



Volume 20, 2021

## ACTIVE LEARNING IN THE CONTEXT OF THE TEACHING/LEARNING OF COMPUTER PROGRAMMING: A SYSTEMATIC REVIEW

---

João Henrique Berssanette*	Instituto Federal do Paraná, Paraná, Brazil	joao.berssanette@ifpr.edu.br
Antonio Carlos de Francisco	Universidade Tecnológica Federal do Paraná, Paraná, Brazil	acfrancisco@utfpr.edu.br

\* Corresponding author

### ABSTRACT

---

Aim/Purpose	This paper presents the results of a systematic literature review that sought to identify the studies that relate the different pedagogical techniques by which active learning is developed in the context of the teaching/learning of computer programming, with the objective to characterize the approaches, the pedagogical techniques used, the application, the contributions, and difficulties of implementation reported by these studies.
Background	The literature has shown that teachers in teaching programming have been less successful than they should and need to be, so dropout and failure rates for students remain high. In this sense, much has been discussed about the possibilities and limitations of using the active learning pedagogical techniques in this context.
Methodology	For this review, an analysis from all studies mentioning active learning in the context of the teaching/learning of computer programming published between 2014 and 2019 was performed, retrieved in WOS, SCOPUS, ScienceDirect, and ACM Digital Library. The selection of studies was based on a set of criteria established to guide the selection process, including alignment with the research questions and evaluating the quality of studies.
Contribution	This study contributes to an overview of the current scenario, characterizing the research studies that associate the different pedagogical techniques of

Accepting Editor Janice Whatley | Received: November 20, 2021 | Revised: January 25, February 21, March 8, April 9, April 20, 2021 | Accepted: May 3, 2021.

Cite as: Berssanette, J. H., & de Francisco, A. C. (2021). Active learning in the context of the teaching/learning of computer programming: A systematic review. *Journal of Information Technology Education: Research*, 20, 201-220. <https://doi.org/10.28945/4767>

(CC BY-NC 4.0) This article is licensed to you under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/). When you copy and redistribute this paper in full or in part, you need to provide proper attribution to it to ensure that others can later locate this work (and to ensure that others do not accuse you of plagiarism). You may (and we encourage you to) adapt, remix, transform, and build upon the material for any non-commercial purposes. This license does not permit you to use this material for commercial purposes.

	active learning in the context of the teaching/learning of computer programming.
Findings	The results showed that the studies' approaches usually occur by intervention/pedagogical experiment or by the development of a tool, instrument or methodology. The flipped classroom methodology has obtained a notable prominence in research. The use of active learning pedagogical techniques results in greater acceptance or positive feedback from students, increasing their satisfaction or motivation to improve the learning experience, learning outcomes, or student performance. However, they require a greater effort/work by the teacher to plan and/or execute the teaching/learning process. It should be highlighted that the contributions observed for the teaching/learning process of computer programming derive from investigations mainly concentrated in the university context, aiming to observe if these contributions can be reproduced in other education levels. The contributions observed in the studies regarding the uses of pedagogical techniques of active learning in the context of computer programming indicate that their use can contribute significantly to the teaching/learning process, showing it to be a viable alternative and consistent with the reduction of the failures in the learning of programming.
Recommendations for Practitioners	Considering that over the years the teaching/learning process of computer programming has been a challenge for students, based on the findings of this research, we recommend that teachers consider restructuring their traditional practices of teaching computer programming, making use of pedagogical techniques of active learning to obtain better learning results of their students.
Recommendations for Researchers	We recommend that fellow scholars consider investigating how the difficulties inherent to teachers related to the teaching/learning process of programming may relate to difficulties concerning students and content, especially with regard to traditional teaching practices.
Impact on Society	This study adds to previous systematic reviews of the literature, specifically studies that relate active learning to the context of teaching/learning of programming. It is hoped that the findings of this article can support other research that addresses the topic, enabling its development and deepening, through the developed basis from which active learning researchers can work.
Future Research	Future studies may investigate the benefits of using different pedagogical techniques for active learning and the costs related to the higher cognitive burden imposed by these techniques for learning computer programming.
Keywords	computer programming, active learning, teaching/learning, teaching methodologies, systematic review

## INTRODUCTION

---

Due to the relevance of computers for society, the promotion of skills related to computer programming has been notable. In this sense, the development of these skills in formal education invariably goes through the process of teaching/learning computer programming. However, when it comes to the teaching/learning process of programming, the literature of the area over the years has shown that, when teaching programming to students, teachers are less successful than they should and need to be (Bennedsen & Caspersen, 2007; Gomes & Mendes, 2014; McGettrick et al., 2005; Simon et al., 2019; Sleeman, 1986; Watson & Li, 2014).

The origin of the problems associated with programming teaching/learning is broad since it involves several variables from different actors involved in the process. The existing literature mainly highlights the difficulties related to students, such as: reading and interpretation; development of logical reasoning; deficient problem-solving skills; low abstraction capacity; low level of knowledge in mathematics; inappropriate study habits; little motivation; lack of persistence or little commitment (Bosse & Gerosa, 2016; Gomes & Mendes, 2007; Medeiros et al., 2019; Qian & Lehman, 2017; Robins et al., 2003).

However, it must be considered that at least part of the problems related to the learning of programming can originate from the teaching practices developed during the teaching/learning process; that is, from the difficulties intrinsic to the teachers' methods. In this sense, Vihavainen et al. (2014), when conducting a systematic review of articles describing introductory programming teaching approaches in order to analyze the effect that various interventions can have on the pass rates of introductory programming courses, concluded that, on average, teaching interventions can improve programming pass rates by nearly one third when compared to a traditional lecture and lab-based approach.

Thus, the traditional teaching based on lectures, which is the pedagogical approach most used in classrooms at different levels and teaching contexts, and its efficiency in the teaching/learning process, has been questioned in different areas of knowledge. Part of the concern with the efficiency of traditional teaching stems from the teaching format that can often place students in a passive role so that they receive the knowledge transferred by the teacher from isolated facts, out of context, in addition to this knowledge being abstract with the significant possibility of being forgotten later.

Much has been discussed in education about the possibilities and limitations of active learning (Freeman et al., 2014; Michael, 2006). It is prudent to highlight that this learning model contrasts in no small extent with the traditional approach of lecture-based teaching. It aims at the active participation of students and their engagement through a wide variety of methodologies, strategies, approaches, and pedagogical techniques centered on the student, so that they are involved in the teaching/learning process, in order to apply their knowledge in meaningful ways, employing higher-order thinking skills and reflecting on their learning to build new knowledge (Bonwell & Eison, 1991; Richardson, 2008).

In this context, the main aim of this systematic review was to identify the studies that relate the different pedagogical techniques (methodologies, strategies, and approaches) by which active learning, in the context of the teaching/learning of programming, is developed in order to characterize the approaches, pedagogical techniques used, applications, contributions, and reported difficulties of implementation in the referred context. In the context of teaching/learning of computer programming, several reviews can be found in the literature (Borges et al., 2018; Luxton-Reilly et al., 2018; Medeiros et al., 2019; Pears et al., 2007; Qian & Lehman, 2017; Robins et al., 2003). However, those reviews focus on more general aspects of the process of teaching/learning of programming, including, for instance, problems, difficulties, and skills related to the learning of programming, as well as teaching methods and tools. In light of this context, we consider that the relevance of this study consists in the specific relationship between the various pedagogical techniques of active learning and the process of teaching/learning of programming, having as the main contribution the development of a study that reflects a current panorama of related research.

## **BACKGROUND**

---

The process of teaching/learning of programming, especially in its introductory context, has been a challenge for students and teachers, resulting in high levels of failure, dropping out, and even abandoning the course, in different levels, courses, and teaching contexts. In this direction, Martins et al. (2018) reveal that the teaching of programming faces serious challenges worldwide. Many students drop out of computer science courses in their early years, because they are unable to understand many of the concepts taught in introductory programming courses.

Studies such as that of Bennedsen and Caspersen (2007), Watson and Li (2014) and Simon et al. (2019) indicate that failure rates in introductory programming subjects are approximately 30%, which means that these subjects can be considered one of the existing bottlenecks in computing, informatics, and other courses that include these subjects in their curricula, hindering or even preventing students from continuing such courses. To reverse this scenario, teachers and researchers have dedicated themselves to studying the causes of these complexities and developing varied proposals, primarily aimed at students who are new to programming. Those proposals aim to make the teaching/learning process more effective.

However, it should be noted that none of the proposals proved to be complete or even generic to the point of solving the problems inherent in learning how to program that still persist (Gomes & Mendes, 2014; Koulouri et al., 2015; Vihavainen et al., 2014). Students continue to struggle with learning how to program, thus investigations should account for the teaching/learning process as a whole, not just the difficulties observed in the students, or their cognitive aspects.

It is worth noting that as the majority of studies focus on observing the difficulties related to students, and not the ones from the teaching/learning process itself, it can be hypothesized that these problems for learning cannot be restricted to students only. Neither their lack of abilities nor competencies, make it necessary to observe the entire teaching/learning process, since the teaching processes in place, as well as the learning theories that guide those processes, must have some impact or influence over them.

Several studies have evidenced that the use of pedagogical techniques of active learning is effective and produces superior results, in comparison to the traditional approach based on lectures, indicating the reduction of failure rates, the promotion of a greater interest and engagement of students, by developing and improving their attitudes, in addition to promoting critical thinking and a deeper understanding. This allows for better learning results through a contextualized teaching/learning process that establishes conditions for students to be responsible for their knowledge (Freeman et al., 2014)

In general, it can be said that active learning develops through any instructional method that requires learners to do meaningful learning activities, combined with reflection on what they are learning and doing, and having as their main characteristic an apprentice-centered approach, in which teachers and students collaborate and interact with the content (Prince, 2004). Active learning takes place in various ways through methodologies, strategies, approaches, or pedagogical techniques. Among these methodologies, the following stand out: flipped classroom, peer instruction, project-based learning, problem-based learning, team-based learning, think-pair-share, gamification, game-based learning, case study, hands-on, and many others.

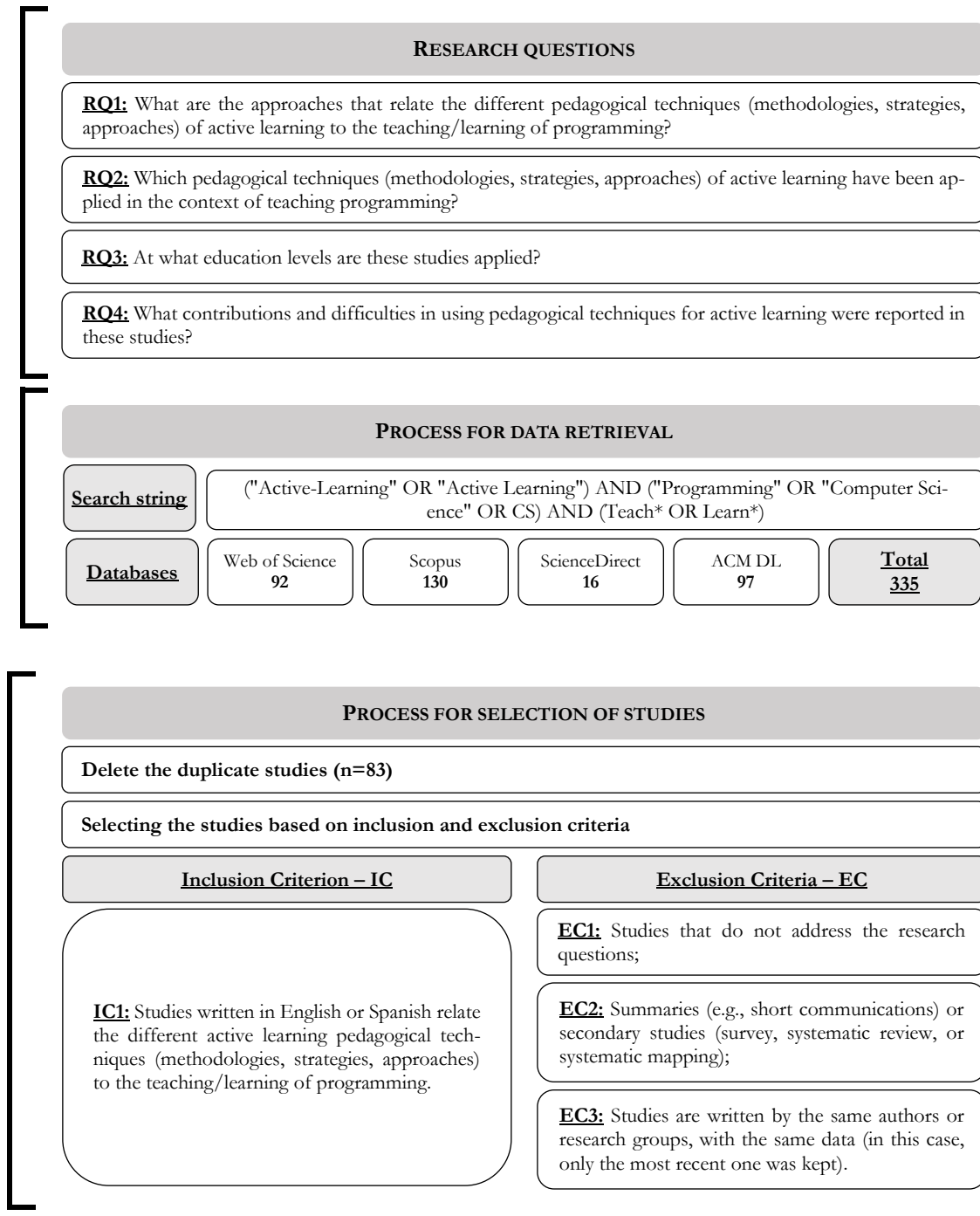
Due to the benefits resulting from the use of pedagogical techniques of active learning, educators started to develop, adapt, and implement a wide variety of them at the most varied levels and contexts of teaching to make the teaching/learning process more efficient and meaningful for their learners. Thus, considering that active learning is an inherent part of the learning of computer programming, due to the application and practical characteristics related to this knowledge, there are prominent attempts in the literature with relevant results, deriving from the use of active learning in the process of teaching/learning of programming, making it possible to hypothesize that the use of the various pedagogical techniques of active learning is a field with promising potential for developing research. In this sense, a systematic review of the literature is conducted in this paper in order to identify and characterize the research studies that relate the different pedagogical techniques by which active learning is developed in the context of teaching/learning of computer programming, to develop a study that reflects a current panorama of the research.

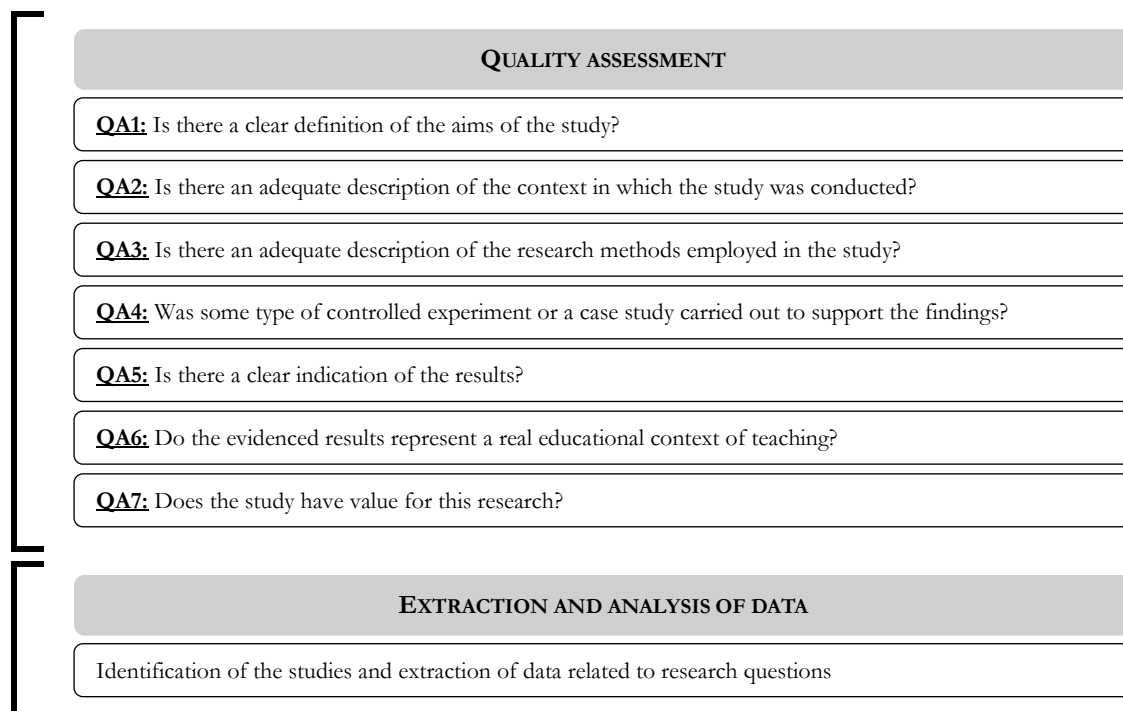
## METHOD

---

This review was developed based on the guidelines for systematic literature reviews presented by Kitchenham (2012), with the aim to identify the studies published between 2014 and 2019 that relate

the various pedagogical techniques of active learning to the context of teaching/learning of programming, in order to characterize the approaches, the pedagogical techniques used, the application, contributions and difficulties reported in the studies. The summary of the protocol developed for this review is presented in Figure 1, together with the research questions.





**Figure 1. Summary of the protocol developed**

As can be seen in Figure 1, to obtain the necessary data to answer the research questions based on the terms that make up the scope of this review (Active Learning; Computer Programming; and Teaching/Learning), an appropriate search string was developed.

In order to verify if the search string developed was adequate, a set of tests was carried out in different databases. After those tests, it was considered that the search string chosen proved to be appropriate for the recovery of studies falling within the scope of this research.

In this way, that search string, given in Figure 1, was implemented automatically through the search engines, having as parameters the title, keywords, and abstract fields, and article as document type, published between the years 2014 and 2019, in the following databases: Web of Science - Core Collection (Clarivate Analytics); Scopus (Elsevier); ScienceDirect (Elsevier); and ACM Digital Library. The search was carried out on September 21, 2020 and retrieved a total of 335 studies in the referred databases.

From the retrieval of the studies, the process of selecting the studies to compose the documentary corpus or basis for review began. Thus, for this review, duplicate studies were initially deleted (n=83). The remaining 252 studies were then submitted to an analysis of the title, keywords, and abstract by the researchers individually, observing the inclusion and exclusion criteria.

As a next step, since the assessment of the quality of existing studies is an underlying concern when performing systematic reviews, the pre-selected studies were assessed for quality. A quality assessment was performed because it allowed for identifying and disregarding studies that were methodologically weak, presented incipient results, or were of low scientific quality.

Thus, as can be seen in Figure 1, to assess the quality of the studies that make up this review, seven questions were developed as an adaptation of the model proposed by Dybå and Dingsøyr (2008) for evaluating the study's quality. To answer those questions, the studies pre-selected based on the selection criteria were read thoroughly. The questions were weighted as follows: 1 (yes, the selected study meets the criterion); 0.5 (partially, the selected study does not report clearly); and 0 (no, the selected

study does not meet the criterion). At the end of the assessment, the sum of the scores obtained by each study was taken based on the quality assessment questions.

The quality assessment questions aim to identify the relevance of the studies regarding their aims, methodology, results, and their reliability and value for this review. Thus, as a cut-off rule for studies to form the basis for this review, it was defined that studies should obtain a score equal to or greater than 4.

For the data extraction process, the studies selected to compose the documentary corpus of this review were identified and made available in a shared spreadsheet, in which one researcher proceeded to extract data related to research questions, with the other verifying this extraction.

## **IMPLEMENTATION OF THE PROTOCOL AND DISCUSSION OF RESULTS**

---

Following the process described for retrieving appropriate papers, 335 studies were initially identified, 83 duplicate studies were discarded from these, leaving 252 studies for later evaluation. The selection process for the 252 recovered studies, based on inclusion and exclusion criteria, removed a further 179 studies, which did not meet the inclusion criteria. Another 33 studies fit into at least one of the exclusion criteria, thus totaling 212 discarded studies, which left 40 pre-selected studies.

The pre-selected studies were then assessed for quality, with a view to analyzing their aims, methodology and results, reliability, and relevance to the aims established for this review, as well as identifying and disregarding studies that were methodologically weak with incipient results, or of low scientific quality. Most of the pre-selected studies received a good score, obtaining an average of six points, which would be somewhat expected, given that these studies were published in scientific journals and peer-reviewed conferences.

Thus, with regard to the quality of the pre-selected studies, only two of the studies scored below 4, and they were discarded. As a result, 38 studies that were pre-selected based on the selection criteria and that obtained a quality score equal to or greater than 4, make up the basis for this review. Details of the 38 studies are given in Appendix A, and the quality assessment can be found in Appendix B.

It is worth noting that, during the process of implementing the systematic literature review protocol in question, there was a good agreement between the researchers, with divergence rates below 10% being registered, both in the selection process and in the quality assessment process. Among these divergences in the selection process, cases can be mentioned related to the exclusion criterion - EC3 - (studies are written by the same authors or research groups, with the same data), or even cases where it was not possible to determine explicitly which are the active learning techniques used in the study, so that one of the researchers considered not including these studies or even classifying them in one of the exclusion criteria. In contrast, the other researcher took the opposite position. In the quality assessment of the studies, the divergences were due to the attribution of different scores in these evaluations.

Thus, for both the selection process and the quality assessment process of studies, meetings were held to discuss, as well as re-analyse, the divergent studies (when necessary) to resolve the differences resulting from the analysis independently of the two investigators for each of the studies.

The following discussion is based around the research questions that guided this review.

### ***APPROACHES THAT RELATE THE DIFFERENT PEDAGOGICAL TECHNIQUES OF ACTIVE LEARNING TO THE TEACHING/LEARNING OF PROGRAMMING (RQ1)***

Based on the studies that form the basis of this review, it can be observed that there are three approaches that relate the various active learning pedagogical techniques to the teaching/learning of programming: pedagogical intervention/experiment; the development of a tool, instrument, or methodology; and the analysis of the teachers' perception. Table 1 identifies the studies within each of these approaches and the number present in the sample.

**Table 1. Approaches that relate active learning pedagogical techniques to teaching/learning of programming**

APPROACH	STUDIES	QUANTITY
Pedagogical intervention/experiment	S01; S02; S03; S04; S05; S08; S09; S10; S11; S12; S14; S16; S17; S18; S19; S20; S21; S22; S23; S24; S25; S27; S28; S29; S30; S32; S33; S34; S35; S37; S38	31
Development of a tool, instrument, or methodology	S06; S07; S13; S15; S26; S31	6
Analysis of teachers' perception	S36	1
	Total	38

As can be seen from Table 1, most studies that relate active learning pedagogical techniques to the teaching/learning of programming, address this relationship through pedagogical interventions or experiments carried out in the classroom. The six studies aim at developing a tool, instrument or methodology; and only one study addresses this relationship through an analysis of the teachers' perception on the use of active learning pedagogical techniques.

It is worth noting that, in studies S09 and S11, the authors conduct a pedagogical intervention, focusing their research on the assessment of active classroom environments; that is, the assessment of physical learning environments developed to promote active learning.

It should be noted that the use of pedagogical interventions or experiments carried out in the classroom to evaluate the different pedagogical techniques of active learning seems to be adequate to collect empirical evidence of the effectiveness of these techniques, as well as the possible contributions and the limitations to their implementation. Also, on the use of pedagogical interventions or experiments in the context of teaching/learning of programming, Vihavainen et al. (2014) show that, on average, teaching interventions can improve programming pass rates by nearly one third when compared to a traditional lecture and lab-based approach.

### ***PEDAGOGICAL TECHNIQUES OF ACTIVE LEARNING APPLIED IN THE CONTEXT OF TEACHING PROGRAMMING (RQ2)***

Regarding the pedagogical techniques applied in the context of teaching/learning of programming (Table 2), it is observed that the Flipped classroom has achieved notable prominence. In addition, the popular and already consolidated active learning pedagogical techniques, such as Project-based learning, Peer instruction, Blended learning, Collaborative learning, Problem-based learning, Game-based learning, Pair programming, Gamification, Hands-on, Inquiry-based learning, Living code, Team-based learning, and Think-Pair-Share, are present in the sample. The basis of reviews includes more



recent pedagogical techniques, such as Process Oriented Guided Inquiry Learning (POGIL), and the adaptation of others, such as Peer-teaching with videos.

**Table 2. Active learning pedagogical techniques (methodologies, strategies, approaches)**

PEDAGOGICAL TECHNIQUES	STUDIES	QUANTITY
Flipped classroom	S01; S03; S04; S05; S16; S18; S19; S20; S21; S22; S28; S29; S30; S31; S33; S34; S35	17
Project-based learning	S06; S10; S12; S24; S28; S32	6
Peer instruction	S02; S09; S37; S38	4
Blended learning	S03; S25; S34	3
Collaborative learning	S12; S14; S23	3
Problem-based learning	S02; S03; S20	3
Game-based learning	S08; S13	2
Pair programming	S22; S23	2
Undetermined	S07; S26	2
Gamification	S15	1
Hands-on	S10	1
Inquiry-based learning	S20	1
Living code	S33	1
Peer-teaching with videos	S27	1
POGIL	S36	1
Team-based learning	S19	1
Think-Pair-Share	S17	1

The active learning methodology, Flipped classroom, implemented by over 40% of the studies that comprise the basis of this review, refers to the concept of inverting roles in the classroom. This was initially proposed by Lage et al. (2000) and was conceived as “Inverted Classroom” and popularized in 2007 by Jonathan Bergmann and Aaron Sams. According to Bergmann and Sams, the concept of the flipped classroom is, basically: “that which is traditionally done in class, is now done at home, and that which is traditionally done as homework is now completed in class” (2012, p. 13).

The second active learning pedagogical technique most implemented by the studies that comprise the basis of this review is Project-based learning. The idea of working with projects as a pedagogical resource in the construction of knowledge has its origin and theoretical foundation in the thinking of the New School’s philosopher John Dewey, based on learning by doing (Noordin et al., 2011; Wang et al., 2016). The Project-based learning technique aims to involve students in obtaining information and skills from the search for solving complex, authentic problems that are planned in search of efficient and dynamic learning in which the students are active in building their knowledge (Peng et al., 2017).

The pedagogical strategy, peer instruction, implemented by four studies, was developed in 1991 by Eric Mazur (Mazur, 1997; Mazur & Somers, 1999). Its objective is to promote the learning of the fundamental concepts of the content being studied, through interaction between students,

stimulating exchange and discussion in order to emphasize the processes and results obtained in a collaborative environment capable of influencing the entire process of students' learning, as well as in the teacher-student and student-student relationship. This contributes effectively to developing skills, such as questioning, debating, listening, doing, and teaching.

As shown in Table 2, there are several pedagogical techniques (methodologies, strategies, approaches) for active learning. However, this review is not intended to present and discuss all of these techniques, considering that more information regarding them can be obtained through the studies retrieved by this review. Therefore, only the active learning pedagogical techniques that have a greater representation within the studies that make up the basis of this review have been described.

It should also be clarified that it was not possible to explicitly determine which active learning techniques are the ones used in studies S07 and S26, even though the proposed approach to the teaching/learning process present in the studies clearly is developed based on the concepts of active learning.

### ***EDUCATION LEVELS AT WHICH STUDIES WERE APPLIED (RQ3)***

With regard to the application of studies (Table 3), it is observed that all studies selected to form the basis of this review, in some way, were applied covering almost all levels of education.

**Table 3. Education levels at which studies were applied**

LEVEL OF EDUCATION	STUDIES	QUANTITY
Elementary School	S08	1
High school	S29	1
University/college	S01; S02; S03; S04; S05; S06; S07; S09; S10; S11; S12; S13; S14; S16; S17; S18; S19; S20; S21; S22; S23; S24; S25; S26; S27; S28; S30; S31; S32; S33; S34; S35; S37; S38	34
Graduate studies	S07	1
Teacher training	S36	1
Indeterminate	S15	1

However, the vast majority of studies (over 80%) are concentrated in university/college education, and the other levels of education (elementary, high school, graduate, and teacher training) have one study each. Furthermore, although it has been applied, it is not possible to determine the level of education at which the study S15 was applied.

### ***CONTRIBUTIONS AND DIFFICULTIES REPORTED IN USING PEDAGOGICAL TECHNIQUES FOR ACTIVE LEARNING (RQ4)***

Interpreting the various contributions and difficulties in the use of pedagogical techniques for active learning, in complex studies, with a broad relevance in their findings, is no easy task. The records that follow comprise an attempt to systematize these contributions and difficulties made explicit by the authors regarding the theme of this review.

Table 4 systematizes the contributions reported by the studies that form the basis of this review on the use of active learning pedagogical techniques for the teaching/learning of programming.

**Table 4. Contributions of the use of active learning pedagogical techniques for the teaching/learning of programming**

CONTRIBUTIONS	STUDIES	QUANTITY
Greater acceptance or positive feedback from students, increasing their satisfaction or motivation	S02; S03; S07; S08; S10; S12; S13; S14; S16; S17; S20; S21; S22; S23; S24; S26; S27; S28; S29; S30; S32; S33; S34; S35	24
Improving students' learning experience	S02; S03; S07; S08; S10; S12; S13; S14; S17; S21; S23; S24; S25; S27; S29; S30; S31; S32; S33; S34; S35	21
Improving students' learning outcomes or performance	S03; S04; S09; S11; S12; S13; S14; S17; S18; S19; S25; S31; S33; S34; S37; S38	16
Stimulating students' interest, involvement, or engagement	S02; S04; S07; S12; S22; S23; S24; S25; S26; S28; S30; S31; S34	13
Stimulating the development of soft skills or abilities related to collaborative work and communication	S04; S10; S12; S14; S20; S23; S30; S31	8
Enabling greater flexibility during the teaching/learning process, making students learn at their own pace	S03; S04; S16; S22; S30; S35	6
Increasing students' confidence or self-confidence	S02; S10; S23; S29; S30	5
Optimizing or expanding the classroom timeline	S03; S16; S22; S30	4
Increasing approval rates	S14; S25; S32	3
Helping to bridge the gap between weak and strong students	S22; S31	2

As shown in Table 4, over 60% of the studies report as being a contribution of the use of active learning pedagogical techniques in the context of teaching/learning of programming, greater acceptance or positive feedback from students, increasing their satisfaction or motivation.

Moreover, it is also worth mentioning the following reported contributions: it improves the students' learning experience (n=21); improves learning outcomes or student performance (n=16); stimulates students' interest, involvement, or engagement (n=13); and encourages the development of soft skills or abilities related to collaborative work and communication (n=8). The contributions observed in this study find support in Andres (2017), which shows that active teaching is a positive predictor of the course score and learning motivation of students.

Regarding satisfaction, Freeman et al. (2014), through empirically validated results in regular classrooms, evidences that active learning is students' preferred teaching practice. In this sense, O'Flaherty and Phillips (2015) indicate that there is much indirect evidence of improved academic performance and student and staff satisfaction with the flipped approach.

In general, the contributions mentioned and presented in most of the studies that make up this review are in line with the literature, being evidenced by several studies such as those by Prince (2004), Michael (2006) Cavanagh (2011), Duffany (2015), in addition to those already mentioned above.

With regard to the difficulties reported in the studies that form the basis of this review on the use of active learning pedagogical techniques for the teaching/learning of programming, the results are presented in Table 5.

**Table 5. Difficulties in using active learning pedagogical techniques for the teaching/learning of programming**

DIFFICULTIES	STUDIES	QUANTITY
Greater effort/work is required from the teacher to plan and/or execute the teaching/learning process	S02; S03; S06; S12; S20; S27; S30; S34; S36	9
Student resistance or mistaken study methods	S01; S05	2
Greater effort and involvement of students is required during the teaching/learning process	S19	1

As shown in Table 5, the main difficulty in the use of active learning pedagogical techniques for teaching/learning programming, is that it requires a greater effort/work of the teacher to plan and/or execute the teaching/learning process. This difficulty was mentioned by nine studies that make up the basis of this review. This main difficulty, observed in studies, is in line with the findings of O'Flaherty and Phillips (2015) in a review of the use of flipped classrooms in higher education, where these authors show what is known about the time, cost and staffing required in flipping a class.

## IMPLICATIONS

In this paper, a systematic literature review was conducted in order to identify and characterize the research reflecting the different pedagogical techniques of active learning in the context of teaching/learning of computer programming.

Through the analysis of the data from the studies that form the basis of this review, it was found that the use of active learning pedagogical techniques allows for greater acceptance or positive feedback from students, increasing their satisfaction or motivation to enhance the learning experience, learning outcomes, or student performance during the process of teaching/learning of programming.

Overall, these observed contributions find support in the results obtained by Freeman et al. (2014), who conducted a meta-analysis of 225 studies published before 2010 that reported data on exam scores or failure rates when comparing undergraduate student performance in STEM under traditional lectures versus active learning. Freeman et al. conclude that average exam scores improved by about 6% in the active learning sections and that students in classes with traditional lectures were 1.5 times more likely to fail than students in classes with active learning.

Motivation is a crucial factor for the success of any learning. However, the development of higher cognitive skills and abilities such as those related to programming, which requires the mastery of multiple domains, may not occur unless students are adequately motivated.

In this sense, Gomes and Mendes (2014) evidence that demotivation is common in many novice-programming students who are not able to cope with the natural difficulties associated with programming learning. Duffany (2015) points out that many students do not seem to be highly motivated to learn the contents of introductory programming. Some students just want to pass the course having as their main motivation obtaining a grade and, as a result, many have a poor performance in the course.

Another finding concerns the use of the active learning methodology, Flipped classroom, to the teaching of programming. This methodology has been gaining popularity in recent years. Giannakos et al. (2014) provide a summary of six main advantages with regard to the use of the flipped classroom teaching approach: (i) increases learning performance; (ii) positive attitudes; (iii) increases engagement; (iv) more discussions; (v) enforces cooperative learning; and (vi) better learning habits. In general, the benefits pointed out by Giannakos et al. (2014), in the context of using the flipped classroom, somewhat match those reported in the context of active learning in this study.

Thus, based on the findings of this research, we recommend that teachers consider restructuring their traditional computer programming teaching practices by using active learning pedagogical techniques to elicit the highlighted learning contributions from their students.

Finally, it is worth noting that this review shares the most common limitations of the systematic method, such as research coverage and possible biases introduced during the selection, data extraction, and analysis of studies. Furthermore, the fact that some of the research questions listed require answers that are not binary or objective is a limitation with regard to data analysis. However, it should be noted that these limitations have been addressed following the general recommendations for systematic reviews.

## CONCLUSIONS

---

The main purpose of this study was to identify research studies that relate the different pedagogical techniques by which active learning is developed in the context of teaching/learning of computer programming, in order to characterize the approaches, the pedagogical techniques used, the application, and the contributions and implementation difficulties reported by these studies. To this end, a systematic review of the literature was conducted between the years 2014 and 2019. By running the systematic review protocol, 335 studies were retrieved from Web of Science, SCOPUS, ScienceDirect, and ACM Digital Library.

The studies' selection was based on a set of criteria established to guide the selection process, including alignment with the research questions and assessment of the quality of the studies. Of the 335 studies retrieved initially, 297 were discarded, either because they were duplicates based on the inclusion and exclusion criteria, or because they did not meet a minimum quality score. Thus, 38 studies were selected to form the basis of this review.

From the analysis of the selected studies, it was observed that the approaches that relate active learning to the context of teaching/learning of programming (RQ1) are pedagogical intervention/experiment, or the development of a tool, instrument, or methodology.

With regard to the active learning pedagogical techniques (methodologies, strategies, approaches) applied in the context of teaching programming (RQ2), it was found that the techniques employed are diverse. However, the Flipped classroom methodology has obtained a notable prominence in research, as well as the Project-based learning and Peer Instruction strategies, and their application is mainly concentrated on university/college education (RQ3).

In relation to the contributions in the use of active learning pedagogical techniques reported by the studies in the context of teaching/learning of programming (RQ4), it is noted that, in general, the use of these techniques enables greater acceptance or positive feedback from students, increasing their satisfaction or motivation in order to improve the learning experience, learning outcomes, or student performance. It also stimulates students' interest, involvement, or engagement, as well as the development of soft skills or abilities related to collaborative work and communication. However, it should be highlighted that the contributions observed for the teaching/learning process of computer programming derive from investigations mainly concentrated in the university context, aiming to observe if these contributions can be reproduced in other education levels.

The studies report as the main difficulty to implementing the pedagogical techniques of active learning in the context of teaching/learning of programming is that these require a greater effort/work by the teacher to plan and/or execute the teaching/learning process. However, even if implementing the active learning pedagogical techniques requires a greater effort/work from the teacher to plan and/or execute the teaching/learning process, the findings of this research concerning the contributions arising from the use of these pedagogical techniques indicate their use can contribute significantly in the teaching/learning process, showing to be a viable alternative and consistent with the reduction of the failures in the learning of programming.

Finally, we consider that this study's objective to identify and characterize the studies that relate the different active learning pedagogical techniques in the context of teaching/learning of computer programming has been achieved. Thus, this study adds to previous systematic reviews of the literature, specifically addressing the panorama of studies that relate active learning to the context of teaching/learning of programming. It is hoped that the findings of this article can support other research that addresses the topic, enabling its development and deepening through the developed basis from which active learning researchers can work.

In future studies, we suggest that researchers consider investigating the benefits of using different pedagogical techniques for active learning and the costs related to the higher cognitive load imposed by these computer programming learning techniques. It should be indicated, even more generally, that the action of investigating how the difficulties inherent to teachers, related to the teaching/learning process of programming, may imply difficulties concerning students and content, especially with regard to traditional teaching practices developed by them.

## REFERENCES

---

- Andres, H. P. (2017). Active teaching to manage course difficulty and learning motivation. *Journal of Further and Higher Education*, 43(2), 1-16. <https://doi.org/10.1080/0309877X.2017.1357073>
- Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, 39(2), 32-36. <https://doi.org/10.1145/1272848.1272879>
- Bergmann, J., & Sams, A. (2012). *Flip your classroom: Reach every student in every class every day*. International Society for Technology in Education. [https://www.rcboe.org/cms/lib/GA01903614/Centricity/Domain/15451/Flip\\_Your\\_Classroom.pdf](https://www.rcboe.org/cms/lib/GA01903614/Centricity/Domain/15451/Flip_Your_Classroom.pdf)
- Bonwell, C., & Eison, J. (1991). *Active learning: Creating excitement in the classroom*. ASHE-ERIC Higher Education Reports. The George Washington University.
- Borges, R. P., Oliveira, P. R. F., da R. Lima, R. G., & de Lima, R. W. (2018). A systematic review of literature on methodologies, practices, and tools for programming teaching. *IEEE Latin America Transactions*, 16(5), 1468-1475. <https://doi.org/10.1109/TLA.2018.8408443>
- Bosse, Y., & Gerosa, M. A. (2016). Why is programming so difficult to learn? Patterns of difficulties related to programming language. *ACM SIGSOFT Software Engineering Notes*, 41(6), 1-6. <https://doi.org/10.1145/3011286.3011301>
- Cavanagh, M. (2011). Students' experiences of active engagement through cooperative learning activities in lectures. *Active Learning in Higher Education*, 12(1), 23-33. <https://doi.org/10.1177/1469787410387724>
- Duffany, J. (2015, July). Active learning applied to introductory programming. *Proceedings of the 13th Latin American and Caribbean Conference for Engineering and Technology: Engineering Education Facing the Grand Challenges What Are We Doing?* Santo Domingo, Dominican Republic. <https://doi.org/10.18687/LACCEI2015.1.1.246>
- Dybå, T., & Dingsøy, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10), 833-859. <https://doi.org/10.1016/j.infsof.2008.01.006>
- Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410-8415. <https://doi.org/10.1073/pnas.1319030111>

- Giannakos, M. N., Krogstie, J., & Chrisochoides, N. (2014, November). Reviewing the flipped classroom research: Reflections for computer science education. *Proceedings of the 2014 Computer Science Education Research Conference*, 23-29. <https://doi.org/10.1145/2691352.2691354>
- Gomes, A., & Mendes, A. J. (2007, September). Learning to program-difficulties and solutions. *International Conference on Engineering Education, Coimbra, Portugal*. [https://www.researchgate.net/publication/228328491\\_Learning\\_to\\_program\\_-\\_difficulties\\_and\\_solutions](https://www.researchgate.net/publication/228328491_Learning_to_program_-_difficulties_and_solutions)
- Gomes, A., & Mendes, A. (2014, October). A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. *Proceedings of the IEEE Frontiers in Education Conference*, 1-8. Berlin, Germany. <https://doi.org/10.1109/FIE.2014.7044086>
- Kitchenham, B. A. (2012, September). Systematic review in software engineering: Where we are and where we should be going. *Proceedings of the 2nd international Workshop on Evidential Assessment of Software Technologies*. Lund, Sweden. Association for Computing Machinery. <https://doi.org/10.1145/2372233.2372235>
- Koulouri, T., Lauria, S., & Macredie, R. D. (2015). Teaching introductory programming: A quantitative evaluation of different approaches. *ACM Transactions on Computing Education*, 14(4), 1-28. <https://doi.org/10.1145/2662412>
- Lage, M. J., Platt, G. J., & Treglia, M. (2000). Inverting the classroom: A gateway to creating an inclusive learning environment. *The Journal of Economic Education*, 31(1), 30-43. <https://doi.org/10.1080/00220480009596759>
- Luxton-Reilly, A., Sheard, J., Szabo, C., Simon, Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Ott, L., Paterson, J., & Scott, M. J. (2018, July). Introductory programming: A systematic literature review. *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 55-106. Larnaca, Cyprus. <https://doi.org/10.1145/3293881.3295779>
- Martins, V. F., Concilio, I. de A. S., & Guimarães, M. de P. (2018). Problem based learning associated to the development of games for programming teaching. *Computer Applications in Engineering Education*, 26(5), 1577-1589. <https://doi.org/10.1002/cae.21968>
- Mazur, E. (1997). *Peer instruction: A user's manual*. Prentice Hall. <https://doi.org/10.1063/1.881735>
- Mazur, E., & Somers, M. D. (1999). Peer instruction: A user's manual. *American Journal of Physics*, 67(4), 359-360. <https://doi.org/10.1119/1.19265>
- McGettrick, A., Boyle, R., Ibbett, R., Lloyd, J., Lovegrove, G., & Mander, K. (2005). Grand challenges in computing: Education – A summary. *The Computer Journal*, 48(1), 42-48. <https://doi.org/10.1093/comjnl/bxh064>
- Medeiros, R. P., Ramalho, G. L., & Falcao, T. P. (2019). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, 62(2), 77-90. <https://doi.org/10.1109/TE.2018.2864133>
- Michael, J. (2006). Where's the evidence that active learning works? *Advances in Physiology Education*, 30(4), 159-167. <https://doi.org/10.1152/advan.00053.2006>
- Noordin, K., Nasir, A. N. M., Ali, D. F., & Nordin, M. S. (2011). Problem-Based Learning (PBL) and Project-Based Learning (PjBL) in engineering education: A comparison. *Proceedings of the IETEC'11 Conference*, Kuala Lumpur, Malaysia.
- O'Flaherty, J., & Phillips, C. (2015). The use of flipped classrooms in higher education: A scoping review. *Internet and Higher Education*, 25, 85-95. <https://doi.org/10.1016/j.iheduc.2015.02.002>
- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., & Paterson, J. (2007). A survey of literature on the teaching of introductory programming. *SIGCSE Bulletin*, 39(4), 204-223. <https://doi.org/10.1145/1345443.1345441>
- Peng, J., Wang, M., & Sampson, D. (2017, July). Scaffolding project-based learning of computer programming in an online learning environment. *IEEE 17th International Conference on Advanced Learning Technologies*, 315-319. <https://doi.org/10.1109/ICALT.2017.17>

- Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education*, 93(3), 223-231. <https://doi.org/10.1002/j.2168-9830.2004.tb00809.x>
- Qian, Y., & Lehman, J. (2017). Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education*, 18(1), 1-24. <https://doi.org/10.1145/3077618>
- Richardson, D. (2008). Don't dump the didactic lecture; Fix it. *Advances in Physiology Education*, 32(1), 23-24. <https://doi.org/10.1152/advan.00048.2007>
- Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137-172. <https://doi.org/10.1076/csed.13.2.137.14200>
- Simon, Luxton-Reilly, A., Ajanovski, V. V., Fouh, E., Gonsalvez, C., Leinonen, J., Parkinson, J., Poole, M., & Thota, N. (2019, December). Pass rates in introductory programming and in other STEM disciplines. *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*, 53-71. Aberdeen, Scotland. <https://doi.org/10.1145/3344429.3372502>
- Sleeman, D. (1986). The challenges of teaching computer programming. *Communications of the ACM*, 29(9), 840-841. <https://doi.org/10.1145/6592.214913>
- Vihavainen, A., Airaksinen, J., & Watson, C. (2014, July). A systematic review of approaches for teaching introductory programming and their influence on success. *Proceedings of the Tenth Annual Conference on International Computing Education Research*, 19-26. Glasgow, Scotland. <https://doi.org/10.1145/2632320.2632349>
- Wang, H.-Y., Huang, I., & Hwang, G.-J. (2016). Comparison of the effects of project-based computer programming activities between mathematics-gifted students and average students. *Journal of Computers in Education*, 3(1), 33-45. <https://doi.org/10.1007/s40692-015-0047-9>
- Watson, C., & Li, F. W. B. (2014, June). Failure rates in introductory programming revisited. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, Uppsala, Sweden, 39-44. <https://doi.org/10.1145/2591708.2591749>

## APPENDIX A – SELECTED STUDIES

---

- S01:** Baldwin, D. (2015, February). Can we “flip” non-major programming courses yet? *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 563-568. <https://doi.org/10.1145/2676723.2677271>
- S02:** Caceffo, R., Gama, G., & Azevedo, R. (2018, January). Exploring active learning approaches to computer science classes. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland*, 922-927. <https://doi.org/10.1145/3159450.3159585>
- S03:** Chen, D., & Faichney, J. (2019). Flipping a programming class to improve student performance and student satisfaction. *International Journal of Adult Vocational Education and Technology*, 10(1), 27-39. <https://doi.org/10.4018/IJAVET.2019010103>
- S04:** Clark, R. M., Besterfield-sacre, M., Budny, D., Bursic, K. M., Clark, W. W., Norman, B. A., Parker, R. S., Ii, J. F. P., & Slaughter, W. S. (2016). Flipping engineering courses. *Advances in Engineering Education*, 5(3), 1-39.
- S05:** Dazo, S. L., Stepanek, N. R., Fulkerson, R., & Dorn, B. (2016). An empirical analysis of video viewing behaviors in flipped CS1 courses. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 106-111. <https://doi.org/10.1145/2899415.2899468>
- S06:** Estévez-Ayres, I., Alario-Hoyos, C., Pérez-Sanagustín, M., Pardo, A., Crespo-García, R. M., Leony, D., Parada G., H. A., & Delgado-Kloos, C. (2015). A methodology for improving active learning engineering courses with a large number of students and teachers through feedback gathering and iterative refinement. *International Journal of Technology and Design Education*, 25(3), 387-408. <https://doi.org/10.1007/s10798-014-9288-6>
- S07:** Giacaman, N., & De Ruvo, G. (2018). Bridging theory and practice in programming lectures with active classroom programmer. *IEEE Transactions on Education*, 61(3), 177-186. <https://doi.org/10.1109/TE.2018.2819969>



- S08:** Giannakoulas, A., & Xinogalos, S. (2018). A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students. *Education and Information Technologies*, 23(5), 2029-2052. <https://doi.org/10.1007/s10639-018-9702-x>
- S09:** Greer, T., Hao, Q., Jing, M., & Barnes, B. (2019). On the effects of active learning environments in computing education. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 267-272. <https://doi.org/10.1145/3287324.3287345>
- S10:** Ham, T., Cyrus Rezvanifar, S., Thomas, V. S., & Amini, R. (2018). Using hands-on physical computing projects to teach computer programming to Biomedical Engineering students. *Journal of Biomechanical Engineering*, 140(8), 1-5. <https://doi.org/10.1115/1.4040226>
- S11:** Hao, Q., Barnes, B., Wright, E., & Kim, E. (2018, January). Effects of active learning environments and instructional methods in Computer Science education. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland*, 934-939. <https://doi.org/10.1145/3159450.3159451>
- S12:** Herrera, R. F. (2017). Collaborative Project-Based Learning of Environments Programming From Civil Engineering Projects. *Revista Electrónica Educare*, 21(2), 1. <https://doi.org/10.15359/ree.21-2.10>
- S13:** Hijon-Neira, R. B., Velázquez-iturbide, Á., Pizarro-Romero, C., & Carriço, L. (2014, June). Game programming for improving learning experience. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education, Uppsala, Sweden*, 225-230. <https://doi.org/10.1145/2591708.2591737>
- S14:** Kaila, E., Kurvinen, E., Lökkilä, E., & Laakso, M.-J. (2016). Redesigning an object-oriented programming course. *ACM Transactions on Computing Education*, 16(4), 1-21. <https://doi.org/10.1145/2906362>
- S15:** Kasinathan, V., Mustapha, A., & Yee, H. C. C. (2019). Codecube: Active learning for STEM. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(1.3), 293-299. <https://doi.org/10.30534/ijatcse/2019/5581.32019>
- S16:** Kay, R., MacDonald, T., & DiGiuseppe, M. (2019). A comparison of lecture-based, active, and flipped classroom teaching approaches in higher education. *Journal of Computing in Higher Education*, 31(3), 449-471. <https://doi.org/10.1007/s12528-018-9197-x>
- S17:** Kothiyal, A., Murthy, S., & Iyer, S. (2014, June). Think-pair-share in a large CS1 class. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education, Uppsala, Sweden*, 51-56. <https://doi.org/10.1145/2591708.2591739>
- S18:** Lacher, L. L., Jiang, A., Zhang, Y., & Lewis, M. C. (2018, January). Including Coding Questions in Video Quizzes for a Flipped CS1. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland*, 574-579. <https://doi.org/10.1145/3159450.3159504>
- S19:** Latulipe, C., Rorrer, A., & Long, B. (2018, January). Longitudinal Data on Flipped Class Effects on Performance in CS1 and Retention after CS1. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland*, 411-416. <https://doi.org/10.1145/3159450.3159518>
- S20:** Love, B., Hodge, A., Corritore, C., & Ernst, D. C. (2015). Inquiry-based learning and the flipped classroom model. *PRIMUS*, 25(8), 745-762. <https://doi.org/10.1080/10511970.2015.1046005>
- S21:** Maher, M. Lou, Latulipe, C., Lipford, H., & Rorrer, A. (2015). Flipped classroom strategies for CS education. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 218-223). ACM Press. <https://doi.org/10.1145/2676723.2677252>
- S22:** Mok, H. N. (2014). Teaching tip: The flipped classroom. *Journal of Information Systems Education*, 25(1), 7-11.
- S23:** Moore, D. (2014). Supporting students in music technology higher education to learn computer programming. *Journal of Music, Technology and Education*, 7(1), 75-92. [https://doi.org/10.1386/jmte.7.1.75\\_1](https://doi.org/10.1386/jmte.7.1.75_1)
- S24:** Namratha, M., Rekha, G. S., Akram, S., Selva Kumar, S., & Nayak, J. S. (2018). Active learning approach for python programming. *Journal of Engineering Education Transformations*, 32(1), 15-19.
- S25:** Olelewe, C. J., Agomuo, E. E., & Obichukwu, P. U. (2019). Effects of B-learning and F2F on college students' engagement and retention in QBASIC programming. *Education and Information Technologies*, 24(5), 2701-2726. <https://doi.org/10.1007/s10639-019-09882-7>

- S26:** Park, J., Park, Y. H., Kim, J., Cha, J., Kim, S., & Oh, A. (2018). Elicast. *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, 1-10. <https://doi.org/10.1145/3231644.3231657>
- S27:** Phelps, G., & Liu, Y. (2018). The study and investigation of a new form of active-learning for CS: Peer-teaching with videos. *Journal of Computing Sciences in Colleges*, 33(4), 45-51.
- S28:** Pokorny, K. L. (2015). Creating a computer simulator as a CS1 student project. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 42-47). ACM Press. <https://doi.org/10.1145/2676723.2677210>
- S29:** Sáez López, J. M., & Cózar Gutiérrez, R. (2017). Visual programming with blocks in Primary Education: Learning and creating content in Social Sciences. *Revista Complutense de Educación*, 28(2), 409-426. [https://doi.org/10.5209/rev\\_RCED.2017.v28.n2.49381](https://doi.org/10.5209/rev_RCED.2017.v28.n2.49381)
- S30:** Sarawagi, N. (2014). A flipped CS0 classroom: Applying Bloom’s taxonomy to algorithmic thinking. *Journal of Computing Sciences in Colleges*, 29(6), 21-28.
- S31:** Schwarzenberg, P., Navon, J., Nussbaum, M., Pérez-Sanagustín, M., & Caballero, D. (2018). Learning experience assessment of flipped courses. *Journal of Computing in Higher Education*, 30(2), 237-258. <https://doi.org/10.1007/s12528-017-9159-8>
- S32:** Secules, S., & Lawson, W. (2019). Description and mixed methods evaluation of a novel hardware-based introductory programming course. *Advances in Engineering Education*, 7(3), 1-30.
- S33:** Shannon, A., & Summet, V. (2015). Live coding in introductory computer science courses. *Journal of Computing Sciences in Colleges*, 31(2), 158-164.
- S34:** Sharp, J. H., & Sharp, L. A. (2017). A comparison of student academic performance with traditional, online, and flipped instructional approaches in a C# programming course. *Journal of Information Technology Education: Innovations in Practice*, 16(1), 215-231. <https://doi.org/10.28945/3795>
- S35:** Wang, Y., Huang, X., Schunn, C. D., Zou, Y., & Ai, W. (2019). Redesigning flipped classrooms: A learning model and its effects on student perceptions. *Higher Education*, 78(4), 711-728. <https://doi.org/10.1007/s10734-019-00366-8>
- S36:** Yadav, A., Kussmaul, C., Mayfield, C., & Hu, H. H. (2019, February). POGIL in computer science: Faculty motivation and challenges *Proceedings of the 50th ACM Technical Symposium on Computer Science Education, Minneapolis, MN*, 280-285. <https://doi.org/10.1145/3287324.3287360>
- S37:** Zingaro, D., & Porter, L. (2014a). Peer instruction in computing: The value of instructor intervention. *Computers & Education*, 71, 87-96. <https://doi.org/10.1016/j.compedu.2013.09.015>
- S38:** Zingaro, D., & Porter, L. (2014b). Peer instruction: A link to the exam. *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education, Uppsala, Sweden*, 255-260. <https://doi.org/10.1145/2591708.2591711>

## APPENDIX B – QUALITY ASSESSMENT

ID STUDIES	QA1	QA2	QA3	QA4	QA5	QA6	QA7	QAT
S01	1.0	1.0	1.0	1.0	1.0	1.0	1.0	7.0
S02	1.0	1.0	1.0	1.0	1.0	1.0	1.0	7.0
S03	1.0	1.0	1.0	1.0	1.0	1.0	1.0	7.0
S04	1.0	1.0	1.0	1.0	1.0	1.0	1.0	7.0
S05	1.0	1.0	1.0	1.0	1.0	1.0	1.0	7.0
S06	1.0	1.0	1.0	1.0	1.0	1.0	0.5	6.5
S07	1.0	1.0	1.0	1.0	1.0	1.0	1.0	7.0

ID STUDIES	QA1	QA2	QA3	QA4	QA5	QA6	QA7	QAT
<b>S08</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S09</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S10</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S11</b>	1.0	1.0	1.0	1.0	1.0	1.0	0.5	<b>6.5</b>
<b>S12</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S13</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S14</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S15</b>	1.0	1.0	0.5	0.0	0.5	0.5	0.5	<b>4.0</b>
<b>S16</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S17</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S18</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S19</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S20</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S21</b>	1.0	1.0	1.0	1.0	1.0	1.0	0.5	<b>6.5</b>
<b>S22</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S23</b>	1.0	1.0	1.0	1.0	1.0	1.0	0.5	<b>6.5</b>
<b>S24</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S25</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S26</b>	1.0	1.0	1.0	1.0	1.0	0.5	1.0	<b>6.5</b>
<b>S27</b>	0.5	1.0	0.5	0.5	0.5	1.0	0.5	<b>4.5</b>
<b>S28</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S29</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S30</b>	1.0	1.0	0.5	0.0	0.5	1.0	1.0	<b>5.0</b>
<b>S31</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S32</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S33</b>	1.0	1.0	1.0	1.0	0.5	1.0	1.0	<b>6.5</b>
<b>S34</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S35</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S36</b>	1.0	1.0	1.0	1.0	1.0	0.0	0.5	<b>5.5</b>
<b>S37</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>
<b>S38</b>	1.0	1.0	1.0	1.0	1.0	1.0	1.0	<b>7.0</b>

## AUTHORS

---



**João Henrique Berssanette** is a Professor of Basic, Technical and Technological Education at the Instituto Federal do Paraná (IFPR), Campus Telêmaco Borba, Brazil. Master in Science and Technology Teaching from the Universidade Tecnológica Federal do Paraná (2016). He is currently a PhD student in Science and Technology Teaching from the Universidade Tecnológica Federal do Paraná, Brazil. His research focuses on evaluating the contributions of a pedagogical approach based on the use of the association of Active Learning Methodologies and cognitive load theory for the teaching of Computer Programming.



**Antonio Carlos de Francisco** is a Professor of Basic, Technical and Technological Education of the Universidade Tecnológica Federal do Paraná (UTFPR), Brazil, and full professor in the Postgraduate Program in Production Engineering (Master and Doctorate) at UTFPR, Campus Ponta Grossa. He has a Master in Technology from the Universidade Tecnológica Federal do Paraná (1999) and PhD in Production Engineering from the Universidade Federal de Santa Catarina (2003).