Empirical Evidence of the Usage of Programming Languages in the Educational Process

Mariuxi Vinueza-Morales, Diana Borrego, José A. Galindo, and David Benavides

***Abstract*—*Contribution:* A systematic literature review on the empirical evidence regarding the usage of programming languages for learning purposes is presented. The review analyzes different methods and tools at different educational levels and with different objectives.**

***Background:* Learning programming has gained relevance in the last decade. This is due to the massive presence of programmable elements ranging from computers to toys. Because of this, the interest of researchers on this topic has increased. Questions, such as what to use, in what educational stages to use it, the effectiveness of the method, and the focal objectives for learning programming are questions that do not have obvious answers.**

***Research Questions:* 1) What empirical evidence exists on the use of educational programming languages (EPLs)? 2) In what context is the research performed? 3) How is effectiveness reported in the literature after applying EPLs? 4) What pedagogical goals are achieved by using EPLs?**

***Methodology:* Following a formal protocol, automated searches were performed for primary studies from 2007 to 2018. A total of 62 studies were identified, of which 29 were selected and analyzed since they include some type of empirical evidence.**

***Findings:* After performing the evaluation, the results support the need for better approaches with empirical evidence when reporting research on the usage of EPLs. Some research opportunities are identified which concerns the used programming languages, the areas or stages of their application, or the need to have more empirical evidence in general and more studies in non-WEIRD (Western, educated, industrialized, rich, and democratic) contexts.**

***Index Terms*—Computational thinking (CT), educational programming language (EPL), programming learning.**

I. Introduction

THE EDUCATIONAL tools used to teach and learn computational thinking (CT) skills have attracted the attention of education researchers. This is because most students consider programming to be a difficult subject to learn, and they lack the motivation and interest to gain programming skills. The interest of researchers in subjects related to the learning process of programming leads us to ask questions, such as what pedagogical and research objectives should be the focus of studies or what method is most effective and at what educational level should it be used. Some questions can be answered by performing a literature review of the empirical studies that give the results obtained by applying different methods or tools in different stages or with different objectives. The evidence extracted from this literature review includes an overview of the methods chosen to improve the process of programming learning, the programming languages applied in each research paper, the pursued pedagogical goals, the performed methods for evaluating the effectiveness of programming tools, and the research objectives achieved in each study.

The reviewed literature shows that students have significantly improved their programming and development skills in CT with the use of EPLs as learning instruments. Mainly, this article focus on using block-based visual environments and textual programming languages, which are two of the five categories to develop CT mentioned by Moreno-León [30]. The widespread use of both aforementioned categories could indicate there is still no consensus among which one is the most appropriate to best address the learning of programming.

When evaluating the literature, it is observed that 23 of the 29 research objectives proposed, which is approximately 79% (see Table V), seek to address pedagogical goals. It is also detected that only 4 of 29 studies with empirical evidence focused on early educational stages (primary and kindergarten), representing approximately 14% of the studies. This might indicate that more empirical evidence is needed at these educational levels. It is noticed that two empirical strategies were mainly adopted: 1) case studies and 2) controlled experiments.

In addition, the pedagogical goal that is most mentioned by researchers is problem solving and the most used methods to assess the effectiveness of EPL are questionnaires, surveys, and interviews. However, the maturity level of these studies is relatively low, making it difficult for researchers to replicate the studies. Nevertheless, the vast majority of the papers report that they achieved the objective they pursued with their performed research.

Some research opportunities have been spotted after reading the primary studies. Concretely, they need: 1) to address in which thematic areas of the study program CT is incorporated; 2) to determine the extent to which different programs

Manuscript received February 3, 2020; revised June 24, 2020, September 15, 2020, and October 5, 2020; accepted October 9, 2020. This work was supported in part by EU FEDER Program; in part by the MINECO Project OPHELIA under Grant RTI2018-101204-B-C22; in part by the TASOVA Network under Grant MCIU-AEI TIN2017-90644-REDT; in part by the Junta de Andalucia METAMORFOSIS Project; in part by ECLIPSE Project under Grant RTI2018-094283-B-C33; and in part by the European Regional Development Fund (ERDF/FEDER). *(Mariuxi Vinueza-Morales, Diana Borrego, and José A. Galindo contributed equally to this work.) (Corresponding author: Mariuxi Vinueza-Morales.)*

Mariuxi Vinueza-Morales is with the Facultad Ciencias de la Ingeniería, Universidad Estatal de Milagro, 091050 Milagro, Ecuador (e-mail: mvinuezam@unemi.edu.ec).

Diana Borrego, José A. Galindo, and David Benavides are with the Department of Computer Languages and Systems, Universidad de Sevilla, 41012 Seville, Spain (e-mail: dianabn@us.es; jagalindo@us.es; benavides@us.es).

Digital Object Identifier 10.1109/TE.2020.3030588

are used to improve the learning of programming by relying on block-based languages; 3) to use different alternative programming languages in order to compare their respective; 4) to have more and better quality empirical evidence; 5) to develop more studies and empirical evidence in non-WEIRD (Western, educated, industrialized, rich and democratic [31]) contexts; and 6) to develop studies on the use of textual programming languages at early educational stages.

The remainder of this article is organized as follows. Section II presents the necessary background to understand the scope of this document. Section III covers the methodology used in this systematic literature review (SLR). Section IV presents the results of the analysis of the selected studies. Section V discusses the previous works in the literature in this research area. Section VI presents the research opportunities. Finally, the conclusions are drawn in Section VII.

II. Background

To provide context and introduce the research topic, this section presents existing educational programming tools (including EPLs and other educational environments), which can be used to teach and develop programming skills for children, youth, and adults. Most of the tools are based on programming languages designed with learning programming in mind and are generally based on blocks containing basic elements required in traditional programming.

One of the most common categories considered by researchers to develop CT is block-based visual environments. Also, AgentCubes,[1] Alice,[2] and AppInventor[3] are intuitive and visual programming environments that allow users to create 3-D games and fully functional applications.

It is important to highlight PiktoMir[4] and Scratch,[5] which are useful to learn basic programming principles in preschool and elementary school children. Also, Snap[6] is suitable for an advanced introduction to computer science for high school or college students. Other educational tools used to teach programming are Gidget,[7] which is a multiplatform game, and Jeliot,[8] which is a program viewing application.

III. SLR Process Definition

This study follows the Kitchenham *et al.* guidelines [32] for performing a literature review process. Fig. 1 shows the main steps to conduct a systematic literature review which are depicted as follows.

1) *Plan Review:* This phase includes the activities related to the organization of the development and validation of the protocol.
2) *Conduct Review:* This phase includes the execution of the review protocol and monitoring its progress.

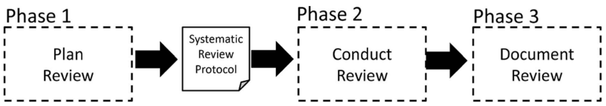

Fig. 1. Overview of the systematic review process [32].

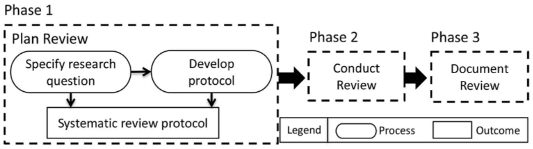

Fig. 2. Plan review phase [32].

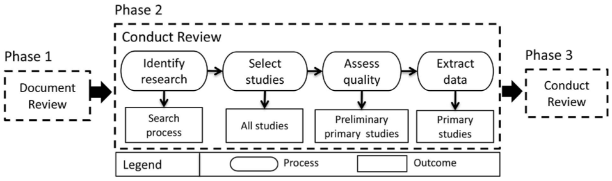

Fig. 3. Conduct review phase [32].

3) *Document Review:* This phase includes activities, such as documenting or reporting the study appropriately for the intended audiences.

Next, the process that guides this SLR is detailed.

A. Phase 1: Plan Review

In this first phase, the way in which the study will be carried out and what will be documented through the review protocol is designed. It involves the execution of two steps of the process, as shown in Fig. 2.

Next, each of these steps are detailed.

1) Specify Research Questions: The objective of this literature review is to answer the following research questions.

1) *RQ1:* What empirical evidence exists about the use of EPLs?
2) *RQ2:* In what context is the research performed?
3) *RQ3:* How is effectiveness reported in the literature after applying an EPL?
4) *RQ4:* What pedagogical goals are achieved by using EPLs?

2) Develop Protocol: In this phase, the systematic review protocol document is developed, which describes all the details on how the review will be carried out. The protocol presented in Kitchenham *et al.* [32] guided covers items, such as the background, research questions (see Section III-A1), search strategies, data extraction, databases, and inclusion/exclusion criteria (see Section III-B) used for the present study. The design of each phase of the protocol is explained in the next section.

B. Phase 2: Conduct Review

In this phase, the plan established in the systematic review protocol is implemented. Fig. 3 presents the activities to perform in this phase.

Next, the performance of each of these steps is detailed.

1) Identify Research: In this phase of the review, the databases to be queried are identified. Concretely, these

[1] Created by AgentSheets https://www.agentcubesonline.com.
[2] Developed at the Carnegie Mellon University https://www.alice.org/.
[3] Created by Google Labs http://appinventor.mit.edu/explore/index-2.html.
[4] Freely distributed https://piktomir.ru/.
[5] Created by MIT Media Lab https://scratch.mit.edu/.
[6] Extended reimplementation of Scratch http://snap.berkeley.edu/.
[7] http://www.helpgidget.org/
[8] http://cs.joensuu.fi/jeliot/

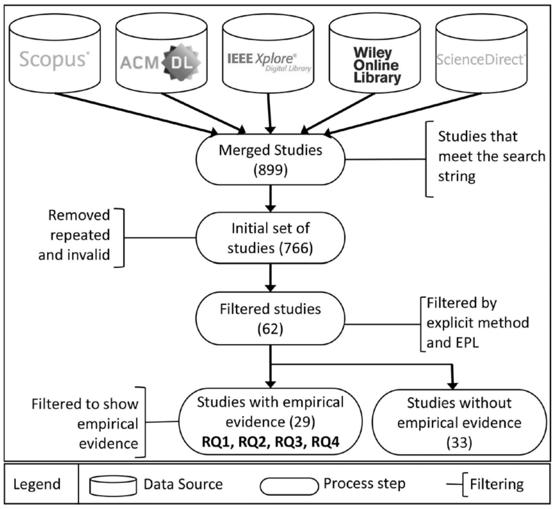
Scopus
ACM DL
IEEE Xplore Digital Library
Wiley Online Library
ScienceDirect
Merged Studies (899)
Studies that meet the search string
Removed repeated and invalid
Initial set of studies (766)
Filtered studies (62)
Filtered by explicit method and EPL
Filtered to show empirical evidence
Studies with empirical evidence (29) RQ1, RQ2, RQ3, RQ4
Studies without empirical evidence (33)
Legend
Data Source
Process step
Filtering

Fig. 4. Search citing publication process.

databases are Scopus,[9] ACM,[10] IEEE,[11] ScienceDirect,[12] and Wiley,[13] which is based on the fact that they are commonly consulted sources that contain the greatest amount of information. Note that the same databases as in the systematic literature review (SLR) study conducted by [33] that are common sources of bibliographic information in the research community have been used.

The database query was conducted in November 2018, and the query was as follows. (“Learning programming” OR “Learning coding” OR “Learning to program” OR “Learning to code” OR “Learning of computer programming”) AND (“Educational software” OR “Educational technology” OR “EPL” OR “Visual programming environment” OR “Block-based languages” OR “Block-based coding”).

Section VII presents the link that contains the database query used for each database. After executing the queries, 899 papers were retrieved from all databases.

2) Select Studies: Fig. 4 shows the process carried out to search for relevant studies.

Inclusion and Exclusion Criteria: After executing the database query, the retrieved studies were subject to an in-depth analysis. Below are the inclusion and exclusion criteria.

Inclusion Criteria: 1) papers published since January 2007 (Scratch development began in 2004 and it was finally released in 2007); 2) the type of document should be a “journal article” or “conference paper”; 3) papers written in English; 4) papers in the following disciplines: computer science, social science, education, and computing; 5) papers that explicitly state the method and the EPL used in the programming learning process; and 6) papers showing empirical evidence.

Exclusion Criteria: 1) papers whose studies did not apply to learning programming; 2) duplicated papers; and 3) invalid papers, i.e., no focus on applying an EPL for learning programming.

TABLE I
NUMBER OF PAPERS IDENTIFIED BY THE RESEARCH, INCLUSION, AND EXCLUSION CRITERIA

Database	#papers identified (1)	#papers removed (2)	#papers filtered (3)	#papers evidence (4)
Scopus	39	31	19	9
ACM	36	25	14	5
IEEE	12	8	2	1
Science Direct	438	390	22	11
Wiley	374	312	5	3
Total	899	766	62	29

Table I presents the number of papers identified in this review, considering the inclusion and exclusion criteria detailed in this section.

The description of the columns is as follows: 1) the number of studies identified when searching in each database using the search query presented in Section III-B1; 2) the number of papers deleted as duplicate or invalid for the research; 3) the number of filtered papers that explicitly state the method that was used in the learning process and EPL; and 4) the number of filtered studies that show empirical evidence of the learning procedure.

3) Assess Quality: First, it should be noticed that the applied inclusion and exclusion criteria, detailed above, are considered as a first quality indicator. It is used to evaluate the contributions by selecting the largest number of studies and making sure that they report research that provides empirical results of the applied methods. Second, and for the remaining research questions considered, the quality evaluation will be carried out later and not as a way to discard articles for analysis. This is because one of the *RQs* (specifically *RQ*2.2) deals with the evaluation of the quality of the reviewed papers; in addition, that is why the result of that assessment is not used to exclude papers from the present study since it is part of the analysis presented in Section IV.

4) Extract Data: Once the previous phases are completed, the data obtained in the last stage of phase two are classified and the following steps are taken.

1) A database is created to store all papers to be classified. Concretely, a bibtex database[14] is used. Later, it is managed using Mendeley.[15] The database records the following attributes for each article: a) the type of document; b) the title; c) the authors; d) the year of publication; e) a summary; and f) the link to access the full document.
2) By applying the selection criteria, 62 items are obtained as a result. A database in Excel is then created by adding the following information: a) the place where the research was conducted (country and educational institution); b) the applied method; c) the software used; d) the objective; and e) the results obtained according to the research questions.
3) From these 62 studies, those that present empirical results are separated, resulting in 29 articles. These latter articles form the second group of papers, and the following information is included for them: a) the educational context in which the method was applied and b) the

[9] http://www.scopus.com

[10] https://dl.acm.org/

[11] https://ieeexplore.ieee.org/Xplore/home.jspg

[12] https://www.sciencedirect.com/

[13] https://onlinelibrary.wiley.com/

[14] http://www.bibtex.org/

[15] https://www.mendeley.com/

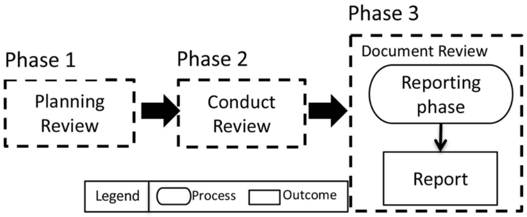

Fig. 5. Document review phase [32].

details of the method used according to the guidelines proposed by Wohlin *et al.* [34] (see Section IV-A).

C. Phase 3: Document Review

In this final phase of the systematic literature review, shown in Fig. 5, the classification and documentation of the information retrieved from the papers considered in the present study (see Section IV) are carried out. To perform this, the research questions defined in Section III-A1 are answered, and the results of the analysis of the 29 articles that present empirical evidence are reported.

IV. Results

This section provides answers to the research questions that were defined in Section III-A1.

A. RQ1: What Empirical Evidence Exists on the Use of EPLs?

In the context of the question, two situations were analyzed as follows: 1) the empirical methods described in the literature and 2) the criteria that were established to determine the quality of the method and to report the results. First, the empirical methods that have been described in the literature and have been used for the learning process of programming were identified. Therefore, to answer this question, the group of 29 papers that presented empirical evidence is used. Some studies use more than one method.

1) *Case Study:* The papers in this category use the case study as an empirical research method, e.g., Runeson *et al.* [35], who investigates an instance of a contemporary phenomenon within its real-life context. The articles in this category are [1]–[15].
2) *Experiment:* The papers in this category use the experiment as an empirical research method, e.g., Wohlin *et al.* [34], who has the goal of manipulating one or more variables to perform statistical analysis. The articles in this category are [16]–[25].
3) *Others:* The papers in this category use other research methods, such as courses or workshops, to apply EPLs in the programming learning process.

It is observed that the most used method in the literature is the *case study* method with 15 papers representing 47% of the total, which is followed by *experiments* with ten items representing 31% of the total. Additionally, the authors of seven papers chose other methods, such as workshop and courses, representing 22% of the total. Based on these results, it can be inferred that researchers consider that case studies are easier to plan and more realistic whereas experiments try to isolate the studied situation from the real-world and are more tedious.

TABLE II
Criteria to Determine the Quality of the Case Study

#	Topic	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
	Introduction	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
1	Problem statement	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	Research objectives	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	Context	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Case study design	4	4	4	3	4	3	2	2	3	3	1	1	1	4	2
4	Research questions	1	1	1		1				1					1	
5	Case and subject selection	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	Data collection procedure	1	1	1	1	1	1		1	1	1				1	
7	Analysis procedure	1	1	1	1	1	1	1			1				1	1
8	Validity procedure															
	Results	3	0	1	1	1	0	1	0	2	2	0	0	0	1	0
9	Case and subject descriptions	1			1			1								
10	Subsection	1		1		1				1	1				1	
11	Evaluation of validity	1								1	1					
	Conclusions and future work	2	2	2	2	3	2	2	0	1	1	1	2	1	1	1
12	Summary of findings	1	1	1	1	1	1	1		1	1	1	1	1	1	1
13	Relation to existing evidence															
14	Impact/implications					1										
15	Future work	1	1	1	1	1	1	1					1			
	Total	12	9	10	9	11	8	8	5	9	9	5	6	5	9	6

Second, the criteria considered when reporting the findings are analyzed. To evaluate the methods, both the structures of the reports for the experiment and case study proposed by Wohlin *et al.* [34] were considered. Each criterion was a binary variable with a value of 1 if the corresponding information was reported and 0 otherwise.

The following aspects correspond to the case study.

1) *Introduction:* Regarding this topic, the definition of the problem statement, research objectives, and context in which the study was conducted were considered to be part of the introduction.
2) *Case Study Design:* It includes the definition of the research questions, case and subject selection, data collection procedure, analysis procedure, and/or validity assessment procedure.
3) *Results:* It entails case and subject descriptions covering execution, analysis and interpretation issues, sections that may be structured, and/or evaluation of validity.
4) *Conclusions and Future Work:* They are composed of a summary of the findings, relations to existing evidence, impact/implications, limitations, and/or future work.

Table II presents the different aspects and identifies 15 studies where the *Case study* method was applied. In 100% of these studies, all the information of the *Introduction* was described. In 55% of the studies, the *Case study design* was reported; only 27% of the studies present information regarding the *Results* of the research. Approximately 38% of the papers present information on the *Conclusions and future work*. Chao [1] and Navarrete [5] are the ones who have reported the most complete studies.

The aspects considered for the experiment are shown as follows.

1) *Motivation:* This item considers if the problem statement, research objectives, and context in which the study was carried out were defined.
2) *Experimental Design:* It includes the hypotheses, design, subjects, objects, instrumentation, data collection procedure, analysis procedure, and evaluation of the validity.
3) *Execution:* It entails the sample, preparation, data collection performed, and validity testing procedure.

TABLE III
CRITERIA TO DETERMINE THE QUALITY OF THE EXPERIMENT

#	Topic	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]
	Motivation	3	3	3	3	3	3	3	3	3	3
1	Problem statement	1	1	1	1	1	1	1	1	1	1
2	Research objectives	1	1	1	1	1	1	1	1	1	1
3	Context	1	1	1	1	1	1	1	1	1	1
	Experimental design	4	3	4	7	5	3	4	3	4	2
4	Hypothesis	1			1	1				1	
5	Design	1		1	1		1	1	1		1
6	Subjects	1	1		1	1	1	1	1	1	1
7	Objects	1	1	1	1	1		1	1	1	
8	Instrumentation		1	1	1	1				1	
9	Data collection procedure				1	1	1	1			
10	Analysis procedure			1	1						
11	Evaluation of validity										
	Execution	2	2	1	3	1	1	1	2	1	1
12	Sample	1	1		1	1	1	1	1	1	1
13	Preparation	1	1		1				1		
14	Data collection performed			1	1						
15	Validity procedure										
	Analysis	0	1	0	3	2	2	1	2	1	2
16	Descriptive statistics		1		1	1	1	1	1	1	1
17	Data set reduction				1	1		1		1	
18	Hypothesis testing				1		1				
	Interpretation	2	1	1	4	3	1	1	1	2	1
19	Evaluation of results and implications	1	1	1	1	1	1	1	1	1	1
20	Limitations of study	1			1	1				1	
21	Inferences				1	1					
22	Lesson learn				1						
	Conclusions and future work	2	2	2	3	2	2	2	1	2	1
23	Conclusions	1	1	1	1	1	1	1	1	1	1
24	Relation to existing evidence										
25	Impact										
26	Limitations				1						
27	Future work	1	1	1	1	1	1	1		1	
	Total	13	12	11	23	16	12	13	11	14	9

4) *Analysis:* It contains the descriptive statistics, dataset reduction, and hypothesis testing.
5) *Interpretation:* It involves the evaluation of the results and implications, limitations of the study, inferences, and lessons learned.
6) *Conclusions and Future Work:* Finally, the conclusions are related to the existing evidence, impacts, limitations, and future work.

Table III identifies ten authors who performed experiments in their research. From the criteria established to evaluate the quality of the studies, the aspects where more information has been reported corresponds to the *Motivation* with 100% of the papers. Following this, 49% of the papers reported on the *Experimental design*, 47% reported on the *Analysis*, 43% reported on the *Interpretation*, and finally 38% presented information about the *Execution* and the *Conclusions and future work*. With respect to the authors, Feng and Chen [19] is the one that has reported more information about their performed study, followed by Lee and Ko [20] and Wang *et al.* [24]. The remaining authors have not reported much information.

In general, it is observed that there is no consistency in what has been reported and that there is a lack of maturity in the subject. Also, there is a lack of having a complete report that encourages discipline in writing the results. It is necessary to know the details to be able to replicate the respective works in similar environments and to be able to detect if the reader can clearly and easily determine what is being studied and how the results presented were obtained.

B. RQ2: In What Context Is the Research Performed?

To answer this question, the context in which the empirical research has been carried out is analyzed, including the continent and the educational level of the study participants, and only the 29 articles that present empirical evidence were considered.

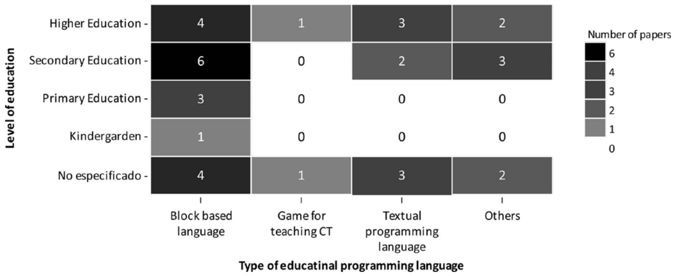

Fig. 6. Visualization of the systematic map.

In general, it is observed that the largest number of papers have been published by European organizations, summing up 13 articles ([3], [6]–[10], [16], [18], [26], [27], and [23]–[25]). This is followed by North America with five articles ([2], [5], [13], [20], and [28]) and Asia with four articles ([1], [14], [19], and [29]). However, the authors of seven papers have not specified where the research was conducted.

It looks like this kind of study is mainly carried out in Europe because researchers count on more information and resources. However, it would be necessary to analyze the factors that influence the research at one site or another and what research opportunities originated.

In addition, the educational stages have been considered and established according to the organization responsible for education and professional training in Spain, to classify the 29 articles. The largest number of empirical studies were conducted on secondary education (12–18 years old) with nine studies ([2], [3], [5], [7], [8], [13], [15], [21], and [28]); secondary education is followed closely by higher education (over 18 years old) with eight studies ([9], [14], [16], [18], [25]–[27], [29]). This could be because students over 18 may have some previous experience in programming concepts, which would make it easier to perform the experiment or the case study at higher levels.

However, few studies have been conducted on primary education (6–12 years old) with three studies ([6], [10], and [19]) and one study was conducted on kindergarten (3–6 years old) ([23]). Likewise, there are eight papers ([1], [4], [11], [12], [17], [20], [22], and [24]) where the educational stage in which the study was conducted was not specified.

Fig. 6 depicts the gaps in the research by identifying the educational levels at which the empirical studies are carried out and the type of EPL used.

C. RQ3: How Is Effectiveness Reported in the Literature After Applying EPL?

In the context of this question, the way in which the authors report on the effectiveness after the application of the EPLs is identified. To answer this, only the 29 articles that present empirical evidence are considered. In [6], two alternatives for effectiveness evaluation are reported while the rest of the authors only reported one. A distribution of these activities is shown in Table IV.

TABLE IV
ACTIVITY FOR REPORTING THE EFFECTIVENESS OF THE EPL

Activity	Citation	#Papers
Data collection instruments	[18], [26], [6], [7], [9], [29], [10], [11], [12], [14], [15], [25]	12
Assessment	[28], [19], [20], [21], [4], [23], [24], [13]	8
Exercise development	[17], [1], [3], [5], [8], [22]	6
Evaluation of a final project	[16], [27], [2], [6]	4

1) *Data Collection Instrument:* For this category, the contributions that used: a) questionnaires; b) interviews; and c) surveys as data collection techniques to know the opinions of students after the use of EPL were considered.
2) *Assessment:* Papers in this category include activities, such as: a) programming tests and b) self-assessments to determine achievements in understanding and learning the programming concepts of the participants.
3) *Exercise Development:* The papers in this category analyzed the skills obtained in developing programming exercises to reach the learning objectives proposed in the application of an EPL.
4) *Evaluation of Final Project:* The papers in this category examined the projects designed and developed by the participants to evaluate the learning of programming concepts.

Of the activities mentioned in Table IV, the most commonly used activity to determine the effectiveness of the application of an EPL is the use of data collection instruments. This was the technique used in 12 articles of the literature; it has been concluded that questionnaires, interviews, and surveys are considered to be valid, reliable, and practical basic mechanisms to obtain data.

D. RQ4: What Pedagogical Goals Are Achieved by Using EPLs?

This question aims to determine the pedagogical goals in the primary studies. To classify the pedagogical goals according to the acquisition of programming skills, both the 29 previously mentioned papers and some others mentioned below were analyzed to find existing classifications.

First, bloom's taxonomy of educational objectives [36] was created in 1956 to promote high forms of thinking in education. However, regarding the present literature review, this classification is very generic since some skills that must be acquired when learning programming languages do not fit exactly in one category, but rather they fit in several.

It might be better to provide a classification more in line with the skills that might be acquired when learning programming. To define the aforementioned classification, a detailed analysis and discussion of previous computer science education research were used. In addition to the 62 papers analyzed in this SLR, Kurland *et al.* [37] performs an empirical study identifying the lack of information on both the achieved skills and the methodology students follow to achieve them. The proposal in [38] carries out a literature review that aims to answer questions to uncover the pedagogical content knowledge of informatics education. In addition, the contribution in [39] analyzes many of the factors involved in the learning process of programming. Considering all the research, the following skills, among others, to be acquired were identified: 1) algorithmic thinking that involves sequencing, analysis, and testing processes in time and space; 2) creative play, innovation, and exploration through entertainment, communication, and social applications; 3) application of the creativity and acquired skills to solve problems; 4) mastery of the fundamental concepts and practices; and 5) teamwork and building supportive partnerships.

Likewise, in addition, motivation factors are necessary when dealing with the issues associated with the fundamentals of programming within practical scenarios.

Therefore, the final classification would be the following, which are the five high-level pedagogical objectives in the context of this systematic literature review (SLR).

1) *Problem Solving:* Use of a structured problem-solving process to help address new problems, see the challenges as problems that can be solved, and be able to break down larger problems into smaller elements. The papers classified in this category ([1]–[6], [8]–[13], [15], [20]–[23], [25], [28], and [29]) pose that the resolution of computational problems helps develop skills for learning programming as a pedagogical goal. This category includes items, such as: a) problem-based learning (PBL); b) understanding CT; c) computer programming training; and d) identification of the logic of a program.
2) *Persistence:* Consider that mistakes are a productive and natural part of problem solving, stay firm and constant in the search for solutions even if setbacks occur, and perform iterations to improve partial solutions. The papers in this category ([7], [16]–[19], [24], and [27]) propose significantly increasing the constancy of the participants in the learning and improving the understanding of the programming concepts as a pedagogical goal. This category includes items, such as: a) increased programming performance; b) improved programming learning; c) self-regulation; and d) increased retention.
3) *Creativity:* Conduct the work considering your own interests or ideas, consider various possible approaches and experiment with new ideas, and build, extend and even improve upon the ideas or projects of colleagues. The papers in this category ([5], [13], and [25]) set the creative thinking of students to understand programming concepts with an imaginative view as a pedagogical goal. This category includes items, such as: a) creativity and innovation and b) sustained systematic reasoning.
4) *Collaboration:* Work together with others to develop better solutions that incorporate everyone's work; help the team, even by mediating disagreements to get a common solution; and actively contribute to successful group projects. The papers in this category ([2], [5], [25], [26], and [28]) set working collaboratively to help participants develop common solutions by applying programming concepts as a pedagogical goal.
5) *Communication:* Structure and document the work done so that others can easily understand it, be empathetic to the audience regarding their level of knowledge when

presenting the work, and accept and provide constructive feedback to improve the work. The papers in this category ([5] and [18]) set communication as a way to develop a sense of trust and confidence within the participants in their programming learning process as a pedagogical goal.

Motivation is a subject mentioned in the literature that affects the attainment of the five aforementioned goals. In [26], the results of interviews with students on the learning methods used establish that motivation is a problem that is partially related to the learning method. Likewise, Nikou and Economides [6] mentioned that computer science integrates all disciplines of engineering; therefore, all the students need to be motivated to learn the basic principles of computer science. However, it is quite difficult to activate and maintain students' motivation to learn to program. Also, in the results of their studies, Nikou and Economides [6], Reardon and Tangney [9], and Rozali and Zaid [29] established that through the implementation of PBL, students develop problem-solving skills and are motivated to learn programming.

E. Summary of Results

Table V shows a summary of the primary sources with empirical evidence (29 papers) together with all the data extracted in the review process. The columns represent the authors classified by the empirical method used in the research (see Section IV-A).

First, the EPLs mentioned in the literature are identified. Some authors used more than one EPL in their research. Among the languages most used by the authors, Scratch is used approximately 24% of the time and is followed by AppInventor that is used approximately 11% of the time. This can happen because Scratch is an easily accessible and very useful tool to get started in programming. The languages used are classified as follows.

1) *Block-Based Language:* The papers classified in this category use programming languages that allow one to visually program a sequence of commands that break blocks or pieces of code that fit together.
2) *Textual Programming Language:* The papers classified in this category use languages in which it is necessary to write commands with a specific syntax.
3) *Games for Learning CT:* The papers classified in this category use computer games for CT development.
4) *Other Educational Tools:* The papers classified in this category use tools different from the previously mentioned tools to learn programming.

From among the categories mentioned, it is observed that most of the studies use *block-based languages* for learning programming (eight contributions), followed closely by *textual programming languages* (seven papers). Therefore, there is not enough evidence to determine which type of EPL dominates the learning processes. From the data obtained, it seems that this could be due to the context and objectives of the study since some textual languages serve one purpose and visual languages serve another, although it is often argued that visual languages serve to introduce the textual programming learning by developing skills, such as creativity, problem-solving abilities or teamwork.

In addition, Table V also shows the research objectives addressed by the authors (some of them have more than one). Each objective is explained as follows.

1) *Address Pedagogical Goals:* The papers categorized as this objective use some EPL as an alternative to conventional programming languages (for more information, see Section IV-D).
2) *Evaluate an EPL or Technique Used:* The papers categorized as this goal evaluate the use of EPLs for their inclusion in the study program.
3) *Increase Motivation in Computer Science:* The papers classified as this objective are intended to promote the positive attitudes of students toward computers.
4) *CT Across the Study Program:* The papers in this category seek to identify the benefits of the cross-implementation of EPL in the study program.

From the aforementioned objectives, it is noticed that the most important aspect considered by the authors is to *address pedagogical goals*. This means that in 23 of the 29 articles presented in this investigation (approximately 79%), the authors seek to meet the pedagogical goal by using an EPL in the programming learning process. This may be because most people working in this area have teaching as their vocation, and therefore they are looking for students to improve their grades and reach the desired skills and knowledge according to their level.

Regarding the pedagogical goals, the percentage of compliance with the quality criteria of the empirical methods and the educational level observed in the table, they are analyzed in Section IV and specifically in Sections IV-A, IV-B, and IV-D, respectively.

V. Related Work

In this section, the studies and the authors who conducted literature reviews [33], [40]–[45], are depicted and shown in Table VI. The period used in the reviews carried out by the authors varies according to the context of the study. Likewise, the number of contributions analyzed by each author is also shown, as well as a comparison about the research questions made in this study and in the reviews done by the authors.

The comparative analysis was performed with respect to four aspects that make up the scope of this research and that correspond to the chosen method ($RQ1$), the research context ($RQ2$), the effectiveness of EPLs ($RQ3$), and the pedagogical goals ($RQ4$).

Regarding the empirical method selected for the research ($RQ1$), the comparative table shows that Costa and Miranda [40], Lye and Koh [41], and Moreno-León and Robles [33] reported the methods used in their publications. Nevertheless, it should be pointed out that the evaluation of the quality of the method (also considered in this question) is only presented in this article, which can be taken as a reference so that the authors of subsequent publications may consider the inclusion of this topic in their future research.

Concerning the research context ($RQ2$), Lye and Koh [41], Medeiros *et al.* [42], and Noone and Mooney [43] do not report

TABLE V
SUMMARY OF EPL AND OBJECTIVES

	EXPERIMENT										CASE STUDY															OTHERS			
	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[28]	[26]	[27]	[29]
BLOCK-BASED LANGUAGE																													
Agent Cubes																							✓						
Alice																								✓					
App Inventor							✓									✓	✓	✓											
Erlang																												✓	
Piktomir								✓																					
Scratch				✓		✓				✓		✓		✓		✓			✓	✓	✓								
Snap																										✓			
Visual problem solving											✓																		
TEXTUAL PROGRAMMING LANGUAGE																													
Action Script															✓														
Adobe Flash CS6																													✓
Java									✓										✓										
Language C													✓									✓							
Learning Objects																						✓							
Logo																											✓		
Python					✓																								
GAME FOR LEARNING																													
Gidget					✓																								
Second life		✓																											
OTHER EDUCATIONAL TOOLS																													
Flash design															✓														
Jeliot		✓							✓																				
LECGO													✓																
PILet	✓																												
RUR-PLE																									✓				
Unity																								✓					
RESEARCH OBJECTIVES																													
Address pedagogical goals	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	✓			✓	✓	✓	✓
Evaluate an EPL						✓		✓												✓	✓				✓				
Increase motivation					✓											✓				✓				✓			✓		✓
CT in study program																				✓						✓			
PEDAGOGICAL GOALS																													
Problem solving					✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓			✓
Persistence	✓	✓	✓	✓					✓								✓											✓	
Collaboration										✓		✓			✓											✓			
Creativity										✓					✓								✓						
Communication			✓												✓														
Quality criteria (%)	48	44	41	85	59	44	48	41	52	33	80	60	67	60	73	53	53	33	60	60	33	40	33	60	40				
Educational level	H		H	P		S		K		H		S	S		S	P	S	S	H	P			S	H	S	S	H	H	H

H:Higher education S:Secondary education P:Primary education K:Kindergarten

TABLE VI
COMPARISON OF RELATED REVIEWS

Review work	Year	Period	#papers studied	RQ1	RQ2	RQ3	RQ4
Costa *et al.* [40]	2016	2000–2014	6	✓	✓		
Lye *et al.* [41]	2014	2009–2013	27	✓			
Medeiros *et al.* [42]	2018	2010–2016	89				
Moreno *et al.* [33]	2016	2007–2015	15	✓	✓		
Noone *et al.* [43]	2017	2007–2015	15				
Salleh *et al.* [44]	2013	2005–2011	45		✓		✓
Ting *et al.* [45]	2018	2006–2017	120		✓		
This paper	2018	2007–2018	62	✓	✓	✓	✓

the country and the educational stage where their studies were conducted.

Finally, in relation to the analysis of achieved pedagogical goals (*RQ*4), it is found that only Salleh *et al.* [44] reported their pedagogical goals. It should be pointed out that the effectiveness (*RQ*3) is only presented in this article, which can be taken as a reference so that the authors of future publications may consider the inclusion of this topic in their research.

Regarding the analysis of the authors presented in Table VI, it can be seen that only Medeiros *et al.* [42], as well as the present study, made a comparison with results of previous studies. Regarding the results reported by the authors, through the Systematic literature review (SLR), it is concluded that the use of an EPL has a positive impact on student performance [40], [41], the use of an EPL results in a better understanding and resolution of programming problems [42], the choice of the first programming language has or does not have a profound effect on student performance and interest [43], and what is the effectiveness of the techniques and tools used by Salleh *et al.* [44] and Ting-Chia *et al.* [45]. Likewise, this study shows that the use of EPLs helps the development of skills for programming learning and CT. Additionally, the methods and techniques that the authors have used have been collected, which for the most part agree with the conclusions of the authors in Table VI.

VI. RESEARCH OPPORTUNITIES

Based on the research questions (see Section III-A1), some ideas for contributing to the research are detailed as follows.

1) A low number of reports that provide empirical evidence and are critical to guiding and promoting the research in this field have been found. Reporting studies with a clear empirical strategy can help practitioners to improve the usage of programming languages in the educational process.
2) Latin American countries are underrepresented among the reviewed works according to this study. However, this can be seen as more of an opportunity than a problem. The lack of studies in non-WEIRD contexts [46], where students from the rural sector face significant challenges due to their low academic preparation, geographic isolation, and poor socioeconomic conditions [47], [48], can motivate further research in this topic. These studies would soften the bias of only performing studies on similar countries.
3) There are few studies showing why textual programming languages are not well suited for learning programming. More efforts to learn programming in early educational stages (kindergarten and primary education) are required to understand the difficulties that students have and how to overcome those in both the early and later stages of the educational process.
4) In this review, there is no unique evaluation method. Additionally, there are no clear pros and cons of each method. Having a catalog of methods can help researchers to and discuss and improve these methods.
5) In some authors' research, CT is incorporated transversally in the curricula, mainly at the secondary level. However, it is unclear in which thematic areas of the curriculum CT is being used. This information can be useful for curriculum guideline creators who can then consider where CT is more useful and why.
6) It is still unclear to what extent different programs are used for learning programming relying on block-based languages. Additionally, does the syntax of these languages affect the outcomes of the learning process? Such questions can encourage and promote the research on the topic while helping to understand which types of languages are best suited for each type of student.

VII. Conclusion

This work presented a literature review of scientific publications on the use of EPLs for learning programming. The pedagogical goals to be achieved through the application of EPLs, the method to evaluate the effectiveness of the tools, the main research objectives that are sought in each investigation in recent years, and the time of the contributions were identified.

Through the performed SLR, it is verified that most of the studies have been conducted at the secondary level by applying an EPL considering that it has had a positive impact on the development of programming learning skills. In addition, the results revealed that command and block-based programming languages facilitate learning and allow the development of other types of skills and abilities besides programming.

Specifically, certain research opportunities were identified, such as the need to: 1) detect in what thematic areas of the study program CT is used; 2) use different alternatives to block-based and textual languages in order to compare the results; 3) have more and better quality empirical evidence; 4) develop more studies and empirical evidence in non-WEIRD nations; and 5) develop studies regarding the use of textual programming languages in early educational stages. The objective of this study was to guide new research using these most important findings that identify future research by analyzing a specific language, such as *Alice* to establish the impact on programming learning by performing controlled experiments at a high educational level.

Materials

The results of searching in digital libraries and processing these searches were stored in the Mendeley bibliography system. To access to these results, review https://doi.org/10.5281/zenodo.3569716.

Acknowledgment

The authors thank Dr. J. Moreno for his valuable comments on early versions of this work.

References

[1] P.-Y. Chao, "Exploring students' computational practice, design and performance of problem-solving through a visual programming environment," *Comput. Educ.*, vol. 95, pp. 202–215, Apr. 2016.

[2] Y. Kafai and V. Vasudevan, "Hi-lo tech games: Crafting, coding and collaboration of augmented board games by high school youth," in *Proc. ACM IDC*, 2015, pp. 130–139.

[3] M. Kordaki, "A drawing and multi-representational computer environment for beginners' learning of programming using C: Design and pilot formative evaluation," *Comput. Educ.*, vol. 54, no. 1, pp. 69–87, 2010.

[4] A. Merkouris, K. Chorianopoulos, and A. Kameas, "Teaching programming in secondary education through embodied computing platforms: Robotics and wearables," *ACM Trans. Comput. Educ.*, vol. 17, no. 2, pp. 1–22, May 2017.

[5] C. C. Navarrete, "Creative thinking in digital game design and development: A case study," *Comput. Educ.*, vol. 69, pp. 320–331, Nov. 2013.

[6] S. A. Nikou and A. A. Economides, "Transition in student motivation during a scratch and an app inventor course," in *Proc. EDUCON*, 2014, pp. 1042–1045.

[7] A. O. Garcacia, A. Ruiz-Martínez, and R. Valencia-García, "Using app inventor for creating apps to support *m*-learning experiences: A case study," *Comput. Appl. Eng. Educ.*, vol. 26, no. 3, pp. 431–448, 2018.

[8] S. Papadakis and V. Orfanakis, "The combined use of lego mindstorms NXT and app inventor for teaching novice programmers," *Adv. Intell. Syst. Comput.*, vol. 560, pp. 193–204, Mar. 2017.

[9] S. Reardon and B. Tangney, "Smartphones, studio-based learning, and scaffolding: Helping novices learn to program," *ACM Trans. Comput. Educ.*, vol. 14, no. 4, pp. 1–15, 2014.

[10] J.-M. Sáez-López, M. Román-González, and E. Vázquez-Cano, "Visual programming languages integrated across the curriculum in elementary school: A two year case study using scratch in five schools," *Comput. Educ.*, vol. 97, pp. 129–141, Mar. 2016.

[11] E. Tanrikulu and B. C. Schaefer, "The users who touched the ceiling of scratch," in *Proc. WCETR*, vol. 28, 2011, pp. 764–769.

[12] G. Tuparov, D. Tuparova, and V. Jordanov, "Teaching sorting and searching algorithms through simulation-based learning objects in an introductory programming course," in *Proc. WCES*, vol. 116, 2014, pp. 2962–2966.

[13] D. C. Webb, "Troubleshooting assessment: An authentic problem solving activity for it education," in *WCLTA*, vol. 9, 2010, pp. 903–907.

[14] L. S. Woei, I. H. Othman, and C. K. Man, "Learning programming using objects-first approach through folktales," *Jute Fabrics J.*, vol. 75, no. 3, pp. 47–53, 2015.

[15] I. Yoon, J. Kim, and W. Lee, "The analysis and application of an educational programming language (RUR-PLE) for a pre-introductory computer science course," *J. Crohn's Colitis*, vol. 19, no. 1, pp. 529–546, 2016.

[16] B. Alshaigy, S. Kamal, F. Mitcchell, C. Martin, and A. Aldea, "PILeT: An interactive learning tool to teach python," in *Proc. WiPSCE*, Nov. 2015, pp. 76–79.

[17] M. Ben-Ari *et al.*, "A decade of research and development on program animation: The Jeliot experience," *J. Vis. Lang. Comput.*, vol. 22, no. 5, pp. 375–384, 2011.

[18] M. Esteves, B. Fonseca, L. Morgado, and P. Martins, "Improving teaching and learning of computer programming through the use of the second life virtual world," *Brit. J. Educ. Technol.*, vol. 42, no. 4, pp. 624–637, 2011.

[19] C.-Y. Feng and M.-P. Chen, "The effects of goal specificity and scaffolding on programming performance and self-regulation in game design," *Brit. J. Educ. Technol.*, vol. 45, no. 2, pp. 285–302, 2014.

[20] M. J. Lee and A. J. Ko, "Comparing the effectiveness of online learning approaches on CS1 learning outcomes," in *Proc. ACM ICER*, 2015, pp. 237–246.

[21] A. Merkouris and K. Chorianopoulos, "Introducing computer programming to children through robotic and wearable devices," in *Proc. ACM WiPSCE*, 2015, pp. 69–72.

[22] F. Rahman, "Leveraging visual programming language and collaborative learning to broaden participation in computer science," in *Proc. SIGITE*, 2018, pp. 172–177.

[23] I. Rogozhkina and A. Kushnirenko, "PiktoMir: Teaching programming concepts to preschoolers with a new tutorial environment," in *Proc. WCETR*, vol. 28, 2011, pp. 601–605.

[24] P. Wang, R. Bednarik, and A. Moreno, "During automatic program animation, explanations after animations have greater impact than before animations," in *Proc. ACM Koli*, 2012, pp. 100–109.

[25] E. Yukselturk and S. Altiok, "An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming," *Brit. J. Educ. Technol.*, vol. 48, no. 3, pp. 789–801, 2017.

[26] R. Horváth and S. Javorský, "New teaching model for java programming subjects," in *Proc. WCES*, vol. 116, 2014, pp. 5188–5193.

[27] F. Huch, "Learning programming with Erlang," in *Proc. ACM ERLANG*, 2007, pp. 93–99.

[28] A. Feng, M. Gardner, and W.-C. Feng, "Parallel programming with pictures is a Snap!" *J. Parallel Distrib. Comput.*, vol. 105, pp. 150–162, Jul. 2017.

[29] N. F. Rozali and N. M. Zaid, "Code puzzle: Actionscript 2.0 learning application based on problem based learning approach," in *Proc. ISPC*, Jan. 2017, pp. 1–4.

[30] J. Moreno-León, "On the development of computational thinking skills in schools through computer programming with scratch," Ph.D. dissertation, Escuela Internacional de Doctorado, Universidad De Murcia, Murcia, Spain, 2018.

[31] A. Mesoudi, K. Magid, and D. Hussain, "How do people become w.e.i.r.d.? Migration reveals the cultural transmission mechanisms underlying variation in psychological processes," *PLoS ONE*, vol. 11, no. 1, pp. 1–17, Jan. 2016.

[32] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*. London, U.K.: CRC Press, 2015.

[33] J. Moreno-León and G. Robles, "Code to learn with scratch? A systematic literature review," in *Proc. EDUCON*, Apr. 2016, pp. 150–156.

[34] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, and A. Wessln, *Experimentation in Software Engineering*. Cham, Switzerland: Springer, 2012.

[35] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case Study Research in Software Engineering—Guidelines and Examples*. Hoboken, NJ, USA: Wiley, Feb. 2012.

[36] B. S. Bloom, M. B. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, *Taxonomy of Educational Objectives. The Classification of Educational Goals. Handbook 1: Cognitive Domain*. New York, NY, USA: Longmans Green, 1956.

[37] D. M. Kurland, R. Pea, C. Clement, and R. Mawby, "A study of the development of programming ability and thinking skills in high school students," *J. Educ. Comput. Res.*, vol. 2, no. 4, pp. 429–459, Jan. 1986.

[38] M. Saeli, J. Perrenet, W. Jochems, and B. Zwaneveld, "Teaching programming in secondary school: A pedagogical content knowledge perspective," *Informat. Educ.*, vol. 10, pp. 73–88, Apr. 2011.

[39] D. Smith, P. Lameras, and N. Moumoutzis, "Using educational programming languages to enhance teaching in computer science," in *Proc. EDGE Conf. Use Technol. K-12 Post Secondary Educ.*, Oct. 2010.

[40] J. M. Costa and G. L. Miranda, "Relation between Alice software and programming learning: A systematic review of the literature and meta-analysis," *Brit. J. Educ. Technol.*, vol. 48, no. 6, pp. 1464–1474, 2017.

[41] S. Y. Lye and J. H. L. Koh, "Review on teaching and learning of computational thinking through programming: What is next for K-12?" *Comput. Human Behav.*, vol. 41, pp. 51–61, Dec. 2014.

[42] R. P. Medeiros, G. L. Ramalho, and T. P. Falcao, "A systematic literature review on teaching and learning introductory programming in higher education," *IEEE Trans. Educ.*, vol. 62, no. 2, pp. 77–90, May 2019.

[43] M. Noone and A. Mooney, "Visual and textual programming languages: A systematic review of the literature," 2017. [Online]. Available: arxiv.abs/1710.01547.

[44] S. M. Salleh, Z. Shukur, and H. M. Judi, "Analysis of research in programming teaching tools: An initial review," *Procedia Soc. Behav. Sci.*, vol. 103, pp. 127–135, Nov. 2013.

[45] H. Ting-Chia, C. Shao-Chen, and H. Yu-Ting, "How to learn and how to teach computational thinking: Suggestions based on a review of the literature," *Comput. Educ.*, vol. 126, pp. 296–310, Jul. 2018.

[46] A. Luxton-Reilly *et al.*, "Introductory programming: A systematic literature review," in *Proc. ACM ITiCSE*, 2018, pp. 55–106.

[47] S.-y. Byun, J. L. Meece, and M. J. Irvin, "Rural-nonrural disparities in postsecondary educational attainment revisited," *Amer. Educ. Res. J.*, vol. 49, no. 3, pp. 412–437, 2012.

[48] T. R. Morton, N. A. Ramirez, J. L. Meece, C. Demetriou, and A. T. Panter, "Perceived barriers, anxieties, and fears in prospective college students from rural high schools," *High School J.*, vol. 101, no. 3, pp. 155–176, 2018.

Mariuxi Vinueza-Morales received the master's degree in business administration and management from the Universidad Tecnológica Empresarial de Guayaquil, Guayaquil, Ecuador, in 2008. She is currently pursuing the Ph.D. degree in computer engineering with the University of Seville, Seville, Spain.

She has worked in software development in both the public and private sectors. She has been an Associate Professor with the University of Milagro, Milagro, Ecuador, since 2001. Her research interests include educational programs and development methodologies applied to software engineering.

Diana Borrego received the Ph.D. degree in computer science from the University of Seville, Seville, Spain, in 2012.

She is a Lecturer with the University of Seville, where she is a Member of the IDEA Research Group. Her works have appeared in international conferences and journals, including *Information and Software Technology*, *Data and Knowledge Engineering*, and the *Journal of Systems and Software*. Her research interests include the verification and diagnosis of business processes through automatic reasoning for their quality improvement.

José A. Galindo received the Ph.D. degrees (Hons.) from the University of Seville, Seville, Spain, and the University of Rennes, Rennes, France, in 2015.

He has developed his professional activity in the United States, France, and Spain. He is currently working as a Lecturer with the University of Seville, where he continues his research on configuration, testing, and the evolution of highly configurable systems. His research areas are product-line software and the configuration of such products.

Dr. Galindo received the award for the Best National Thesis by SISTEDES.

David Benavides received the B.S. degree in information systems from the Institute Superieur d'Electronique de Paris, Paris, France, in 2000, and the M.Sc. degree in computer engineering and the Ph.D. degree in software engineering from the University of Seville, Seville, Spain, in 2001 and 2007, respectively.

He has been an Associate Professor with the University of Seville since 2010. His main research interests include software product lines, configurability, variability management, opensource, and artificial intelligence applied to engineering education.