Article **Journal of Critical Pedagogy**
June 2023, Vol.7 No.17 6-11

The skills of thinking and computational programming

Las habilidades de pensamiento y la programación computacional

AGUILAR-ESPINOSA, María Guadalupe†, CISNEROS-LOPEZ, Hilda Lucia, RUBIO-RIVERA, Rocío and VACA-GONZALEZ, Francisco Javier

Universidad de Guanajuato/Escuela de Nivel Medio Superior de Salvatierra

ID 1st Author: *María Guadalupe, Aguilar Espinosa* / **ORC ID**: 0009-0007-7561-9421

ID 1st Co-author: *Hilda Lucia, Cisneros Lopez* / **ORC ID**: 0000-0003-3245-0279

ID 2nd Co-author: *Rocío, Rubio Rivera* / **ORC ID**: 0009-0005-8194-8681

ID 3rd Co-author: *Francisco Javier, Vaca Gonzalez* / **ORC ID**: 0009-0006-4014-7113

DOI: 10.35429/JCP.2023.17.7.6.11 Received March 15, 2023; Accepted June 30, 2023

Abstract

This article presents the results obtained in the research carried out at the Salvatierra High School, in which the impact of the use of computer programming on the development of higher order skills in students was measured, for such purpose. For this purpose, a quantitative route was followed, applying Likert scale instruments and questionnaires to an intentional sample of students who began programming from the basic education level, either formally or through social learning. The results refer to an important recognition on the part of the students about the critical, analytical, reflective, creative skills, among others, that allow them to generate deep thought processes in any context of their lives, and therefore allow them to have an academic development different from the rest of their classmates, giving them competitive advantages that are reflected in their academic life. The foregoing highlights that teachers must include in our study plans and programs transversal projects that include computer programming to solve problems of daily life and also awaken or promote thinking skills.

Skills, Technology, Strategies

Resumen

El presente artículo presenta los resultados obtenidos en la investigación realizada en la Escuela de Nivel Medio Superior de Salvatierra, en la que se midió el impacto que tiene el uso de la programación computacional en el desarrollo de habilidades de orden superior en los estudiantes, para tal efecto se siguió una ruta cuantitativa, aplicando como instrumento un cuestionario con escala de Likert a una muestra intencional sobre los estudiantes que desde el nivel de estudios básicos comenzaron con la programación ya sea de manera formal o a través de un aprendizaje social. Los resultados refieren un reconocimiento importante por parte de los estudiantes sobre las habilidades críticas, analíticas, reflexivas, creativas, entre otras, que les permiten generar profundos procesos de pensamiento en cualquier contexto de su vida, y por lo tanto les permiten académicamente contar con un desenvolvimiento distinto al resto de sus compañeros, dándoles ventajas competitivas que se reflejan en su vida académica. Lo anterior destaca que los docentes debemos incluir en nuestros planes y programas de estudio proyectos transversales que incluyan la programación computacional para resolver problemas de la vida cotidiana y además despierten o fomenten las habilidades de pensamiento.

Habilidades, Tecnología, Estrategias

Citation: AGUILAR-ESPINOSA, María Guadalupe, CISNEROS-LOPEZ, Hilda Lucia, RUBIO-RIVERA, Rocío and VACA-GONZALEZ, Francisco Javier. The skills of thinking and computational programming. Journal of Critical Pedagogy. 2023. 7-17: 6-11

† Researcher contributing as first author.

Introduction

According to Anderson (1983) there is a difference between declarative and procedural knowledge, on the one hand, declarative knowledge includes factual and verbal information, memories based on signs, senses and perception. Additionally, it encompasses facts and concepts related to everyday life events up to organised and interrelated conceptual knowledge. Procedural knowledge, in contrast, comprises the ability to monitor one's own cognitive processes (metacognition), as well as the automated ability to transform and organise new methods to solve problems.

For Montaño (2021), thinking processes are sets of actions related to the elaboration of knowledge based on internal and external stimuli, in order to process information. According to this author, learning is related to the processes of observation, comparison, classification, analysis or description which are activated to build new knowledge and through which the ability to think, process information or have a critical attitude is built. Basic thinking processes relate to the way in which the individual perceives the reality around him/her, while integrative thinking processes emerge once the basic thinking processes have been completed.

It is useful to distinguish between the cognitive skills that are used to execute the process of encoding, storing, retrieving and transforming information, and the higher-order (metacognitive or executive) processes that are needed to implement the lower-order processes and to monitor the outcome of the transformations and responses generated by these processes. The terms used to distinguish between lower- and higher-order thinking skills vary widely, but there is some consensus that these metacognitive skills are among the most transferable mental competencies (Warner and Sternberg, 1984).

The concept of higher order cognitive skills relates directly to mental processes needed in the analysis of complex activities. At the end of the 20th century, Bloom established what these processes were, which is why Bloom's taxonomy became a reference in the educational world.

In a traditional classroom, there are usually three moments (explanation, practice and correction), which only encourage the development of three cognitive thinking skills: remembering, understanding and applying. One of the objectives pursued with the introduction of ICT in teaching and learning processes is to develop the higher level of cognitive thinking (Fernández, 2018).

The teaching and learning of programming seeks the formation and development of skills in students to favour the resolution of problems in school, professional or everyday life (Diaz, Fierro and Muñoz, 2018).

For Cuny, Snyder and Wing (2010), computational thinking is "the thinking process involved in formulating problems and their solutions so that the solutions are represented in a way that can be performed by an information processor".

The value of this concept lies in the application of this concept in the representation of the solution of a problem through sequences of instructions and algorithms. In this way, abstract thinking would be used to identify relevant aspects and a sequence of processes to develop a model.

From this perspective, this concept is transferable and of great relevance in the field of education, where technology not only encourages students' learning, but also fosters their interest in areas of knowledge related to science, technology, engineering, mathematics and computer science. The inclusion of computational thinking in educational programmes encourages students to develop competences that will enable them to move from being consumers to producers of technology.
The implementation in teaching and learning processes of activities that promote the development of computational thinking generates the following advantages for students' learning (UNIR, 2021):

- Stimulates creativity.
- Encourages reasoning and critical thinking skills.
- Develops and reinforces numerical and linguistic skills.
- Encourages leadership and teamwork skills.

However, the challenge is to have teachers capable of incorporating activities related to computational thinking into their teaching practice. Globally, the implementation of computational thinking activities in educational institutions is a trend, as it is considered to favour innovation and is associated with a long-term investment in the economic and social structure of a country.

Methodology to be developed

The research route defined in this work was quantitative, defining a purposive sample with students enrolled in the speciality of Physics-Mathematics at the High School of Salvatierra, Guanajuato, Mexico, this sample consists of 58 students who take the Learning Unit of Programming Languages and on the other hand a sample of 20 students of first and third semester who have learned to program under social learning and not by a formal learning.

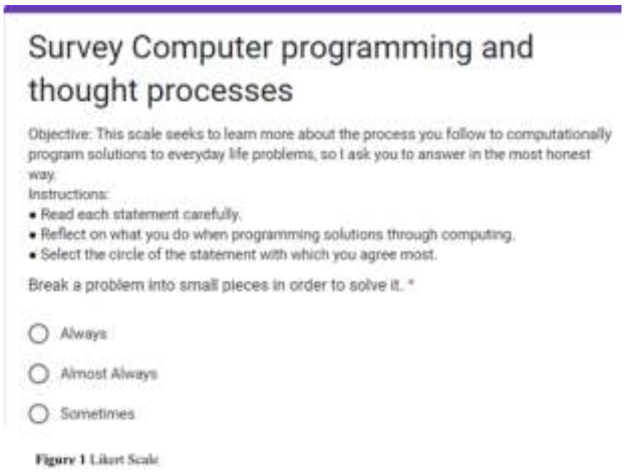
In this context, the hypothesis is defined: "Students who solve problems through computer programming develop reasoning, critical thinking, numerical, linguistic and other skills, called higher order skills. For this reason, the following objectives are derived:

To measure the impact that the use of computer programming has on students' higher order skills.

Particular objectives

To identify fifth, third and first semester students who use computer programming to solve problems.

To measure through a Likert scale with a set of items validated by the body of researchers, the impact that the use of programming has on students. For this purpose, we worked with Google Forms (Figure 1).



Survey Computer programming and thought processes

Objective: This scale seeks to learn more about the process you follow to computationally program solutions to everyday life problems, so I ask you to answer in the most honest way.

Instructions:

- Read each statement carefully.
- Reflect on what you do when programming solutions through computing.
- Select the circle of the statement with which you agree most.

Break a problem into small pieces in order to solve it. *

○ Always

○ Almost Always

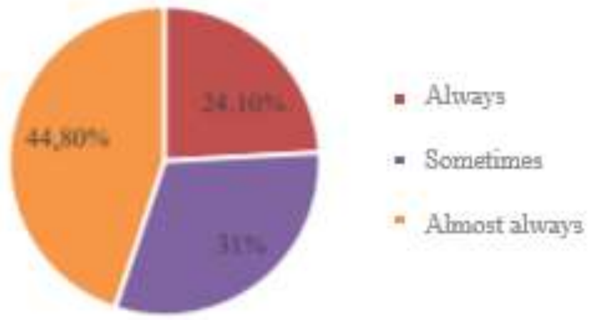
○ Sometimes

Figure 1 Likert Scale

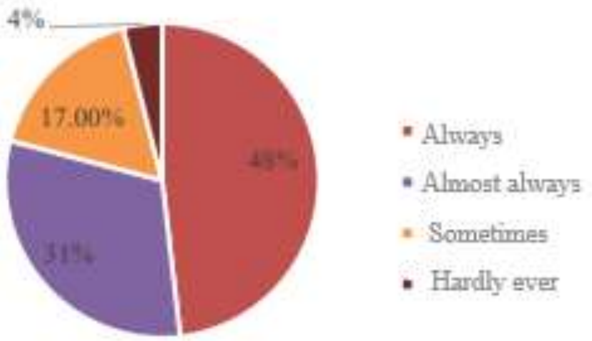
Figure 1 Likert Scale

Results

The following figures show the most outstanding results of the Likert scale and the questionnaire applied to the entire population sample defined in the methodology, regardless of the way in which the knowledge about computer programming has been acquired, i.e. formal learning or social learning.

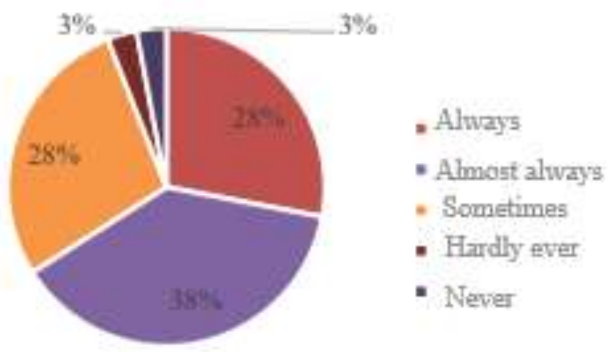


Graph 1 Problem analysis

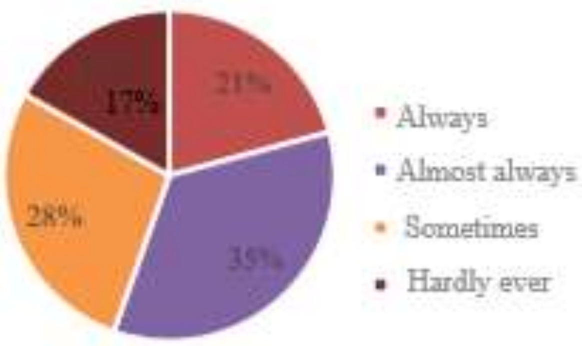


Graph 2 Reading comprehension of the problem

In Graph 1, more than 72% of students report that in order to solve a problem they break it down into small problems, which they say gives them a broad perspective of the elements that make up the problem. As we can see in Graph 2, more than 79% of students report that reading a problem several times leads to a better understanding of the elements of the problem and how they can solve it. The survey talks about reading comprehension processes, which helps in subjects such as physics, chemistry, English, Spanish, among others.



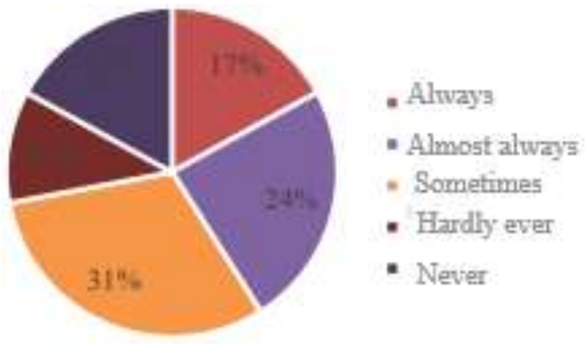
Graph 3 Use of methodologies



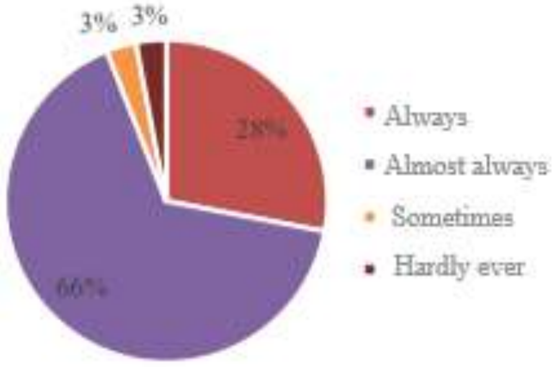
Graph 4 Use of algorithms

According to Graph 3, 65% of students report using a computational methodology to be able to analyse and model the problem before programming a solution. As we can see in figure 5, 55% of students make use of algorithms as a problem analysis methodology, followed by flowcharts and finally pseudocode.

The questionnaire addresses the use of methodologies to achieve the process of analysis in problem solving, being pressing in subjects such as mathematics and experimental sciences.

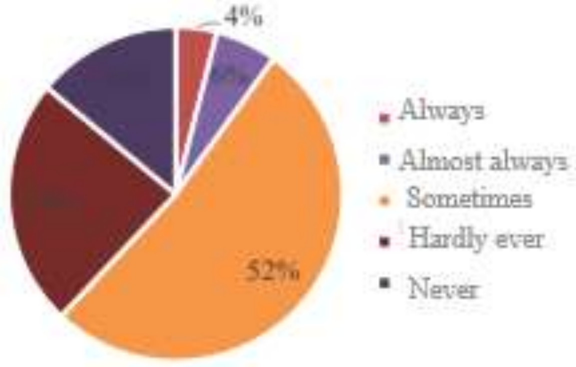


Graph 5 Validation of algorithms



Graph 6 Consideration of possible scenarios

According to Graph 5, 41% of the students say that before starting to programme a solution they make use of the prior validation of the algorithm, i.e. they verify that the proposal proposed will be successful by validating each of the processes defined in the algorithm. Figure 7 shows that more than 90% of the students mention that when analysing the solution they consider all the possible scenarios that could occur in the process. The questionnaire addresses the consideration of different routes or paths that a problem may take, which is very useful in subjects such as experimental sciences.



Graph 7 Logic when running a programme

At least 50% of the students, according to figure 8, mention having logic problems when executing a solution programmed by themselves.

Although in the survey they refer that at the beginning of the programming the failure of the logic was above 50% and the other percentage to syntax problems with the passage of time have been improving, having much less logic errors, as they refer to be improving with practice.

Something also important in the survey was to recognise that their academic performance is acceptable, mainly referring to having fewer learning problems in the learning units of mathematics, physics, chemistry and communication languages mainly, in addition to the fact that they observe various opportunities to make inroads such as olympiads, competitions, science fairs, among others.

Conclusions

The use of computer programming at any age in a person's life is very useful, and even more so if it is encouraged from childhood, in what is now called early programming. The use of technologies is another important factor to consider, for the generations of students currently in the classroom called digital natives, who seek to solve problems using technology.

Therefore, derived from the concern to achieve graduation profiles in upper secondary level students, mainly in logical-mathematical skills, critical thinking, reflective thinking, initiative, creativity, among others, which fall into what is called higher thinking skills, it is necessary to look at technologies that contribute greatly to developing the graduation requirements of students.

The results of the study, from the students' perspective, emphasise the importance of the use of computer programming for the development of these skills and also the need to incorporate it into any learning unit, not as just another subject but as a transversal learning that supports the resolution of problems in any environment that may arise, and that in parallel enriches the desired profiles.

Likewise, the results show that the students' perception of programming in their academic training is very noticeable, as they observe a different way of proposing solutions to the problems they are presented with, generating complex analysis processes that lead to the achievement of different higher order skills such as those already mentioned, thereby providing more participative, creative, enterprising and reflective students, generating deeper teaching and learning processes with a significant transversality. As Steve Jobs pointed out, programming a computer teaches you how to think.

References

Anderson, J. R. (1983). The architecture of cognition. Cambridge: Harvard University Press.

Cuny, J., Snyder, L., y Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript, referencedinhttp://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

Díaz, K. I., Fierro, E. R. y Muñoz, M. A (2018). Empleo de los entornos virtuales de aprendizaje en la formación de profesionales de la educación Una experiencia cubana. Disponible en: https://www.researchgate.net/publication/328107562

Fernández, P. (2018). Las Habilidades Cognitivas de Orden Superior. Actividades prácticas para su desarrollo en el aula. Blog: Un Aula para todos. TAC, competencia digital, inclusión y mucho más. https://unaulaparatodos.wordpress.com/2018/06/03/las-habilidades-cognitivas-de-orden-superior-actividades-practicas-para-su-desarrollo-en-el-aula/

Montaño, J. (2021). Procesos del pensamiento: qué son y cuáles son (con ejemplos). Lifeder. Recuperado de https://www.lifeder.com/procesos-del-pensamiento/

UNIR. (2021) ¿Qué es el pensamiento computacional?. UNIR REVISTA. Recuperado de https://www.unir.net/educacion/revista/pensamiento-computacional/

Wagner, R. K. y Sternberg, R. J. (1984). Alternative conceptions of intelligence and their implications for education. Review of Educational Research. 54 pp. 179-224.