A Survey on Problems related to the Teaching of Programming in Brazilian Educational Institutions

Anderson S. Marcolino, Ellen Barbosa
Institute of Mathematics and Computational Sciences (ICMC)
University of São Paulo (USP)
Avenida do Trabalhador São–Carlense, 400, São Paulo, Brazil.
Email: andersonmarcolino@gmail.com, francine@icmc.usp.br

***Abstract*—Almost all college and university courses have integrated disciplines for the teaching of programming in their curricula. The importance of knowing "how to program" has contributed towards increasing private and public initiatives for the teaching of programming in different countries. Many of such initiatives include changes in the curriculum of primary and high schools, replacing disciplines by programming classes. Despite the increasing needs for the learning of programming, many limitations and problems as (i) learning to program, (ii) application of programming concepts, (iii) understanding of programs, (iv) process of refactoring and factoring programs, (v) students' motivation, and (vi) teachers' difficulties have been imposed on this domain. In this perspective, this paper presents a survey conducted with Brazilian teachers and learners, from private and public institutions, aiming at: (i) identifying the relevance of the international problems in Brazilian higher education, (ii) identifying the main solutions that have been adopted in programming disciplines and their limitations, including software solutions, and (iii) collecting problems in teaching of programming in learner's perspective. Based on the results of the survey, a set of main challenges to be solved for providing a better adoption of programming disciplines in Brazilian curricula are presented and discussed. The challenges are complemented with discussions about the economic and technological limitations of the country. Moreover, means to mitigate the problems presented in both scenarios, national and international, are discussed as well.**

I. Introduction

Disciplines related to the teaching of programming are present in almost all undergraduate courses in Brazilian and international institutions [1], [2] and are not restricted to technological courses.

The teaching of programming is important not only for the acquisition of technical knowledge by learners, but also for their professional career. Several initiatives that facilitate both learning and comprehension of complex structures and the logical skills involved have been developed. Such initiatives are not limited to software, but include specialized schools in the teaching of programming for different ages [3].

Despite these possible solutions and the several proposals reported in the literature [4], [5], [3], the teaching of programming still faces limitations, such as [6] high level of abstraction, traditional educational methodologies, diversity of learners, difficulties related to individual monitoring, necessity of more practical studies, and complexity of programming languages.

In this perspective, Marcolino and Barbosa [4] conducted a systematic mapping for the identification of software that supports the teaching of programming and mitigation of problems in this domain. Aureliano and Tedesco [5] investigated software and applications for the teaching of programming to beginners and discussed the Brazilian initiatives undertaken in primary and secondary schools. However, problems in the teaching of programming in Brazilian educational institutions must be identified and solutions that attenuate them must be proposed.

This study addresses a survey conducted with Brazilian teachers and learners from private and public institutions for identifying: (1) the relevance of the international problems in Brazilian higher education, (ii) the main solutions adopted in programming disciplines and their limitations, including software solutions, and (iii) problems in the teaching of programming from the learner's perspective.

The answers contributed towards a discussion on some open challenges to be overcome for improving the teaching of programming and the reasons for the small number of initiatives in its domain in the difficult of primary and secondary schools. Moreover, we discuss the way the current research has been conducted for the reduction of problems in the programming domain in international and national scenarios.

The paper is organized as follows: Section 2 presents the main concepts and discussions on the initiatives for the teaching of programming worldwide; Section 3 describes the methodology, planning and execution of the survey; Section 4 discusses the results and presents the open challenges; finally, Section 5 presents the conclusions and future work.

II. Background

Since the advent of computers and their use in different life contexts, as industries, residences and educational institutions, the way they can help individuals to improve their routine has been a concern [7], [8], [9]. Mainly in education area, this concern reflects the emergence of

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

several educational theories that support the adoption of technological solutions [10], [11], [12], such as the 21th century theories [12]. This is a direct consequence of the creation of new technologies and their adoption in the classroom for traditional, blended or distance education [13].

Reductions in the costs of information and communication technologies (ICT) and the use of such technologies in social and economic sectors have led to their adoption in education and appearance of new learning modalities, as electronic learning (e-learning), digital–television–based learning (t-learning) and mobile learning (m-learning) [7], [14]. New tools are therefore required, so that their maximum potential can be investigated for improving the processes of teaching and learning. While new hardware and technological concepts, such as virtual reality glasses, low-cost programming microcontrollers (e.g. Raspberry pi, Arduino), Internet of the Things (IoT) and Big Data have arisen, a wide base of skills and knowledge that deals with such innovations in the technological area is required [15], [16]. As a consequence, the software industry claims for more practitioners with great technical and programming skills.

A. Worldwide Initiatives

Several publications have reported an alarming situation in the technological area, i.e., the lack of software professionals until 2020[1] [17], [16].

In this perspective, the United States government is expected to reach one million programmers by 2020[2], therefore, many educational initiatives have been undertaken[3] [3]. The United Kingdom (UK) has proposed to teach children to program, mainly due to complaints of technological companies that have claimed for more qualified graduates.

In Australia, the government has proposed new digital technology curricula. The main Australian curriculum change refers to the replacement of history and geography by coding classes. The Australian government has justified the insertion of technological education in primary and high schools through projections that show a lack of qualified workers. If this situation is not changed now, it may result in a big problem for the development of the country in 20 years' [3].

The Australian new curricula are based on successful programs implemented in the United States, as "Code.org" and "Hour of Code" – a non-profit initiative supported by Google and Microsoft[4] [3].

There are also initiatives with other aims [3], such in Estonia, where the objective is the intellectual enhancement. They believe people that using technology, computers and the Web become smarter[5].

Despite the visible expansion of the technological market and undergraduate courses in Brazil, many actions must be taken for integrating the teaching of programming in primary and high schools, as reported in the systematic mapping (SM) conducted by Aureliano and Tedesco [5]. Such an SM aimed at identifying the positioning of researchers regarding the teaching and learning of programming for children in the Brazilian literature.

The authors were motivated by the lack of discussion on the topic in the Brazilian scenario, which is considered important from an international perspective. They also described some software solutions for supporting the teaching of programming [5].

Aureliano and Tesdesco [5] identified considerable efforts by Brazilian researchers towards the teaching of programming to beginners until 2009. However, after that period, the interest has stagnated. The authors showed the substantial difference in research developed with undergraduate students and students from secondary and technical education in the programming area – 18 studies against 5, respectively.

While minimal efforts have been devoted by public institutions towards proposing the integration of teaching to program in their curricula, private institutions have proposed initiatives as *Desafio do Código* [6] and Yadaa [7] to teach it to different ages. However, such a concern is recent and the initiatives are also new. More long-term research is necessary, so that the real benefits to the countries and learners that have adopted such initiatives can be identified [3].

B. Why the teaching of programming?

The benefits of learning to program are not restricted to the potential development of the technological sectors of countries, hence, their economy. They also encompass the improvement of individual skills in other areas and domains. For the increase of mathematical thought, for instance, the programming activity: (i) justifies the formal mathematical rigor: (ii) encourages children to study mathematics through exploratory activities; and (iii) provides key insights into certain mathematical concepts, a context for problem solving and a language through which students can describe their own problem solution [10].

Furthermore, children with some contact with programming activities can produce more original and creative ideas in comparison with children that have never written codes [18], [19].

Learners can also develop the ability to work in teams through different approaches, as pair programming, in

[1]https://goo.gl/ptmYMvandhttps://goo.gl/T1lUL1
[2]https://code.org/
[3]http://goo.gl/ztgJ9d
[4]https://goo.gl/M3nRO1, https://goo.gl/h4citZ and http://goo.gl/aaoLlo
[5]https://goo.gl/67Smtc
[6]http://desafiodocodigo.com.br/
[7]http://www.yadaa.com.br/

TABLE I
Programming Problems Categories [6].

Id	Category	%	Description
P1	Learning of programming	38%	Limitations in learning programming concepts, such as recursion, index, repeated structures, classes and objetcs.
P2	Application of programming concepts	24%	Difficulties in applying the learned programming concepts.
P3	Understanding of programs	11%	Difficulties in reading and understanding programs.
P4	Refactoring and factoring programs	3%	Difficulties in dividing a program into modules, functions, classes, etc.
P5	Motivation	22%	Difficulties related to the lack of interest or discouragement of learners to perform programming activities.
P6	Teachers' difficulties	2%	Difficulties of teachers and instructors in teaching programming concepts, creating materials, activities and assessing programming homework.

which two learners work collaboratively towards solving a problem [20].

Expert programmers develop individual skills that can be used in different fields, as mathematics. They are good at recognizing, using and adapting patterns or schemes, and faster, more accurate and able to draw a wide range of examples from different sources of knowledge and apply effective strategies [21].

Because of such benefits and the importance of learning programming concepts, several educational software solutions have been proposed, as reported in the SM conducted by Marcolino and Barbosa [4]. However, most solutions are related to very specific problems in the institutions where they were developed and despite supporting traditional education, they do not implement mechanisms that provide a more precise feedback for mitigating learners' problems in their programming learning process and support teachers.

Additionally, some ICT for supporting the teaching and learning do not present many adoptions. A large number of mobile devices have been used worldwide, however, few studies have investigated their adoption in programming education. According to the SM conducted [4], among 81 educational programs, only two adopted mobile devices and m-learning, despite the advantages of the platform, as more personal learning environments, education in geographically remote places and lower cost [4].

Duncan et al [3] described several software and applications for the teaching of programming to learners of different ages (2–14 years and up), called initial learning environments (ILEs). Despite all computer programs and applications that support the teaching of programming, several problems are still faced in the domain.

C. Problems in the Teaching of Programming

Souza et al [6] performed an SM for identifying and classifying programming problems in the international scenario, the idea was to provide a comprehensive summary of these problems and the solutions proposed to minimize them. They selected 70 primary studies dated from 2010 to 2014 from SCOPUS electronic base [8] to propose a problem classification. Each category concentrate several problems related with each topic. The problems categories are showed in Table I.

Category **Learning of Programming (P1)** are related with the limitations in learning programming concepts for both synthetic and semantic. Category **Application of Programming Concepts (P2)** are related with the application of the concepts mentioned in previous category, it concerns in how the concepts can be used. Category **Understanding of Programs (P3)** corresponding with the difficulties in reading and also understanding the programs, concerns in what the program do and/or need to do. Category **Refactoring and Factoring Programs (P4)** corresponding with the difficulties in dividing a program in small parts, as modules, functions and classes, which allows the use of the learned concepts incrementally until solve the problem by complete. Category **Motivation (P5)** concerns the interest or discouragement of learners to do their programming tasks, how attractive are the tasks and how motivated are the learners to do them. Category **TeacherÂt's difficulties (P6)** are corresponding the difficulties in teaching the programming concepts, creating materials, creating learning activities, monitoring the performance of learners and assessing them. The two main problems were categories P1, which corresponds to 38% of the selected primary studies, and P2, which corresponds to 24% of the primary studies.

As such categories reflect the main problems in the teaching of programming in the international scenario and are considered important to be mitigated in such a domain, we conducted a survey with Brazilian teachers and learners to check if those problems also occurred in Brazilian institutions. The survey also enabled the collection of information on the solutions adopted for mitigating the problems reported.

The aim is the establishment of a set of problems that encompass problems from both Brazilian and international realities, identifying possible differences and allowing for proposing solutions for both realities.

[8] https://www.scopus.com/

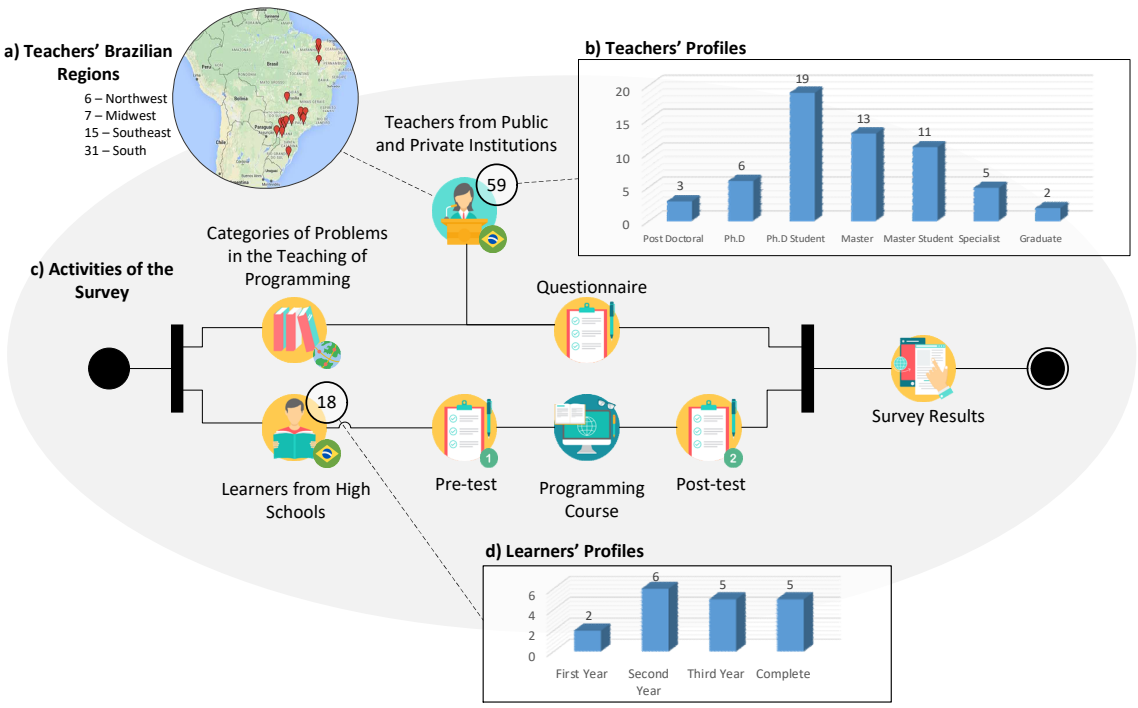

Fig. 1. Survey Information: a) Teachers' Brazilian Regions, b). Teachers' Profile, c). Activities of the Survey and d). Learners' Profiles.

III. Methodology, Planning and Execution

Prior to the online survey, the following steps were taken: (i) definition of the questions; (ii) validation by researchers from CAEd – Applied Computing in Education lab and at LabES – Software Engineering lab at the University of São Paulo, Brazil and (iii) execution of a pilot application.

A teachers' form was made available and shared in e-mail lists, whereas a learners' form was made available in two moments – as a pre-test prior to the web programming course and at the end of the course, as a post-test. The basic web programming course was free for high school learners. The annual course is taught by undergraduate students through the *Codifique* initiative [9] at the University of São Paulo.

The survey for the teachers comprehended three questions. In the first, the participants evaluated the six problem categories in the teaching of programming. The rank scale ranged from least incident (5) to most incident (1). The second question regarded possible strategies or software solutions adopted for mitigating the problems faced by the participants in the classroom. Finally, the last question concerned other possible problems or limitations not included in the six categories considered. Apart from those three questions, a characterization questionnaire was also answered. The teachers' answers were collected in the second semester of 2015 – 59 teachers answered the questionnaire.

[9]http://cursocodifique.weebly.com/

The research applied to learners comprehended five questions in the pre–test and eight questions in the post–test. From the eight questions, four are addressed in this paper; the other four questions gave feedback about the course.

The learners' pre–test asked about (i) their reasons to learn to program, (ii) importance of programming, (iii) their intentions in adopting mobile devices to learn to program, (iv) strategies to learn to program, and (v) other possible ways for learning to program.

The post-test addressed (i) topics difficult to be learned, (ii) the learners' opinions on why they faced difficulties, (iii) their intentions in adopting mobile devices to learn to program, and (iv) their intentions in continuing to learn to program through applications of mobile devices. The answers were collected in the second semester of 2016. 14 learners answered the pre–test and 14 learners answered the post–test, which totaled 18 different participants – ten answered both tests, four answered the first one, and another four answered the second test.

In both surveys, teachers and learners were invited to answer the questionnaire, being free to do so or not. Their different profiles allowed the collection of heterogeneous data and, even with different profiles, the analysis results showed same initial evidences. The different teacherÂt's level of expertise also contributed with the results, since their experience reflect in the adoption of different solutions and strategies in their classes, some of them are older than others, but, most of them direct the efforts for mitigating similar set of problems, as showed in next

section, in the analysis of the results.

Figure 1 shows a) the Teachers' Brazilian Regions, b) teachers' degrees, c) the main activities planned executed in the survey and d) learners' degree profiles.

IV. Analysis of Results

A. Survey with Teachers

The 59 programming teachers were from the Northwest (6), Midwest (7), South (31) and Southeast (15) of Brazil. The Northern region did not have participants. Southeast and South had the largest number of participants. Such regions also provided most initiatives for the teaching of programming for beginners, according to Aureliano and Tedesco's SM [5], and are ranked first and third in the demographic distribution according to Brazilian Geographic and Statistics Institute (IBGE) [10]. It is highlighted that the survey was sent for lists of discussions in informatics in education, being the subscribed teachers from all regions of Brazil. In addition, none of the regions received more invitations, due the participants were free for answering or not the survey.

Among the teachers, 19 are Ph.D. Students, 13 are masters, 11 are master students and 16 are specialists. Their average time of experience in programming was 7.20 years.

The teachers' answers are discussed below.

Question 1 - Do the categories of problems in the Teaching of Programming reflect the problems that occur in your classes?

The question aimed at identifying if the categories also reflected the problems in Brazilian universities, since they were extracted from a set of international primary studies [6]. Answers to the question suggested proposals of general solutions to problems of both international and national institutions. Table II shows the scores given by the participants ranged from most incident problem (1) to least incident problem (5).

P2 was considered the most incident, which suggests most common problems are related to the application of programming concepts. P3, i.e. understanding of programs, was ranked second and P6, related to teachers' difficulties, was the third. The graphic in Figure 2 shows the categories and their classification in six positions ordered from 1 to 6, from the highest to the lowest, respectively.

[10]ftp://ftp.ibge.gov.br/Estimativas_de_Populacao/Estimativas_2016/

TABLE II
Ranking Results of the Teaching of Programming Problems.

Id	Category	Position		% by Incidence Level - Survey					
		SM	Survey	1	2	3	4	5	Total
P1	Learning of programming	1	4	32.2%	32.2%	15.25%	15.25%	5.1%	100%
P2	Application of programming concepts	2	1	***35.6%***	28.8%	16.95%	13.55%	5.1%	100%
P3	Understanding of programs	4	2	15.25%	***35.6%***	***42.38%***	6.77%	0%	100%
P4	Refactoring and factoring programs	5	5	15.25%	23.72%	37.31%	16.95%	6.77%	100%
P5	Motivation	3	6	23.72%	25.42%	22.04%	20.33%	8.49%	100%
P6	Teachers´ difficulties	6	3	15.25%	8.49%	33.9%	***23.72%***	***18.64%***	100%

Fig. 2. Ranking of Results by Category of Problems.

The participants could classify the categories with the same level of incidence, which led to statistical ties, as in P3 and P4 categories, in italics. Such ties were considered in the definition of the final order of the most relevant problems.

To avoid biases related of those categories and the absence of other possible faced problems, not encompassed on them, an extra question (Question 3) was included for the identification of other related issues. A reduced set of categories provides an easier way to propose solutions and also their use in future work.

Question 2 - Do you adopt strategies and/or software solutions for mitigating problems in the teaching of programming?

The question aimed at identifying strategies and software solutions adopted towards reducing the problems faced by teachers in their programming classes. The solutions can be adopted by other teachers or improved, when possible, for mitigating problems in the programming domain.

The answers showed several solutions to be adopted. The participants were university teachers and most came from technical schools. The main strategies reported in more than one answer are highlighted next:

- Adoption of easy programming languages, i.e., languages with fewer complex structures, weak variable typing and less reserved words, as Python language, for instance;

- Adoption of ideas and learning materials from online programming courses, as Udacity [11] and Coursera [12];
- Adoption of different teaching approaches, as problem-based, several exercises and learning activities, examples based on analogies of real scenarios, collaborative activities, competitions of programming (e.g., programming marathon);
- Adoption of software as a complement to the activities developed in the classroom, such as Scratch [13], Greenfoot [14], VisualLog [15], Integrated Development Environments (IDEs), Coding DOJO [16], BlueJ [17] and URI Online Judge [18] and;
- Adoption of different content presentations, as videos, slides and funny figures.

The most indicated solution was the application of several lists of programming exercises for motivating learners to practice the concepts learned in the classroom and quickly identifying their doubts. The solutions were adopted as a complement and used as a tool that enables students to apply what they learned in classroom.

Question 3 - Are there other problems related to the teaching of programming that were not considered in the six categories previously identified?

The question aimed at identifying problems not specified in the six categories, for a possible improvement in their categorization and better proposals of solutions.

Most answers agreed the six categories were complete enough to represent the main problems in the teaching of programming. Although the results cannot be generalized for all Brazilian institutions, they provide preliminary evidences the categorization of the teaching and programming problems represents real national scenarios that are not limited to international institutions. On the other hand, the sample can be increased in future iterations of the survey.

The participants reported the following specific difficulties in dealing with classrooms:

- "Learners are not prepared to program and their knowledge in mathematics and logical thinking is not deep".
- "Learners do not know the differences between use of a programming language and application of logical concepts. Programming must be taught in a specific program language and programming fundamentals should focus on logical thinking and algorithms without the application of a language";
- "Learners are not engaged. They prefer to use cellphones and the Internet to perform other activities instead of learning" and "since many technologies are available (e.g. games, social networks, etc.), their concentration is hampered".
- " Learners are not interested in performing their learning tasks. All tasks are considered difficult, mainly those that consume time, e.g., watching videos longer than 15 minutes ".
- " Learners copy answers (plagiarism). They prefer to copy answers to their activities from other learners than solve the tasks by themselves. ".
- "Programming tools and languages provide little visual stimulation".
- "Learners have problems in expressing their doubts".
- " Learners have problems in interpreting problems reported in more than four lines (functional illiteracy)".
- "Learners cannot seek information for themselves. There is a lack of autonomy and initiative".
- "Learners do not like to work in groups, they have difficulties in forming groups".
- "Learners very frequently give upon the courses".
- "Program disciplines are often fixed and, somehow, outdated".
- "Open source and free tools are limited".
- "Learning topics already assimilated by learners becomes demotivating when applied in exercises. The customization and creation of exercises would be a good solution".
- "Materials do not simulate functions (e.g. methods and repetition structures) in the introduction to programming".

All reports are related to the basic knowledge necessary for learners to understand the programming concepts involved, their demotivation and distractions with other technologies. Some problems in the tools and software solutions do not meet the teachers' needs or support them. In some cases, outdated curricula do not adopt new solutions for supporting both teachers and students in programming classes.

B. Survey with Learners

The 40 learners enrolled in the course were from São Carlos (São Paulo state), located in the Brazilian Southeastern region, where the basic web programming course was offered. All participants were invited to answer the online questionnaire, however, only 18 answered it – 14 answered the pre–test and 14 responded to the post–test. Ten answered both tests, four answered exclusively the pretest and the other four learners answered the post–test.

Regarding the pre–test, two participants were enrolled in the first grade of high school, four in the second grade, five in the third and three had already finished it. In the post-test, two attended the first grade of high school, six

[11] https://www.udacity.com/
[12] https://www.coursera.org/
[13] https://scratch.mit.edu/
[14] www.greenfoot.org/
[15] www.apoioinformatica.inf.br/produtos/visualg
[16] http://dojopuzzles.com/
[17] www.bluej.org/
[18] https://www.urionlinejudge.com.br/

the second grade, four the third grade and two had finished it.

The set of questions of the pre–test was aimed at identifying the learners' expectations, their intentions to adopt mobile devices to learn and the strategies they expected to see in learning software. The implications about the adoption of mobile devices for learning and the pre–test questions and answers are reported next.

Why do you want to learn to program?
Most learners answered they intended to follow a career in the area and, besides this, technology is present everywhere; therefore, if they knew how to program, they could create and improve their environment.

This unanimous answer highlights the concern of students and the importance they give to programming. Due to such a small sample, this initial evidence suggests teenagers know the importance of learning to program, which can lead Brazilian educators and the government to reconsider the inclusion of this discipline in the curricula of primary and secondary schools.

Added to this motivation, the efforts seen in international scenario and also in several studies presented previously [4], [5], [3].

Why learning to program is important for your future?

The answer is the same, i.e. the learners said learning to program enables the understanding of the technological world we live in and programming is essential for their careers.

If you could, would you adopt mobile devices to learn to program?

As mobile devices are a low–cost and widespread technology, including in Brazil, they could be adopted to teach learners to program [22]. Understanding the learners' acceptance for mobile devices to learn can lead to their adoption as a way to engage learners and motivate them.

13 learners answered they would adopt mobile devices, mainly because they can be accessed and used anywhere and anytime. Only one participant answered "no" and justified computers support a more complete development environment.

According to IBGE [19], from 53,052,621, 31,445,902 Brazilian homes had mobile cellphones in 2005. In the same year, 59.7% of that population could access cellphones and only 18.5% had access to computers. Although the possible support provided by those devices to learning applications must be identified, the number of cellphones in Brazil is enough to show the potential of the platform. Moreover, if the geographic extension of the country is considered, several significantly distant regions can benefit from such a use.

Which strategies would you prefer to use in software for learning to program?

As one of the main challenges of educational software and applications is the guarantee of learners' engagement, the answers may help to improve their engagement. Four learners answered videos, two said games, and eight preferred a mix of videos, animations, texts, figures and games. One learner said strategies with fewer texts would be more engaging than those with long texts.

Which other means would you like to adopt to learn to program?

Six learners were interested in mobile applications, four in robots, two in sites, two in games and two would adopt software. The results show a diversity of options that could be adopted for the teaching of programming.

At the end of the three-month course, a Post–test questionnaire was available for the learners. The questions were focused on the knowledge acquired and the learners' problems, which were compared with the six categories of problems for the identification of possible lacks and improvements.

Which topics are more difficult to learn?

Among the answers, one indicated repetition structures, one regarded conditional structures, seven vectors, seven functions and one learner reported no difficulty which highlights difficulties were related to the topics taught in the course. The problems related to the teaching and learning process are showed next.

Why do you have such problems?

The difficulties were justified by nine learners as problems in following the theory; two pointed out the level of difficulty of the activities and three did not explain the reasons.

The first and second questions indicate the six categories of problems in the teaching of programming also include the problems described by learners.

The participants were asked to develop activities at home with no support of learning platforms, where they could check they doubts. All doubts were solved in the classroom by the instructor. Another consideration was the instructor's difficulty in identifying learners' limitations, since each learner could tell their difficulties to the instructor. If a learning platform was adopted, this need would be reduced due to the support in the learning process.

If possible, would you adopt mobile devices to learn to program?

The question was repeated, so that possible changes in the adoption of mobile devices could be identified.

[19] ftp://ftp.ibge.gov.br/Estimativas_de_Populacao/Estimativas_2016

In the pre–test, thirteen learners answered they would adopt a mobile device platform and one said he/she would not. In the post–test, again, thirteen learners answered they would adopt mobile devices and one said no. From the thirteen answers, ten learners that had answered both tests confirmed the acceptance of adoption of a mobile platform for learning.

Do you have any suggestions for the course and the topic learned?

Two main suggestions were highlighted: (i) the need of teaching programming fundamentals; and (ii) the adoption and teaching of other programming languages, besides JavaScript.

Basic concepts of programming web with JavaScript were taught. However, the learners considered the topics somewhat advanced for their knowledge. In this perspective, they expressed the need to learn programming fundamentals prior to a programming language, as JavaScript.

The answers from 59 teachers and 18 learners and the SMs identified[3], [4], [5] enabled the identification of some open challenges in the teaching of programming, summarized next.

C. Open Challenges

- Provide of a better training to learners, mainly in mathematics and logical skills, prior to the teaching of programming;
- Conception of more personalized software solutions that aggregate precise feedback, identification of limitations and doubts, short content and learning activities, more visual simulation strategies, collaboration and cooperation interactions;
- More tools for supporting teachers during and after classes and monitoring of the learners' performance;
- Adoption of solutions for engaging and motivating learners.
- Development of the same functionalities and facilities that learners find in social network applications, games and mobile gadgets, for instance;
- Creation of content through different strategies of presentation, taking into account the separation of logical and algorithm concepts from specific programming languages, and;
- Adoption of mobile devices that support teaching and learning.

Such challenges are related to the programming reality and are not restricted to Brazil [4]. However, in comparison with other countries and their initiatives for improving the technological sector, as inclusion of programming disciplines in their elementary curricula, many changes are still required in Brazil. Among them, we can highlight changes in the teacher's thinking regarding the adoption of innovative technologies for engaging learners and enriching their lectures. Teachers need more motivation (and training) for effectively adopting new technologies in their programming classes [5]. A deeper investigation must be also conducted, so that changes in the curricula for public and private primary and high schools can be proposed.

D. Conclusions and Future Work

This paper has addressed worldwide initiatives taken regarding the teaching of programming and problems in classrooms in some Brazilian institutions. The answers of a survey conducted enabled the identification of initial evidences the problems of worldwide educational institutions are the same in Brazil. The survey also suggested considerable initiatives for mitigating the problems listed.

From the different contributions provided by the 59 teachers and 18 learners, we have listed a series of opened challenges. Despite the several limitations on the teaching of programming, there is a lack of software solutions that mitigate them.

The solutions suggested by the participants focus mainly on tasks that are not supported by software, such as the adoption of activities and lists of exercises. Software and related applications are used to support teachers in classes, being considered as complementary tools since they do not provide mechanisms that automate manual activities, as activities correction, or more accurate feedback.

Motivated by the issues and initiatives addressed for the teaching of programming to undergraduate students and primary and secondary courses, we have worked on the establishment of an infrastructure (software product line [23]) that promotes the development of mobile learning solutions [24] and mitigation of the problems reported.

Acknowledgment

The authors acknowledge Brazilian funding agencies – FAPESP, CNPq and CAPES (DS-8907173DT) – for the financial support provided to this research.

References

[1] SBC, "Sociedade Brasileira de Computação. Currículo de Referência da Sociedade Brasileira de Computação para Cursos de Graduação em Computação e Informática (in portuguese)," Availble in http://www.sbc.org.br/index Acessed on January, 04th, 2017, 2003.

[2] C. S. P. IEEE and P. ACM, "Joint Task Force for Computing Curricula." Available in http://www1.acm.org/education/curricula.html Acessed on May, 4th, 2016., 2005.

[3] C. Duncan, T. Bell, and S. Tanimoto, "Should your 8-year-old learn coding?" in *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, ser. WiPSCE '14. New York, NY, USA: ACM, 2014, pp. 60–69.

[4] A. S. Marcolino and E. F. Barbosa, "Softwares Educacionais para o Ensino de Programação: Um Mapeamento Sistemático (in portuguese)," *Simpósio Brasileiro de Informática na Educação*, vol. 1, pp. 190–199, 2015.

[5] V. C. O. Aureliano and P. C. A. R. Tedesco, "Ensino e Aprendizagem de Programação para Iniciantes: uma Revisão Sistemática da Literatura focada no SBIE e WIE (in portuguese)," *Simpósio Brasileiro de Informática na Educação*, 2012.

[6] D. M. Souza, M. H. S. Batista, and E. F. Barbosa, "Problemas e Dificuldades no Ensino de Programação: Um Mapeamento Sistemático (in portuguese)," in *Revista Brasileira de Informática na Educação.*, vol. 24, 2016, pp. 1–14.

[7] A. Craig, J. Coldwell-Neilson, A. Goold, and J. Beekhuyzen, "A Review of E-learning Technologies–Opportunities for Teaching and Learning," in *Int. Conf. on Computer Supported Education (CSEDU)*, 2012, pp. 29–41.

[8] S. Bennett and K. Maton, "Beyond the âĂŸdigital nativesâĂŹ debate: Towards a more nuanced understanding of students' technology experiences," *Journal of computer assisted learning*, vol. 26, no. 5, pp. 321–331, 2010.

[9] H. Nissenbaum, *Privacy in context: Technology, policy, and the integrity of social life.* Stanford University Press, 2009.

[10] R. D. Pea and D. Kurland, "On the Cognitive Effects of Learning Computer Programming," *New Ideas in Psychology*, vol. 2, no. 2, pp. 137 – 168, 1984.

[11] J. J. V. Merrienboer and F. G. Paas, "Automation And Schema Acquisition In Learning Elementary Computer Programming: Implications For The Design Of Practice," *Computers in Human Behavior*, vol. 6, no. 3, pp. 273 – 289, 1990.

[12] L. Harasim, *Learning theory and online technology.* Routledge, 2012.

[13] Ambient Insight Reasearch, "Ambient Insight Reasearch. The Worldwide Market for Self–paced e–Learning Products and Services: 2011-2016 Forecast and Analysis," Available in: http://goo.gl/r1moXN Accessed on January, 04th, 2017, 2013.

[14] N. Rubens, D. Kaplan, and T. Okamoto, "E-Learning 3.0: Anyone, Anywhere, Anytime, and AI," in *New Horizons in Web Based Learning.* Springer, 2014, pp. 171–180.

[15] L. Johnson, S. Adams, and M. Cummins, "Nmc horizon report: 2014 higher education edition," Available in: ://www.nmc.org/publications/2014-horizon-report-higher-ed, The New Media Consortium, Austin, Texas Trendreport, Tech. Rep., 2014.

[16] ——, "Nmc horizon report: 2016 higher education edition," Available in: http://cdn.nmc.org/media/2016-personalized-learning-student-success-summit-report_public.pdf Accessed on January, 04th, 2017, The New Media Consortium, Austin, Texas Trendreport, Tech. Rep., 2016.

[17] A. Cushing and J. F. Gantz, "Skills Requirements for Tomorrow's Best Jobs – Helping Educators Provide Students with Skills and Tools They Need," Available in: https://news.microsoft.com/download/presskits/education/docs/IDC_101513.pdf Accessed on January, 04th, 2017, Microsoft, Tech. Rep., 2013.

[18] Y. C. Liao and G. W. Bright, "Effects of Computer Programming on Cognitive Outcomes: a Meta-Analysis." *Journal of Educational Computing Research*, pp. 251–266, 1991.

[19] A. Cockburn and A. Bryant, "Do it This Way: Equal Opportunity Programming for Kids," in *Computer-Human Interaction, 1996. Proceedings., Sixth Australian Conference on*, Nov 1996, pp. 246–251.

[20] A. Cockburn and L. Williams, "Extreme programming examined," G. Succi and M. Marchesi, Eds. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001, ch. The Costs and Benefits of Pair Programming, pp. 223–243.

[21] A. Robins and N. Rountree, J. an Rountree, "Learning and Teaching Programming: A Review and Discussion." *Computer Science Education*, pp. 137–172, 2003.

[22] N. Alalwan, A. Alzahrani, and M. Sarrab, "M-Learning the Next Generation of Education in Cyberspace," *Int. Congress World and Academic of Science, Engineering and Technology (WASET), Rio de Janeiro, Brazil.*, no. 75, pp. p. 642–645, 2013.

[23] R. Capilla, J. Bosch, P. Trinidad, A. Ruiz-Cortes, and M. Hinchey, "An overview of dynamic software product line architectures and techniques: Observations from research and industry," *Journal of Systems and Software*, vol. 91, no. 0, pp. 3 – 23, 2014.

[24] A. S. Marcolino and E. F. Barbosa, "Towards a software product line architecture to build m-learning applications for the teaching of programming." *In: 50th Hawaii International Conference on System Sciences (HICSS-50)*, pp. 6264–6273, 2017.