. Conclusion and Suggestions

Strictly following a protocol, this systematic literature review compiles articles that improve our understanding of mobile learning tools for programming logic.

After analyzing all the information from the articles, we could see the increase in the emergence of mobile teaching tools in the areas of computer science, mainly in the academic environment, to promote teaching and research, and the considerable effort to increase engagement, motivation, and improvement in student learning and retention.

Another critical point is the need for studies with opposing positions, citing problems, suggestions for improvement, or usage bugs. In addition, the need for multi-language applications (portability) is evident.

Additionally, we perceived a lack of papers on programming for early ages, demonstrating that the tools only appear or are being created by university teachers to teach computational programming content, such as programming languages or algorithms. Even though using these tools from an early age may be essential for children's formation and understanding of logic.

We also noticed that only one work dealt with the pseudo-language format, even though pseudocode is more comfortable and easier for users because it resembles the spoken language. However, many tools and subjects focus on teaching computational logic or algorithms through market languages, hindering learning due to the nuances of learning specific syntaxes for each action. Thus, there is a concern about training students capable of market programming, but this needs to start with more simplistic approaches, techniques, and skills.

We can see that the user interfaces (UI) used by the students, authors, and creators were the visual block programming style, maybe because it is easy to accept and use, and the correlation of visual style with the capability to learn or use the tool. Thus, showing that applications that use visual blocks have greater acceptance.

A technical aspect is that all tools need the Internet to run, with no offline operation or mode, thus limiting their usage and experience. All the papers found in this research were based on problem analysis or problem-solving regarding programming logic. This result may indicate that the definition of programming logic is perceived as problem analysis or problem-solving, and the studies approached their research questions with this definition in mind. Indeed, there are several attempts to create new tools, to understand and explain learning difficulties in the academic field.

We can see that different tools and ways of teaching are evolving. However, we need to pay attention to the form of analysis, application, and acceptability of these tools. Even with more experimental cases open to more users, we know that mobile tools in the classroom may contain some didactic and/or application bias.

We noticed that several tools were created to solve specific problems, learning was not standardized, and there was no standard application. Finally, only seven tools are still online within the ten years searched by this SLR (2011-2022), showing the need for new tools and methodologies for teaching and learning.

We suggest for future studies:

- To develop reliable tools to measure the levels of programming logic learning, besides the technical skills of those tools;

- To correlate the lack of programming language studies with current problems faced by high school and university students
- To analyze the relationship between computational thinking and programming logic learning with mobile tools.

References

- Alepis, E., Troussas, C. (2017). M-learning programming platform: Evaluation in elementary schools. *Informatica (Slovenia)*. 41. 471-478. *America Transactions*, vol. 16, no. 6, pp. 1825-1830, June 2018, doi: 10.1109/TLA.2018.8444405.
- Amro, J.S., Romli, R. (2019). "Investigation on the Learning Programming Techniques via Mobile Learning Application," 2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE), 2019, pp. 1-7, doi: 10.1109/ICRAIE47735.2019.9037764.
- Benjamin, N. & Sottolare, R., Ragusa, C., Hoffman, M. (2014). *Defining Instructional Challenges, Strategies, and Tactics for Adaptive Intelligent Tutoring Systems*.
- Borges, R.P., Oliveira, P.R.F., Lima, R.G. da L., Lima, R.W. (2018). "A Systematic Review of Literature on Methodologies, Practices, and Tools for Programming Teaching," in *IEEE Latin America Transactions*, vol. 16, no. 5, pp. 1468-1475, May 2018, doi: 10.1109/TLA.2018.8408443.
- Bosse, Y., Gerosa, M.A. (2017). Why is programming so difficult to learn?: Patterns of Difficulties Related to Programming Learning Mid-Stage. *ACM SIGSOFT Software Engineering Notes*. 41. 1-6. 10.1145/3011286.3011301.
- Burke, P. F., Kearney, M., Schuck, S., Aubusson, P. (2021). Improving mobile learning in secondary mathematics and science: Listening to students. *Journal of Computer Assisted Learning*. doi:10.1111/jcal.12596.
- Butler, D. L., Winne, P. H. (1995). Feedback and self-regulated learning: A theoretical synthesis. *Review of Educational Research*, 65, 245–281. <https://doi.org/10.3102/00346543065003245>.
- Butler, M., Morgan, M. (2007). "Learning challenges faced by novice programming students studying high level and low feedback concepts," in *Proceedings of ASCILITE - Australian Society for Computers in Learning in Tertiary Education Annual Conference*, 2007.
- Carlson, C., Peterson, G., Day, D. (2019). Utilizing Portable Learning Technologies to Improve Student Engagement and Retention. *IEEE Transactions on Education*, 1–7. doi:10.1109/te.2019.2941700.
- Cavus, N. (2020). Evaluation of MoblrN M-learning system: Participants' attitudes and opinions. *World Journal on Educational Technology: Current Issues*. 12. 150-164. 10.18844/wjet.v12i3.4978.
- Condé, M.Á., Peñalvo, F.G., Alier, M., Casany, M.J., Piguillem, J. (2013). Mobile devices applied to Computer Science subjects to consume institutional functionalities through a personal learning environment (2013) *International Journal of Engineering Education*, 29 (3), pp. 610-619.
- Correa, P.V., Pulido, J.M., Betancur, M.T. (2021). A systematic mapping review of context-aware analysis and its approach to mobile learning and ubiquitous learning

- processes, *Computer Science Review*, Volume 39, 2021, 100335, ISSN 1574-0137, <https://doi.org/10.1016/j.cosrev.2020.100335>.
- Criollo, S.C., Moscoso-Zea, O., Guerrero-Arias, A., Jaramillo-Alcazar, Á., Luján-Mora, S. (2021). "Mobile Learning as the Key to Higher Education Innovation: A Systematic Mapping," in *IEEE Access*, vol. 9, pp. 66462-66476, 2021, doi: 10.1109/ACCESS.2021.3076148.
- Elfeky, A. Masadeh, T. (2016). The Effect of Mobile Learning on Students' Achievement and Conversational Skills. *International Journal of Higher Education*. 5. 20-31. 10.5430/ijhe.v5n3p20.
- Emran, M., Mezhuyev, V., Kamaludin, A. (2020). Towards a conceptual model for examining the impact of knowledge management factors on mobile learning acceptance, *Technology in Society*, Volume 61, 2020, 101247, ISSN 0160-791X, <https://doi.org/10.1016/j.techsoc.2020.101247>.
- Evrin, B. (2014). A Review of Research on Mobile Learning in Teacher Education. *Educational Technology & Society*. 17. 17-32.
- Fagan, M. (2019). Factors Influencing Student Acceptance of Mobile Learning in Higher Education. *Computers in the Schools*. 36. 1-17. 10.1080/07380569.2019.1603051.
- Flanagan, T. (2013). The scientific method and why it matters. *C2C Journal*, 7(1), 4-6.
- Fokides, E. (2018). Teaching Basic Programming Concepts to Young Primary School Students Using Tablets: Results of a Pilot Project. *International Journal of Mobile and Blended Learning*. 10. 34-47. 10.4018/IJMBL.2018010103. From hypothesis to results. New Jersey, USA: Jossey-Bass.
- Gezgin, D. (2019). The Effect of Mobile Learning Approach on University Students' Academic Success for Database Management Systems Course. *International Journal of Distance Education Technologies*. 10.4018/IJDET.2019010102.
- Gikas, J., Grant, M.M. (2013). Mobile computing devices in higher education: Student perspectives on learning with cellphones, smartphones & social media. *Internet and Higher Education*, 19, pp. 18-26.
- Hwang, G., Tsai, C. (2011). Research trends in mobile and ubiquitous learning: A review of publications in selected journals from 2001 to 2010. *British Journal of Educational Technology*. 42. E65 - E70. 10.1111/j.1467-8535.2011.01183.x.
- Isidora M., Dragana, Ž., Dragan M., Danijel N. (2015). The effects of the intended behavior of students in the use of M-learning. *Comput. Hum. Behav.* 51, PA (October 2015), 207–215. DOI: <https://doi.org/10.1016/j.chb.2015.04.041>.
- Kao, G. Y., Ruan, C. (2022). Designing and evaluating a high interactive augmented reality system for programming learning, *Computers in Human Behavior*, Volume 132, 2022, ISSN 0747-5632, <https://doi.org/10.1016/j.chb.2022.107245>.
- Klein, A.Z., Junior, C.S.F.J., Barbosa, J.L.V. (2015). "M-Learning in Practice: Using SMS for Teaching and Learning in Undergraduate Courses," in *IEEE Latin America Transactions*, vol. 13, no. 1, pp. 321-329, Jan. 2015, doi: 10.1109/TLA.2015.7040665.
- Lindsay, L. (2015). Transformation of teacher practice using mobile technology with one-to-one classes: M-learning pedagogical approaches. *British Journal of Educational Technology*. 47. 10.1111/bjet.12265.

- Macrides, E., Miliou, O., Angeli, C. (2021). Programming in early childhood education: A systematic review. *International Journal of Child-Computer Interaction*. 32. 100396. 10.1016/j.ijcci.2021.100396.
- Mahasneh, O. (2021). Factors that affect University College Students' acceptance and use of Mobile Learning (ML). *International Journal of Instruction*. 14. 861-872. 10.29333/iji.2021.14350a.
- Malik, S.I. (2020). Paper Comparison of E-Learning, M-Learning and Game-based Learning in Programming Education... Comparison of E-Learning, M-Learning and Game-based Learning in Programming Education.
- Martin, R., McGill, T., Sudweeks, F. (2013). Learning anywhere, anytime: Student motivators for m-learning. *Journal of Information Technology Education: Research*, 12 (1), pp. 51-67.
- Martin, W.E. and Bridgmon, K.D. (2012) *Quantitative and Statistical Research Methods: From Hypothesis to Results* (Vol. 42). John Wiley & Sons, Hoboken, NJ.
- Martínez, L.G., Marrufo, S., Licea, Reyes-Juárez, G., Aguilar, L. (2018). "Using a Mobile Platform for Teaching and Learning Object Oriented Programming," in *IEEE Latin America Transactions*, vol. 16, no. 6, pp. 1825-1830, June 2018, doi: 10.1109/TLA.2018.8444405.
- Matthew, L., Bernacki, J., Greene, A., Crompton, H. (2020). Mobile technology, learning, and achievement: Advances in understanding and measuring the role of mobile technology in education, *Contemporary Educational Psychology*, Volume 60, 2020, ISSN.0361-476X, <https://doi.org/10.1016/j.cedpsych.2019.101827>."
- Medeiros, R.P., Ramalho, G.L., Falcão, T.P. (2016). "A Systematic Literature Review on RoBlock - Web app for programming learning. *International Journal of Emerging Technologies in Learning*, 11 (12), pp. 45-53. DOI: 10.3991/ijet.v11i12.6004.
- Ng, E.M.W., Wong, H.C.H. (2020). Comparing university students' mobile learning practices in china. *International Journal on E-Learning: Corporate, Government, Healthcare, and Higher Education*, 19 (2), pp. 181-203.
- Ortiz, O., Alcover, P.M., Sánchez, F., Pastor, J.A., Herrero, R. (2015). "M-Learning Tools: The Development of Programming Skills in Engineering Degrees," in *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 10, no. 3, pp. 86-91, Aug. 2015, doi: 10.1109/RITA.2015.2452531.
- Perera, P., Tennakoon, G., Ahangama, S., Panditharathna, R., Chathuranga, B. (2021). "A Systematic Mapping of Introductory Programming Languages for Novice Learners," in *IEEE Access*, vol. 9, pp. 88121-88136, 2021, doi: 10.1109/ACCESS.2021.3089560.
- Pinheiro, A., Barbosa, A., Carvalho, R., Freitas, F., Tsai, Y., Gasevic, D., Ferreira, R. (2021). Automatic feedback in online learning environments: A systematic literature review. *Computers and Education: Artificial Intelligence*. 2. 100027. 10.1016/j.caeai.2021.100027.
- Sakibayev, S., Sakibayev, R. and Sakibayeva, B. (2019), The educational impact of using mobile technology in a database course in college", *Interactive Technology and Smart Education*, Vol. 16 No. 4, pp. 363-380 <https://doi.org/10.1108/ITSE-12-2018-0103>.
- Salleh, S.M., Shukur, Z., Judi, H.M. (2013). Analysis of Research in programming Teaching Tools: An Initial Review, *Procedia - Social and Behavioral Sciences*, Volume 103, 2013, Pages 127-135, ISSN 1877-0428.

- Sergio, M., German, L. (2020). Introduction to Programming Using Mobile Phones and MIT App Inventor. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*. PP. 1-1. 10.1109/RITA.2020.3008110.
- Sobral, S. (2021). Strategies on Teaching Introducing to Programming in Higher Education. 10.1007/978-3-030-72660-7_14.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., & Regnell, B. (2012). *Experimentation in Software Engineering*. Springer.
- Ypsilandis, G. (2002). Feedback in distance education. *Computer Assisted Language Learning*, 15, 167–181.
- Zimmerman, B. J. (2008). Investigating self-regulation and motivation: Historical background, methodological developments, and future prospects. *American Educational Research Journal*, 45(1), 166-183. <https://doi.org/10.3102/00028312073129>.

R. Célia Coelho is a professor at the Department of Science and Technology, Federal University of São Paulo (UNIFESP), Brazil. She received a B.Sc. degree in Computer Science from State University Julio de Mesquita Filho - UNESP in 1989, an M.Sc. degree in Computer Science from Federal University at São Carlos in 1993, and a Ph.D. degree in Computational Physics from the Institute of Physics, University of São Paulo, in 1998. She is one of the founders of the Biomedical Computing Laboratory in operation since 2012 at UNIFESP. Her current research interests include biomedical signal and image processing, computer graphics, simulation, and computer vision.

M. F. P. Marques is a master's student in Technological Innovation at the Federal University of São Paulo (UNIFESP), Brazil. He received a B.Sc. degree in Information Technology with Emphasis on Databases from the Faculty of Technology of the State of São Paulo – FATEC in 2009. He was an adjunct professor at the Faculty of Human Sciences of Cruzeiro, Brazil, acting in the areas of administration, accounting, and production engineering. He is currently a professor at the Technical School of São Paulo (ETEC), Brazil, teaching programming logic, computer maintenance, networks, and databases. He is also IT Technologist with an Emphasis on Databases and Education. His current research interests include programming languages and mobile tools in education.

T. Oliveira received a B.Sc. degree in Computer Science and a Ph. D. degree in Electrical Engineering both from State University Julio de Mesquita Filho - UNESP in 2002 and 2008, respectively. His research interests cover curriculum design, the development of educational technologies, and the proposal of educational approaches. Since 2017 he has dedicated efforts to design, elaborate and implement new ideas and student academic trajectories aimed to modernize the Computer Engineering curriculum. Besides that, he has been working with undergraduate and graduate students to analyze the state-of-the-art and propose a variety of software and system designs to come up with technological products that are suitable to address some educational needs and requirements. Some examples are: (1) the implementation of an artificial neural network to investigate the possibility of personalizing and at the same time automating the process of assigning scores to technical reports; (2) the project of an interactive web software aimed to teach and evaluate students about concepts of academic plagiarism, guiding the students to what they should do to avoid it; (3) the implementation of FPGA remote laboratories for teaching and learning of Digital Systems; (4) the design of an educational software to identify student's multiple intelligences and learning styles, contributing to their cognitive self-knowledge and providing specialized information for teachers about their students classes. Furthermore, he has been proposing and applying active methodologies structured in a combination of online peer assessment and scoring rubric techniques in some computer undergraduate and graduate courses that are bringing a more favorable educational environment for project designs, stimulating creativity and student motivation, influencing students to produce better technical reports - developing written communication skills - and a deeper theoretical and practical knowledge retention. Since 2009 he is currently an associate professor at Institute of Science and Technology / Federal University of São Paulo.