v. 4, n. 1, 2024

Journal of
Engineering Research

RUBRICAS FOR LEARNING COLLABORATIVE EVALUATION IN INTRODUCTORY PROGRAMMING - A CASE STUDY WITH PROTOTYPING

Dirson Santos de Campos
Universidade Federal de Goiás (UFG)
Goiânia-GO
http://lattes.cnpq.br/1356268761766006

Deller James Ferreira
Universidade Federal de Goiás (UFG)
http://lattes.cnpq.br/1646629818203057

Abstract: Rubrics are used to evaluate a wide variety of aspects in the pedagogical process. Rubrics are important for measuring aspects of collaborative learning. In this research, aspects of this type of learning will be explored, namely, co-regulation and group regulation of learning involving cognition, behavior, motivation and emotions, in situations of temporary coordination of regulation with colleagues or teachers. Therefore, the objective of this research is to uncover the extent to which introductory programming students apply co-regulation and shared regulation strategies during programming. An exploratory study involving 198 students found evidence that mapping a rubric can help more accurately measure results. A prototype was developed using the Web framework and the MTV design pattern to implement the exploratory study rubric.

Keywords: rubric, assessment, programming learning, co-regulation of learning, shared regulation

INTRODUCTION

The learning process in computing education is complex. It is necessary for students to develop different skills such as high cognitive capacity for abstraction, problem solving and logical thinking (Calderon et. al. 2021).

The rubric is a pedagogical tool that helps standardize aspects that are difficult to measure in learning, as it requires its creators to explain pedagogical criteria in a fair and standardized way. In complex pedagogical activities to measure, such as correlation and shared regulation of learning, an explicit rubric, a priori and not posteriori. The well-defined and outlined assessment criteria and their desirable performance levels can serve as a guide to the learning process and as a parameter in the development, review and judgment of this learning process (Reddy and Andrade, 2010).

Education and technology have caused changes in the relationships between teachers and students, this connection producing new experiences and skills. Consequently, new technologies and tools help to build and monitor this relationship (Aguiar et. al. 2021). In this work, a prototype was made, the Rubric Learning Assessment System, developed by us, to evaluate the exploratory study.

The rubric proposed in this research deals with regulation in collaborative group learning. It involves two aspects that are co-regulation and shared regulation. Coregulation refers to the dynamic metacognitive processes through which a student helps regulate cognition, behavior, motivation, and emotions, providing support in a transitional and flexible way (Hadwin et al., 2018). Shared regulation is understood as the social regulation of learning, in which students temporarily regulate their cognition, behavior, motivation and emotions in situations of temporary coordination of regulation with peers or teachers (Hadwin et al., 2018).

The main contribution of this research is to seek evidence about student learning strategies involving correlation and shared regulation carried out in an exploratory study and how to map it to a rubric. The exploratory study involved 198 students which found evidence that mapping a rubric can help to more accurately measure the results of learning processes.

The motivation is that the job market requires having the skills necessary for teamwork, which requires the ability to cooperate and solve problems in a group. Although students acquire notable theoretical and practical programming knowledge, they lack skills such as those related to soft skills. Collaborative learning brings some advantages over individual learning, mainly with the possibility of exchanging ideas and clarifying doubts due to the interaction between students in a collective and social scenario.

RELATED WORK

Related works represent research in which the rubric was used in teaching pedagogical activities involving co-regulation and shared regulation tasks and which types of rubrics were effectively used. The taxonomy of rubrics adopted in the articles cited in this session is not always explicit, in this case, the classification was made by us, based on the essential characteristics of the rubrics described in academic literature.

The adaptive rubrics are modified throughout the evaluation process, according to the need to measure teaching activities and pedagogical objectives (Reddy and Andrade, 2010).

Cook et al. (2020) found evidence of the reasons why a certain group pedagogical task is related to learning through collaboration with other students, the inference was made through feedback given, in pairs, through the dimensions of the rubric.

Assessment rubrics, as the name suggests, are used to simply evaluate an exercise, task or project. They must define assessment criteria, quality criteria and a clear scoring strategy for the items in the rubric (Reddya and Andrade 2010). Customized assessment rubrics by defining and collecting metadata that help extract evidence of student learning. Metadata refers to certain items of the evaluation rubric whose objective is to adapt relevant information about the decision limits used in teaching evaluation to classify student activities. Assessment rubrics were used to measure the co-regulation of tasks in groups (Aivaloglou and Meulen 2021), including in broader tasks such as academic projects.

Multiple rubrics appear in research that uses more than one type of rubric where each of them is used to measure different aspects of teaching or learning in the classroom context, for example, to review the artifacts of a subject classified internationally as CS1 (introductory programming courses) to measure evidence of group learning (Schmidt 2020).

Pedagogical rubrics are holistic in relation to the pedagogical objectives of the assessed task, measuring the didactic activity predominantly as a whole, analyzing the set of pedagogical activities (Ramírez et al 2018).

Training rubrics focus on aspects of the student's training process that go beyond the acquisition of intrinsic knowledge of a discipline, one must also continue with the development of student critical thinking, in order to generate new learning that integrates it with ethics and responsibilities of everyday life, in massive student courses this type of rubric is very useful, especially involving aspects of shared learning regulation (Kulkarni et al. 2013).

Standardizable rubrics are a subdivision of evaluative rubrics by direct measurable metrics with the aim of supporting the data analysis process in populations whose samples are in a sampling distribution of a specific statistic, for example, the performance of students in massive programming courses can be objectively scored using a rubric standardized for this purpose (Lerís et al. 2017).

RESEARCH METHODS

PROCEDURES

The first step was to create a more appropriate rubric model for the task of measuring collective learning, namely, correlation and shared regulation of learning. The choice is based on the literature in section 2. In this research, measurement involves characteristics of more than one type of rubric, thus, a new type of rubric was developed with characteristics of a pedagogical type of rubric (Ramírez et al 2018), formative (Kulkarni et

Journal of Engineering Research ISSN 2764-1317 DOI: https://doi.org/10.22533/at.ed.1317412409017

al 2013) and standardizable (Lerís et al 2017) whose criteria, scales and weights were defined by us, and implemented as prototyping.

The methodology used to implement the rubric to measure the collaborative learning strategies and skills of students in introductory programming subjects was to use prototyping.

The prototyping methodology for developing interactive systems aims to clarify ideas to help designers produce to achieve the best solutions for a given research need (Obayes and Hamzah, 2022). The main objective of prototyping is to develop and analyze the information flow and functionalities necessary to collect information from students and analyze these results in a statistically consistent way.

Data collection was done through questionnaires at the end of the course. Questions about the correlation and shared regulation of learning form the criteria (dimensions) of the rubric (Figure 1). The criteria (dimensions) of the rubric clearly explain the criteria for exploratory measurement of this aspect of learning, in order to infer, analyzing in a systematic and standardized way the students' responses that can be divided into components of the learning strategy and which of these components are the most important by attributing weights to each of these.

The questionnaire itself was created using the Google Forms tool. It was applied to 198 students from ``Universidade Federal de Goiás`` (UFG), anonymously, after completing several introductory programming courses, and was used for data collection and subsequent analysis.

PARTICIPANTS

198 undergraduate students from the computer science, computer engineering, medical physics, physical engineering, statistics and electrical engineering courses at UFG who were in face-to-face classes, post-COVID-19 pandemic, responded to the questionnaire.

INSTRUMENTS FOR DATA ANALYSIS

The first step was to statistically verify the reliability of the questionnaire, we applied the famous Cronbach's alpha statistical test. It is possible to objectively measure the reliability of an instrument such as a questionnaire by interpreting the result of the Cronbach's alpha test, which is the objective measure most used to verify reliability in these cases (Tavakol and Dennick, 2011). In this research, the aforementioned test was used to verify consistency between the various criteria (dimensions) of the rubric assessment (Figure 1), where each line of the rubric represents one of the 16 questions about co-regulation and shared regulation, so that this set of questions can be tested to see if they are consistent with each other.

The questionnaire was designed with the aim of collecting a set of data mapped to the rubric criteria. The objective is to capture students' perception about the use of correlation strategies and shared regulations. The questionnaire was developed to measure group regulation in introductory programming courses. The questions were based on the Adaptive Instrument for Emotion Regulation (AIRE) (Järvenoja et al., 2013) adapted to the context of introductory programming subjects by ourselves.

Rubrics need to describe levels of performance or competencies, in this research, for each question the level of performance and weights are presented in the columns of the rubric (Figure 1) which are described in the Likert scale (Likert, 1932) of 5 factors which is a scale that has been widely used in academic research for decades.

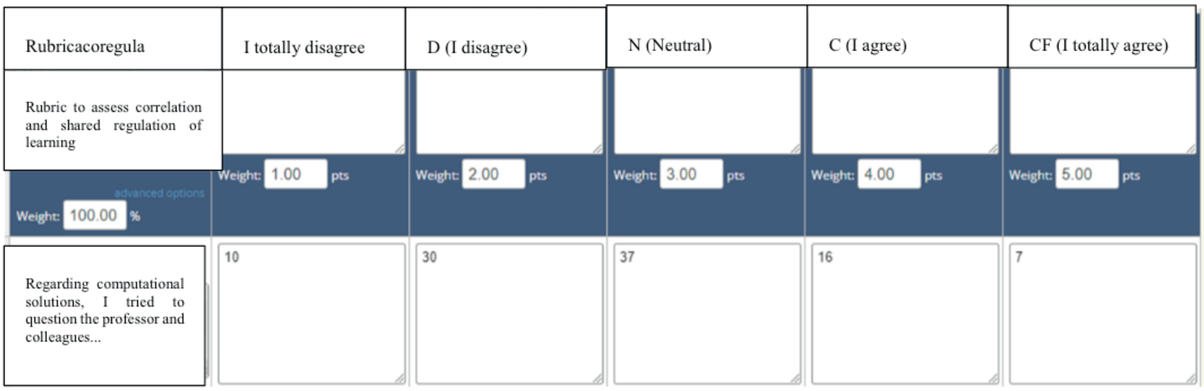

Figure 1. Rubric Elements for Co-Regulation and Shared Regulation

Source: Prepared by the authors (2023)

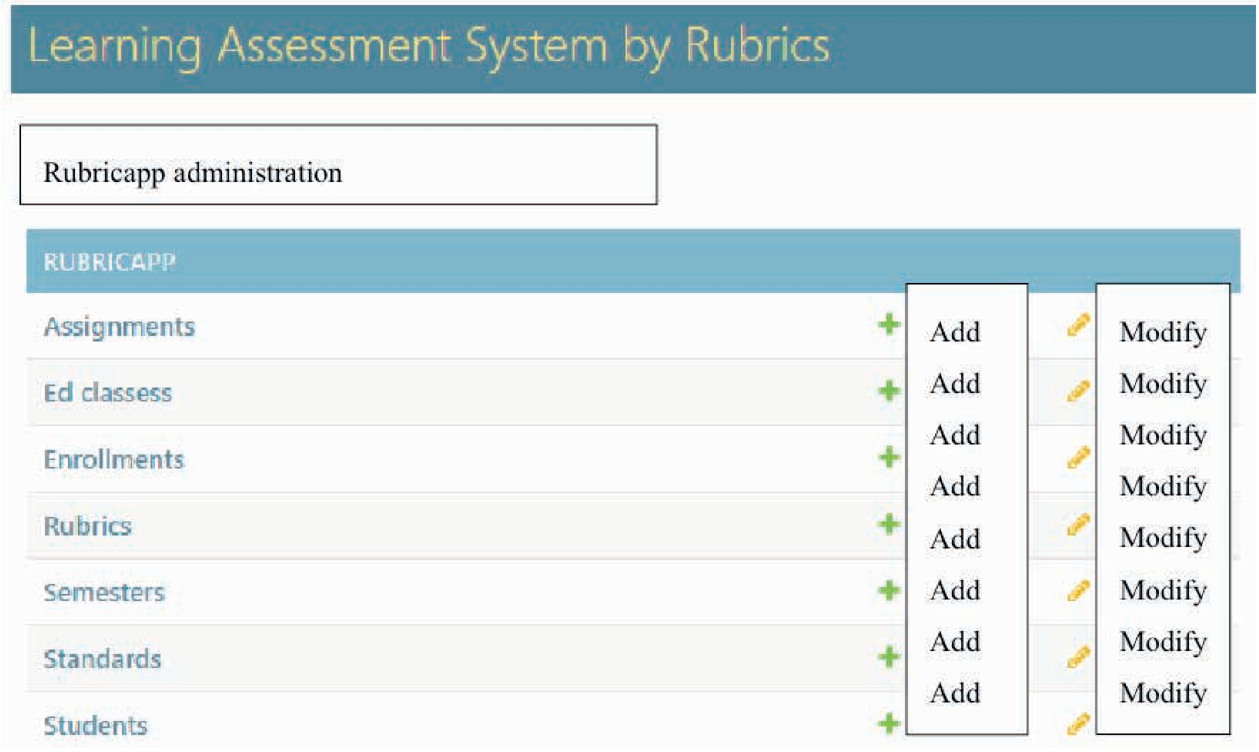

Figure 2. Main Features of the Rubric Learning Assessment System prototype

Source: Prepared by the authors (2023)

DEFINITION OF THE RUBRIC

DEFINITION OF RUBRIC CRITERIA (DIMENSIONS)

Based on the research methodology (section 3), the mainly pedagogical rubric was created in the iRubric tool as can be seen in Figure 1.

Figure 1, in its lines, shows criteria (dimensions) for the evaluation of the rubric for the first question (QP1). For reasons of size and visibility of Figure 1, the remaining 15 criteria that represent the questions self-assessed by students are described as follows: QP2. Have you used social media and other forms of technology to communicate with classmates? QP3. In group projects, did I try to motivate colleagues so that everyone contributed to building the programs? Qp4. Did I contribute to a good work climate during joint programming, facing difficulties with good humor? QP5. Did I value parts of my colleagues' code and contribute to improvements? QP6. Did I treat my colleagues with respect and use positive phrases like "Very good! Keep it up! Thank you! You've helped us a lot now!"? QP7. Have I tried to reconcile your goals, priorities and learning style with those of my colleagues? QP8. Was the group's work organized jointly, seeking to reconcile members' preferences? QP9. Have any time management strategies been used for group projects, such as Kanban or Scrum? QP10. Was any tool used to manage collaborative programming, such as Trello or GitHub? QP 11. Did the group use the "divide and conquer" strategy, thinking about each part of the program in different modules? QP12. In group projects, was the commitment of all group members to compliance with standards and participation in programming activities monitored and appropriate measures taken? QP13. In group projects, were roles assigned to be played by students during the writing of the program, such as writer, consultant, editor and reviewer? QP14. Was any joint programming strategy used, such as Coding Dojo? QP15. In group programming projects, was there reflection on the quality of group interactions and performance, and were actions taken when necessary? QP16. Have group interactions positively influenced your personal performance?

DEFINING PERFORMANCE LEVELS OR RUBRIC COMPETENCIES

In this research, the levels of performance or competence in the skill of co-regulation and shared regulation of learning were collected in 5 skill levels using the Likert scale that represents the columns of Figure 1. The weights of all questions are 100%, each question has the same weight 6.25%, in total the 16 questions form 100% (6.25% x 16). An explanation was provided to the students, a priori, in a text that they read, before filling out the questionnaire, which was anonymous and online, so the degrees of domains were defined by the students' own perception. These performance levels in Figure 1 are the labels DF (Strongly Disagree), D (Disagree), N (Neutral), C(Agree), and CF (Strongly Agree).

CALCULATION OF EACH CELL THAT FORMS THE RUBRIC

The rubric cell stores the results of calculating the percentage of responses from all students participating in the survey, for each of the questions that involve student perception of the strategies and skills used by themselves in co-regulation and shared regulation of their own learning.

When calculating each cell of the rubric, a technique proposed by Tastle and Wierman (2007) was used, which allows identifying for each proposed statement, through a score, the direction of all students' responses towards agreement or disagreement. Thus, firstly, for each of the response alternatives (options), a

Journal of Engineering Research ISSN 2764-1317 DOI: https://doi.org/10.22533/at.ed.1317412409017

different weight (P) is determined (see Figure 1), being, respectively, for strongly disagree (DF), disagree (D), neutral (N), agree (C) and strongly agree (CF), the weight values are 1, 2, 3, 4 and 5 respectively.

Then, to identify the score for each question, the following formula given by Equation 1 is applied:

Score = ((nDF / ntotal) x 1)) + ((nD / ntotal) x 2)) + ((nN / ntotal) x 3)) + ((nC / ntotal) x 4)) + ((nCF / ntotal) x 5)).

Equation 2. Calculation of the Score for each question proposed to students

Equation 2 represents the final score for each of the questions (student responses) and the final result of this equation is the content stored in each of the rubric cells as shown in Figure 1.

Final Score = sum of the score for each of the five answer options (DF; D; N; C; CF), which is obtained by the percentage of responses (number of alternative responses divided by the total number of responses), multiplied by P corresponding (weight).

Equation 2. Calculation of the Final Score for each proposed question

To interpret the results found in the score, a statement is considered to have a "high" score when the value is greater than or equal to four, as indicates evidence of partial or total agreement, while a "low" score is considered to have a value less than four, represents disagreement with the proposed statement. The closer the score value is to five, the greater the tendency of participants to completely agree with the statement and, consequently, the closer the value is to one, the greater the probability that participants will completely disagree with the statement.

RUBRIC PROTOTYPING

The prototype of the Rubric Learning Assessment System was built using a web framework for development using the Python language called Django (Django, 2023a). The interface of the Application that manages rubric manipulation (RubricApp) can be seen in Figure 2.

Prototype development in the Django web Framework used the following languages:

- Python for developing programming logic.

- HTML/CSS/Javascript for Web interface that is seen in the browser.

- SQL using Object-Relational-Mapper (ORM) that maps Model definitions in Django code to the underlying database structure. In the case of this prototype, the Postregre SQL Database Management System (DBMS) (PostgreSQL, 2023).

The prototype was developed with the MVT (Model-Template-View) Architectural Design Pattern, which is the Design Pattern under which the Dijango Web framework was built. The MVT pattern is a variation of the MVC (Model-View-Controller) pattern used in other Web frameworks. The Model (model), View (view) and Controller (Control) architecture facilitates the exchange of information between the user interface and the DBMS, making responses faster and more dynamic.

The structure of the MVT architecture pattern has the following three parts:

- Model: the model will act as an interface for your data. You are responsible for maintaining the data. It is the logical data structure behind the entire application and is represented by a DBMS.

- View: the View is the user interface, it represents what is seen in the browser when rendering an application website.

- Template: A template consists of static parts of the desired HTML output, as well as some special syntax that describes how the dynamic content will be inserted.

DESCRIPTION OF THE MAIN FEATURES OF THE PROTOTYPE

The main module of the prototype is the RubricApp of the Learning Assessment System by Rubrics described in Figure 2 are described below:

- Task (Assignment). This functionality allows the user to associate a task with each class. The task must be linked to the task name and edclass name. The task can also have a keyrubric associated with it.

- Classes (Ed classes). This functionality that represents a single entity of a class. Class creation requires a unique identification number, a subject, a course number (maximum four characters), a section number and a teacher associated with the course.

- Enrollment. This functionality represents the enrollment of a specific student in a class and semester. If the rubric needs to be edited after submission, you will need to deactivate the completed rubric and resubmit it.

- Rubrics. This feature allows the teacher to create a rubric to be used for each course. The rubric template consists of lines where the user can edit the line name and add descriptive text for each line/column intersection. The user must enable the "Template" box to allow the rubric to be used in a course. The template box differentiates the blank rubric from the student's completed rubric (both based on the same template). Rubric lines cannot be edited after creation; however, lines can be added to rubrics after the rubric is created.

- Semester. This functionality represents a specific semester registration.

- Standards. A model that represents patterns associated with rubric lines. The user can choose more than one pattern per line or none at all.

- Students. This functionality represents an instance of a student. Student attributes include first name, last name, and student identification (enrollment) number.

In the prototype, it is possible to build different types of rubrics with criteria and dimensions contextualizing the teacher's assessment needs.

RESULTS

VALIDITY AND RELIABILITY OF RUBRIC CRITERIA (DIMENSIONS)

Reliability was tested with regard to the internal consistency of the questionnaire. Internal consistency assesses the reliability of summed scores derived from a Likert scale. Internal consistency refers to the degree to which there is compatibility and correlation between responses to the multiple items that make up this scale. Cronbach's alpha statistical test was applied to the questionnaire for each question that covers students' co-regulation and shared regulation, to find out whether the questions are cohesive. Table 1 shows the interpretation of Cronbach's alpha coefficient.

The Cronbach's Alpha coefficient values for the rubric criteria are described in Table 2. The rubric criteria represent the questions about learning for student self-assessment, which are a total of 16 questions described in section 4.1.

0.9 <= Alpha	Excellent
0.8<= Alpha < 0.9	Good
0.7<= Alpha < 0.8	Acceptable
0.6<= Alpha < 0.7	Questionable

Table 1. Interpretation of Cronbach's alpha coefficient (Tavakol and Dennick, 2011)

Journal of Engineering Research ISSN 2764-1317 DOI: https://doi.org/10.22533/at.ed.1317412409017

Coefficient	Co-regulation and Shared Regulation
Cronbach's alpha	0,881

Table 2. Cronbach's alpha coefficient for the criteria (dimensions) of the rubric (lines in Figure 1)

Therefore, according to Table 2, the value of Cronbach's alpha coefficient calculated, according to Cronbach's formula for the 16 questions about co-regulation and shared regulation among students, is 0.881. Thus, according to the interpretation of Table 1, it can be said that issues relating to co-regulation and shared regulation are correlated at a good level, indicating that they are internally consistent.

PROTOTYPE VERIFICATION, VALIDATION AND TESTING

Verification, Validation and Testing (VV&T) activities are fundamental to the development of high-quality software products. Automated software testing implies the use of specific tools and frameworks to implement and execute test cases (García et al., 2023).

The VV&T activities of the prototype were carried out by applying automated tests using test scripts developed in Python using the following tools.

- Django Unit Test Framework (DJANGO, 2023b). The Django Unit Test Framework is a part of the Django web framework (DJANGO, 2023a) that provides support for creating and running unit tests for Django applications. Unit tests are a way to verify that specific parts of code (usually functions, methods, or classes) are working as expected. They run automatically to ensure that code changes don't break existing functionality.

- Unittest tool (Unit Testing Framework) (Unitest, 2023). This tool contains the framework's core classes that form the basis of specific test cases and suites (TestCase, TestSuite, etc.), and also a text-based utility class for running the tests and reporting the results.

- Selenium Framework for Web Applications (Selenium, 2023). Selenium tool is the set of tools for automated software testing for web systems with support for writing and running tests. Selenium commands support tests related to various forms of interaction with a system, such as window size, mouse position, alerts, Ajax functionality, pop-up windows, event handling, and many other properties characteristic of a Web system.

Automated test scripts were developed in py format, the great advantage of which is saving time and minimizing tester errors when conducting the testing activity. Automated testing saves the tester time, as some tasks, generally repetitive, that he must do will be performed by a computer program. On the other hand, for this to happen, there is an initial investment, which also takes time. In other words, there is an investment in selecting and implementing a test environment (Maldonado et. al. 2018).

Automated test scripts were developed in py format, the great advantage of which is saving time and minimizing tester errors when conducting the testing activity. Automated testing saves the tester time, as some tasks, generally repetitive, that he must do will be performed by a computer program. On the other hand, for this to happen, there is an initial investment, which also takes time. In other words, there is an investment in selecting and implementing a test environment (Maldonado et. al. 2018).

The test scripts are not exhaustive and were developed in Python to test the MVT pattern of the Django framework, namely, models, the templates that are the forms interface, and the views.

STUDENT PERCEPTIONS OF THE USE OF REGULATORY STRATEGIES

Each cell in Figure 1 presents the percentage of responses to the co-regulation and shared regulation questions. Due to the size of the figure, the result of each cell will be placed for each answer in Table 3 where "QN" means "question number". In Table 3, each line corresponds to a co-regulation and shared regulation question from Figure 1. The scores were calculated according to equations 1 and 2, described in section 4.3.

Considering the questions described in section 4.1 and the data summarized in Table 3, which were mapped to one or more questions, we can infer some statistics about the learning strategies of correlation and shared regulation.

Regarding the emotional strategies of co-regulation and shared regulation, 34% of students are neutral, disagree or strongly disagree, which contribute to a good work climate during joint programming, facing difficulties with good humor. 33% of students are neutral, disagree or strongly disagree that group interactions positively influence their personal performance. 21% of students are neutral, disagree, or strongly disagree who use social media and other forms of technology to communicate with peers. Only 40% of students strongly agree and agree that they try to motivate their colleagues so that everyone contributes to the construction of programs in group projects. 21% of students are neutral, disagree or strongly disagree with the effect of expressing positive sentences.

Regarding behavioral co-regulation and shared regulation strategies, only 10% of students strongly agree and agree that they use joint programming strategies. Only 38% of students strongly agree and agree that they reflect on the quality of group interactions and performance and act when necessary during group projects.

Only 13% of students strongly agree or agree that they apply the time management strategy to group projects. Only 18% of students strongly agree or agree that they use tools to manage collaborative scheduling. These results reveal that a significant number of students do not apply co-regulation and shared regulation behavioral strategies during introductory programming. Regarding contextual strategies of co-regulation and shared regulation, 54% of students are neutral, disagree, or strongly disagree that group commitment agrees with group rules and monitors participation in programming activities and acts if necessary. 20% of students are neutral, disagree or strongly disagree with the group working together, trying to reconcile members' preferences.

QN	DF	D	N	C	CF
1	7	16	37	30	10
2	7	4	10	35	44
3	14	11	35	27	13
4	7	1	26	53	13
5	13	2	22	50	13
6	8	4	19	44	25
7	8	12	29	38	13
8	12	8	26	35	19
9	57	19	1	9	4
10	51	16	15	14	4
11	13	12	30	33	12
12	11	11	32	33	13
13	43	16	20	16	5
14	54	17	19	7	3
15	21	11	30	26	12
16	8	1	24	50	17

Table 3. Final percentage of responses for each cell in Figure 1

Only 10% of students strongly agree or agree that, in group projects, roles are assigned to be played by students during the writing of the

Journal of Engineering Research ISSN 2764-1317 DOI: https://doi.org/10.22533/at.ed.1317412409017

program, such as writer, consultant, editor and reviewer. These results show that a significant number of students do not use contextual co-regulation and shared regulation strategies when learning introductory programming.

Table 4 shows an overall score for the co-regulation and shared regulation questions, revealing that students perceive that they are even worse at using co-regulation and self-regulation strategies during introductory programming.

Scoring of all questions on co-regulation and shared regulation	3,08

Table 4. Global scores of student self-assessment of co-regulated learning and shared regulation

The results discussed in this section reveal that students have difficulties using emotional co-regulation and shared regulation strategies.

CONCLUSIONS

The present study highlights the importance of using rubrics to measure learning in a standardized and systematic way, defining objective criteria to evaluate student learning strategies. The 16 qualitative questions were defined by us based on the assumptions of the Adaptive Instrument for Emotion Regulation (AIRE) (Järvenoja et al., 2013) and the different types of rubrics discussed in section 2 using the research methods described in section 3.

Co-regulated and shared regulated learning, which is understood as the social regulation of group learning. The exploratory results in section 5 allow us to infer that traditional programming teaching and learning approaches do not prioritize skills aligned with co-regulation and shared regulation. Students trying to learn to program do not always receive explicit training or support to develop the regulatory skills necessary for group programming.

The main objective of the present study was to explore students' perspectives on the use of regulation strategies during introductory programming courses and how to map this perspective to pedagogical rubrics.

The exploratory study carried out found evidence that novices in programming use regulation strategies in a limited way, drawing attention to a demand for the development and application of teaching approaches to promote collaborative learning. In this work, co-regulation and regulation were analyzed shared in introductory subjects in programming courses. Understanding students' perspectives on the use of group activity strategies during programming is an important addition to studies in this area, as the results of this exploratory study can broaden the understanding of the group learning approach. The results of this work will help in the design of future teaching and learning approaches in collective or pair activities.

The use of the prototyping methodology allowed us to create models for rubrics that may have different criteria and dimensions from the contextualized rubric for evaluating aspects of collaborative learning described in section 4. At the same time, prototyping allowed us to create a rubric whose criteria are the 16 questions defined in section 4.1 and the dimensions are those defined in Section 4.2. Prototyping was important because it allowed the implementation of the mapping of criteria and dimensions of a contextualized rubric to measure aspects of collaborative learning for students in introductory programming subjects.

REFERENCES

AGUIAR, F., MELLO, R., FURTADO, A., NASCIMENTO, A., MIRANDA, P. **Learning Analytics e Problem-Based Learning: Mapeamento Sistemático da Literatura**. In Anais do VI Congresso sobre Tecnologias na Educação (Ctrl+E), pp. 110-119, 2021. Porto Alegre: SBC.

AIVALOGLOU, E. AND MEULEN, A. V. D. **An Empirical Study of Students' Perceptions on the Setup and Grading of Group Programming Assignments**, ACM Trans. Comput. Educ. 21, 3, Article 17 (September 2021), 22 pages. https://doi.org/10.1145/3440994.

CALDERON, I, SILVA, W. E FEITOSA, E. **Um Mapeamento Sistemático da Literatura sobre o uso de Metodologias Ativas durante o Ensino de Programação no Brasil**. X Congresso Brasileiro de Informática na Educação (CBIE 2021). Anais do XXXII Simpósio Brasileiro de Informática na Educação (SBIE 2021), pg. 1152-1161. DOI: 10.5753/sbie.2021.217564.

COOK, A., DOW, S. AND HAMMER, J. **Designing Interactive Scaffolds to Encourage Reflection on Peer Feedback**. In Proceedings of the 2020 ACM Designing Interactive Systems Conference (DIS '2020). Association for Computing Machinery, New York, NY, USA, 1143–1153, 2020. https://dl.acm.org/doi/10.1145/3357236.3395480.

DJANGO. **Django Documentation**. Disponível em: <https://docs.djangoproject.com/en/5.0/>. Acesso em: 30 novembro 2023.

__________. **Django Unit Test Framework: Testing in Django.** Disponível em: <https://docs.djangoproject.com/en/5.0/topics/testing/>. Acesso em: 30 novembro 2023.

HADWIN, A., JÄRVELÄ, S., & MILLER, M. **Self-regulation, co-regulation, and shared regulation in collaborative learning environments**. In D. Schunk, & J. Greene, (Eds.). Handbook of self-regulation of learning and performance (2nd ed.). New York, NY: Routledge, 2018.

GARCÍA, B., RICCA, F., ALAMO, J. M., LEOTTA, M. **Enhancing Web, Observability through Instrumented Automated Browsers**. Journal of Systems and Software", Volume 203, 2023, 111723, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2023.111723 .

KULKARNI, C., WEI, K. P., LE, H., CHIA, D., PAPADOPOULOS, K., CHENG, J., KOLLER, D. AND KLEMMER, S. R. **Peer and self-assessment in massive online classes**. ACM Trans. Comput.-Hum. Interact. 20, 6, Article 33 (December 2013), 31 pages. https://doi.org/10.1145/2505057.

JÄRVENOJA, H., VOLET, S., & JÄRVELÄ, S. **Regulation of emotions in socially challenging learning situations: An instrument to measure the adaptive and social nature of the regulation process**. Educational Psychology, 33(1), 31–58, 2013.

LIKERT, R. A. **Technique for the measurement of attitudes**. Archives in Psychology, 140, 1–55, 1932.

LERÍS, D., SEIN-ECHALUCE, M. L., HERNÁNDEZ, M., BUENO, C. **Validation of indicators for implementing an adaptive platform for MOOCs**. Computers in Human Behavior, Volume 72, 2017, Pages 783-795, ISSN 0747-5632, https://doi.org/10.1016/j.chb.2016.07.054.

MALDONADO, J., DELAMARO, M., VINCENZI, A.M.R X **Automatização de teste de software com ferramentas de software livre**. Editora Elsevier. ISSN 9788535288162, 2018.

OBAYES, K. A., HAMZAH, A. **Using of prototyping in develop an employee information management, Measurement: Sensors**. Volume 24, 2022, 100557, ISSN 2665-9174. https://doi.org/10.1016/j.measen.2022.100557.

PostgreSQL. **PotgreSQL Documentation**. Disponível em: <https://www.postgresql.org/docs/ >. Acesso em: 30 novembro 2023.

RAMÍREZ, R. JUÁREZ, C. X., NAVARRO, V., TAPIA, R. I., OLVERA, M. AND GARCÍA, C. G. **What is Programming? Putting all Together - A Set of Skills Required**. 6th International Conference in Software Engineering Research and Innovation (CONISOFT), 2018, pp. 11-20, doi: 10.1109/CONISOFT.2018.8645956.

REDDYA, M. Y. AND ANDRADE B., H. **A Review of Rubric Use in Higher Education**. Assessment & Evaluation in Higher Education, vol. 35, no. 4, pp. 435- 448, 2010. DOI: 10.1080/02602930902862859.

SCHMIDT, J. Y. **Reviewing CS1 Materials through a Collaborative Software Engineering Exercise: An Experience Report**. Proceedings of the 51st ACM Technical Symposium on Computer Science Education. Association for Computing Machinery, New York, NY, USA, 379–385, 2020. https://doi.org/10.1145/3328778.3366932.

SELENIUN. **Selenium Automates Browsers**. Disponível em: <https://www.selenium.dev/>. Acesso em: 30 novembro 2023.

TASTLE, W. J., & WIERMAN, M. J. **Consensus and dissention: a measure of ordinal dispersion**. International Journal of Approximate Reasoning, 45, 531–545, 2007. https://doi.org/10.1016/j.ijar.2006.06.024.

TAVAKOL M, DENNICK R. **MAKING SENSE OF CRONBACH'S ALPHA**. Int J Med Educ. 2011 Jun 27; 2:53-55. doi: 10.5116/ijme.4dfb.8dfd. PMID: 28029643; PMCID: PMC4205511.

UNITEST. **Unittest — Unit Testing Framework**. Disponível em: <https://docs.python.org/3/library/unittest.html>. Acesso em: 30 novembro 2023.