- ORIGINAL ARTICLE -

Experiences in Learning Problem-Solving through Computational Thinking

Experiencias en el Aprendizaje de la Resolución de Problemas mediante Pensamiento Computacional

Jacqueline M. Fernández, Mariela E. Zúñiga, María V. Rosas and Roberto A. Guerrero

Laboratorio de Computación Gráfica, Departamento de Informática, FCFMyN,
Universidad Nacional de San Luis, Ejército de los Andes 950 (San Luis, Argentina)
{jmfer, mezuniga, mvrosas, rag}@unsl.edu.ar

Abstract

Computational Thinking (CT) represents a possible alternative for improving students' academic performance in higher level degree related to Science, Technology, Engineering and Mathematics (STEM). This work describes two different experimental proposals with the aim of introducing computational thinking to the problem solving issue. The first one was an introductory course in the Faculty of Physical, Mathematical and Natural Sciences (FCFMyN) in 2017, for students enrolled in computer science related careers. The other experience was a first attempt to introduce CT to students and teachers belonging to not computer related faculties at the National University of San Luis (UNSL). Both initiatives use CT as a mean of improving the problem solving process based on the four following elementary concepts: Decomposition, Abstraction, Recognition of patterns and Algorithm.

The results of the experiences indicate the relevance of including CT in the learning problem solving issue in different fields. The experiences also conclude that a mandatory CT related course is necessary for those careers having computational problems solving and/or programming related subjects during the first year of their curricula. Part of this work was presented at the XXIII Argentine Congress of Computer Science (CACIC).

Keywords: Computational Thinking, Higher Education, Problem Solving.

Resumen

El Pensamiento Computacional (PC) representa una alternativa posible para la mejora del rendimiento académico de los estudiantes de nivel superior en áreas relacionadas a la Ciencia, Tecnología, Ingeniería y Matemáticas (STEM; por sus siglas en inglés). Este trabajo describe dos propuestas experimentales diferentes con el propósito de introducir el pensamiento computacional en la resolución de problemas. En el primero se describe un curso introductorio realizado a comienzos del 2017 en la Facultad de Ciencias Físico Matemáticas y Naturales (FCFMyN) destinado a estudiantes matriculados en carreras relacionadas a las ciencias de la computación. La otra experiencia fue una primera aproximación para introducir conceptos del PC a estudiantes y docentes pertenecientes a facultades de la UNSL que no están relacionadas directamente a la computación. Ambas iniciativas se valen del PC como medio para mejorar el proceso de resolución de problemas basado sobre cuatro pilares elementales: Descomposición, Abstracción, Reconocimiento de patrones y Algoritmo. Los resultados de las experiencias dan cuenta de la relevancia de incluir el PC en el aprendizaje de la resolución de problemas en diferentes áreas. Además, se evidencia la necesidad de contar con un curso introductorio en modalidad obligatoria en aquellas carreras que tengan materias relacionadas a la resolución de problemas computacionales y/o programación durante el primer año de su plan de estudio. Parte de este trabajo fue presentado en el XXIII Congreso Argentino de Ciencias de la Computación (CACIC).

Palabras claves: Pensamiento Computacional, Educación Superior, Resolución de problemas.

1. Introduction

The economic, political, technological and cultural context presents new challenges to the educational

Citation: J. M. Fernández, M. E. Zúñiga, M. V. Rosas and R. A. Guerrero. "*Experiences in Learning Problem-Solving through Computational Thinking*", Journal of Computer Science & Technology, vol. 18, no. 2, pp. 136-142, 2018.
DOI: 10.24215/16666038.18.e15
Received: February 15, 2018 **Revised:** April 21, 2018 **Accepted:** May 14, 2018.

system causing changes and restructurings in its pedagogical proposals. These changes become more evident at higher level education. The current trend determines that new professionals are trained to deal with a wide variety of situations, also to face problems that may not yet exist. Therefore, educational institutions need to adapt to the requirements of the actual world, preparing the future generations in thinking skills as a complement to specific contents which were the teaching axis so far [1].

The university, as part of the educational system, cannot be apart to current challenges, having to coach its future graduates in a more generalist way. Focusing them on problem-solving practices and encouraging creativity to perform satisfactorily in their professional field. The working world is undergoing transformations; emerging new unknown or unimaginable professions until a short while ago. The gap between what is expected of new professionals and how they graduate from the university can be reduced by focusing efforts on training students in routines and thought practices and not on specific knowledge, and it should be a constant from the beginning. In this context, it could be considered that carrying out these actions from the first year of any university career would contribute to the adaptation and permanence of the students, improving their academic performance during the whole grade formation process [2,3]. In addition, different authors have observed the recurrent problem of dropping out in the first years and the lack of motivation in careers related to Science and Engineering [4,5]. To deal with this situation different interventions have been proposed: introductory courses with specific tools [7,4] and programming practices [6], the articulation of middle and high level education [8], among others. These initiatives, using practices limited to specific problems and with objectives mostly oriented to the learning of programming, have showed satisfactory results.

On the other hand, the CT has developed a strong theoretical advance in recent years. Several authors state the relevance of promoting its development from a very early age, so this could improve significantly the students' abilities to face and solve different kinds of problems: academic, personal, social, among others [9,10,11]. Jeannette Wing defines the CT as the set of thought processes involved in the definition of problems and the representation of their solutions, so that these solutions can be effectively executed by an information processing agent (human, computer or combination of both). Therefore, the CT is based on solving problems using basic concepts of programming [12,13].

During the problem-solving process the CT comes into play when:

1) The problem is broken down into smaller subproblems; called the **Decomposition** process. It consists of splitting a complex problem (situation or task) into smaller and more manageable subproblems whose combined solutions provide the solution to the general problem.
2) The attention is focused on the most important characteristics; called **Abstraction** process. The aim is to capture the essence of the problem by filtering the non-fundamental characteristics and preserving the most relevant features, in order to create a simplified representation or model.
3) The knowledge of alike problems solved previously is used; called the **Recognition of Pattern** process. It consists of looking for similarities between different problems and within the same problem. It is about finding patterns of a complex problem (or subproblem) with an analogous one already analyzed and solved effectively. The more patterns are recognized, the easier and faster the general task of solving problems will be.
4) An action plan to be executed is elaborated: **Algorithm.** It consists of a set of clear and precise instructions, which are identified and planned in a certain order for the resolution of a problem.

Since computer teaching has been defined as strategic for the educational system from a very early age, it is considered relevant to train new generations of teachers to the development of CT.

Therefore, at the moment, from the University two working lines must coexist for different target groups:

- prospective learners and the first-year students without training in CT.
- teachers from middle and high levels and teacher-training students.

The experiences presented in this work consisted of promoting practices for the development of thinking skills in both lines.

The first of these was carried out within the framework of the research project "Strategies for the Improvement of the Teaching of Programming for Recruitment of Science and Engineering Careers" and in accordance with the objectives set out in its definition [1,14,15]. Whereas the second initiative appears as an alternative aimed to the presentation of the basic notions of CT and its possibility of application for solving problems in domains not related to Computer Science. Both proposals focused on problem-solving activities that develop the CT core pillars, using as support different programming tools and incorporating the ludic model as an element of motivation at different levels of complexity.

The organization of this paper is as follows: Section 2 describes how the course to incoming students of FCFMyN was designed and implemented. The characteristics of the sample, data collection instruments and analysis of the results obtained from questionnaires carried out are also described in this section. The specifications of the workshop modality for teachers and students from other disciplines not related to Computer Science are given in Section 3. Finally, the main conclusions and future works are summarized in Section 4.

2. Course for FCFMyN prospective learners

The "Start to Programming" (SP) course was designed to investigate what CT related skills the students unconsciously activate in learning programming in Science and Engineering careers. It was intended for FCFMyN admitted students in 2017, with the aim of introducing them in the practice of solving problems using CT fundamentals.

2.1. Course Description: Tools and Methods

During admission period, students took two courses (leveling and diagnostic assessments) in a mandatory way. Due to these courses, the B-learning modality was chosen for the SP course organization. It was structured in three whole weeks of activity, finishing each one with an in-person class. A virtual classroom on the Moodle platform[2] covered the e-learning mode [16]. It facilitated the communication, organization, and distribution of the resources needed for the course.

Teachers responsible for the course developed the study material and made it available weekly to students. The material set out the objectives to achieve, the theoretical framework and the activities to be carried out to consolidate the concepts. Likewise, different digital resources (audio, video, web link) complementing the study material were incorporated.

The programmed activities involved an integrating modality and increasing complexity to implement the contents and promote the development of the skills proposed in the objectives. Communication between participants of virtual classrooms was encouraged through forums to debate and exchange ideas, knowledge and doubts. Use of private messaging was specially designed for communication among the teaching team and each student.

Contents and activities were organized as follows:

Week 1

- Computational and non-computational problems resolution.
- Computational Thinking Pillars: decomposition, pattern recognition, abstraction and algorithm

Week 2

- Problem-solving. Strategies.
- Control Structures: Sequential, Repetitions (simple and conditional).

Week 3

- Control Structures: Conditional.
- Variables. Parameterized Programming.

In order to obtain the necessary information to carry out a balance of the achievements attained throughout the course and evaluate the impact and relevance of the experience, two questionnaires were planned (Pre and Pos questionnaires are described in Section 2.2 below). One questionnaire was taken before starting SP course activities and the other after finishing them. The first questionnaire intended to investigate the students' prior knowledge in order to focus activities to be developed during the course. The second questionnaire intended to observe the appropriate use of the terminology addressed in the course, as well as the understanding of the new concepts. At the end of the third and last meeting, a dialogue space was set up where participants could verbally assess their experience. From this exchange, some criteria were recorded to take into account the contents studied, their relevance, the proposed modality and the integration of the contents of this course with the curricular subjects of the first year in each career.

2.2. Course Evaluation

The experience was planned for the 2017 academic year, taking a group of 429 pre-enrolled students as a population (Column **C1** - Table 1). A student gets a pre-registered status since the moment he initiates the registration process by Web, with no need to complete the process or confirm the enrollment.

A first sample version came from the analyzed and filtered data (after eliminating repeated or erroneous records) declared from potential students. Then, only those people who had expressed their intention to study careers having some subject related to programming and/or problem-solving in the first year of their curricula were selected.

The new sample was composed of 100 persons

[2] UNSL Virtual Campus: http://campus.unsl.edu.ar

who were contacted by email (Column **C2** - Table 1). From this, 50% of the students answered agreeing to participate voluntarily in the experience (Column **C3** - Table 1). Table 1 is completed with the number of students who started the course and those who finished it, Column **C4** and **C5** respectively in Table 1.

Table 1 Sample size in the different instances

C1	C2	C3	*C4*	*C5*
429	100	50	40	22

As was said, data collection was done in two different moments and through two questionnaires, one at beginning of the course (*Pre* questionnaire) and the other at the end of the course (*Pos* questionnaire). Both questionnaires maintained similar characteristics and were structured in eight practical exercises about CT pillars presented in the course. Each pillar had two practical exercises and the answer to each one was rated according to the following criteria: *Complete*, *Incomplete*, *Incorrect* and *No answer*. The analysis was organized by levels from a general evaluation of the experience impact to a more specific study on each proposed exercise.

Figure 1 shows the resulting percentages organized by type of answer in the *Pre* (a) and *Pos* (b) questionnaires. From the comparison between (a) and (b), it could be observed that values achieved in the *Pos* questionnaire improved with respect to those obtained in the *Pre* questionnaire. The percentage of unresolved (*No answer*) or *Incorrectly* resolved exercises was reduced by approximately 50%, increasing the percentage of *Complete* or *Incomplete* exercises. The *Complete* answer rate increased by 6%, while the value of *Incomplete* answers increased by 3%. The students tried to solve 90% of the exercises which means that they understood the statement of the problem. A very significant percentage of the students knew how to solve the problem or understood the problem definition but did not reach the solution.

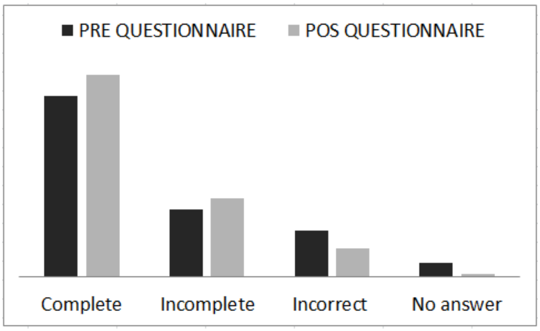
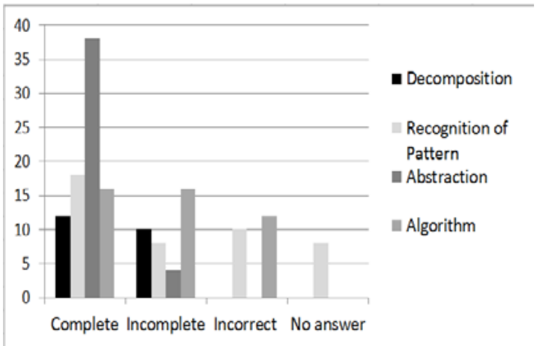
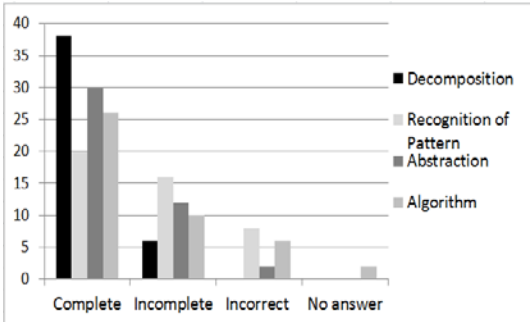


Fig. 1 Comparison between Pre and Pos questionnaires, global totals by answers type

Figure 2 presents the results of *Pre* questionnaire (a) and *Pos* questionnaire (b) grouped by answer type identifying the different pillars. As shown–in graph (a), the answers of type *Complete* for the exercises associated with the **Abstraction** pillar were clearly the best solved. Meanwhile, in **Decomposition**, **Recognition of Patterns** and definition of the **Algorithm** it is noticeable that, although students found greater difficulty in solving them, their performance was similar and relatively good. In graph (b), the *Complete* type answers report a more homogenous distribution in different pillars than in graph (a) case, and improving the performance in all of them. **Decomposition** is the first pillar in which students improved significantly followed by **Abstraction** where they got better performance. From this, it is reasonable to infer that students enhanced their analysis of the problem, which allowed them to identify subproblems of the whole problem and ensure a good performance in remaining pillars.



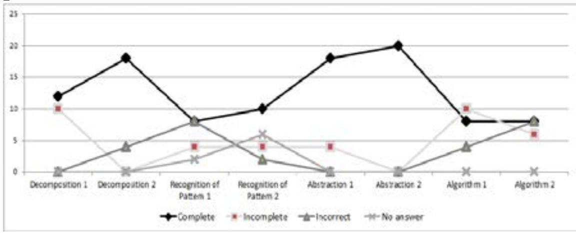
(a)



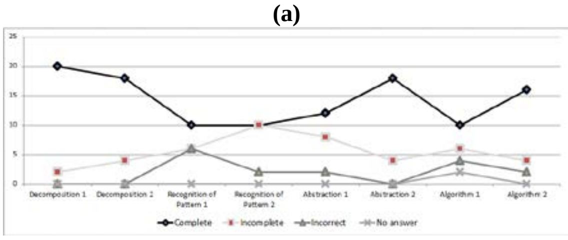
(b)

Fig. 2. Comparison between Pre(a) and Pos(b) questionnaires, totals by each pillar and answer type

Figure 3 shows the number of answer types in the *Pre* (a) and *Pos* (b) questionnaires where the results by exercise can be interpreted in more detail. At a first look, the *Complete* answer type is highlighted in the *Pos* questionnaire, since they are separated from the rest of answer types in most of the exercises. In the same way, it is clearly noted that the *No answer* type in the *Pos* questionnaire was recorded in a single exercise. Deeper analyzes comparing it in respect to each exercise allows visualizing those that showed a drop in the *Pos* questionnaire (according to the number of answers of all type). This leads to the need to investigate possible causes.



(a)



(b)

Fig. 3. Comparison between Pre and Pos questionnaires each exercise by answer type

3. Workshop for teachers and students of other disciplines, not computer science related

According to Jeannette M. Wing, "*Teaching computational thinking could not only inspire future generations to enter the field of Computer Science given the intellectual adventure, but it would benefit people in all fields*" [12]. From this, teachers must know the significance of CT, *what* contents have to be taught, and *how* they have to be taught.

The proposed workshop, called "*Thinking Computationally: how, when and where? And ... who?*" was held in two days of three hours each at the "IV Latin American Congress of Art, Education, Communication and Speeches" developed at UNSL in August, 2017.

The sample size of the workshop was relatively small, consisting of 21 participants mostly young people. 80% of the participants were female. This workshop group was formed by teachers from middle and high levels and teacher-training students. None of the participants belonged to the Computer Sciences, but came from the humanistic and mathematics areas. All of them had no experience in programming neither knowledge about CT.

The workshop presented the basic notions of CT and its possible applicability for problems solving in domains not related to Computer Science. CT competencies are not exclusive to computer experts or students from Informatics related areas [17]. Every person, day-to-day and implicitly, gives a solution to situations in a natural way using CT concepts.

The developed concepts were:

● Theoretical and operational concepts of CT, understanding this as the set of thought processes involved in the definition of problems and the representation of their solutions. Those solutions can be effectively executed by an information processing agent (human, computer or combination of both).

● Introduction to the CT pillars, working on the decomposition of a problem, the recognition of problem (or sub problems) patterns with similar ones, using abstraction to find the most relevance characteristics of the problem, modeling a representation and, based on it, creating and executing an algorithm for the effective solution.

● Practices in **Decomposition**, **Abstraction**, **Recognition of Patterns** and **Algorithms** that include the main CT operational pillars concepts worked in the workshop.

● Generation of a space to think about feasibility and relevance of include CT in different disciplines.

The participants respond a simple questionnaire, which helps to determine previous conceptions of teachers and students about CT and its applicability in the resolution of problems of a general nature. The same questionnaire was taken in two different moments, one at the beginning of the workshop (*PreW*) and other at the end of it (*PosW*). They enabled to observe possible changes in participants' know-how.

The questionnaire included the four questions motivating the selected title for the workshop: *HOW to apply the CT for solving a general and everyday problem*? *WHEN and WHERE is the CT used to solve these types of problems*?, and *WHO can use and develop the CT as a method when solving a problem*?.

From analyzing the first question answers in *PreW* questionnaire, it arises that CT is applied through ordered processes of thoughts, logical processes, procedures based on sequences and data analysis. In *PosW* questionnaire the answers made reference to some or all of the pillars presented in the workshop as a mean for the application of CT in solving problems.

Second question answers didn't show a significant difference between *PreW* and *PosW* questionnaire. Most replies mentioned that, whenever a new problematic situation arises it is possible to apply CT.

The answers to *PreW* questionnaire related to *WHERE the CT is used to solve these types of problems?* The educational field was stated as the only place of application. On the other hand, answers in *PosW* argued that CT can be applied in any environment.

Finally, opinions in the last question mainly pointed that only people who work in a purely computerized context and those who teach and learn subjects related to technology can use CT. This conception was strongly modified in the post-workshop answers since all the participants indicated that all people can use and develop CT in order to solve a problematic situation.

4. Conclusions and future work

We observed a low academic performance, a high percentage of desertion, and a significant dropping out of the students in the first year of careers related to computer science during the period 2013-2016. This should be taken into account due to its impact on last years and the number of professionals who graduate from these careers. The course for prospective learners described in Section 2 intends to determine if the development of CT improves problem-solving. On the other hand, the workshop was addressed to work this point with teachers from middle and high levels and teacher-training students (training teachers of UNSL prospective students).

The analysis of collected data suggests that experiences were positive. During the course, we observed from questionnaires a better answer to problem-solving. Furthermore, after the workshop, the educators indicated that are likely to include CT to solve the problem in different academic areas.

The conclusions derived from those experiences can be categorized considering: our point of view and learners' one.

The experiences enabled us to:

- Identify necessary skills of students enrolled in computer sciences related careers.
- Inquiry the teachers' willingness to involve in educative innovation practices.
- Get to know the institutional requirements to formalize this kind of academic approach.
- Take part in a space for exchange that would enable links with students and teachers. We were able to build an educational relationship with prospective students of our subjects to work introductory core concepts and contribute their university life adapt. We could share with middle school teachers assisting to workshop the feedback between both levels. We exchange ideas about the own skills of the students and those that could be necessary to develop before start university.

The experiences allowed to the assistants:

- enrich their previous knowledge about problem-solving through CT.
- recognize the potential benefits of CT application in different academic areas.

In particular, the workshop assistants indicated the relevance of creating a space where different areas meet to define new tendencies to improve students' academic performance. Finally, we are currently working to design and implement the course described as mandatory for those careers having computational problems solving and/or programming related subjects during the first year of their curricula; like others introductory courses such as Mathematics and Reading Comprehension. The methodology applied and the topics involved will be the same as the course described here but more comprehensive. We plan to do this strengthening the practical part with new exercises and with motivational tools for participation improvement. As well, it would be interesting to implement a follow-up project to register first-year university students' academic performance.

Competing interests

The authors have declared that no competing interests exist.

References

[1] Rosas M., Zúñiga M., Fernández J., Guerrero R., "El Pensamiento Computacional en el Ámbito Universitario". *Proceeding of XIX Workshop de Investigadores en Ciencias de la Computación* (WICC 2017), Instituto Tecnológico de Buenos Aires, Buenos Aires, Argentina. pp.696-699 (2017)

[2] *Bit & Byte*. Referentes de diferentes universidades argentinas opinan sobre la formación de profesionales informáticos. Año 2, no. 4. ISSN: 2468-9564. pp. 23-28 (2016)

[3] Simari G., "Los fundamentos computacionales como parte de las ciencias básicas en las terminales de la disciplina Informática". *Proceeding of VIII Congreso de Tecnología en Educación y Educación en Tecnología*. (TE&ET 2013) UNSE. Santiago del Estero. ISBN 978-987-1676-04- 0. ID 5522. (2013)

[4] Szpiniak D., Rojo G., "Enseñanza de la programación." *Proceeding of TE&ET: Revista Iberoamericana.* ISSN 1851-0086. Vol. 1. Número 1. pp. 71-78. (2006).

[5] Dapozo G., Greiner C., Pedrozo Petrazzin G., Chiapello, J., "Investigación para fortalecer actividades de promoción y retención de alumnos en carreras de Informática". *Proceeding of XX Congreso Argentino de Ciencias de la Computación*. (CACIC 2014) 1a ed. San Justo. Universidad Nacional de La Matanza. Buenos Aires. ISBN 978-987-3806-05-6. pp. 72-83. (2014).

[6] Compañ-Rosique, P., Satorre-Cuerda, R., "Enseñando a programar: un camino directo para desarrollar el pensamiento computacional", *Proceeding of RED-Revista de Educación a Distancia*, 46(11). Available at 15-Sept.-2015 DOI: 10.6018/red/46/11 http://www.um.es/ead/red/46/faraon_et_al.pdf (2015)

[7] Dapozo G., Greiner C., Petris R., "Herramientas lúdicas como apoyo a la enseñanza de la programación." *Proceeding of XXII Congreso Argentino de Ciencias de la Computación* (CACIC 2016). Universidad Nacional de San Luis. San Luis. pp. 1311-1319 (2016)

[8] Dapozo G., Petris C., Greiner C., Espíndola M., Company A., López M., "Capacitación en programación para incorporar el pensamiento computacional en las escuelas". *Proceeding of XI Congreso de Tecnología en Educación y Educación en Tecnología* (TE&ET 2016). ISBN 978-987-3977-30-5. Universidad de Morón. Buenos Aires. pp. 194-203 (2016)

[9] Bravo-Lillo, C., "Pensamiento Computacional: una idea a la que le llegó el momento". *Proceeding of Bits de Ciencia*. Revista del Departamento de Ciencias de la Computación de la Universidad de Chile. pp. 48-51. (2015)

[10] Zapata-Ros, M., "Pensamiento computacional: Una nueva alfabetización digital". *Proceeding of RED-Revista de Educación a Distancia*, Nº 46. (2015)

[11] Wing J.: "Computational Thinking: What and Why?" Available at http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf (2010)

[12] Wing J., "Computational Thinking and Thinking about Computing". *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*. pp. 3717-3725. (2008)

[13] Wing J., *Computational thinking*. Commun. ACM 49, 3 (March 2006), Available at https://doi.org/10.1145/1118178.1118215 pp. 33-35. (2006)

[14] Zúñiga M., Rosas M., Fernández J., Guerrero R., "El Desarrollo del Pensamiento Computacional para la Resolución de Problemas en la Enseñanza Inicial de la Programación". *Proceeding of XVI Workshop de Investigadores en Ciencias de la Computación* (WICC 2014), Universidad Nacional de Tierra del Fuego, Ushuaia, Tierra del Fuego, Argentina. ISBN 978-950-34-1084-4. pp. 340-343 (2014)

[15] Rosas M., Zúñiga M., Fernandez J., Guerrero R., "El Pensamiento Computacional: experiencia de su aplicación en el aprendizaje de la resolución de problemas". *Proceeding of XXIII Congreso Argentino de Ciencias de la Computación* (CACIC 2017). Universidad Nacional de La Plata, La Plata, Buenos Aires. pp. 1152-1161 (2017)

[16] Moodle - Open-source learning platform. Available at: https://moodle.org.

[17] Linn, M., Aho, A., Blake, B., et al.: *Report of a Workshop on the pedagogical aspects of COMPUTATIONAL THINKING*. The National Academies Press. ISBN 978-0-309-21474-2. (2011)