 Published in the 2023 Frontiers in Education Conference. College Station, TX, USA October 18-21
DOI: https://doi.org/10.1109/FIE58773.2023.10343195

Proposals for the Promotion of Computing in K-12 Studies in Spain

David Lopez
Computer Architecture Department
Universitat Politècnica de Catalunya
Barcelona, Spain
david.lopez@upc.edu 0000-0002-7993-3799

Faraón Llorens-Largo
Dept. Computer Science and Artificial Intelligence
University of Alicante
Alicante, Spain
Faraon.Llorens@ua.es 0000-0002-2117-0784

Mercedes Marqués-Andrés
Dept. Computer Science and Engineering
University Jaume I
Castelló, Spain
mmarques@uji.es 0000-0003-4880-6284

J. Ángel Velázquez-Iturbide
Departamento de Informática y Estadística
Universidad Rey Juan Carlos
Madrid, Spain
angel.velazquez@urjc.es 0000-0002-9486-8526

***Abstract*— The information society requires new skills and knowledge about how to cope in a computerized world. Although it is fully assumed in the political and social discourse, it is a neglected subject in formal education. Each nation should analyze how computing is learned in its K-12 education system. This research-to-practice paper presents the report carried out in Spain promoted by two academic associations, which analyzes the state of computing education in Spain in K-12 studies, summarizes some elements of the didactics of computing and analyzes the training received by teachers who will train K-12 students. The report also presents some reflections of the working group in charge, and nine proposals for improvement oriented to policy makers are pointed out.**

***Keywords*— K12 STEM Education, Computing Pedagogy, Teacher Development, Culture of Teaching and Learning**

I. Introduction

The information society requires new skills and knowledge. Knowing how to cope in a computerized environment is today as important as knowing how to calculate, read or write. Although it is fully assumed in the political and social discourse, it is a neglected issue in formal education to the point that the 21st century has seen the re-emergence of a new illiteracy, called digital illiteracy.

Digital competence is not acquired simply by growing up in a digital world, but requires an effort to introduce these skills at school. However, digital competence is not enough for the society of the future, as computing skills are needed from two perspectives. On the one hand, a knowledge of the basics of computing; on the other hand, developing a way of thinking processes and solutions oriented to their resolution with computers.

Any professional of the future must master both digital competence and the basics of computing, since they will interact with computer systems and will have to imagine new ways of doing things in their profession, in what is called digital transformation.

The university degree programs related to computing are inspired by the Computing Curricula developed by the two most important world associations in this field, the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers Computer Society (IEEE-CS) [1]. This has given a common view, an international uniformity to this discipline and the use of a shared vocabulary, which does not exist at the pre-university level.

To find a common framework for the definition of computing education at the pre-university level, it is important to clarify what is computing and what is part of it. Nowadays many branches of engineering and science are claiming as their own some concepts and processes that, in reality, are specific to computing, perhaps because there are many terms that are fashionable or attractive. Computing is a discipline that is influenced by others, but it has evolved into a discipline in its own right.

Politicians and lawmakers may be tempted to follow fashionable trends due to lack of knowledge. There are certain terms that are very attractive, thus, at least in Spain, in some forums it is claimed that pre-university students should learn certain topics such as "artificial intelligence", "robotics", "internet of things", "machine learning", "cloud computing", "deep learning", "virtual reality", "quantum computing", "big data", "industry 4.0", "5G", "blockchain", "cryptocurrencies", all of them being terms that fall under the umbrella of computing. This is why it is necessary to call for a "training in computing" that is adaptable to the evolution of computing and not simply guided by attractive terms and technology that can change rapidly.

Each nation should undertake a critical exercise and analyze how it teaches computing within its pre-university education. The different organizations dedicated to computing in each country, with knowledge of the problems of computing as well as its possible evolution, and the characteristics of education and industry in each country, can help by generating a reasoned framework intended to last for years, which can serve as a basis for policy makers to do their work.

Spain has traditionally had an economic system based on services that has suffered the impacts of the economic crises of recent years. At present, there is a commitment to change to a model based on knowledge, with the project to turn Spain into a technological hub. To this end, the government has planned an agenda called Digital Spain 2025 with a planned investment of 70,000 million euros. However, there is a gap between the number of qualified experts in STEM disciplines and market demands, which leads to pressure on the Spanish education system on the need to train professionals with technological skills, especially in computing.

In 2020 the Spanish government proposed a modification in the law on pre-university education by proposing a new general law, the LOMLOE (Organic Law 3/2020, of December 29, amending Organic Law 2/2006, of May 3, on Education). It is a legislative document that updates, amends and complements the Educational Law of 2006 that is currently in force. Its preamble refers to "integral education" organized around five focuses, namely: children's rights, in accordance with what is laid down in the United Nations Convention on the Rights of the Child; gender equality; a transversal focus on ensuring that all students have guarantees of success in education; consideration of the Sustainable Development Goals (SDG); and finally, consideration of the digital transformation that is happening in our societies.

From the time the first draft of the law was made public until its final approval there were a few months of discussion among different stakeholders of the civil society that provoked a series of amendments by different political parties (including the parties in government). As far as computing education was concerned, there were contradictory opinions from different stakeholders with different interests.

In these circumstances, two associations that host the Spanish computing academic community (SCIE and CODDII, see next section) decided that a proactive policy would be interesting, and a document with an analysis of the current situation, the modifications proposed by the LOMLOE and recommendations for future actions should be available, which could be used as a reference by the Government in the implementation of the law, as well as in future adaptations. An important objective of the report was to inform the political estate that it exists a body of knowledge on computing education and that pre-university teachers should know it for the successful introduction of computing into the pre-university curriculum.

This paper presents that report, as well as the elaboration process, with the recommendation that other national societies develop similar reports adapted to the particular characteristics of their nations.

II. About the Report

A. The initiative promoters

The Scientific Computing Society of Spain (SCIE by its Spanish acronym – *Sociedad Científica Informática de España*) is an association whose objectives are: to contribute to the scientific and technological development of computing in Spain; to act as a qualified interlocutor with both civil society and public authorities in the areas of computing; to promote research, innovation and technology transfer in computing in Spain; and to promote activities with other national and international associations with similar scientific purposes. As a society, it was created in 2005 and it is composed of associations that bring together professionals from different fields of computing in Spain: artificial intelligence, educational computing (university and pre-university), human-computer interaction, computer graphics, computer architecture, natural language processing, image analysis, software engineering and video games.

The Conference of Directors and Deans of Computing Engineering (CODDII by its Spanish acronym – *Conferencia de Directores y Decanos de Ingeniería Informática*) is composed of the heads of the schools and faculties that offer studies leading to Computing Engineering degrees in Spain, both public and private. It was created in 1997 as a forum for collaboration, debate and exchange between the different schools and faculties, and in 2009 it was constituted as an association with the aim of defending and promoting excellence and continuous improvement in Computing Engineering studies, participating in academic, scientific and professional initiatives in the national and international context.

SCIE and CODDII have worked together on many occasions. When discussing the appearance of the LOMLOE, it was thought that it was a good time to present a common position to society and the politicians in charge of education in Spain. As a result, in May 2022 the two societies held a joint meeting to define the main lines of what they wanted to communicate, and four people (the authors of this paper) were commissioned to develop the report. The report was sent in December to the presidencies of SCIE and CODDII. The report was approved by the Standing Committee of CODDII and the Board of Directors of SCIE, and was made public in January 2023 under the title *Informe CODDII/SCIE sobre formación del profesorado y didáctica de la informática en etapas preuniversitarias* (CODDII/SCIE Report on Teacher Training and Didactics of Computing at the Pre-university Level) [37].

B. Methodology

At the joint SCIE-CODDII meeting in May 2022, the objectives of the report were defined, as well as the people to whom it is to be addressed: the associations of primary and secondary computing teachers, those in charge of training these teachers, as well as legislators at both the national and regional levels. The committee in charge of drafting the proposal was also appointed.

The working group carried out some bibliographic research on the teaching of computing in Spain, the legal environment and the formal training received by computing teachers. Also, a study was made on how computing was taught in other countries and about recommendations from various international organizations. Finally, the literature on computing didactics in primary and secondary education was analyzed.

After several working meetings, a first draft was prepared and sent in early November to a group of seven recognized experts in the field for a critical reading of the report. With the comments gathered, the final document was created in December and sent to the presidencies of SCIE and CODDII and approved by both.

C. Report organization

The report was divided into four parts.

- The first part defines the context: what will be considered as computing in the report, and how this concept in considered in the new educational law.
- The second part focuses on the training of current computing teachers in the different educational stages in Spain, analyzes how their training is carried out in other countries and what are the recommendations from different organizations about teachers' training. Finally, a reflection is made on the knowledge involved in educational actions in general.
- The third section presents elements of computing didactics, from pedagogical bases to evaluation systems, including student motivation and teaching methodologies.
- Finally, the fourth section presents the conclusions of the report, with a series of reflections and nine proposals for improvement aimed at educational authorities.

The final document was 30 pages long, excluding cover page, executive summary and table of contents, and contained 67 bibliographic references. The report was published on the websites of both institutions and on social networks, always referencing a permanent URL in order to evaluate the number of visits. The document has been downloaded (May 9, 2023) 349 times. It may not seem much, but considering that the people to whom it is to be addressed are associations of primary and secondary computing teachers, those in charge of training these teachers, as well as legislators at both the national and regional levels, these are not terrible numbers.

III. Contents of the Report

This section will explain the contents of the report. Since the target audience of this paper is not the same as the target audience of the original report, neither the order followed nor the contents here will be exactly those of the published report. The authors believe that some very specific parts of the report, such as the Spanish legal framework, may not be of interest to the readers of this paper, so a mere description of the content and the steps followed to create it will be given. Other sections, such as those referring to computing didactics, will be explained in depth.

A. What is computing?

It is important to establish what exactly is meant when the term computing is used. The report uses the ACM and IEEE CC2020 definition [1], so in this document, the word computing refers to a goal-oriented activity requiring, benefiting from, or associated with the creation and use of computers. It includes a variety of interpretations such as designing and constructing hardware and software systems for a wide range of purposes: processing, structuring, and managing various kinds of information; problem solving by finding solutions to problems or by proving a solution does not exist; making computer systems behave intelligently; creating and using communications and entertainment media; and finding and gathering information relevant to any particular purpose.

The report discusses that some attractive or fashionable terms (such as Artificial Intelligence) can skew the vision of what should be part of the pre-university training in computing, arguing that these terms fall under the umbrella of computing and that we should think of computing education in a broad sense, beyond trends and designed to last over time. The difference between computing and digital competence is also presented, indicating that the latter is not sufficient for the future society. This part is similar to the introduction of this paper, so we will not elaborate further in this section.

B. How is computing addressed in the new law?

The report analyzes the new law, which redefines the curriculum and its basic elements. The authors assume that the analysis of the Spanish law is not of interest to the readers of this paper, so this section will briefly describe the Spanish education system and present some conclusions of analysis of the law.

In Spain, education is divided into several stages: early childhood (before the age of 6), which is not compulsory; compulsory primary education (from 6 to 12 years of age) and compulsory lower secondary education (between 12 and 16 years of age). From the age of 16, young people can enter the labor market, pursue vocational studies or study higher secondary school, oriented for university entrance, which lasts 2 years.

In primary education there are no subjects fully dedicated to computing, the knowledge being included in subjects of knowledge of the natural, social and cultural environment or mathematics. The knowledge covered include digital competence and what is called computational thinking, which includes the organization of data, decomposition into parts, recognizing patterns, generalizing, interpreting, modifying and creating algorithms in a guided way to model and automate everyday situations.

In secondary education there is a technology and digitalization course in at least one of the first three years, which includes some programming and robotics topics. In the fourth year there is an elective subject called digitalization.

In conclusion, the report points out that there are no specific courses on computing or with this word in its title.

C. Training and profiles of pre-university computing teachers in Spain

In Spain, education is a regulated profession that requires certain qualifications in order to work as a teacher. In the case of early childhood and primary education, it is required to have a Degree in Education (with specialties).

In the case of secondary education, a Master's Degree in pedagogical and didactic training of 60 ECTS must be taken. This Degree has a common generic module (20%) and a specific module divided by teaching specialties (e.g., language teaching or mathematics). Two of these specialties are computing and technology. To gain access to this Master's Degree, students must demonstrate mastery of the competencies related to the specialization they wishes to pursue.

Holding a master's degree allows for employment in private schools, while in public schools (the majority in Spain), teachers are recruited by means of opposition contest, consisting of tests of specific knowledge and pedagogical ability.

The analysis carried out by the working group detected two problems. On the one hand, in the Degree of Education different didactics and specific educational methodologies are studied (such as the didactics of mathematics), but neither the didactics nor the educational methodologies of computing or technology are studied. On the other hand, the syllabus of the opposition contest examinations to become a secondary education teacher in computing is too old, so it should be updated.

D. Computing teachers training experiences

The working group conducted a search for successful international experiences with regard to initial and continuous teacher training in computing, following the review by Heinz, Mannila and Färnqvist [14].

In UK there is a community that promotes computing at school, named Computing at School (CAS, http://computingatschool.org.uk). The community allows users to register and participate in discussion forums, and provides teaching resources. In addition, CAS hubs have been created, which are local communities of teachers who wish to share ideas about teaching computer science. The success of the CAS project is due in part to the network of hubs, as local meetings reduce teacher isolation and increase teacher engagement. Teaching resources include lesson plans and guidelines for different levels, starting with Barefoot Computing (https://www.barefootcomputing.org/) for Primary Education and Computing at School (https://www.computingatschool.org.uk/) which provides support for both Primary and Secondary Education. In addition, some teachers have also developed and shared teaching materials (see, for example, http://code-it.co.uk).

In Australia, initiatives such as a professional development MOOC and a systematic review of resources for the computing curriculum have been launched. Similarly, in New Zealand, a postgraduate distance learning course was developed to provide a formal qualification in computing education. To provide suitable teaching material, the CS Field Guide (https://www.csfieldguide.org.nz/es/) was developed. This is an interactive site developed to provide information at the level required by the new computing standards, including material for teachers.

In the United States, the K-12 Computer Science Framework (https://k12cs.org/) helps teacher professional development. The concepts and practices included in the framework help program designers and novices to organize and understand the diverse knowledge in the field of computing. The document was intentionally written with the intent that it would be used by both, faculty familiar with computing and those new to the field.

The Computer Science Teachers Association (CSTA) has addressed the issue in the U.S. through a task force (https://csteachers.org/page/standards-for-cs-teachers). The report recommends forming communities of computing teachers who can convey the nature of the discipline and who may even be mentored by university faculty.

Focusing on the European Union, in Norway a MOOC was prepared and an association called Lær Kidsa Koding has also been created to provide teaching materials as well as contribute to professional development (http://kidsakoder.no/) For teacher preparation in Poland, there is a standard for computing education that is similar to the ISTE standards (http://www.iste.org) There is also an accreditation procedure that assesses each professional's preparedness for teaching computing. In Estonia, HITSA (Information Technology Foundation for Education) developed learning materials and offers professional development courses for teachers. All HITSA's activities were transferred to Estonian Education in 2020 (https://www.educationestonia.org/infosystems/). In Finland, the National Board of Education and the Ministry of Education and Culture fund professional development projects and programs to provide teachers with resources and ways to integrate programming into teaching. In addition to specialized face-to-face courses, online teacher training courses on programming in both Finnish and Swedish were launched in 2015/2016, attracting a large number of participants. In addition to state-supported initiatives, there are also activities offered by private entities, universities and organizations.

In Europe, an active role is being played by the Informatics for All coalition. It was created in 2018 at the initiative of ACM Europe Council, CEPIS Education Committee, and Informatics Europe; IFIP joined in 2020. The coalition's goal is for computing to be considered a core discipline that all students study in school. Computing should be recognized as an essential discipline that plays an important role in 21st century education, as important as mathematics, science and languages. Their report The Informatics Reference Framework for School [15] contains concrete recommendations on how to integrate computing into the school curriculum. Likewise, its report Designing and Implementing a Concrete Informatics Curriculum for School [16] points out the need for computing teachers in compulsory education to receive training in this field, in addition to being able to count on pedagogical and methodological support on the teaching of this discipline. The report presents some examples for developing the computing curriculum. Regarding the general objectives, which should be considered before establishing the contents, and which are a description at a higher level of the goals to be achieved, it is indicated that the aim is to give a global vision of computing and the use of computing, as well as the pedagogical approaches when establishing the learning activities: practical activities with and without computers, teamwork, creativity and abstraction. Equality, diversity and inclusion issues are also crucial when setting these general objectives.

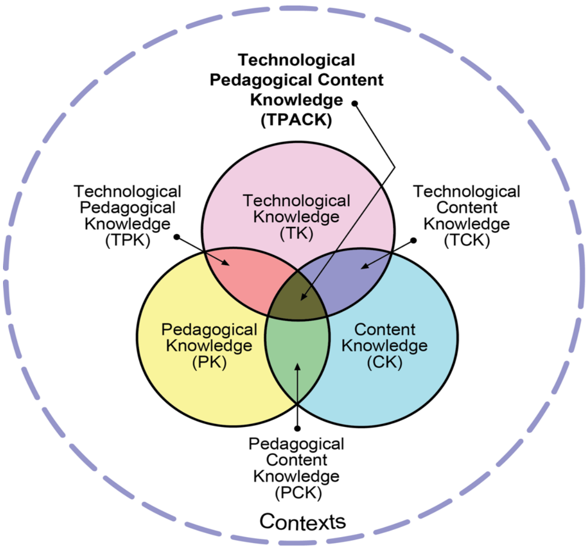

Fig. 1. The TPACK model (Koehler and Mishra, 2009). Reproduced by permission of the publisher, © 2012 by tpack.org

E. Technological, Pedagogical and Content Knowledge

Universities offer bachelor's and master's degrees to train future teachers in a generic way but also in a specific way for different subjects: languages, mathematics, physical education, etc. It is assumed that it is not enough to know a discipline, but that one must know how to teach it. This simple idea has been concretized in several models, especially the PCK and TPACK models. The Pedagogical and Content Knowledge model, PCK [32], distinguished between content knowledge and pedagogical knowledge. Subsequently, the Technological, Pedagogical and Content Knowledge (TPACK) model [18] has been developed and is presented below, as it provides a more detailed breakdown of the different types of knowledge involved in educational action. Similar proposals had previously been made in the field of computing education, although without actually formulating a theoretical framework [12]. The TPACK model distinguishes between content knowledge, pedagogical knowledge and technological knowledge. These types of knowledge can be formulated separately, but they also intersect (see Fig. 1).

Content knowledge is the teacher's knowledge of the subject matter to be taught. This knowledge must be consistent with the knowledge accepted by subject matter experts, and varies with the educational stage and specialty of study. It includes facts, concepts, theories and frameworks, ways of determining evidence and truth, and working methods. For example, learning programming includes programming concepts and languages, as well as elements of programming style or basic testing techniques, among others.

Pedagogical knowledge is the teacher's knowledge of educational processes and methods, both of student learning and teacher teaching. It includes issues such as an understanding of the learning process in students, instructional planning, classroom management, teaching methods and assessment. For example, teachers must know active learning methods, such as problem-based learning or collaborative learning, and the conditions under which they should be applied so that the learner plays an active role cognitively and not only behaviorally.

Pedagogical content knowledge enables teachers to use pedagogical knowledge effectively for learning a particular subject. For example, pair programming can be used for collaborative learning or Parsons problems for individual study or assessment of programming. It also enables teachers to reinterpret the subject matter for effective teaching in specific contexts, finding alternative ways to present it, and adapting instructional materials to address students' preconceptions and misconceptions. This ability is important in adapting to learners of different ages or genders or with different cognitive or motivational profiles. For example, a computing teacher may use a "traditional" programming language (such as Java or C) to teach programming to computing students, but may prefer the use of Processing or Python if his/her students are in fine arts or statistics majors, respectively. Likewise, knowledge of the difficulties that each learner usually encounters in learning the operation of assignment or recursion allows the learner to anticipate them, adapting the content and corresponding educational activities.

Technological knowledge is the type of knowledge that changes most rapidly, especially in the use of digital technologies. This knowledge is comparable to digital competence in the use of digital media, but other technologies are also used. Some technologies in daily use have been around for centuries, such as the use of paper for writing, while others are more recent, such as the use of programmed cards to access premises.

Technological content knowledge is the teacher's knowledge of how technology has affected a subject, especially the representation and manipulation of information. For teaching computing, hardware and software technologies are often used. For example, one way to present programs and algorithms in a more tangible and, in principle, more understandable way is through visualization, resulting in program visualization or algorithm animation systems.

Technological pedagogical knowledge is the knowledge of effective ways of using a technology for teaching a certain subject, as well as its limitations. In particular, this knowledge is key to the effective use of various applications developed for education, such as simulators, serious games, etc. For example, algorithm animations are only effective if each student is cognitively active, which can be achieved in various ways, such as allowing students to control the forward and backward movement of an animation, allowing them to enter their own input data, or forcing them to answer questions during the animation.

Technological pedagogical content knowledge is the intersection of content, pedagogical and technological knowledge. Teachers who manage to position themselves at this intersection are able to develop effective educational action. There is no single way to combine the three aspects in a subject, but each person can do it differently. The combined use of the three aspects is dynamic and any change in one of these aspects forces a rebalancing of the other two. Technological change tends to be more frequent, but other changes are also frequent. For example, advances in computing technology may make a programming language or environment obsolete, in which case a change of programming language or environment may be required. Likewise, a new version may provide new possibilities that affect the exercises or practices of the subject. Likewise, if a change of content, such as programming language, is decided, it will force a restructuring of the technological and probably the pedagogical aspect.

The report is aimed to provide an introduction to the intersecting sectors of Fig. 1 referring to the teaching of computing, trying to illustrate these types of knowledge in their application to computing education, without focusing on any particular educational stage. Obviously, their application at a particular stage requires detailed analysis.

F. Didactics of computing

The section of didactics in the report is aimed at computing teachers in pre-university education (and their trainers, as well as those responsible for the design of such training), and presents key elements of the didactics of computing. Taking into account that this section occupies one third of the report, only a summary of the ideas is presented here, with special emphasis on explaining what were the objectives that led the working group to develop this section in this way.

1) Knowledge construction

The first part was oriented to talk about constructivism and active methodologies, but from a critical point of view since it is not enough to use active learning methods in any way, but they must be used in an appropriate way to be effective [28]. For example, the learner must be cognitively active for knowledge construction to occur, and not just behaviorally active [21]. Consequently, pure discovery learning methods are ineffective, and it is preferable to provide minimal guidance to the learner. In general, it cannot be said that there are better or worse methods. All learning methods have advantages and disadvantages [19]. What is desirable is for teachers to be aware of learning methods and to use them appropriately.

During the learning process, each learner builds mental schemas of knowledge and problem solving that are specific to the subject matter [35]. The availability of subject-specific problem-solving schemas is what differentiates experts and learners. The transition from one to the other is a long process, lasting years [9]. It has been conjectured that the time frame is ten years [38], so even a recent graduate is still far from reaching the degree of expert in his or her field.

Knowledge construction is not an easy process. On the one hand, knowledge always has a large memory component, but it must also have some degree of understanding so that it is not inert knowledge. Moreover, knowledge is built on previous knowledge and in a gradual way, so that between initial ignorance and final knowledge there are stages in which the knowledge under construction can be incomplete, ambiguous and even incoherent [24], which can be confusing for students. Therefore, two elements must be emphasized: motivation and detection by teachers of misconceptions developed by students.

2) Motivation

Computing is often perceived as difficult and abstract, as well as being burdened by a series of stereotypes that must be overcome with information. In different studies carried out by asking young people about their vision of computing, the words that appear most frequently when defining a computing professional are "man", "intelligent", "shy", "nerd", "geek" and "asocial". Regarding the work of people who are professionally involved in computing, the words that appear are "boring", "lonely", "cerebral", "dedicated" and "obsessive" [2][36][39].

Much of the stereotypes are associated with gender. Sinclair and Kavala [31] showed that the two main problems of gender bias in computing studies are: gender stereotypes themselves (computing is for boys) and cultural expectations (that family, or faculty discourage girls from studying this career). Working on eliminating these gender biases has been shown to have a very positive influence on female enrollment in STEM studies [6].

Despite these stereotypes, what most influences the choice of computing as a future career is how each person sees themselves doing these studies the so-called self-efficacy. According to Bandura, self-efficacy is made up of people's judgments about their abilities to achieve certain levels of performance [4]. As computing concepts have a reputation for being difficult, there is a view that to be able to engage in computing requires mathematical, programming, logical thinking, organizational, and entrepreneurial skills [27]. A person who does not visualize himself or herself with these skills will hardly choose this profession.

The study by Alshahrani, Ross & Wood [2] tells us that, in order to have a positive view of computing and influence the self-efficacy of each person, it is of vital importance to have lived a positive experience in computing subjects. The experience lived basically depends on the teacher, the approach of the subject and the rest of the students.

The motivation of students for any subject depends on them having a vision of its usefulness and feeling attracted to it. Therefore, if we want to train professionals in computing or in any field with knowledge of computing, we must attack stereotypes, work to ensure that students see themselves as capable of solving problems in a computing environment and develop well-planned subjects, with trained teachers who can create positive experiences with computing. Likewise, we must think about people who are not going to work professionally in computing but who are going to live their daily lives surrounded by computing and interacting with it. These people need training in the basics of computing, but they also need to see themselves as designers and managers, not just users. To do this they need to be motivated by computing, seeing themselves as capable of learning a basic knowledge of computing and of interacting with ease and confidence with devices and software (self-efficacy).

3) Mental models

In addition to motivation, the second problem to be considered by teachers is that each learner internally constructs his or her knowledge by building a mental model incrementally, so it is normal that this is incomplete, ambiguous and incoherent on many occasions. In contrast, the teacher's explanation of the subject matter is usually complete, precise and coherent. Moreover, it must be compatible with the consensus of experts in the subject matter (although it may be modified for better presentation, for example, in a simplified form or expressed by means of metaphors). Such an explanation is the conceptual model that is presented to the students.

It is difficult to determine the mental model of each student, since it is not easily observable. Fortunately, the number of mental models that can be constructed on a certain subject is limited. For example, the mental models that are constructed for understanding recursion have been studied [13]. Apart from the mental model corresponding to the actual execution of recursion in programming languages (a double "back and forth" process), there are other mental models that differ from each other in the parts of the execution that are not well understood by the learner, e.g., forgetting the back process or changing the order of operations to be performed between the results of successive recursive calls.

Since mental models are not tangible, a person's mental model is not said to be correct or not, but viable or non-viable to understand or predict certain phenomena. In fact, a mental model may be viable in some cases, but not in others. For example, a learner who has constructed a mental model of recursion similar to the execution of a loop will have a viable mental model for final (or tail) recursion, but not for other forms of recursion.

4) Difficulties in learning programming.

Programming is one of the most difficult skills to acquire [33][22][20], but it is still basic to learning computing. In order to make good learning possible, there are issues that need to be emphasized.

First, it is necessary to give greater importance to explaining the dynamic behavior of the programming language. It is not enough to describe the syntax and semantic constraints of the various elements of the language, but it is necessary to clearly describe its dynamic semantics, i.e., its effect on a hypothetical abstract or virtual machine [10][34].

It is also necessary to provide learners with patterns and antipatterns of code. That is, instead of leaving them without any guidance for program development, offer students patterns, which are general solutions for frequent tasks, e.g., how to traverse an array to perform some operation with all its elements [7]. The development of some mastery by learners responds to the development of mental models in the form of programming schemes that help them solve problems. The goal, therefore, of explicitly presenting code patterns is to assist learners in the development of such mental models. Also useful are code patterns to avoid, usually for responding to a bad programming style. These patterns are called anti-patterns.

The relationship between different ways of interacting with code has been investigated. There is evidence that students must first be able to predict the effect of parts of a program and to write the trace of their behavior. They should also be able to abstract the purpose of part of a code. Once successful in both tasks, learners are better able to develop programs [8][11]. Instructional methodologies based on this precedence of instructional activities have even been proposed and validated [40].

5) Cooperative learning, teaching methods and assessment

Since Piaget and other educational psychologists, it has been known that the interaction between peers who learn in a more symmetrical relationship is as important as the asymmetrical relationship between teachers and students. Cooperative learning is based on the work of small teams of students, generally of heterogeneous composition in performance and ability, using a structure of activity that ensures maximum equal participation, maximizing simultaneous interaction (that all people learn and learn to the best of their ability), in addition to learning to work as a team. One of the examples in the computing environment is using pair programming as a cooperative activity [23], but there are many others. Spencer Kagan [17] designed about 150 cooperative structures from which cooperative teamwork activities can be created. Project-based learning, problem-based learning and challenge-based learning can also be organized with teamwork, promoting autonomy and increasing motivation.

The student is the protagonist of the tasks, has the capacity of choice and, in addition, works in contextualized situations in everyday situations.

Programming is an exercise at the highest level of Bloom's revised taxonomy [3], but for incremental learning it is necessary to have exercises at lower levels. There are some methods developed specifically for the discipline, such as predictive exercises [20], Parsons problems [26] or scaffolded programming exercises (e.g. Lister et al. [20]). This list is not exhaustive, and several taxonomies have been proposed to classify programming exercises, showing a wide range of exercises [5][29][30].

Another issue of great importance is assessment, which must be aligned with educational objectives and instructional activities. The importance of feedback is well known, as each student needs to know whether what he or she is doing is correct or not and why in order to learn from his or her own mistakes.

In addition, feedback must be fast so it is interesting to work with self-assessment or peer assessment (usually with the use of rubrics), in addition to automatic correction systems [25] through tests with data sets.

G. Recommendations form the working group

Some reflections of the working group in charge of the report are here presented, divided into three blocks: those related to computing as a discipline and its training across the world, those related to the didactic aspects of computing and, finally, those related to teacher training in computing, which has been detected as one of the main obstacles (both at national and global level) for the integration of computing in the K-12 curriculum.

In relation to computing as a discipline and universal training in computing, the working group wants to highlight:

- Use of the right term when speaking about computing. The exact term is *Informática* in Spanish, *Computing* in US English while *Informatics* is the common word used in Europe. Lately there has been a tendency to use other terms as substitutes for computing, such as computational thinking, artificial intelligence or digitalization. At the general and academic level, the term computing (*informática* in Spanish) is much more precise, so its use should be actively vindicated. Please, notice that across this paper we have used the English term *Computing* as it is the one recommended by the Computing Curricula of ACM and IEEE, but each country should clarify which is the most appropriate word in its environment.
- A differentiation between digital competence and computing. Despite the clear difference between the two terms, they continue to be confused, sometimes out of ignorance, sometimes in an interested way.
- Promote teaching computing from an early age. Education in computing should be learned from an early age, in an appropriately modulated way for each age. Let us remind that the demand for the study of computing is mainly for the general good of society and citizenship, broadening their general culture.
- Need for computing specialists, with a greater presence of women. A secondary objective of a greater knowledge of computing in the pre-university age group is to facilitate the awakening of technological vocations. This issue is especially critical among women, whose presence in computing studies has been on a downward trend, and is currently very low.

Regarding didactic aspects, both general and specific to computing, the working group want to propose:

- Increase motivation for computing. Motivation for any subject is a fundamental factor for its learning. Knowledge of computing would reduce the existing stereotypes about it. A well-trained teacher could be more motivated towards computing teaching and could design positive computing-related experiences that reduce such stereotypes among students, and that include a variety of interests and tastes.
- Use active methodologies. While these learning methods can be applied to any subject, they are especially relevant in computing. Although memorization is necessary, it should be remembered that computing is basically a problem-solving discipline.
- Promote correct computing learning. Good computing training for teachers will lead to better computing teaching, and therefore a proper response to students' preconceptions and prevention of the development of misconceptions.

Teacher training in computing is one of the main obstacles detected, becoming a crucial aspect. The working group therefore suggests:

- Train teachers in content, pedagogical and technological knowledge. Teacher training in Computing requires their training in the three components of content knowledge, pedagogical knowledge and technological knowledge, and all their intersections.
- Didactics of computing and programming should be integrated in Early Childhood and Primary Education Degrees, just as the didactics of other subjects is introduced in these Degrees.
- Develop actions for teachers of Secondary Education. These teachers opt for a specialty according to the degree in which they have graduated. An action from the computing field itself would be to integrate optional subjects in education in the Computing Engineering Degrees, to encourage vocations among this group, traditionally distanced from education. In addition, the Master's Degree in Education should include a specific Computing Module, with its associated pedagogical training and practicum.

- An Update of syllabi for competitive examinations. These syllabi should incorporate scientific advances and new approaches that have arisen in recent decades.

H. Nine proposals for improvement oriented to policy makers

1.- Differentiate between digital competence and computing. Both are necessary although there is in the current curriculum an emphasis on digital competence.

2.- To reclaim the use of the term "computing", avoiding the use of attractive terms that can quickly go out of fashion.

3.- To teach computing from an early age and in compulsory courses, adapted to each age group.

4.- To move towards a fairer, more inclusive and social computing.

5.- To train computing teachers in K-12 studies in three types of knowledge: content (what is computing), technological (how to use it in their environment) and pedagogical (how to teach it).

6.- To promote the use of active learning methods for the teaching-learning process of computing.

7.- To incorporate the didactics of computing as part of the curriculum in the Primary Education Degree (the studies that, in our country, lead to qualify teachers to become teachers of these studies).

8.- Promote the incorporation of a Computing Didactics subject in the undergraduate studies in Computing Engineering Degree.

9.- To update the syllabus of the national exams for access to the qualification to become a secondary education teacher in order to have a deep knowledge of the new approaches arising from computing.

IV. Conclusions

Nowadays there is a consensus on the importance of computing in society and the need to incorporate it into education at all levels. This incorporation is carried out differently according to the legislation of each country. The associations of computing professionals (and especially academic computing education) should reflect on how to help governments and legislators to implement computing education in the best possible way given the characteristics of education in each country.

This paper presents the reflection document carried out by two of the major Spanish academic societies related to computing. Both the process carried out and the result may be useful for other national associations wishing to carry out similar work.

Regarding future work, it is necessary to follow the implementation of the new Education Law in Spain and to study the degree of alignment between the report and the implementation of the Law, analyzing the impact of the of the Law on computing knowledge in Spanish students.

Acknowledgment

The authors would like to thank the presidencies of SCIE and CODDII for entrusting us with the drafting of this document. We would also like to thank the seven experts (Antonio Bahamonde Rionda, Sanja Dabic Radisc, Juan Manuel Dodero Beardo, Ángel Fidalgo Blanco, Antonio Hernández Fernández, Francisco Javier Soriano Camino, and Francisco José Vico Vela) for the valuable comments received, which have improved the document in terms of rigor and readability. This work has been funded by the Departament de Recerca I Universitats de la Generalitat de Catalunya – 2021 SGR 01412

References

[1] ACM-IEEE (2020). Computing Curricula 2020. https://doi.org/10.1145/3467967

[2] Alshahrani A., Ross, I. & Wood, M.I. (2018). “Using social cognitive career theory to understand why students choose to study computer science”. ICER’18. https://doi.org/10.1145/3230977.3230994.

[3] Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich,P. R., Raths, R. & Wittrock, M. C. (2001) A Taxonomy for Learning, Teaching and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives. Pearson Education, 2001.

[4] Bandura, A. (1986). Social foundations of thought and action: A social cognitive theory. 1986: Prentice-Hall.

[5] Bower, W. (2008) “A taxonomy of task types in computing”. En Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2008), pp. 281285. https://doi.org/10.1145/1384271.1384346.

[6] Charlesworth, T.E.S. & Bana-ji, M.R (2019). “Gender in Science, Technology, Engineering, and Mathematics: Issues, Causes, Solutions” Journal of Neuroscience. 2019, 39 (37):7228-7243. https://doi.org/10.1523/JNEUROSCI.0475-18.2019.

[7] Clancy, M. J. & Linn, M. C. (1999) “Patterns and pedagogy”. En Proceedings of the 30th SIGCSE Technical Symposium on Computer Science Education (SIGCSE 1999), pp. 37-42. https://doi.org/10.1145/384266.299673.

[8] Clear, T., Whalley, J., Robbins, P., Philpott, A., Eckerdal, A., Laakso, M., & Lister, R. (2011). “Report on the final BRACElet workshop”. Journal of Applied Computing and Information Technology, 15(1). http://www.citrenz.ac.nz/jacit/JACIT1501/2011Clear_BRACElet.html.

[9] Dreyfus, S.E. & Dreyfus H.L. (1986) Mind over Machine. Nueva York, NY: Free Press.

[10] du Boulay, B., O’Shea, T. & Monk, J. (1981) “The black box inside the glass box: Presenting computing concepts to novices”, International Journal of Man-Machine Studies, 14(3):237-249, https://doi.org/10.1016/S0020-7373(81)80056-9.

[11] Fowler, M., Smith IV, D.H., Hassan, M., Poulsen, S., West, M. & Zilles, C. (2022) “Reevaluating the relationship between explaining, tracing, and writing skills in CS1 in a replication study”, Computer Science Education, 32(3):355-383, https://doi.org/10.1080/08993408.2022.2079866.

[12] Gal-Ezer, J. & Harel, D. (1998) “What (else) should CS educators know?”, Communications of the ACM, 41(9):77-84. https://doi.org/10.1145/285070.285085.

[13] Götschi, T., Sanders, I. & Galpin, V. (2003) “Mental models of recursion”. En Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2003), pp. 346-350 https://doi.org/10.1145/611892.612004.

[14] Heintz, F., Mannila, L., & Färnqvist, T. (2016). “A review of models for introducing computational thinking, computer science and computing in K-12 education”. 2016 IEEE Frontiers in Education conference (FIE) (pp. 1-9). https://doi.org/10.1109/FIE.2016.7757410.

[15] Informatics for All. The Informatics Reference Framework for School. Permanent URL: https://www.informaticsforall.org/the-informatics-reference-framework-for-school-releasefebruary-2022/

[16] Informatics for All. Designing and Implementing a Concrete Informatics Curriculum for School. Permanent URL: https://www.informaticsforall.org/designing-and-implementing-a-concrete-informatics-curriculumfor-school/

[17] Kagan, S. (2001) Kagan Structures and Learning Together: What is the Difference? San Clemente, CA: Kagan Publishing. Kagan Online Magazine. www.KaganOnline.com.

[18] Koehler, M.J. & Mishra, P. (2009) “What is technological pedagogical content knowledge?”, Contemporary Issues in Technology and Teacher Education, 9(1):60-70 https://www.learntechlib.org/primary/p/29544/.

[19] Laurillard, D. (2012) Teaching as a Design Science: Building Pedagogy Patterns for Learning and Technology. Londres, RU: Routledge.

[20] Lister, R., Adams, E.S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J.E., Sanders, K., Seppälä, O., Simon, B. & Thomas, L. (2004) “A multi-national study of reading and tracing skills in novice programmers”, ACM SIGCSE Bulletin, 36(4):119-150, Dec. 2004 https://doi.org/10.1145/1041624.1041673.

[21] Mayer, R.E. (2004) “Should there be a three-strikes rule against pure discovery learning? The case for guided methods of instruction”, American Psychologist, 59(1):14-19. https://doi.org/10.1037/0003066X.59.1.14.

[22] McCracken, M., Kolikant, Y.B.-D., Almstrum, V., Laxer, C., Diaz, D., Thomas, L., Guzdial, M., Utting, I., Hagan, D. & Wilusz, T. (2001) "A multinational, multi-institutional study of assessment of programming skills of first-year CS students", ACM SIGCSE Bulletin, 33(4):125-140, Dic. 2001. https://doi.org/10.1145/572133.572137.

[23] McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). “Pair programming improves student retention, confidence, and program quality”. Communications of the ACM, 49(8), 90-95. https://doi.org/10.1145/1145287.1145293.

[24] Norman, D. (1983) “Some observations on mental models”. En: Mental Models. D. Gentner, A. Stevens (eds), pp. 7-14. Hillsdale, NJ: Erlbaum. Novak, M., Joy, M. & Kermek, D. (2019). “Source-code similarity detection and detection tools used in academia: A systematic review”, 19(3):27, may 2019, https://doi.org/10.1145/3313290.

[25] Paiva, J. C., Leal, J. P. & Figueira, Á. (2022). “Automated assessment in computer science education: A state-of-the-art review”, ACM Transactions on Computing Education, 22(3):34, june 2022, https://doi.org/10.1145/3513140.

[26] Parsons, D. & Haden, P. (2006) “Parsons’ programming puzzles: A fun and effective learning tool for first programming courses”. En Proceedings of the 8th Australasian Conference on Computing in Education (ACE 2006), pp. 157-163 https://dl.acm.org/doi/abs/10.5555/1151869.1151890. 29

[27] Peters. A-K. ”Students’ Experience of Participation in a Discipline—A Longitudinal Study of Computer Science and IT Engineering Students”. ACM Transactions on Computing Education. 19(1):1-28 2019. https://doi.org/10.1145/3230011

[28] Prince, M. (2004) “Does active learning work? A review of the research”, Journal of Engineering Education, 93(3):223-231. https://doi.org/10.1002/j.2168-9830.2004.tb00809.x.

[29] Ragonis, N. (2012) “Type of questions – The case of Computer Science”, Olympiads in Informatics, 6:115-132. https://www.researchgate.net/profile/NoaRagonis/publication/241011964_Types_of_Questions_in_Computer_Science_Education/links/56683d2f08ae8905db8f7099/Types-of-Questions-in-Computer-Science-Education.pdf

[30] Ruf, A., Berges, M., Hubwieser, P. (2015). “Classification of programming tasks according to required skills and knowledge representation”, En Brodnik A, Vahrenhold J (eds.) Proceedings of the 8th International Conference on Informatics in Schools (ISSEP 2015). Springer, LNCS 9378, pp. 57–68. https://doi.org/10.1007/978-3-319-25396-1_6.

[31] Sinclair, J. & Kalvala, S. (2015). “Exploring societal factors affecting the experience and engagement of first year female computer science undergraduates”. Koli Calling ‘15. https://doi.org/10.1145/2828959.2828979.

[32] Shulman, L.S. (1986) “Those who understand: Knowledge growth in teaching”, Educational Researcher, 12(2):4-14. https://doi.org/10.2307/1175860.

[33] Soloway, E. (1983) "Cognitive strategies and looping constructs: An empirical study", Communications of the ACM, 26(11):853–860, Nov. 1983. https://dl.acm.org/doi/pdf/10.1145/182.358436.

[34] Sorva, J. (2013) “Notional machines and introductory programming education”, ACM Transactions on Computing Education, 13(2):8, 2013, https://doi.org/10.1145/2483710.2483713.

[35] Sweller, J., Ayres, P. & Kalyuga, S. (2011) Cognitive Load Theory. Nueva York, NY: Springer.

[36] Taylor-Smith, E., Smith, S. & Smith, C. (2019). "Identity and Belonging for Graduate Apprenticeships in Computing: The experience of first cohort degree apprentices in Scotland." ITiCSE 2019. https://doi.org/10.1145/3304221.3319753.

[37] Velázquez-Iturbide, J.A., Llorens-Largo, F., López, D. and Marqués Andrés, M. “Informe CODDII/SCIE sobre formación del profesorado y didáctica de la informática en etapas preuniversitarias” (In Spanish). Permanent URL: https://rua.ua.es/dspace/handle/10045/131535

[38] Winslow, L.E. (1996) “Programming pedagogy - A psychological overview”, SIGCSE Bulletin, 28(3):1722 25. https://doi.org/10.1145/234867.234872.

[39] Wong, B. (2017). “‘I’m good, but not that good’: digitally-skilled young people’s identity in computing”, Computer Science Education, 26(4):299-317. 2017 https://doi.org/10.1080/08993408.2017.1292604.

[40] Xie, B., Loksa, D., Nelson, G.L., Davidson, M.J., Dong, D., Kwik, H., Hui Tan, A., Hwa, L., Li, M. & Ko, A.J. (2019) “A theory of instruction for introductory programming skills”, Computer Science Education, 29(2-3):205-253, https://doi.org/10.1080/08993408.2019.1565235