XVII CONGRESSO INTERNACIONAL DE TECNOLOGIA NA EDUCAÇÃO

Dilemas e desafios de um futuro presente: o que esperar da educação?

22 e 23 | setembro | 21

COMPARATIVE EVALUATION OF THE PERFORMANCE OF STUDENTS AROUND COMPUTATIONAL THINKING

Autor: Cecir Barbosa de Almeida Farias
Email: cecir.almeida@gmail.com

Resumo - O pensamento computacional tem sido considerado uma das principais competências para prosperar em um mundo tecnológico, e diversas iniciativas têm surgido visando o seu desenvolvimento desde a educação básica. Sua inserção nas escolas possibilita a aquisição de habilidades necessárias à resolução de problemas, podendo apoiar e relacionar-se com outras ciências. Nesse contexto, o presente trabalho objetivou quantificar o desempenho prévio e posterior ao curso "ProgramAção", realizado com alunos pertencentes ao ensino médio, mediante a aplicação de um teste. O referido teste, foi desenvolvido por Román-González (2015), este é composto por 28 questões, as quais tentam identificar conhecimentos de formação e capacidade de solução de problemas, baseando-se nos conceitos fundamentais da Computação. Apesar do tamanho da amostra não ser estatisticamente significativo, os resultados apontam a eficácia da oficina em relação à aprendizagem de conteúdos de computação o que pode ser usado de forma a melhorar o ensino da computação na Educação Básica.

Palavras-chave: Ensino. Metodologia ativa. Programação em blocos. Pensamento computacional. App Inventor.

1. INTRODUCTION

The incessant changes in society and the arrival of more and more complex computational systems have brought with them new challenges that demand new approaches to solve them.

In Brazil, multiple initiatives have appeared in different regions e have been published in conference proceedings from events held by the Congresso Brasileiro de Informática na Educação (CBIE) [Brazilian Congress of Computing in Education], like the Workshop de Ensino em Pensamento Computacional Algoritmos e Programação (WAlgProg) and Algorithms and Programming]. There are still those published in the proceedings of the Workshop sobre Educação em Computação (WEI), Workshop about Education in Computing] and in magazines like the Revista Brasileira de Informática na Educação (RBIE) and the Revista Novas Tecnologias na Educação (RENOTE).

It is possible to observe that countless actions trying to include fundamental concepts of computing - knowledge of programming logic, algorithmic structure and the abstraction necessary for solving computational problems - have been proposed to be included in the curriculum of elementary and high school institutions (Fernandes e Menezes 2011).

Currently, researches in computational thinking present multiple proposals for curriculums and short courses for the development or insertion of the PC in the level of basic education and it is believed that these skills could contribute to the cognitive development of students.

With this in mind, tests were created to measure and quantify the development of students taking part in the “PromgramAção” course. One test was taken prior to the course and aimed to measure students' prior knowledge, the other after the course, trying to find an increase in their development.

2. Problens, questions, objectives

Computational Thinking includes methods for problem solving based on the fundamentals and techniques of computer science, and is seen as one of the ways of developing logical reasoning.

> The development of problem solving skills and critical thinking is necessary thanks to the social transformations e growing competitiveness in the modern world. The inclusion of Computational Thinking in the school environment contributed to the social and scholarly life of the student because, through the improvement of this skill, he can more easily face the daily challenges of his life (Valente, 2016, page 35).

Waiselfisz (2007) considers it important that teachers evolve and appropriate digital solutions, with autonomy in thought and actions, being necessary still to recognize the teaching binomial and learning as a relation between subjects.

3. METHODOLOGY

A pedagogical project was developed, in the form of a introductory programming course where the methodological procedures utilised were: (a) Introduction of the course to students of the public school Professor José Gonçalves de Queiroz in Sumé, in the brazillian state of Paraíba; (b) Enrollment of students interested in the course; (c) Preparation of the IT Laboratory from the Federal University of Campina Grande - Center of Sustainable Development for the Semi-arid; (d) Application of a quiz for the purpose of verifying student's profiles, the extent of their computation knowledge and their respective difficulties; (e) Application of classes made up of enrolled students in high school; (f) Stimulus of rational thinking and creativity through small programming projects; (g) Application of a quiz at the end of the course, with the purpose of comparing the student's computational knowledge and identifying if there was significant growth.

3.1. Exploratory Fase

The course named “ProgramAção” was taught by scholarship students and extension workers from Probex, the scholarship extension program from the Federal University of Campina

Grande. The class was composed of 22 students. Classes occured in weekly meetings, with the duration of 2 hours per class. The planning of these classes was based in the following stages:

-The First stage: consisted of the application of the computational thinking quiz developed by Román González et al. (2015), which had the goal of identifying the previous knowledge of students.

-The Second stage: was made up of the most long lasting cycle of classes, as well as the most relevant ones, as it was in them that the fundamental concepts of programming.

-The Third stage: consisted of teaching the concepts of block-based programming, approaching its logic, followed by the introduction of App Inventor.

-The Fourth stage: consisted of utilizing the platform in a practical way, presenting its tools, resources and existing components.

-The Fifth stage: composed of individual and group development of mobile apps, based on the knowledge learned in the last stage.

-The sixth stage: consisted again of the application of a computational thinking quiz developed by Román González et al. (2015), which tried to identify increases in the students performance.

In harmony with all the stages, an analysis was made, through a quantitative approach, based on the initial and final quizzes.

3. 2. Evaluation instrument

The quiz utilized in the research was chosen due to its quantitative and appropriate approach and because it has already gone through an extensive process of validation. The questions that make up the test include concepts from the four pillars of computational thinking: abstraction, decomposition, pattern recognition and algorithms. It is composed of 28 multiple choice questions with each question having 4 possible answers.

4. Data Analysis

In general, the inicial quiz had an average of eight (~8,35; 36,98%) correct answers, while the final quiz had an average score of twelve and a half (~12,46; 56,65%) in questions related to abstraction skills and computational thinking. Figure 1 shows the average performance of students in relation to their abstraction skills. The decrease in performance from question to question relates to the increase in complexity as the test goes on.

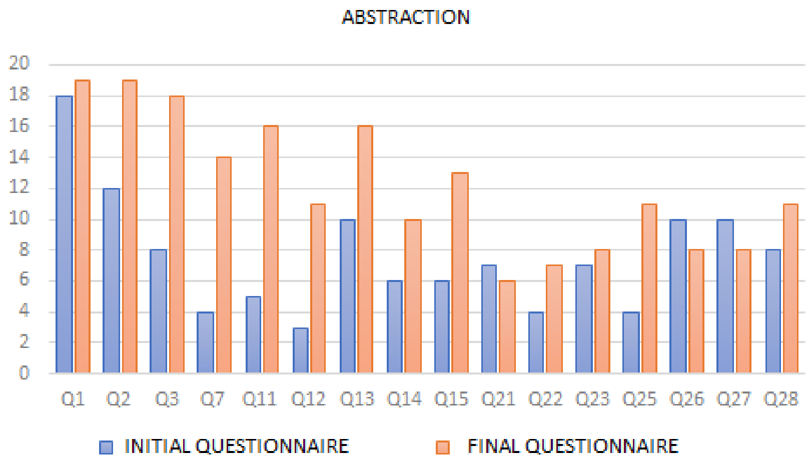

Figure 1. Graphical representation of the correct answers for each question in the initial (blue) and final test (orange), testing the computational skill of “abstraction”

The evaluation of the average capacity of students to decompose problems showed an average improvement of 20,26 percentage points in the area of fragmenting big problems into smaller ones. The increase mounted up to a final percentage of 58,86% average of correct answers, as can be seen in figure 2.

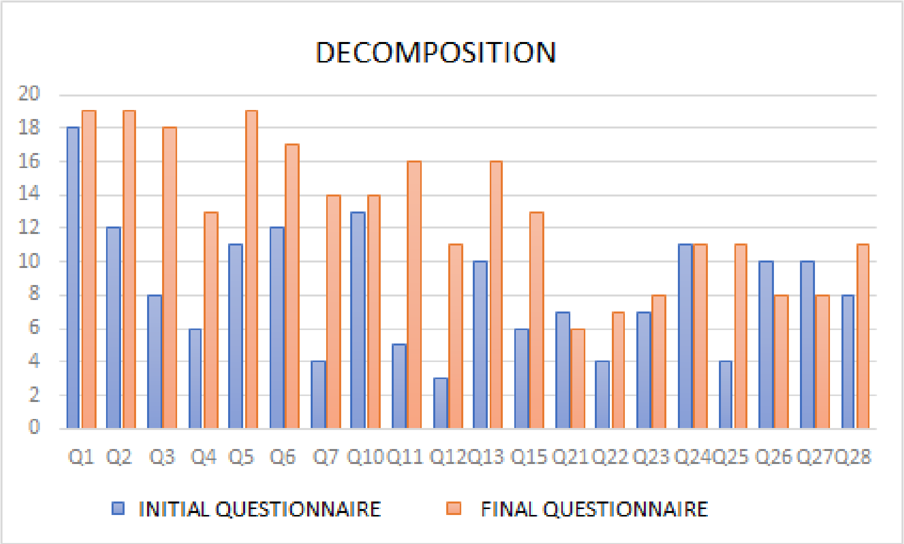

Figure 2. Graphical representation of the correct answers for each question in the initial (blue) and final test (orange), testing the computational skill of “decomposition”

The questions referring to pattern recognition showed the least average final porcentual increase after the block-programming course. However, an increase of 17,7 percentage points was found in the students averages, hitting a global average of correct answers of >50% (53,8%). A representation of the relation between average scores in the initial and final tests can be found on picture 3.

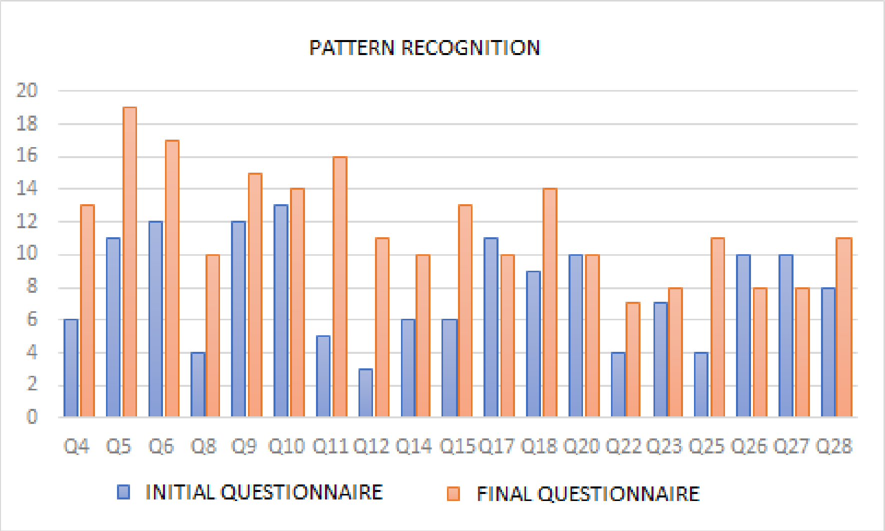

Figure 3. Graphical representation of the correct answers for each question in the initial (blue) and final test (orange), testing the computational skill of “pattern recognition”

The average comprehension capacity of students in relation to algorithms was measured by the amount of correct answers, which numbered a total of 37,98% of correct answers prior to the test. This average increased by 18,68 percentage points, reaching 56,65% correct answers.

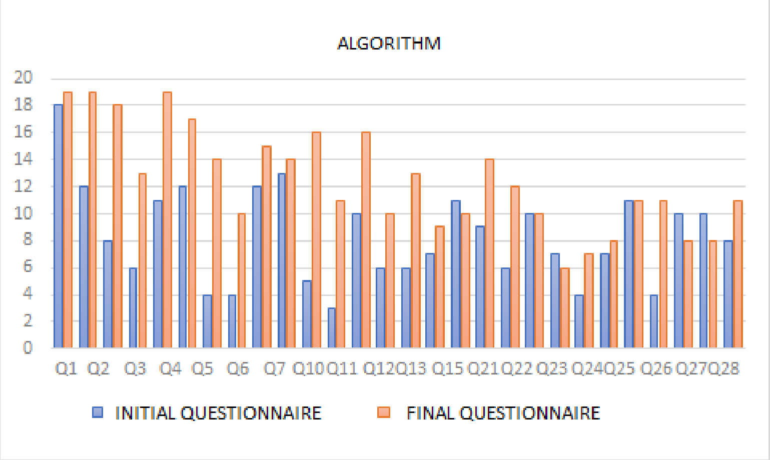

Figure 4. Graphical representation of the correct answers for each question in the initial (blue) and final test (orange), testing the computational skill of “algorithm”

The global average of correct answers in the initial test numbered ~8,085, while the average for the final test was approximately 12,36. The average capacity of students to utilize the skills of “Abstraction”, “Decomposition”, “Pattern recognition” and “Algorithms” to solve the presented problems grew noticeably bigger after students learned the concepts of block-programming in the course. The final average skill to think with computational thinking grew approximately 19,4%, adding up to a global average of close to 56,15/100.

5. CONCLUSION

The sample of students selected to partake in the block-programming course managed to, throughout the course, develop the skills of computational thinking. It is possible to observe an increase in each of the four skills of computational thinking. The quiz created by Román-González (2015) showed itself useful as a measure of the aforementioned skills. Block-programming showed its capabilities as an educational device in the service of teaching introductory programming.

References

GAL-EZER, J.; HAREL, D. **Curriculum for a high school computer science curriculum**. Computer Science Education 9, 1999. 2v.

Fernandes, C. S.; Menezes, P. B. **Metodologia do ensino de ciência da computação: uma proposta para criança**. In: Anais do Workshop de Informática na Escola. 21 a 25 de novembro. Fortaleza, CE, p.103-107, 2011.

Ministério da Educação de Ontário – MEO. Currículo para o ensino de Ciência da Computação nas escolas. 2008. Disponível em: <www.edu.gov.on.ca>. Acessado em: 02/10/20.

Román-González, M. **Computational thinking test: Design guidelines and content validation**. EDULEARN15, p. 2436–2444, 2015.

Valente, J. A. **Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno**. PUC, São Paulo, p. 35-39, 2016.

Waiselfisz, J. J. **Lápis, borracha e teclado: tecnologia da informação na educação**. Brasília: Rede de Informação Tecnológica Latino-americana (ritla), p. 119-123, 2007.