Informatics in Education, 2023, Vol. 22, No. 2, 233–255

DOI: 10.15388/infedu.2023.12

A Second Experimental Study the Application of a Teaching Plan for the Algorithms Subject in an Undergraduate Course in Computing using Active Methodologies

Fabrício Wickey da Silva GARCIA[1,2],
Sandro Ronaldo Bezerra OLIVEIRA[1], Elielton da Costa CARVALHO[1]
[1]*Graduate Program in Computer Science, Federal University of Pará (UFPA), Belém, Brazil*
[2]*Federal Rural University of Amazon (UFRA), Capitão Poço, Brazil*
e-mail: fabricio.garcia@ufra.edu.br, srbo@ufpa.br, elielton.carvalho@icen.ufpa.br

Received: December 2021

Abstract. The contents taught in the programming subjects have a great relevance in the formation of computing students. However, these subjects are characterized by high failure rates, as they require logical reasoning and mathematical knowledge. Thus, establishing knowledge through the subject of algorithms can help students to overcome these difficulties and absorb the contents and skills required. Thus, this work aims to present and discuss the results of a second experiment on the application of a teaching plan composed of several active methodologies (Virtual Learning Environments, Coding Dojo, Gamification, Problem-Based Learning, Flipped Classroom and Serious Games) in an algorithms subject. Based on this experiment, it was evaluated whether there were learning gains compared to the learning acquired with the traditional method. Finally, an analysis was performed using the two-tailed Student-t approach, used for independent samples, which presented statistically significant results.

Keywords: active methodologies, coding dojo, gamification, problem-based learning, virtual learning environment, flipped classroom, serious games, programming class, learning and teaching, experimental study.

1. Introduction

Computer programming, in addition to being a mandatory subject in computing courses, is the basis for training professionals in the area (Ding, 2019). In view of the importance of this subject for the training of students, organizations, institutes and societies related to the area have developed and maintain guidelines that guide universities in the elaboration of their syllabus, as is the case of the Association for Computing Machinery (ACM), the Institute of Electrical and Electronic Engineers (IEEE) and the Brazilian Computer Society (SBC). In Brazil, for example, the National Curriculum Guidelines (DCN) guide universities on the importance of building a solid base of knowledge regarding programming and logical reasoning skills (Brasil, 2016).

Programming a computer requires students to have skills ranging from cognitive to metacognitive skills, as it requires a combination of logical reasoning and creativity, in addition to the task of learning one or more programming language syntaxes (Eteng *et al.*, 2022). Teague (2011) states that computer programming is complex in nature. Taking all these points into account, Oroma *et al.* (2012) state that, for an apprentice to become an experienced programmer, it takes an average of 10 years, depending on their own interest. It is not by chance that programming subjects have a high rate of failure, as reported (Eteng *et al.*, 2022). As a consequence of failures, Hoed (2016) states that dropout from courses has a higher rate among those who failed than those who managed to pass.

For Sajjanhar and Faulkner (2019), the high numbers of failures reported by higher education institutions is an issue that must be taken seriously, since this percentage reaches 40% in some cases. In addition, the authors emphasize that there is a systemic flaw in the curricula of universities and that, therefore, solving these problems will lead to a reduction in failure rates and, consequently, a reduction in dropout, leading more students to complete computer courses.

In view of this, researchers undertake time and effort to minimize the high failure rates in introductory programming subjects (algorithms) (Vieira, Júnior and Vieira, 2015). These researchers propose to develop techniques, methods and strategies that integrate these subjects with the many pedagogical approaches that aim to overcome these challenges (Garcia, 2021). Therefore, researching, applying and evaluating new algorithmic teaching approaches are necessary to improve student performance. Studies such as (Giraffa and Müller, 2017; Vasconcelos *et al.*, 2019; Fonseca and Brito, 2021) recommend the adoption of active methodologies that encourage student engagement and make them the center of the teaching-learning process.

Thus, this work aims to present the results of a second experiment related to the application of a teaching plan in an algorithms subject in an undergraduate computing course. The teaching plan is composed of a set of active methodologies, namely: Virtual Learning Environments (VLE), Coding Dojo, Gamification, Problem-Based Learning (PBL), Flipped Classroom and Serious Games. In addition to applying the teaching plan, this work compares the traditional method applied in universities with the method proposed in the plan. It is important to mention that this work is a second experiment, the first one being carried out with a total of 68 undergraduate students (Garcia *et al.*, 2022). Meanwhile, this second experiment was carried out with 72 undergraduate students.

In addition to this introductory section, this work is composed as follows: Section 2 presents a theoretical foundation, that is, concepts and definitions on the topic addressed in the research, Section 3 brings the works related to this work, Section 4 deals with the methodological steps used in this work and presents the strategy for applying the teaching plan, Section 5 presents the data analysis of the experiment performed, Section 6 presents the discussion of the results, Section 7 addresses threats to validity, Section 8 discusses the main threats to the validity of this work and Section 9 presents the conclusions of the work.

2. Background

Computer programming has become so important that many practitioners have come to consider that in the future the digital illiterates will be those who do not develop the skills to program. The computer programming market will have a 22% increase in the number of vacancies between 2020 and 2030 (US Bureau of Labor Statistics, 2022), and, according to CIO (White, 2022), computer programming is among the 10 most wanted jobs in the technology area. However, programming is not a trivial task and is far from accessible to everyone. Learning to program requires time and effort, in many cases, previous skills. These skills, in turn, are acquired through the teaching of algorithms, that is, from the base of programming (Gallego-Durán *et al.*, 2021).

Programming knowledge involves learning algorithms, since this subject is the basis for the development of skills and abilities that will be useful throughout a programmer's life. Knowing this, the subject of algorithms is usually offered in universities in the first semesters of computing courses, with a workload that can vary between 60 and 80 class hours (Falkembach *et al.*, 2003; Saito *et al.*, 2015; Bulcão, 2017). In Brazil, organizations such as the Brazilian Computer Society (SBC) and guidelines such as the National Curriculum Guidelines (DCN) on Computing work strongly in the preparation of documents that serve as a basis for the curricular construction of computing courses.

The documents prepared by the SBC and the DCN guide universities around the skills that students must acquire throughout the subjects and, consequently, throughout the course (Calsavara *et al.* 2018). Zorzo *et al.* (2017) point out that the subject of algorithms can help students to identify problems that can be solved algorithmically, in problem solving through programming environments, in the recognition that computational thinking has significant importance in everyday life and which can be applied in different contexts, in the formulation of problems whose resolution requires the application of logical reasoning, mathematical and computational knowledge, and, finally, in the understanding of problems that can be solved by computers.

As already mentioned, the teaching of algorithms is focused on the development and establishment of a solid base of knowledge related to computer programming. This subject aims to stimulate logical and mathematical reasoning, so that students can solve computational problems (Barros and Santos, 2018). However, even with the difficulties reported by several works (Sajjanhar and Faulkner, 2019; Gallego-Durán, 2021; Prasad *et al.*, 2022), algorithm teaching is still taught in a mechanized way, that is, teaching focuses only on memorization and retention of concepts, which makes students simple spectators (Ramos *et al.* 2015; Amaral *et al.* 2017). It is also worth mentioning that, in addition to the mechanized form, another factor that corroborates the learning difficulties is linked to the misinterpretation of computational problems, little mathematical knowledge and lack of motivation (Moraes, Neto and Osorio, 2020).

2.1. *Reference Curriculum – SBC and ACM/IEEE*

When preparing a pedagogical project, course coordinators usually follow the guidelines of the References for Training (RF). In Brazil, the SBC is one of the institutions that constantly work on the elaboration of the RF and that guides universities in the elaboration of pedagogical projects (Calsavara *et al.* 2018). According to Zorzo *et al.* (2017), the RF are elaborated based on the notion of competence and skills that students need to acquire, so that it is possible to apply this knowledge in real day-to-day situations.

According to the 2016 DCNs, it is necessary to develop a solid and adequate foundation of content that develops the skills required by computer programming, such as logical reasoning and mathematical knowledge (Brasil, 2016). With this, Zorzo *et al.* (2017) highlight some skills that can be developed with the teaching of algorithms and that are present in RF, they are:

- Identify problems can be solved algorithmically.
- Solve problems through programming environments.
- Recognize that computational thinking has significant importance in everyday life, as well as its different possibilities of application in different domains.
- Use logical reasoning, mathematical and computational knowledge to formulate and solve problems.
- Understand problems and design solutions that can be solved computationally.

In addition, the good practices contained in the ACM and IEEE (ACM/IEEE 2020) Computing Curriculum 2020 (CC2020) served as a basis, from which recommendations and the main concepts referring to the basic units of knowledge and the programming pillars were taken covering the following topics:

- Develop an algorithm that can illustrate iterative and recursive functions, use divide and conquer techniques, in addition to using some programming language to implement, test and debug the algorithm that solves a simple problem.
- Identify data components and their behavior by decomposing a program that has abstract data. Also, implementing some kind of data that is coherent.
- Carry out the process of designing, implementing, testing and debugging a given program, which includes simple input and output methods, file manipulation, conditional and iteration structures, functions and parameter passing.
- Show which are the main costs and benefits related to the implementation of static and dynamic data structures, choosing the most appropriate structure to solve a problem.
- Contribute to the readability and maintainability of the software by applying consistent documentation and conducting a code review using a provided checklist.
- Point out coding, construction and debugging errors considered common, through libraries available by the adopted programming language.
- Refactor code and identify opportunities to apply abstract methods.

The most current version of the ACM/IEEE curriculum (CC2020) adopted in its structure the cognitive levels defined by Bloom's Taxonomy, which indicate the skills that are necessary and expected to perform a task in a given teaching unit. According to (Ferraz and Belhot, 2010), Bloom's Taxonomy is structured as follows:

- **Remember:** this cognitive level deals with the recognition and reproduction of ideas and content.
- **Understand:** this cognitive level concerns the establishment of links between new knowledge and previously acquired knowledge.
- **Apply:** this cognitive level is related to the act of executing and using some procedure in a specific situation, whether new or not.
- **Analyze:** this cognitive level seeks to divide information into relevant and non-relevant parts, as well as their relationships.
- **Evaluate:** this cognitive level seeks to make judgments based on quantitative and qualitative standards and criteria, and in certain cases these criteria may be related to efficiency and effectiveness.
- **Create:** this cognitive level concerns the insertion of elements in order to build a new vision, through preexisting knowledge and skills.

3. Related Works

Freire *et al.* (2019) present a plan of sixteen tutoring meetings, based on active methodologies for introductory programming subjects. The PBL was applied to 16 tutoring meetings, which were focused on the teaching units of the algorithms subject, there was positive feedback from the students regarding the use of tutoring as a way to help the teaching and learning of programming introduction, so that 88% of the participants reported that the activities contributed significantly to their learning.

Sousa e Leite (2020) use gamification to promote introductory programming learning. The proposal is based on the use of online judges applied to a programming course lasting 60 class hours. The authors carried out a comparative study of two classes, the first using an approach based on active methodologies, and the second class using an approach based on traditional teaching. The results showed that the first class achieved pass rates of 81% while the second class achieved a pass rate of 42%.

Alves *et al.* (2019) carried out a study that consisted of the use of Coding Dojo in the teaching of introductory programming. It was noticed that the use of the active methodology contributed positively to increasing student engagement, so that students were more motivated with the application of the Coding Dojo. The authors also point out that there is a little resistance from some students with tasks related to peer programming.

Silva (2017) carried out the application of the active methodologies Flipped Classroom, Collaborative Learning and Gamification in the teaching of programming guided by the use of a virtual collaborative learning environment that allowed the teaching and learning of programming. For the validation of the proposal, experiments were carried out with the participation of two groups of students, the first, the experimental, which makes use of the proposed approach, while the second group, the control, makes use of the approach traditional teaching. The results obtained with the experiment showed that the experimental group obtained higher grades when compared to the control group that used the traditional teaching approach.

As reported in Section 1, this work is a second experiment referring to the application of a teaching plan composed of active methodologies, in an undergraduate course in computing. This means that, to reach this second experiment, other steps were completed and as a result, other works were published in scientific circles. In (Garcia *et al.*, 2020) a quasi-Systematic Literature Review (qSRL) was carried out and several approaches applied in the teaching of algorithms and programming were identified, as well as their benefits, forms of use and difficulties of use in the classroom of class. In (Garcia *et al.*, 2021a) a strategy was developed for the application of VLE, Coding Dojo, Gamification, PBL, Flipped Classroom and Serious Games in algorithms or equivalent subjects. In (Garcia *et al.*, 2021b) a teaching plan is developed consisting of six approaches that were identified from the qSRL (Garcia *et al.*, 2020) and which makes use of the strategy adopted in (Garcia *et al.*, 2021a). In (Garcia *et al.*, 2022) the first experiment was conducted with 68 students, divided into two equal groups of 34 students, where one of the groups was a control group. For one of the groups, the subject of algorithms was taught following the traditional teaching method and the second group was the experimental one, that is, the subject was conducted following the teaching plan with active methodologies.

The works presented are part of a process that culminated in the elaboration, application and evaluation of a teaching plan that aims to implement a new method of teaching algorithms based on active methodologies. Thus, this work presents a second experiment carried out with 72 students, differentiating itself from the others by including a greater number of participants in its scope, in order to ratify the positive results presented in the previous works.

4. Research Methodology

This is an applied research, as it seeks, through experimental studies, to extract knowledge that enables the improvement or solution of a specific problem (Gil, 2010). In this sense, the present study seeks, through experiments, to evaluate the effects of applying a teaching plan for algorithms supported by active methodologies. A quantitative approach was adopted, because, according to Jacobsen (2009), this characterization is concerned with quantifying the data collected through statistical analysis. Therefore, the data obtained from the experiment were analyzed using statistical tests, the two-tailed Student-t was used to evaluate the difference between two distinct populations, the Equal Variance Test (Brow-Forsithe) to evaluate the variance and data normality and the Shapiro-Wilk test for normality assumptions.

Regarding the objectives or purpose of the research, it was classified as explanatory, because, according to Vergara (2006), explanatory research is based on experiments that are based on hypotheses related to a particular problem or real process. The technical procedures of the research were classified as experimental, because, according to Gil (1991) and Lakatos and Marconi (2011), this type of research is carried out in a controlled environment and involves the direct action of researchers in the process of analyzing the effects from the variables applied to the study objects, the study control rules

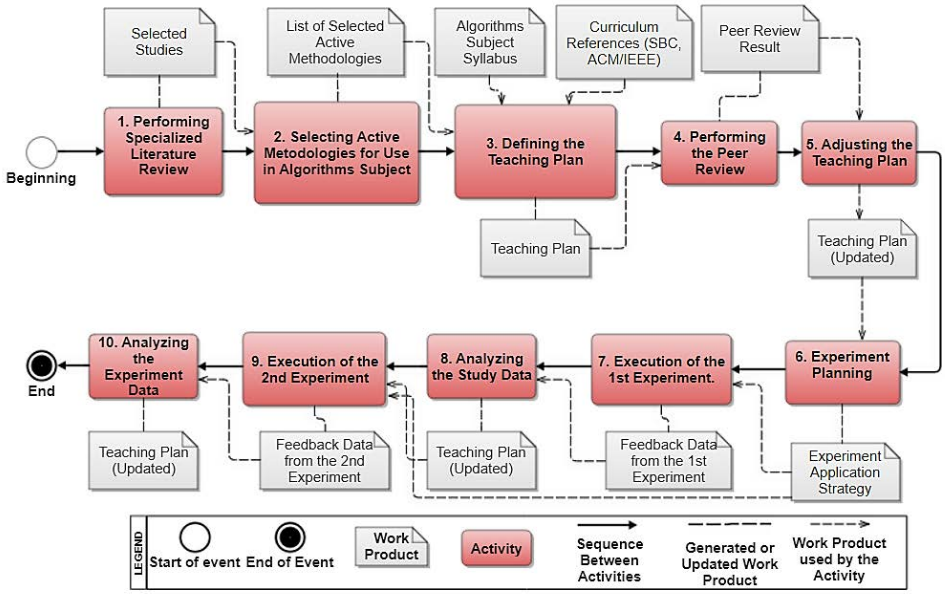

Fig. 1. Study Execution Steps (Garcia *et al*, 2022).

are defined, as well as ways of analyzing and observing the effects that the variables will produce on the analyzed objects.

In this sense, some steps were defined for the successful completion of this study, as shown in Fig. 1.

The study started with a literature review, with the objective of carrying out a survey of data regarding the use of active methodologies in programming teaching, focusing on introductory courses. The review was classified as quasi-Systematic, since all the formalism of a Systematic Review of Literature (SRL) was used, but without making comparisons between the analyzed works, therefore, this type of review can be classified as a qSLR (Ramos *et al.*, 2015). A total of 1014 scientific studies were analyzed and what were the active methodologies currently used in programming teaching, as well as their benefits, difficulties of use and forms of evaluation. The details of the review are described in (Garcia *et al.*, 2020).

After identifying the active methodologies that can be worked on in introductory programming subjects, some methodologies that could be used together within a teaching plan were chosen. The methodology use strategy was evaluated by professors specialized in programming teaching and in the use of active methodologies, the evaluation took place through the peer review technique, the review details can be consulted in (Garcia *et al.*, 2021a). Table 1 presents the active methodologies that were analyzed by the experts and their use objectives in the subject of algorithms.

After the selection of active methodologies, a teaching plan for the subject of algorithms was elaborated, which has its teaching units related to the methodologies selected in the previous step. The choice of units was based on the guidelines of the SBC, ACM and IEEE curriculum frameworks. The relationship between the contents of the teach-

Table 1

Teaching Units related to Active Methodologies (Garcia *et al.*, 2022)

Virtual Learning Environment – VLE	To establish a form of remote communication that allows the availability of materials, creation of forums, sending tasks and sharing knowledge.
Coding Dojo	To promote the exchange of knowledge, sharing of ideas, encourage teamwork and student autonomy in decision-making aimed at solving problems, increase engagement.
Gamification	To stimulate student protagonism in the search for knowledge and the ability to assimilate content, making student participation more active in the classroom.
Problem Based Learning – PBL	To promote content learning through an approach focused on practice through problem solving.
Flipped classroom	To encourage the development of skills related to protagonism and student autonomy to search and filter content that is relevant to the learning of a particular teaching unit.
Serious Games	To provide an attractive way to learn specific content of the algorithms subject through game logic combined with block programming, aiming to simplify the learning of key content of the subject.

Table 2

Teaching Units related to Active Methodologies, adapted from (Garcia *et al.*, 2022)

Modules	Teaching Units
• **Initial Evaluation Module (Pre-Test)** Duration 2 class hours	
• **Teaching Module I** Duration 18 class hours • **Active Methodologies** Gamification, Problem Based Learning, Coding Dojo Virtual Learning Environment.	• What is an algorithm? • What is a programming language? • Types to represent an algorithm, • Variables, • Operators (Logical and Arithmetic), • Linearization of expressions.
• **Teaching Module II** Duration 24 class hours • **Active Methodologies** Problem Based Learning, Coding Dojo, Gamification, Serious Games, Virtual Learning Environment.	• Introduction to the Scratch Environment, • Scratch variables, operators and main commands, • Conditional Structures, • Repetition Structures, • Lists (Arrays).
▪ **Teaching Module III** Duration 22 class hours • **Active Methodologies** Problem Based Learning, Coding Dojo, Gamification, Flipped Classroom, Virtual Learning Environment.	• Introduction to C Programming Language, • Variables, operators and main commands of the C programming language, • Conditional structures in C programming language, • Repetition structures in C programming language, • Arrays in C programming language.
• **Final Evaluation Module (Post-Test)** Duration 2 class hours	

ing plan and the active methodologies is described in Table 2, which was adapted from (Garcia *et al.*, 2022).

At the beginning of the course, there is a pre-test lasting two hours, the objective is to identify the profile of the students as well as their previous skills in introductory programming. Then, the Teaching Module I begins, which lasts 18 hours, the focus of this module is on the development of the students' logic, therefore, basic concepts that are related to the subject of algorithms are addressed.

The Teaching Module II focuses on serious games and seeks to deepen the knowledge acquired in the previous module through the use of repetition, decision and array structures. At this moment the students come into contact with programming in blocks through the Scratch tool and through it can interact with simplified games, as well as develop their own solutions based on the serious games approach.

In Teaching Module III students work on the contents covered in the previous teaching module, but with a focus on the use of a structured programming language. At this point in the course, the use of the flipped classroom approach is highlighted, which allows developing student autonomy outside the classroom.

In all teaching modules, the PBL methodology is used as a way to promote dynamics related to problem solving, as well as the Coding Dojo, which allows the creation of a practical programming environment focused on problem solving. The virtual environment is also present throughout the subject as a tool to facilitate communication, exchange materials and centralize discussions and debates. The teaching plan was evaluated by experts who used the peer review technique, details of the evaluation can be found in (Garcia *et al.*, 2021b).

After the evaluation of the teaching plan by peer review, adjustments were made based on the observations made by the evaluators and the planning stage of applying the teaching plan through experiments began. A strategy for applying and evaluating the data obtained from the experiments was defined.

It was decided to use students linked to higher education institutions, who were allocated into two distinct groups, the first being called an experimental group, which makes use of teaching through the traditional approach, while the second group (experimental) uses the use of the approach proposed in this work, which is based on the use of active methodologies. It was defined that both groups would have specific professors who would work on the contents of the algorithms subject, and that each professor had an auxiliary team composed of 3 members with experience in teaching algorithms, who were responsible for carrying out corrections of the activities carried out in both groups. This strategy was adopted to minimize interference in the corrections of activities, and, consequently, reduce the bias of the results obtained. Fig. 2 illustrates the study design.

The content covered in the experimental and control groups were grouped into 3 parts, which were entitled Evaluation 1, 2 and 3 for the control group and Teaching Module I, II and III for the experimental group, and the content covered in each part was the same for both groups, as shown in Table 3. In addition, due to health restrictions resulting from the COVID-19 pandemic, the experiments had to be carried out in a virtual format.

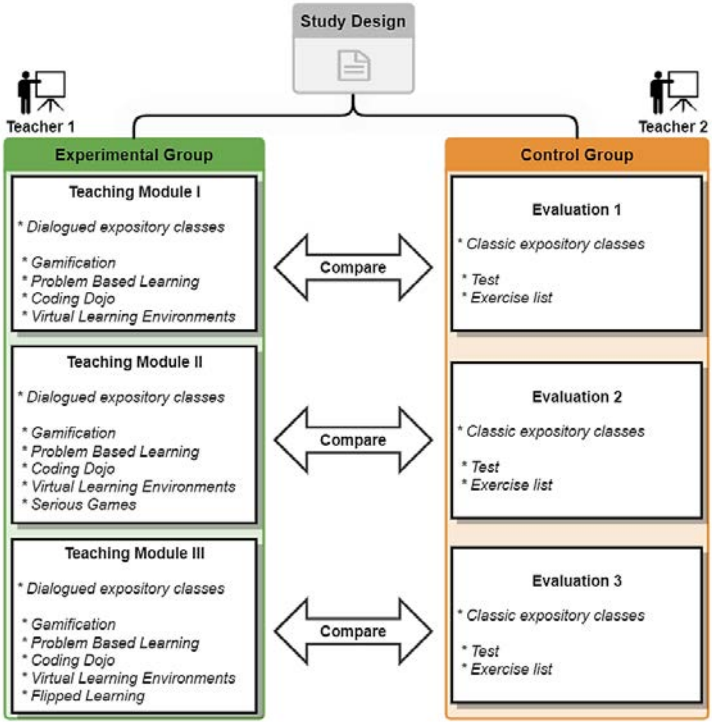

Fig. 2. Study Design.

Table 3

Teaching Units, adapted from (Garcia *et al.*, 2022)

Comparasion	Teaching Units
Teaching Module I **x** **Evaluation 1**	**Unit 1: Algorithms and Programming Languages** • Solving Computational Problems • Informal Algorithm Examples • Formal Concept of Algorithms • Programming Languages and Paradigms
	Unit 2: Computer Programming Basics • Primitive Data Types • Identifiers, Variables and Constants • Input and Output of Data • Operators and Defined Functions • Sequential Structure
Teaching Module II **x** **Evaluation 2**	**Unit 3: Selection Control Structures** • Simple Selection • Composite Selection • Chained Selection • Selection with Multiple Choice Variable
	Unit 4: Repetition Control Structures • Repetition with Control Variable • Repetition with Test at Start • Repetition with Test at End

Continued on next page

Table 5 – continued from previous page

Comparasion	Teaching Units
Teaching Module III **x** **Evaluation 3**	**Unit 5: Program Modularization** • Procedures and Functions • Global and Local Variables • Passing Parameters by Value • Passing Parameters by Reference • Storage Classes • Recursiveness

Table 4

Study Objectives, Research Questions and Hypotheses, adapted from (Garcia *et al.*, 2022)

Study objective 1
Research question 1 (RQ1): What is the effectiveness of learning in Teaching Module I when the proposed approach of using active methodologies for teaching Algorithms is adopted in relation to the traditional approach?
Hypothesis H01: In Teaching Module I, there will be no difference between the scores obtained by the Experimental and Control groups at the Apply level.
Study objective 2
Research question 2 (RQ2): What is the effectiveness of learning in Teaching Module II when the proposed approach of using active methodologies for teaching Algorithms is adopted in relation to the traditional approach?
Hypothesis H02: In Teaching Module II, there will be no difference between the scores obtained by the Experimental and Control groups at the Apply level.
Study objective 3
Research question 3 (RQ3): What is the effectiveness of learning in Teaching Module III when the proposed approach of using active methodologies for teaching Algorithms is adopted in relation to the traditional approach?
Hypothesis H03: In Teaching Module III, there will be no difference between the scores obtained by the Experimental and Control groups at the Apply level.

Hypothesis research questions were defined (as can be seen in Table 4) that served as a framework for the evaluation of the study in the experimental and control groups. The hypotheses were based on the level of application of Bloom's revised taxonomy, which consists of using the knowledge and skills acquired throughout the teaching and learning process, and applying it to problem solving (Ferraz and Belhot, 2010).

4.1. *First Application of the Teaching Plan*

The first experimental application was carried out at the beginning of the second semester of 2021, the study had the voluntary participation of 68 students, who were allocated into two groups (Experimental and Control) with 34 students each.

The results showed that in the first experiment there was a significant statistical gain of the experimental group when compared to the control group, as shown in Table 5. The details of the application of the first experiment can be consulted in (Garcia *et al.*, 2022).

Table 5

Results obtained in the first experiment, adapted from (Garcia *et al*, 2022).

Variables	Experimental Group	Control Group	Statistical Tests
	Evaluation	Evaluation	Results
Teaching Module I x Evaluation 1			
Sample Size	34	34	Shapiro-Wilk test, with a result of P = 0.096.
Minimum	5.4	2	
Maximum	9.6	9.3	Equal Variance Test (Brown-Forsythe), with a result of P = 0.159.
Sum of Points	265.1	219.2	
Median	7.64	6.75	Student-t two-tailed test with P-value = 0.000451 < 0.05.
First Quartile	6.85	5.50	
Third Quartile	8.96	8.00	
Average	7.79	6.44	
Standard Deviation	1.256	1.726	
Note: The experimental score is Δ = 1.35 higher than the control, indicating a possible increase in learning for this group. The results of the two-tailed Student-t test indicate that there were significant statistical gains in the scores of the experimental group.			
Teaching Module II x Evaluation 2			
Sample Size	34	34	Shapiro-Wilk test, with a result of P = 0.226.
Minimum	6.200	1.200	
Maximum	9.900	9.700	Equal Variance Test (Brown-Forsythe), with a result of P = 0.116.
Sum of Points	265.680	216.020	
Median	7.555	6.450	Student-t two-tailed test with P-value = 0.000594 < 0.05.
First Quartile	6.720	5.135	
Third Quartile	9.057	7.625	
Average	7.814	6.354	
Standard Deviation	1.250	2.002	
Note: The experimental score is Δ = 1.46 higher than the control, indicating a possible increase in learning for this group. The results of the two-tailed Student-t test indicate that there were significant statistical gains in the scores of the experimental group.			
Teaching Module III x Evaluation 3			
Sample Size	34	34	Shapiro-Wilk test, with a result of P = 0.448.
Minimum	6.350	4.000	
Maximum	10.00	9.500	Equal Variance Test (Brown-Forsythe), with a result of P = 0.057.
Sum of Points	287.650	219.800	
Median	8.590	6.500	Student-t two-tailed test with P-value = 0.0000000295< 0.05.
First Quartile	7.580	9.287	
Third Quartile	5.400	7.500	
Average	8.460	6.465	
Standard Deviation	1.052	1.523	
Note: The experimental score is Δ = 2.00 higher than the control, indicating a possible increase in learning for this group. The results of the two-tailed Student-t test indicate that there were significant statistical gains in the scores of the experimental group.			

4.2. *Second Application of the Teaching Plan*

A second experiment was carried out in the middle of the second semester of 2021, the study had the voluntary participation of 72 students, who were allocated into two groups (Experimental and Control) with 36 students each. The schedule used in carrying out the second experiment was based on the strategy adopted in (Garcia *et al.*, 2022), and its details are described in Table 6.

Table 6

Experiment Schedule (Garcia *et al.*, 2022)

Days	Control Group	Experimental Group
Day 1 Inaugural Class	Presentation of the subject, how it is conducted and the teaching plan used. Availability of support material.	Presentation of the subject and the functioning of the teaching plan used (Garcia *et al.*, 2021b). Availability of support material.
Days 1 to 5	**Classic lectures:** About the topics of teaching units 1 and 2 present in Table 3. **Test:** Evaluation activity with multiple choice and discursive questions about the content taught in teaching units 1 and 2.	**Dialogue lectures:** about teaching module I – Development of the Programming Logic: With the topics of Unit 1 and Unit 2 present in Table 3, supported by active teaching methodologies: **Gamification:** Using game elements to guide the scoring of course tasks and activities. **Problem Based Learning:** Dynamics focused on the active learning of students, which guide the conduct of Challenges. **Virtual Environment:** Tool where class materials are made available, in addition to serving as a central place for students' virtuais meetings to share knowledge and send out activities. **Coding Dojo 1:** Pseudocode construction challenge using the VisuAlg Tool. All students participate through alternating roles: pilot, copilot and audience. The grade given to the participants takes into account the amount of challenges performed and their correctness. **Extraclass Challenge 1:** Problems that involve the subjects present in the content of all classes of the current teaching unit, these must be carried out outside the classroom. **Challenges:** Problems that involve the subjects present in the content of each class of the current teaching unit and that are carried out after the dialogued expository classes. **Mission 1:** Evaluation activity carried out at the end of teaching module I and which includes all the content covered in the current teaching unit.
Days 6 to 11	**Classic lectures:** About the topics of teaching units 3 and 4 present in Table 3. **Test:** Evaluation activity with multiple choice and discursive questions about the content taught in teaching units 3 and 4.	**Dialogue lectures:** about teaching module II – Software Construction with Scratch: With the topics of Units 3 and 4 present in Table 3, supported by active teaching methodologies: **Gamification:** Using game elements to guide the scoring of course tasks and activities. **Problem Based Learning:** Dynamics focused on the active learning of students, which guide the conduct of Challenges. **Serious Games:** Use of the theme of games with a focus on learning the content covered in the current module. The theme of serious games will be inserted in the activities of the current module: Challenges, Extraclass Challenge 2, Coding Dojo 2 and Mission 2.

Continued on next page

Table 6 – continued from previous page

Days	Control Group	Experimental Group
		Virtual Environment: Tool where class materials are made available, in addition to serving as a central place for students' virtual meetings to share knowledge and send out activities. **Coding Dojo 2:** Block programming activity aimed at creating simplified games that involve the contents covered in the current teaching module using Scratch. All students participate through alternating roles: pilot, copilot and audience. The grade given to the participants takes into account the amount of challenges performed and their correctness. **Extraclass Challenge 2:** Problems with the theme of serious games that involve the subjects present in the content of all classes of the current teaching unit, these must be carried out outside the classroom. **Challenges:** Problems with the theme of serious games that involve the subjects present in the content of each class of the current teaching unit and that are carried out after the dialogued expository classes. **Mission 2:** Evaluation activity based on the theme of serious games, carried out at the end of teaching module II and which includes all the content covered in the current teaching unit.
Days 12 to 17	**Classic lectures:** About the topics of teaching unit 5 present in Table 3. **Test:** Evaluation activity with multiple choice and discursive questions about the content taught in teaching unit 5.	**Dialogue lectures:** on teaching module III – Software Construction with programming language: With the topics of Unit 5 present in Table 3, supported by active teaching methodologies: **Gamification:** Using game elements to guide the scoring of course tasks and activities. **Problem Based Learning:** Dynamics focused on the active learning of students, which guide the conduct of Challenges. **Flipped Classroom:** Dynamics focused on the autonomous learning of students, which will guide the acquisition of knowledge of the teaching units worked on in the current module. **Virtual Environment:** Tool where class materials are made available, in addition to serving as a central place for students' virtual meetings to share knowledge and send out activities. **Coding Dojo 3:** Code construction activity using the C programming language, involving the contents covered in the current teaching module. All students participate through alternating roles: pilot, copilot and audience. The grade given to the participants takes into account the amount of challenges performed and their correctness. **Extraclass Challenge 3:** Problems that involve the subjects present in the content of all classes of the current teaching unit, these must be carried out outside the classroom. **Challenges:** Problems that involve the subjects present in the content of each class of the current teaching unit and that are carried out after the dialogued expository classes. **Mission 3:** Evaluation activity carried out at the end of teaching module III and which includes all the content covered in the current teaching unit.
Day 17 Feedback	Content Perception Questionnaire.	Content Perception Questionnaire. Questionnaire on teaching approaches

5. Data Analysis

In this section, the data obtained from the execution of the experiment presented in this work are presented. The analysis will be carried out from the research questions RQ1, RQ2, and RQ3 defined in this work.

5.1. *Analysis of Research Question 1*

To analyze RQ1, comparisons were made between the average scores obtained in the experimental and control groups in the Teaching Module I versus Evaluation 1 in order to identify statistical evidence that would allow us to refute H01. The two-tailed Student-t test was used to evaluate the difference between two different populations, in addition to the success of the test, it was necessary to evaluate the variance and normality of the data, therefore, the Brow-Forsithe and Shapiro-Wilk Equal Variance Tests were used for normality assumptions.

It was observed that the experimental group in the Teaching Module I scored 7.90 ± 1.271 in the evaluation, while the control group in Evaluation 1 scored 6.52 ± 1.553. It can be noted that the experimental group has a score $\Delta = 1.38$ higher than the control, this shows that there is a significant difference between the average scores of the groups, thus indicating a possible increase in the learning of the experimental group, corroborating the results found in (Garcia *et al.*, 2022).

Before performing the Student-t test, data normality was verified using the Shapiro-Wilk test, which obtained a result of $P = 0.136$, indicating that the data have a distribution that meets normality assumptions. In addition, we also sought to verify the variance of the data, using the Equal Variance Test (Brown-Forsythe), which obtained a result of $P = 0.492$, indicating that the data have the same variance.

The results of the Student-t test ($\alpha = 0.05$) showed that there are positive effects of the use of active methodologies in the experimental group, as can be seen in the test result that obtained P-value = $0.0000995 < 0.05$. Thus, the significance level derived from the test provided statistical evidence to reject H01, as it was identified that there are significant differences between the data of the evaluated groups. Table 7 summarizes the results obtained for QP1.

Table 7

Comparison of Learning Effectiveness between Participating Groups (Student-t) in Teaching Module I x Evaluation 1

Variables	Experimental Group	Control Group
	Evaluation	Evaluation
Sample Size	36	36
Minimum	6.20	3.30
Maximum	10.00	9.50
Sum of Points	284.40	234.70
Median	7.80	6.80
First Quartile	6.80	5.50
Third Quartile	9.18	7.78
Average	7.90	6.52
Standard Deviation	1.271	1.553

Table 8

Comparison of Learning Effectiveness between Participating Groups (Student-t) in Teaching Module II x Evaluation 2

Variables	Experimental Group	Control Group
	Evaluation	Evaluation
Sample Size	36	36
Minimum	6.00	2.30
Maximum	10.00	9.70
Sum of Points	303.80	236.90
Median	8.50	6.60
First Quartile	7.68	5.73
Third Quartile	9.33	7.60
Average	8.44	6.58
Standard Deviation	1.092	1.720

5.2. *Analysis of Research Question 2*

For the analysis of RQ2, comparisons were made between the average scores obtained in the experimental and control groups in the Teaching Module II versus Evaluation 2, in order to identify statistical evidence that would allow us to refute H02, following the same pattern of statistical analysis. used in RQ1, through the use of Student-t, Equal Variance Test (Brown-Forsythe) and Normality Test (Shapiro-Wilk).

The experimental group in Teaching Module II obtained a score of 8.44 ± 1.092 in the evaluation, while the score obtained by the control group in Evaluation 2 was 6.58 ± 1.720. It was observed that the score of the experimental group is $\Delta = 1.86$ higher than that of the control group, which may indicate a possible learning gain in the experimental group.

The Shapiro-Wilk test obtained a result of $P = 0.291$, indicating that the data have a distribution that meets the assumptions of normality and the results obtained with the Equal Variance Test (Brown-Forsythe) had a result of $P = 0.063$, indicating that the data have the same variance.

The results of the Student-t test ($\alpha = 0.05$) showed that there are positive effects of the use of active methodologies in the experimental group, as can be seen in the test result that obtained P-value $= 0.000000649 < 0.05$. Thus, the significance level derived from the test provided statistical evidence to reject H02, as it was identified that there are significant differences between the data of the evaluated groups. Table 8 summarizes the results obtained for RQ2.

5.3. *Analysis of Research Question 3*

For the analysis of RQ3, comparisons were made between the average scores obtained in the experimental and control groups in the Teaching Module III versus Evaluation 3 in order to identify statistical evidence that would allow refuting H03, following the same

Table 9
Comparison of Learning Effectiveness between Participating Groups (Student-t) in Teaching Module III x Evaluation 3

Variables	Experimental Group	Control Group
	Evaluation	Evaluation
Sample Size	36	36
Minimum	5.30	4.00
Maximum	9.70	9.60
Sum of Points	278.80	240.40
Median	7.55	6.75
First Quartile	6.90	6.00
Third Quartile	8.93	7.50
Average	7.74	6.68
Standard Deviation	1.247	1.393

pattern of statistical analysis used in RQ1 and RQ2, using Student-t, Equal Variance Test (Brown-Forsythe) and Normality Test (Shapiro-Wilk).

The experimental group in the Teaching Module III obtained a score of 7.74 ± 1.247 in the evaluation, while the score obtained by the control group in Evaluation 3 was 6.68 ± 1.393. It was observed that the score of the experimental group is $\Delta = 1.06$ higher than that of the control group, which may indicate a possible gain in the learning of the experimental group.

The Shapiro-Wilk test obtained a result of $P = 0.367$, indicating that the data have a distribution that meets the assumptions of normality and the results obtained with the Equal Variance Test (Brown-Forsythe) had a result of $P = 0.634$, indicating that the data have the same variance.

The results of the Student-t test ($\alpha = 0.05$) showed that there are positive effects of the use of active methodologies in the experimental group, as can be seen in the test result that obtained P-value = $0.00104 < 0.05$. Thus, the level of significance derived from the test provided statistical evidence to reject H03, as it was identified that there are significant differences between the data of the groups evaluated. Table 9 summarizes the results obtained for RQ3.

6. Discussion of Results

The results obtained with the experiment presented in this study indicate that there is a learning effectiveness in the use of the form of intervention proposed in this work when compared with the traditional teaching method used in introductory programming subjects, such as algorithms. This can be observed in from the analysis of hypotheses H01 “*In Teaching Module I, there will be no difference between the scores obtained by the Experimental and Control groups at the Apply level*”, H02 “*In Teaching Module II, there will be no difference between the scores obtained by the Experimental and Control* *groups at the Apply level*" and H03 "*Hypothesis H03: In Teaching Module III, there will be no difference between the scores obtained by the Experimental and Control groups at the Apply level*", which were rejected, as it was found from statistical analyzes that there is a significant gain in the scores achieved in the experimental groups when compared with the control groups, corroborating the results obtained in (Garcia *et al.*, 2022).

The experience generated with the use of active methodologies can be very positive in the teaching of algorithms, the learning gains identified in this work are similar to the reports in (Marinho *et al.*, 2016; Giraffa and Müller, 2017; Oliveira, *et al.*, 2017; Fonseca and Brito, 2021), as it can be noted that during the experiment the students of the experimental group were interacting more and more and their engagement was increasing significantly. The classes in the experimental group were based on the strategy defined in (Garcia *et al.*, 2021b), the entire form of intervention was previously presented to the students and the activities performed in the experimental group were student-centered so that there was an equivalence of at least minus 50% of theoretical and practical workload.

In the control group, classes were based on the traditional way of teaching, with a theoretical workload that was superior to practice, reaching an average of 60% to 70% of theory. The practical activities consisted of carrying out lists of exercises in computer labs and evaluations at the end of each teaching stage. The high theoretical workload of the control group may be one of the reflections in the lower result than the experimental group, presented through the analysis of hypotheses H01, H02 and H03.

During the execution of the experiment, some weaknesses arising from its execution in the remote teaching format were identified. During the classes, some students from both the experimental group and the control group had problems connecting to the internet, problems with their computers and also problems related to power outages. It is worth mentioning that the experiment has not yet been conducted in the face-to-face format due to health factors arising from the COVID-19 pandemic, whose restrictions made it difficult to carry out the experiments face-to-face.

It was observed that the students started to get dispersed when the lectures (theoretical) exceeded the time of one hour in duration. To avoid the loss of the students' focus, the professors of the experimental group used the active strategies of the teaching plan to promote the engagement and interaction of the students, through the performance of dynamics that promote the exchange of students' knowledge.

Another point observed is the increase in the interaction of students in the experimental group, which grows as the active methodologies are used, so that on the first school day there is a certain difficulty in promoting interaction, dialogue and debates, but as the classes go on, the limitations related to the interaction and exchange of knowledge of the students are overcome.

7. Threats to Validity

The results presented in this research, although positive, must be interpreted with caution, as the studies were carried out from controlled experiments. In this sense, this section presents some threats that have been identified and that, if not dealt with properly, can influence the results. Therefore, it is advisable that the interpretation of the results or the replication of this study consider the limits created from the threats that will be presented.

7.1. *Internal Validity*

For Vasconcelos (2016), internal validity is related to the veracity of the study, so that the results obtained are in line with the procedures adopted in conducting the research and that they are not the result of methodological errors.

In this sense, the analyzed groups (experimental and control) had the same number of participants and had the same profile of students, who were linked to a higher education institution and who had not yet taken the subject of algorithms in their respective courses. To reduce the bias of the results, the students were randomly distributed in the groups, thus allowing the reduction of the confounding factor in addition to allowing the achievement of the similarity of the groups. The students' participation in the experiments was voluntary and all of them filled out a consent form to carry out the study.

With regard to internal threats related to the use of different approaches in the control and experimental groups, the students' search for knowledge was not limited to the materials used and made available, so it is possible that there is a threat related to the maturation of the study. However, as a way to reduce possible impacts related to this threat, the experiment used content standardization, that is, despite the use of different approaches in the experimental and control groups, the content taught was the same.

Regarding a possible threat related to the different teaching strategies used in the groups, this threat was mitigated through the use of independent experts who performed the corrections of the activities of both groups. The activities were not identified, that is, the process evaluation process was not influenced by professors or students.

7.2. *External Validity*

For Vasconcelos (2016), external validity seeks to ensure that the results obtained from a sample under study from a type of analysis can be generalized to a wider population that have the same sample characteristic.

In this sense, the study was carried out with samples of 36 in the experimental group and 36 students in the control group, all of them enrolled in higher education courses in computing. Therefore, it is recommended that generalizations or replications of the study consider the sample profile used.

7.3. *Construction Validity*

For Raymundo (2009), construction validity seeks to ensure that the instruments used in a study are adequate and can assess the data that will be collected. To mitigate this type of threat, research questions and hypotheses were created that allowed analyzing the efficiency of the experimental group when compared to the control group. All activities carried out in the experimental group were planned based on the cognition levels of Bloom's revised taxonomy (Ferraz and Belhot, 2010).

7.4. *Conclusion Validity*

For Travassos, Gurov and Amaral (2002) and Lima, Neto and Emer (2014), the construction validity allows reaching the conclusions of an experiment from the proper use of statistical analysis methods, correct measurements in the treatments used and of the interpretation of the results obtained in a research.

In this sense, to reduce the bias generated by a small number of participants, which can generate low statistical power in the distribution of samples, a strategy recommended by some authors such as (Furtado, 2020; Chaves *et al.* 2015; Wangenheim *et al.*, 2009; Pfahl *et al.* 2003) was used, which consists of the use of statistical tests that accept small samples without impairing the quality of the results.

8. Conclusion

This work presented the results of an experiment carried out with 72 students, where in this experiment the students were submitted to a teaching plan composed of several active methodologies. According to the results presented, it is possible to conclude that, compared to the traditional teaching method, the approach presented in this work proved to be statistically relevant and effective.

In addition to applying the approach, a correlation was made between Bloom's Taxonomy and the algorithms teaching units. Although the results of this work are promising, it is important to emphasize the need for caution and not to generalize the results, as the audience for this study is still considered small. Furthermore, the teaching plan can be applied in different scenarios and with even more participants.

In this way, as a future work, it is intended to evolve the teaching plan and collect the experiences with its application. In addition, the proposed approach may include analysis from the perspective of professors. Finally, we intend to use this teaching plan, with the necessary adaptations, in other programming subjects, since the algorithms subject is just an introductory subject.

References

Alves, G., Rebouças, A., Scaico, P. (2019). Coding dojo como prática de aprendizagem colaborativa para apoiar o ensino introdutório de programação: Um estudo de caso. In: *XXVII Workshop sobre Educação em Computação* (pp. 276–290). SBC.

Amaral, E., Camargo, A., Gomes, M., Richa, C.H., Becker, L. (2017). ALGO+ Uma ferramenta para o apoio ao ensino de Algoritmos e Programação para alunos iniciantes. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. p. 1677.

Barros, D.S.A., Santos, J.R.A. (2018). Técnicas de estudo e gestão do tempo no auxílio a aprendizagem de fundamentos de algoritmos e lógica aplicada a computação. *CIMATech*, 1(5).

Brasil. Ministério da Educação (2016). *Diretrizes Curriculares Nacionais para os cursos de Computação*. Brasilia: MEC, 2016. Available in: `http://portal.mec.gov.br/docman/novembro-2016-pdf/52101-rces005-16-pdf/file`

Bulcão, J.D.S.B. (2017). *Uma Análise Curricular das Disciplinas de Algoritmo e Lógica Computacional em Cursos de Licenciatura em Informática e Computação*. Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – IFRN. 2017.

Calsavara, A., Serra, A.P.G., Zampirolli, F.A., Carvalho, L.S.G., Jonathan, M., Correia, R.C.M. (2018). Método baseado nos Referenciais de Formação da SBC para reestruturação de descritivos de disciplinas de Ciência da Computação em conformidade com as DCN de 2016. In: *XXVI Workshop sobre Educação em Computação*. SBC.

Chaves, R.O., Wangenheim, C.G.V., Furtado, J.C., Oliveira, S.R.B., Santos, A., Favero, E.L. (2015). Experimental Evaluation of a Serious Game for Teaching Software Process Modeling. *IEEE Transactions on Education*, vol. PP, no. 99.

Sousa, R.R., Leite, F.T. (2020). Usando gamificação no ensino de programação introdutória. *Brazilian Journal of Development*, 6(6), 33338–33356.

Ding, H. (2019). Teaching Research on Computer Programming based on Cultivation of Computational Thinking. In: *2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI)*. IEEE. `https://doi.org/10.1109/csei47661.2019.8938909`

Eteng, I., Akpotuzor, S., Akinola, S.O., Agbonlahor, I. (2022). A Review on Effective Approach to Teaching Computer Programming to Undergraduates in Developing Countries. *Scientific African*, Artigo e01240. `https://doi.org/10.1016/j.sciaf.2022.e01240`

Falkembach, G., Amoretti, M.S., Tarouco, L.M. (2003). Uma experiência de resolução de problemas através da estratégia ascendente: Ambiente de aprendizagem adaptado para algoritmos (a4). In: *Challenges. International Conference of Information and Communication Technologies in Education and 5th SIIE– International Symposium in Educational Computing*, 3., Proceedings.

Ferraz, A.P.C.M., Belhot, R.V. (2010). Taxonomia de Bloom: Revisão Teórica e Apresentação das Adequações do Instrumento para Definição de Objetivos Instrucionais. *Gestão & Produção*. Handbook I, Cognitive Domain: Longmans.

Fonseca, J.G., Brito, C.A.F. (2021). Percepção dos alunos do curso técnico em desenvolvimento de sistemas após vivências com o método de ensino peer instruction. *Research, Society and Development*, 10(3), e10710312382–e10710312382.

Freire, L., Coutinho, J., Lima, V., Lima, N. (2019). Uma proposta de encontros de tutoria baseada em Metodologias Ativas para disciplinas de Programação Introdutória. In: *Workshops do Congresso Brasileiro de Informática na Educação*, (Vol. 8, No. 1, p. 298).

Furtado, J.C.C. (2020). *Uma Abordagem para o Ensino do Controle Estatístico de Processos em Cursos Superiores de Computação*; orientador, Sandro Ronaldo Bezerra Oliveira. – 2020. 177f. Tese (Doutorado) – Universidade Federal do Pará. Instituto de Ciências Exatas e Naturais. Programa de Pós-Graduação em Ciência da Computação. Belém.

Gallego-Durán, F.J., Satorre-Cuerda, R., Compañ-Rosique, P., Villagrá-Arnedo, C.J., Molina-Carmona, R., Llorens-Largo, F. (2020). A low-level approach to improve programming learning. *Universal Access in the Information Society*. `https://doi.org/10.1007/s10209-020-00775-y`

Garcia, F.W.S., Carvalho, E.C., Oliveira, S.R.B. (2020). A survey of teaching methods for a programming subject: A literature review. In: *CONTECSI International Conference on Information Systems and Technology Management*. Brazil. `https://doi.org/10.5748/17contecsi/pse-6454`

Garcia, F.W.S., Carvalho, E.C., Oliveira, S.R.B. (2021a). A strategy for the application of active teaching methodologies for the algorithms subject: an exploratory study on a way to use the methodologies. In: *CONTECSI International Conference on Information Systems and Technology Management*. Brazil.

Garcia, F.W.S., Carvalho, E.C., Oliveira, S.R.B. (2021b). Use of active methodologies for the development of a teaching plan for the algorithms subject. In: *2021 IEEE Frontiers in Education Conference (FIE)*. IEEE. `https://doi.org/10.1109/fie49875.2021.9637482`

Garcia, F.W.S, Oliveira, S.R.B., Carvalho, E.C. (2022). Application of a Teaching Plan for Algorithm Subjects Using Active Methodologies: An Experimental Report. *International Journal of Emerging Technologies in Learning (iJET)*, 17(07), 175–207. `https://doi.org/10.3991/ijet.v17i07.28733`

Garcia, M.B. (2021). Cooperative learning in computer programming: A quasi-experimental evaluation of Jigsaw teaching strategy with novice programmers. *Education and Information Technologies*, 26(4), 4839–4856.

Gil, A.C. (2010). *Como elaborar projetos de pesquisa*. 5. ed. São Paulo: Atlas.

Gil, A.C. (1991). *Como elaborar projetos de pesquisa.* São Paulo: Atlas.
Giraffa, L.M.M., Müller, L. (2017). Metodologia baseada em sala de Aula invertida e Resolução de Problemas relacionado ao cotidiano dos estudantes: uma proposta para ensinar programação para iniciantes. *International Journal on Computational Thinking (IJCThink)*.
Hoed, R.M. (2016). *Análise da Evasão em Cursos Superiores: O Caso da Evasão em Cursos Superiores da Área de Computação* (Dissertação de Mestrado). Universidade de Brasília. Brasilia.
Lakatos, E.M., Marconi, M.A. (2011). *Metodologia Científica*. 6aed. São Paulo: Atlas.
Lima, V.C.M., Neto, A.G.S.S., Emer, M.C.F.P. (2014). Investigação experimental e práticas ágeis: ameaças à validade de experimentos envolvendo a prática ágil Programação em par. *Revista Eletrônica de Sistemas de Informação*, 13(1).
Marinho, C., Moreira, L., Coutinho, E., Paillard, G., de Lima, E.T. (2016). Experiências no uso da metodologia coding dojo nas disciplinas básicas de programaçao de computadores em um curso interdisciplinar do ensino superior. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 5, No. 1, p. 1097).
Morais, C.G.B., Neto, F.M.M., and Osório, A.J.M. (2020). Dificuldades e desafios do processo de aprendizagem de algoritmos e programação no ensino superior: uma revisão sistemática de literatura. *Research, Society and Development*, 9(10), e9429109287–e9429109287.
Oliveira, M.G., Neves, A., Lopes, M.F.S., Medeiros, H.F., Andrade, M.B., Reblin, L.L. (2017). Um curso de programação a distância com metodologias ativas e análise de aprendizagem por métricas de software. *RENOTE*, 15(1).
Oroma, J., Wanga, H., Ngumbuke, F. (2012). *Challenges of Teaching and Learning Computer Programming in Developing Countries: Lessons from Tumaini University*. DOI: 10.13140/2.1,3836.
Pfahl, D., Laitenberger, O., Dorsch, J., Ruhe, G. (2003). An externally replicated experiment for evaluating the learning effectiveness of using simulations in software project management education. *Empirical Software Engineering*, Kluwer Academic, The Netherlands, v.8, p.367–395.
Prasad, A., Chaudhary, K., Sharma, B. (2022). Habilidades de programação: Visualização, interação, linguagem de casa e resolução de problemas. *Educação e Tecnologias da Informação*, 27(3), 3197–3223.
Ramos, V., Wazlawick, R., Galimberti, M., Freitas, M., Mariani, A.C. (2015). A Comparação da Realidade Mundial do Ensino de Programação para Iniciantes com a Realidade Nacional: Revisão sistemática da literatura em eventos brasileiros. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. p. 318.
Raymundo, V.P. (2009). Construção e validação de instrumentos: um desafio para a psicolinguística. *Letras de hoje*, 44(3).
Saito, D., Washizaki, H., Fukazawa, Y. (2015). Work in progress: A comparison of programming way: Illustration-based programming and text-based programming. In: *2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. IEEE, 2015. p. 220–223.
Sajjanhar, A., Faulkner, J. (2019). Second life as a learning environment for computer programming. *Education and Information Technologies*, 24(4), 2403–2428.
https://doi.org/10.1007/s10639-019-09879-2
Silva, G.A.P.D. (2017). *Flipped Classroom, Aprendizagem Colaborativa e Gamification: Conceitos Aplicados em um Ambiente Colaborativo Para Ensino de Programação*. Master's thesis, Universidade Federal de Pernambuco.
Teague, D. (2011). *Pedagogy of Introductory Programming: A People-First Approach*.
Travassos, G.H., Gurov, D., Amaral, E.A.G. (2002). *Introdução à Engenharia de Software Experimental*. Rio de Janeiro: COPPE/UFRJ.
U.S Bureau of Labor Statistics (2022). *Department of Labor, Occupational Outlook Handbook, Software Developers, Quality Assurance Analysts, and Testers*. Accessed 12 July 2022:
https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm
Vasconcelos, A.C., De Andrade Souza, G.L., Brainer, S.A.B., Soares, R.M., Dos Santos Barbosa, L.D., De Souza Campos, P.I. (2019). As estratégias de ensino por meio das metodologias ativas. *Brazilian Journal of Development*, 5(5), 3945–3952.
Vasconcelos, B.C. (2016). Importância da validade externa na pesquisa científica. *Revista de Cirurgia e Traumatologia Buco-maxilo-facial*, 16(2), 04–05.
Vieira, C.E.C., de Lima Junior, J.A.T., de Paula Vieira, P. (2015). Dificuldades no Processo de Aprendizagem de Algoritmos: uma Análise dos Resultados na Disciplina de AL1 do Curso de Sistemas de Informação da FAETERJ – Campus Paracambi. *Cadernos UniFOA*, 10(27), 5–15.
Wangenheim, C.G.; Thiry, M.; Kochanski, D. (2009). Empirical evaluation of na educational game on software measurement. *Empirical Software Engineering*, 14(4), 418–452.

White, S. K. (2022). *CIO United States*. Accessed 11 July 2022:
`https://www.cio.com/article/3235944/hiring-the-most-in-demand-tech-jobs-for-2021.html`

Zorzo, A.F.; Nunes, D., Matos, E.; Steinmacher, I., Leite, J., Araujo, R.M., Correia, R., Martins, S. (2017) *Referenciais de Formação para os Cursos de Graduação em Computação*. Sociedade Brasileira de Computação (SBC). 153p, ISBN 978-85-7669-424-3.

F.W.S. Garcia is PhD in Computer Science with emphasis in Software Engineering from Graduate Program in Computer Science (PPGCC) at Federal University of Pará (UFPA), Brazil. He is Professor at Federal Rural University of Amazon (UFRA), Brazil. His research areas are: Algorithms, Software Engineering, Software Quality and Informatics Education.

S.R.B. Oliveira is PhD in Computer Science with emphasis in Software Engineering and did his postdoctoral internship at the Informatics Center, Federal University of Pernambuco, Brazil. He is Associate Professor and Researcher at the Faculty of Computing (FACOMP) and Graduate Program in Computer Science (PPGCC) at Federal University of Pará (UFPA), Brazil. He is the Lead Coordinator of the SPIDER research project, which has won many scientific awards and has already graduated many doctoral, master, graduate and scientific initiation students in Computer Science. He is consultant, appraiser and instructor of the MPS.BR and CMMI software and service quality models. His research areas are: Informatics in Education, Software Engineering and Software Process Improvement.

E.C. Carvalho is Master student in Computer Science with emphasis in Software Engineering from Graduate Program in Computer Science (PPGCC) at Federal University of Pará (UFPA), Brazil. His research areas are: Project Management, Software Engineering and Software Quality.