ITS 2019
Jamaica
3-7 JUNE

An Adaptive Approach to Provide Feedback for Students in Programming Problem Solving

Priscylla Silva: Federal Institute of Alagoas, Brazil
Evandro Costa: Federal University of Alagoas, Brazil
Joseana Araújo: Federal University of Campina Grande, Brazil

Agenda

1. Introduction
2. System with Adaptive Feedback
3. Approach to Provide Adaptive Feedback
4. Methodology of evaluation
5. Concluding Remarks and Future Perspectives

Introduction

- The skill of computer programming is one of the fundamental skills in computing courses [11].
- Many tools have been developed since 1960 with the goal of assisting the student [4];
- Many of this tools provide automatic feedback [9].

Introduction

- The most common type of feedback is provided only after the student completes a solution and submits it for evaluation [7].
- There are many students that can not create a complete solution to submit in the first try.
- Provide feedback during the process of developing the solution can help these students [6].

Introduction

- Providing feedback messages to the student to assist in coding is not a trivial task.
- The feedback message should also be consistent with the current state of the student's solution.
- The growing number of programming learning systems has produced new research opportunities and challenges.
- One of the challenges is how to provide adaptive and automatic feedback.

Introduction

- This work came from the idea of helping students with problems in the coding process.
- This research presents an approach to provide adaptive feedback during the resolution of programming exercises.
- The feedback messages contain helping them to fix mistakes and how to take the next steps to complete the solution.

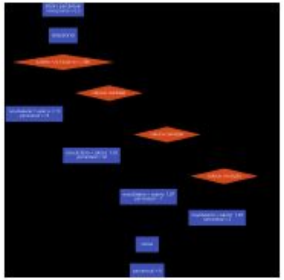
System with Adaptive Feedback

- We developed an ITS for students to practice programming with coding questions.
- The students can use the C language for programming.
- The adaptive feedback is based on the status of the student' solution

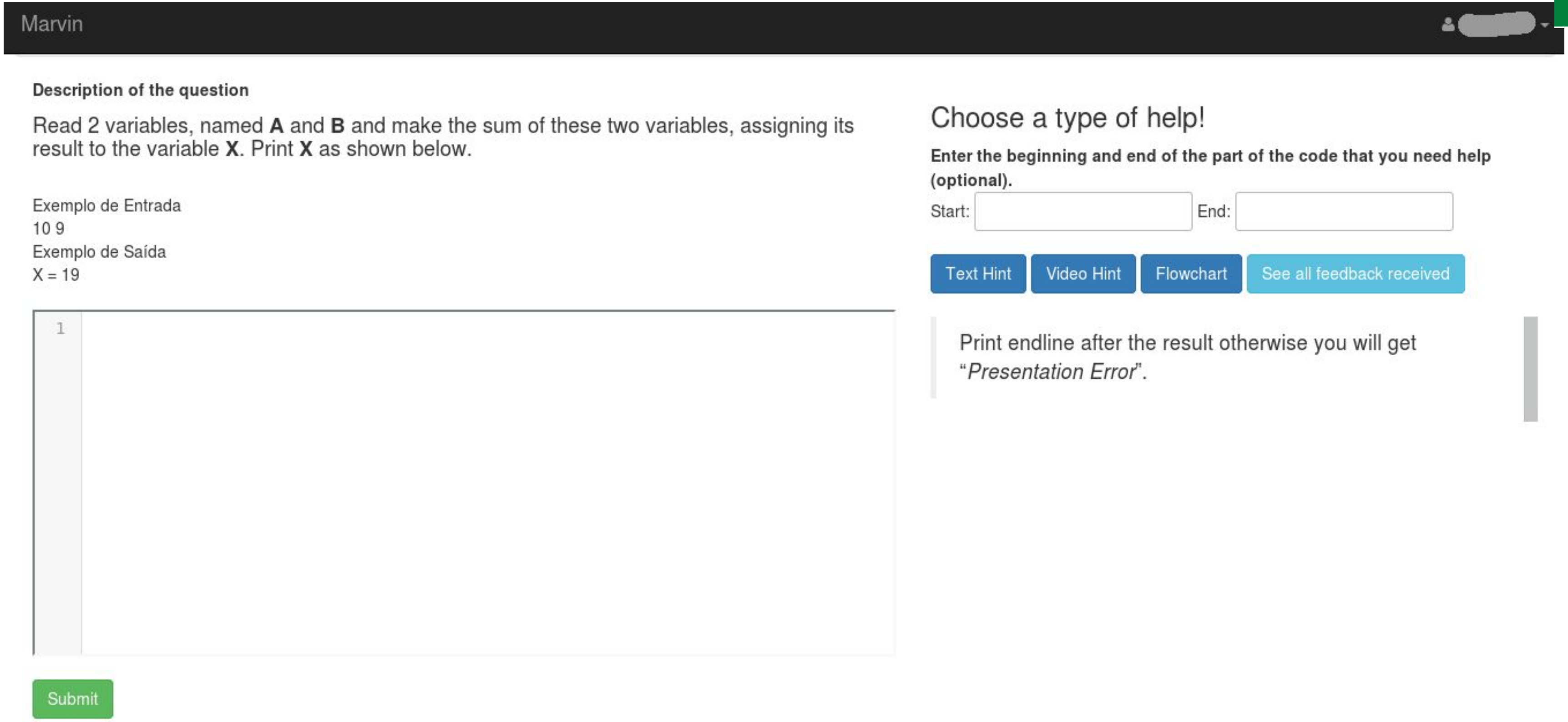
System with Adaptive Feedback

- When the student is solving a question, the system analyzes the code produced by the student.
- If the student request help, the system selects the feedback message based on the analysis of the code, even if the code is not complete or if the syntax is wrong.
- Our system presents three types of feedback: text, videos, and flowcharts.

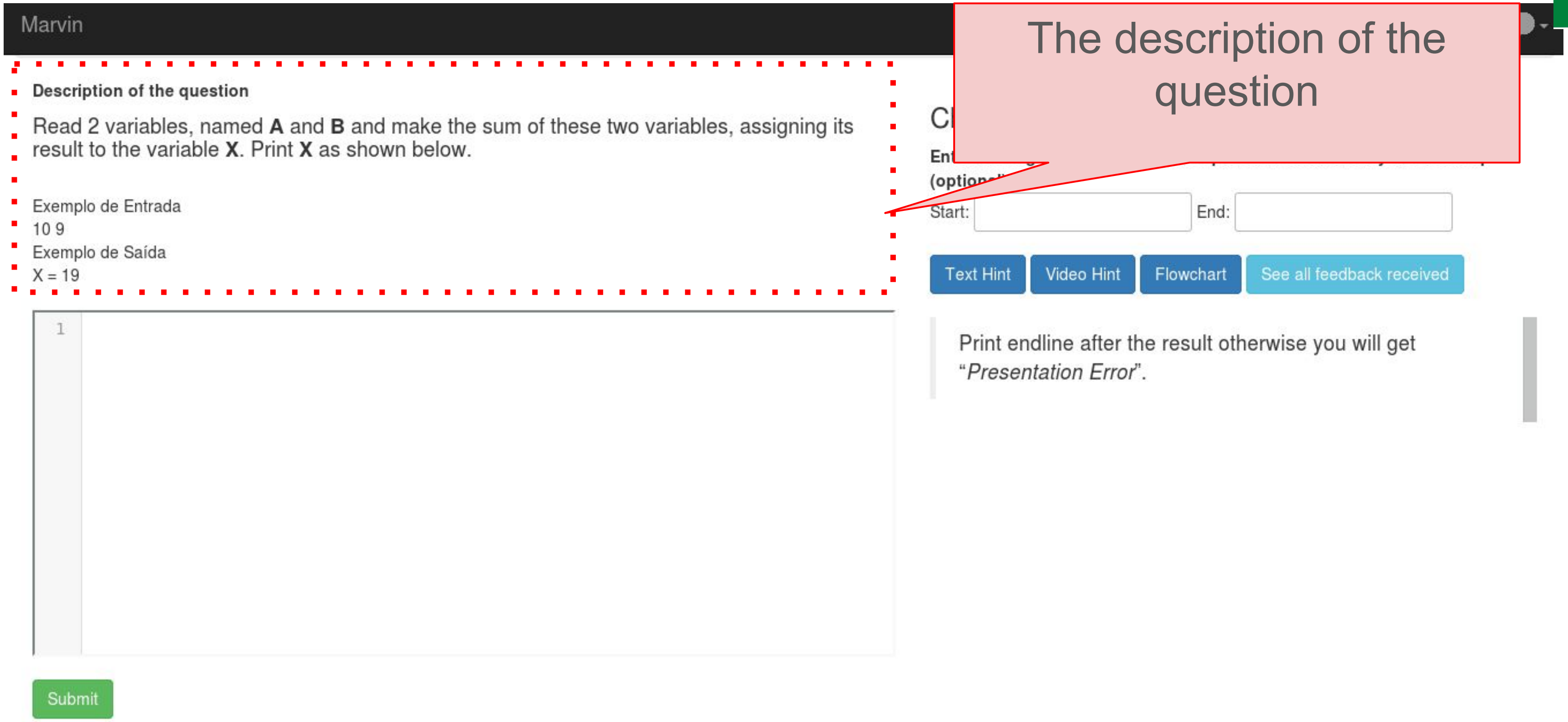
> Print endline after the result otherwise you will get "*Presentation Error*".



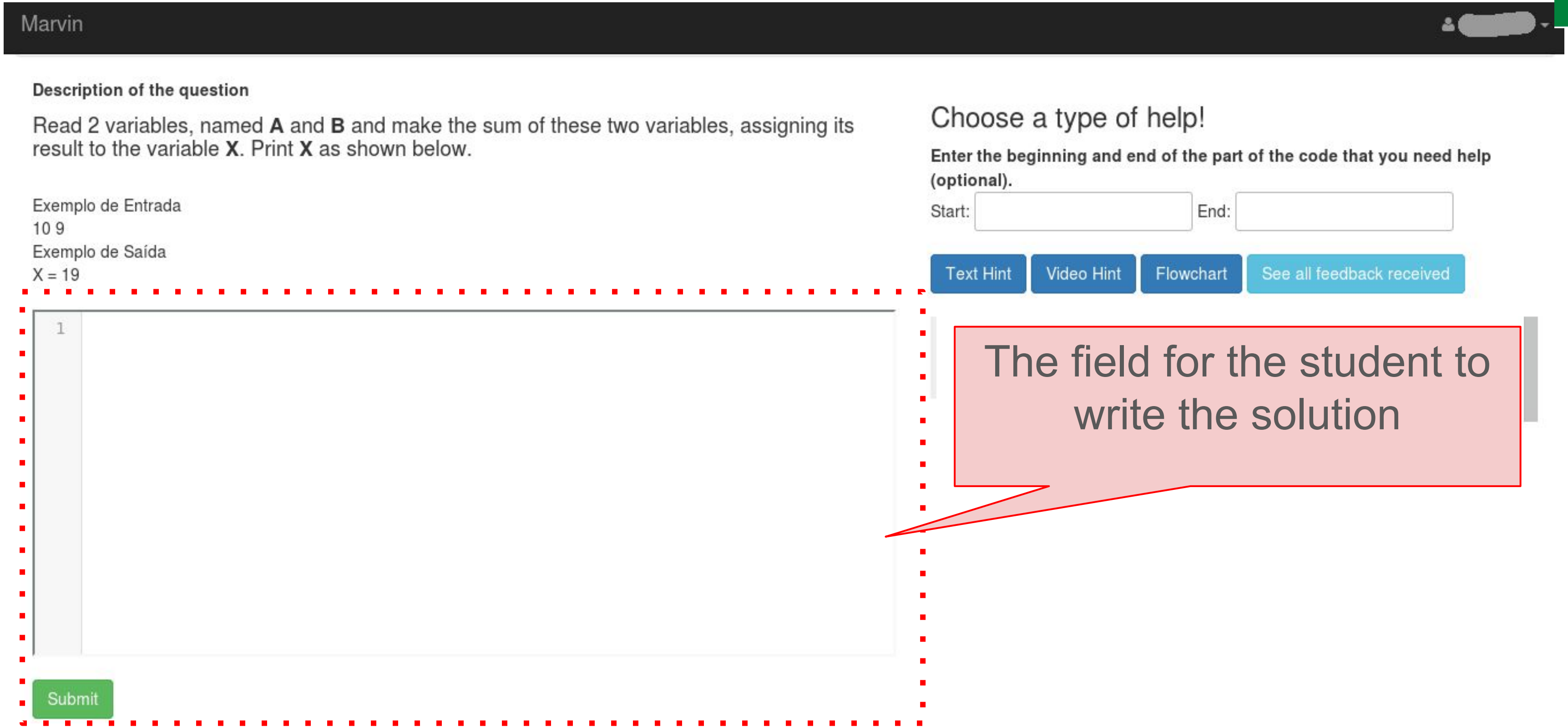
System with Adaptive Feedback



System with Adaptive Feedback

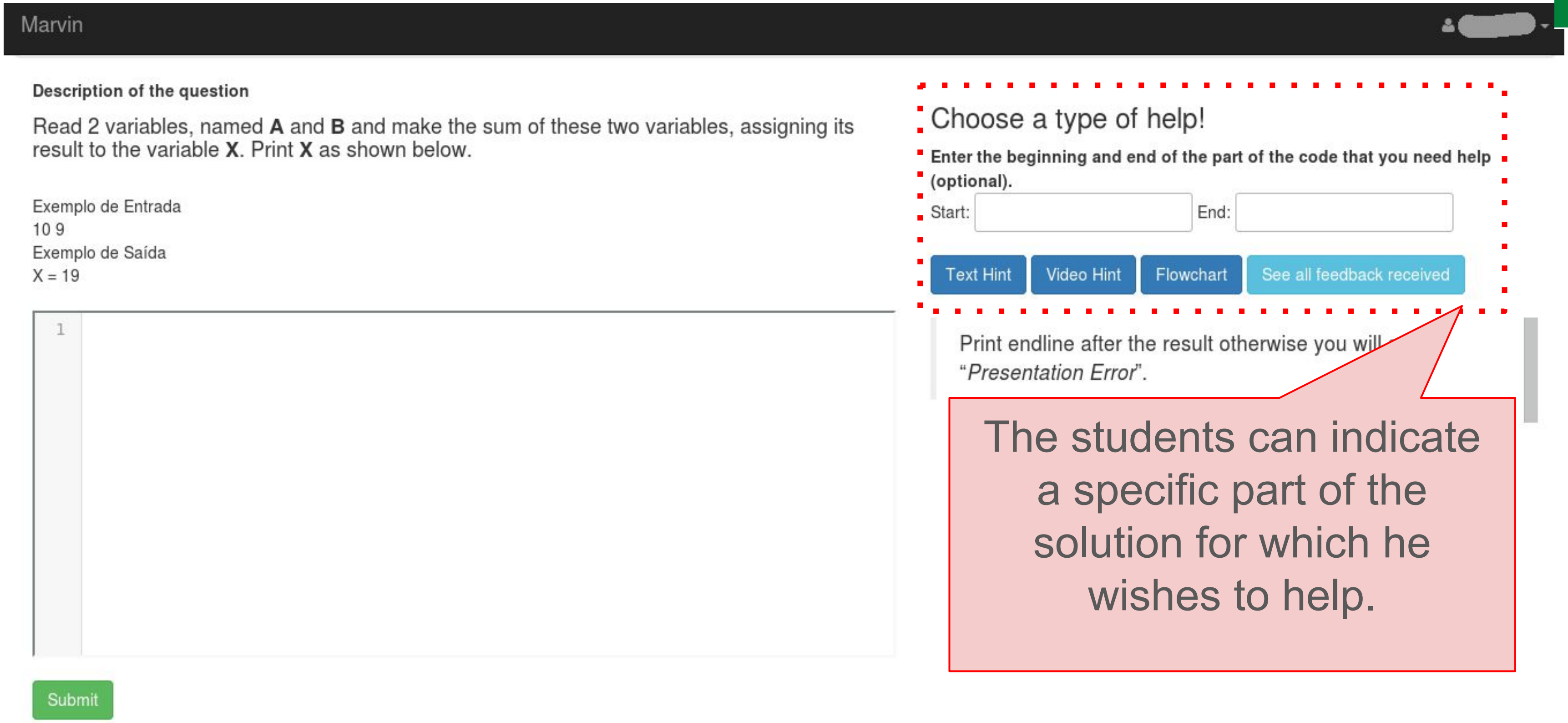


System with Adaptive Feedback



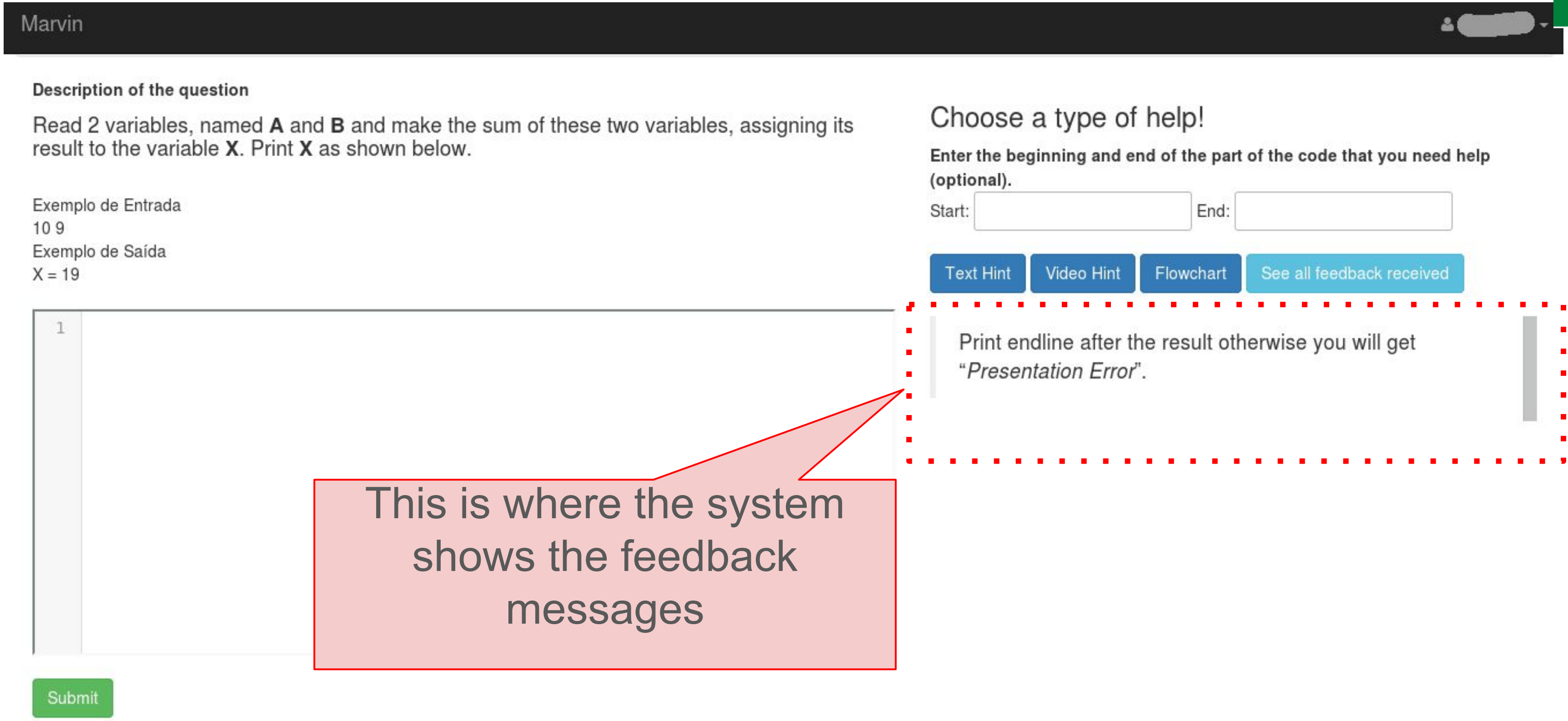
System with Adaptive Feedback

ITS 2019 Jamaica 3-7 JUNE



The students can indicate a specific part of the solution for which he wishes to help.

System with Adaptive Feedback



Approach to Provide Adaptive Feedback

- We used model solutions registered by the teacher in the system.
- The teacher can associate feedback messages to different parts of the solution or the whole solution.
- The teacher can create feedback messages in three formats: text, video, and flowchart.
- One question may have several model solutions associated with it.
- At the moment, the student request help, the system begins the process of selecting and presenting an adequate feedback message.

Approach to Provide Adaptive Feedback

- Initially, the student has told which part of the solution he/she needs help with it. If no part of the code has been specified, the system will use the current state of the student solution to search for general feedback.
- If the student has specified the part of the solution for which he needs help, then the system will look for specific feedback.

General Feedback	It is a message whose content has been created and associated with a complete model solution.
Specified Feedback	It is a message whose content has been created and associated with some part of the code of the model solution.

Approach to Provide Adaptive Feedback

Model Solution

```
1
```

Part of the code

Trecho do código

Feedback type

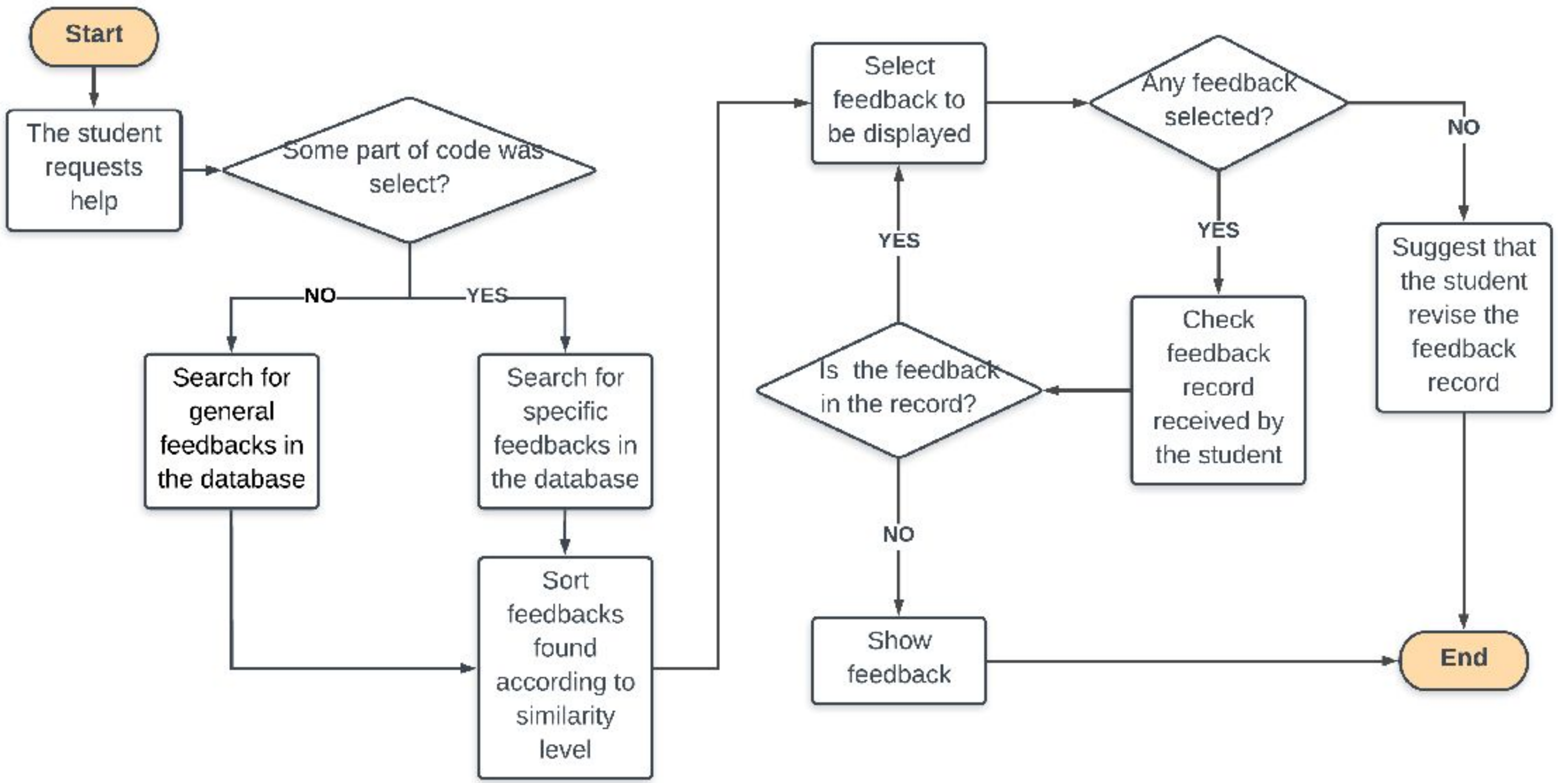
-- Escolha uma Opção --

Content

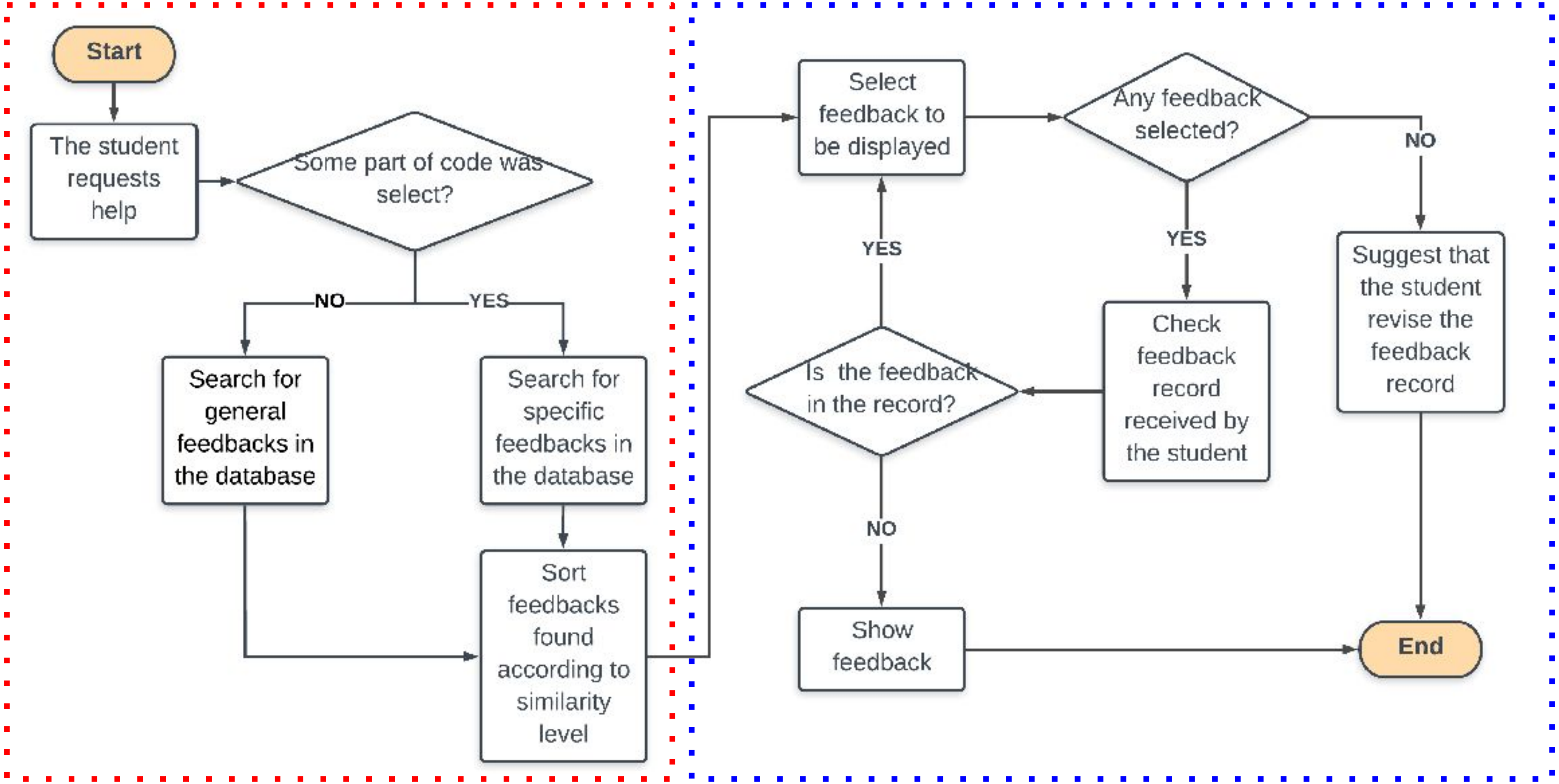
Paragraph B I

Save

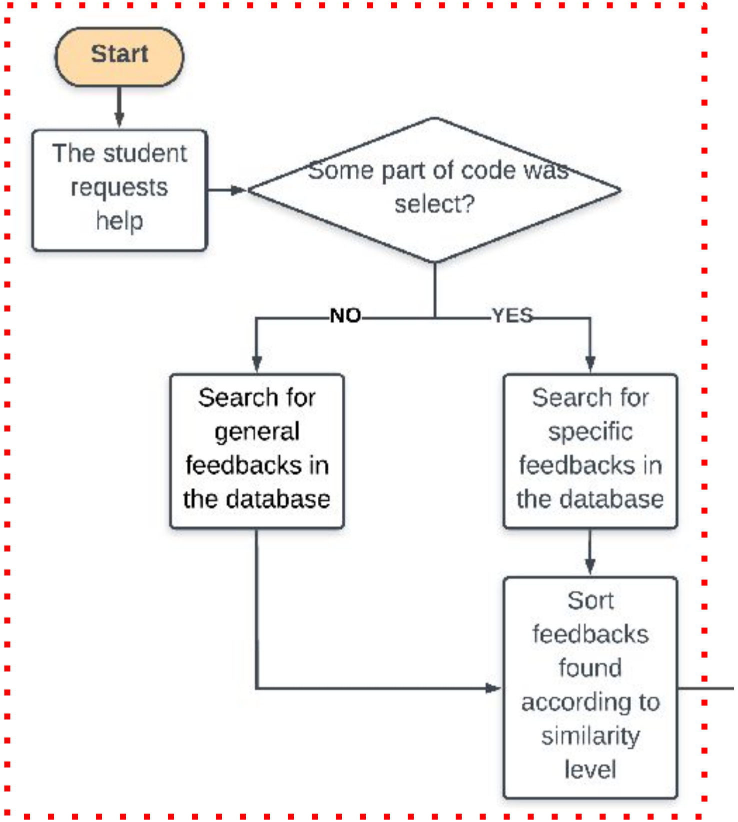
Approach to Provide Adaptive Feedback



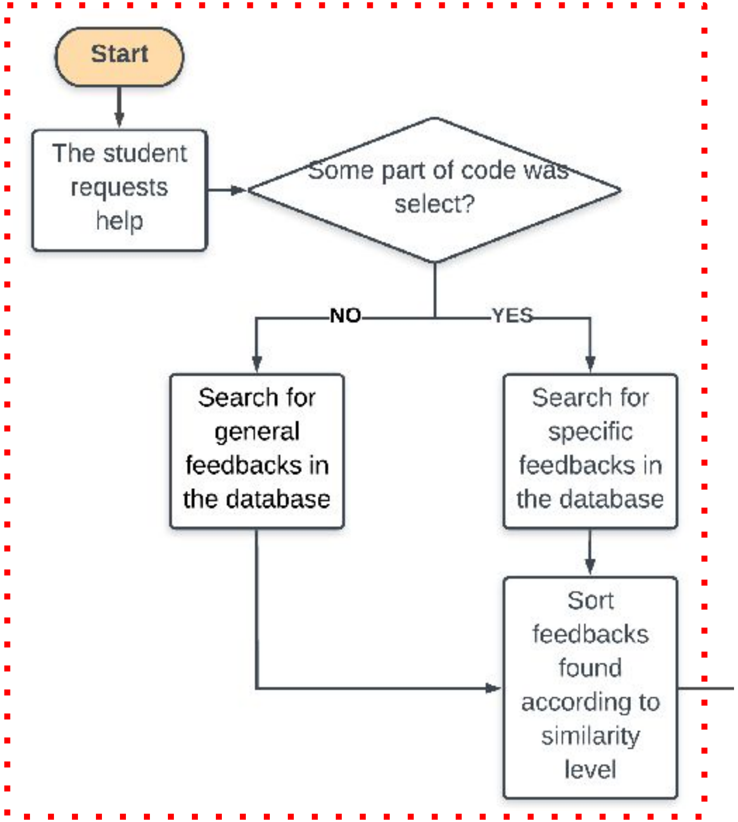
Approach to Provide Adaptive Feedback



Approach to Provide Adaptive Feedback



Approach to Provide Adaptive Feedback

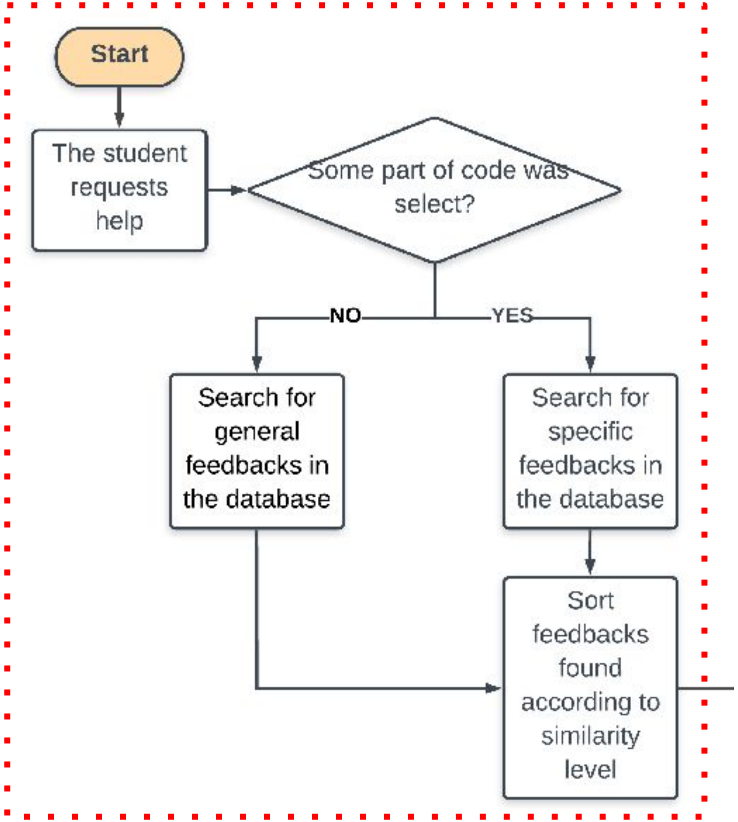


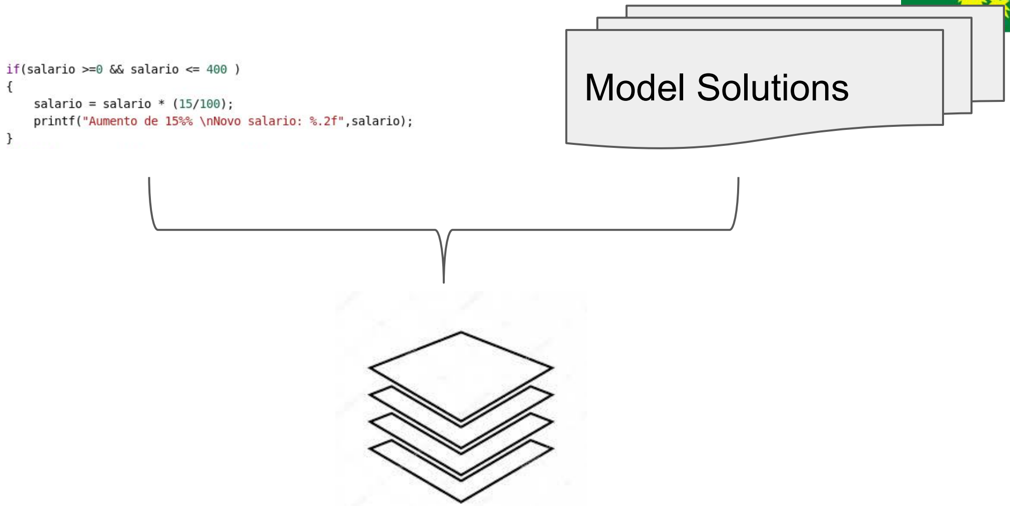

```
if(salario >=0 && salario <= 400 )
{
    salario = salario * (15/100);
    printf("Aumento de 15%% \nNovo salario: %.2f",salario);
}
```

The system performs a similarity calculation between the part of the student solution and the parts extracted from the model solution.

All feedback messages whose similarity calculation is different from zero are returned.

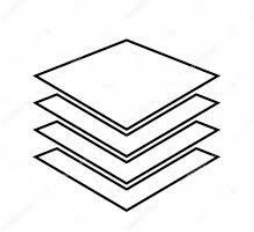
Approach to Provide Adaptive Feedback

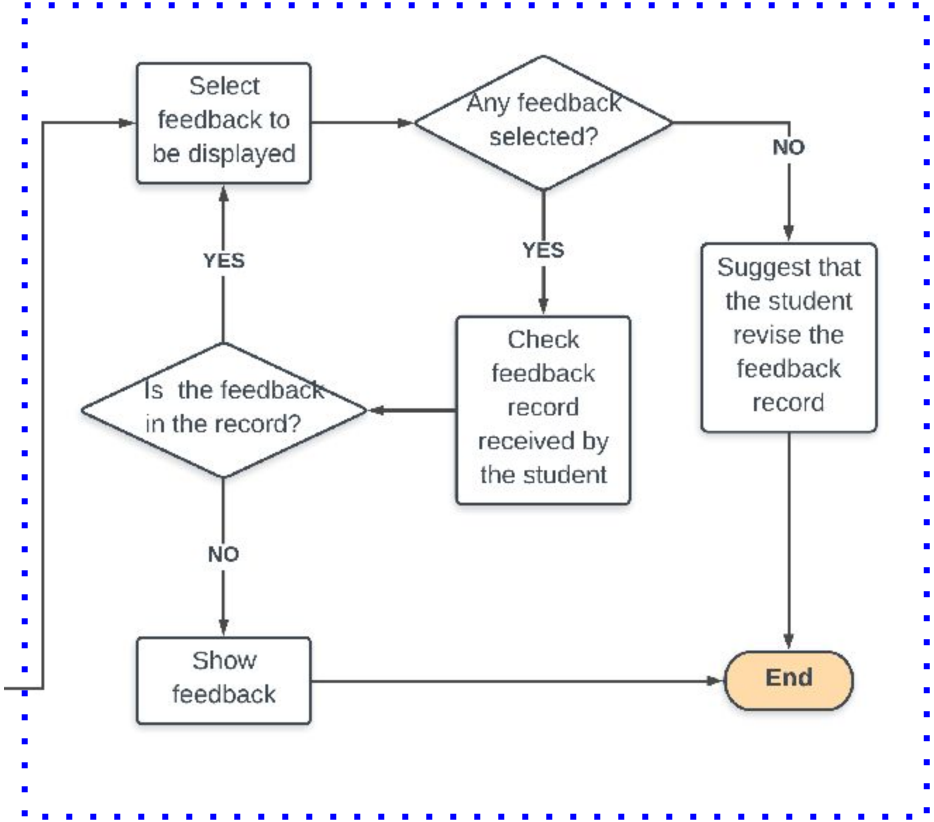




At the end the search process will return a sorted list of descending way of feedback messages.

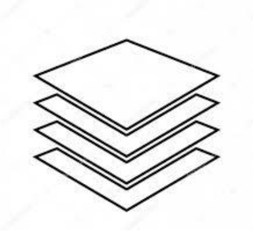
Approach to Provide Adaptive Feedback

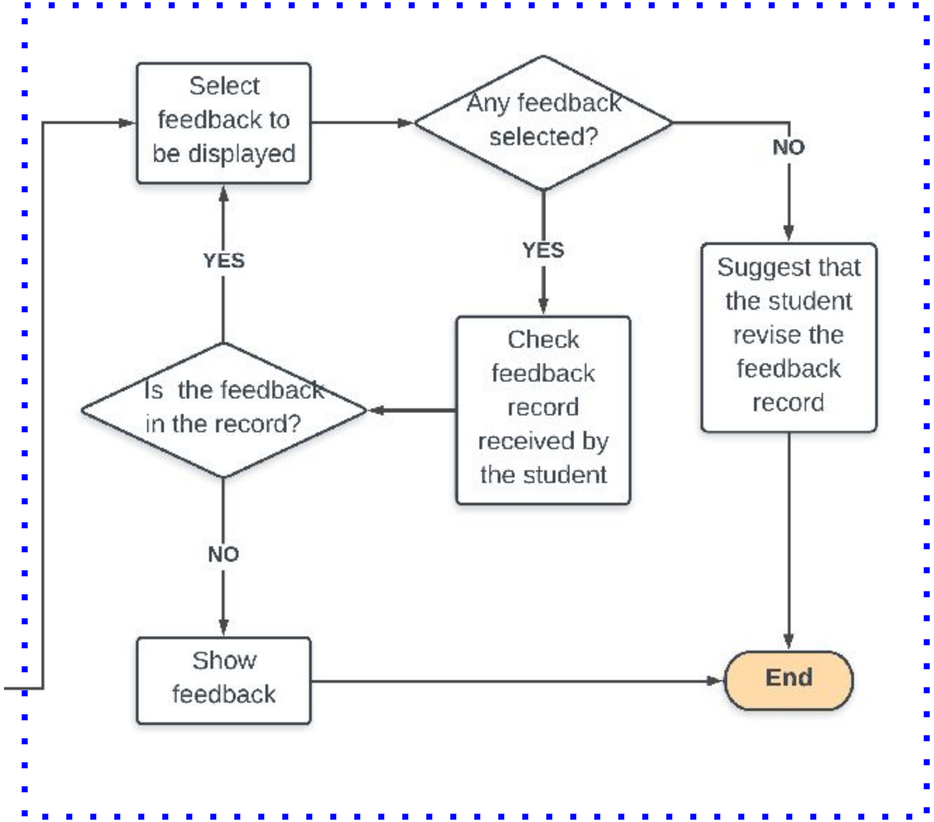




Approach to Provide Adaptive Feedback

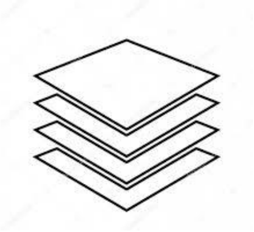
The list can be empty. If this happened, there is no feedback to present for the student, and the system will suggest that the student revise the feedback record

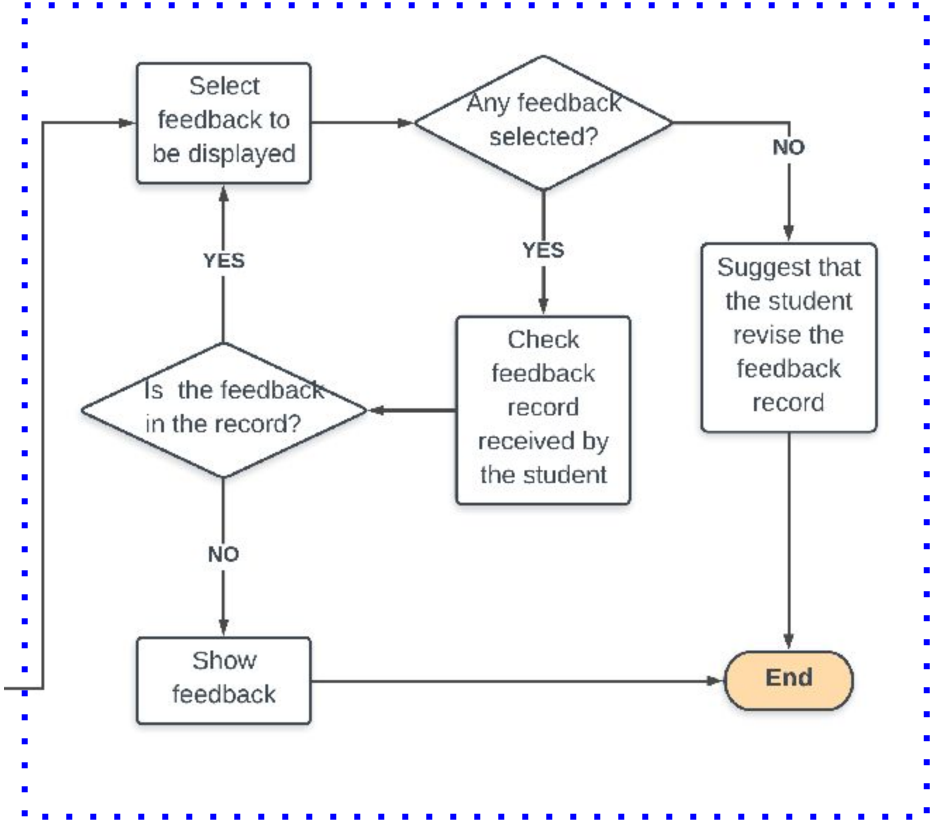




Approach to Provide Adaptive Feedback

If the list is not empty. The system takes the first message from the list and verifies if the message has already been presented to the student.

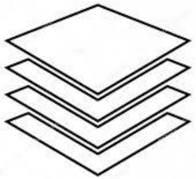


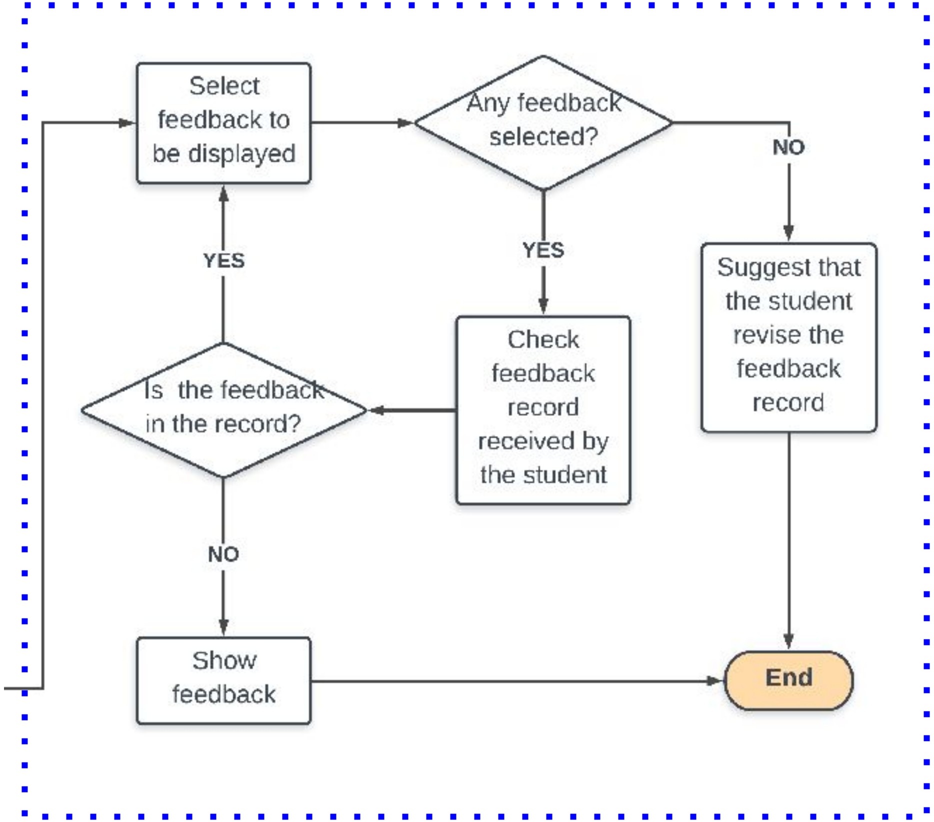


Approach to Provide Adaptive Feedback

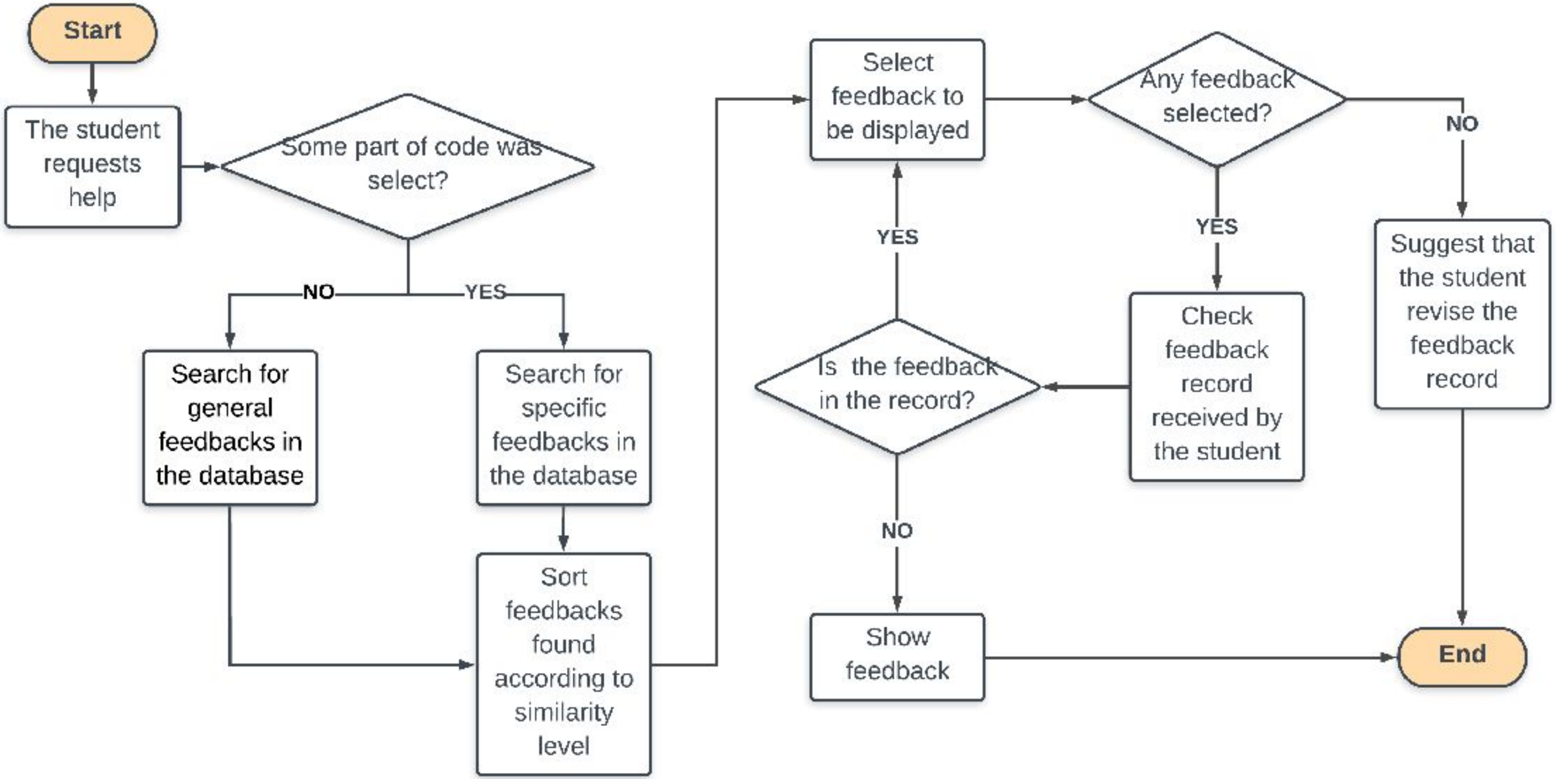
If the answer is yes, the system will select the next message from the list.

If the answer is no, the system will show the message to the student.





Approach to Provide Adaptive Feedback



Methodology of evaluation

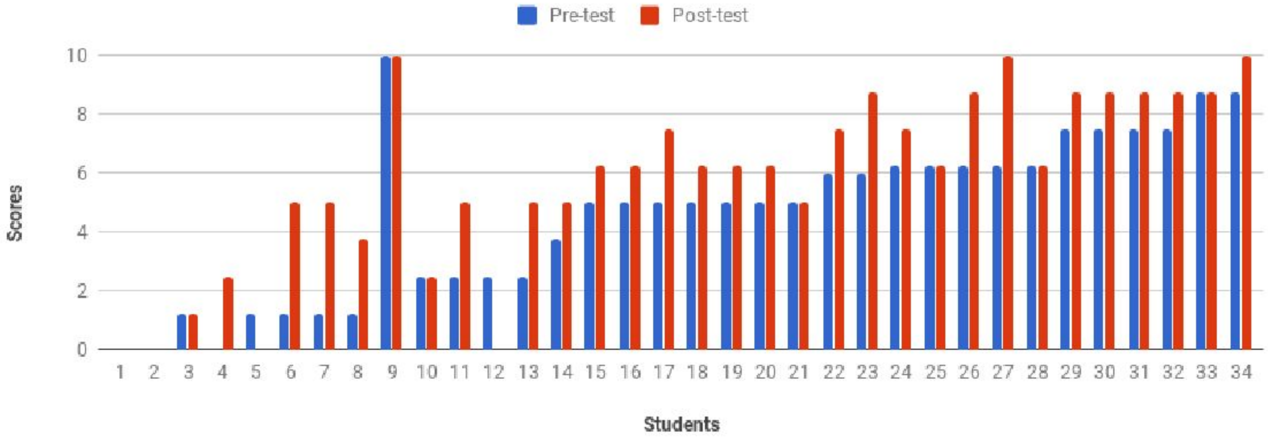
- We conducted a quasi-experiment to determine if the approach has a positive impact on the student learning experience.
- **Participants**
  - This study involves 34 participants (13 female and 21 males).
  - They are students from an introductory programming course in Brazil.
  - The students were from the same class, and this was the first programming course of all them.

Methodology of evaluation

- **Experiment Design**
- The learning impact was evaluated using a pre-test/post-test approach
  1. During the first session (60 min), the students answered the pre-test.
  2. During the second session (120 min), they used the system with our feedback approach individually on a desktop computer. The students were instructed to answer three selected questions on the system, and if they needed help with the questions, they could only use the feedback provided by the system.
  3. During the last session (60 min), they answered the post-test.
- The pre-test and post-test were composed of one programming question. The tests were evaluated using test cases, and the scores determinate between zero to ten.

Methodology of evaluation

- **Results and Discussion**
  - The students' performance in the post-test (M = 5.80, Mdn = 6.25, SD = 3.06) was better than in the pre-test (M = 4.58, Mdn = 5.00, SD = 2.77).
  - These results are providing good evidence that the use of our feedback approach aided the students to improve them programming learning.



Methodology of evaluation

- **Experiment Design**
- Paired T-Test was used to assess whether the difference between pre and post-test was significant.
- The t-test applied resulted in a p-value $< .001$.
- This indicates that the students' scores in the post-test were significantly higher than the pre-test.
- These results indicate that our feedback approach obtained a positive impact on the students learning.

Methodology of evaluation

- **Experiment Design**

Table 1. Statistical data of feedback requests made by students during the experiment.

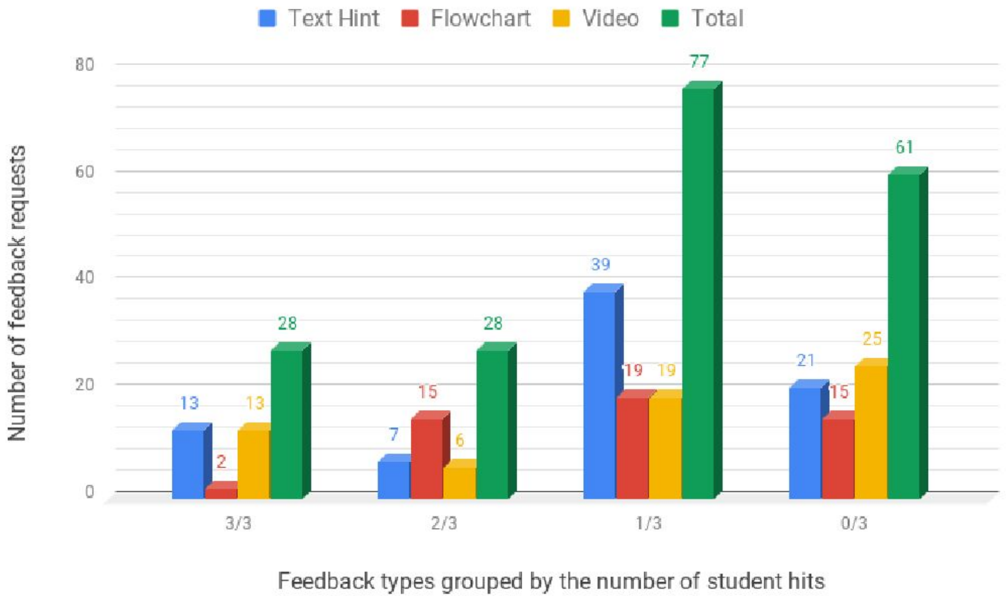
Feedback format	N	%	M	Mdn	SD
Text	80	41.2%	2.35	1.0	3.0
Flowchart	51	36.3%	1.50	0.5	2.2
Video	63	32.5%	1.85	1.0	2.5
Total	194	100%	5.70	1.0	2.6

Methodology of evaluation

- **Experiment Design**
- We analyze the students' interaction log with the system to check their behaviors during feedback requests.
- We divided the students into four groups:
  1. The group that submitted correct solutions to the three questions
  2. The group that hit two questions
  3. The group that got just one question right
  4. The group that failed to submit any correct solution.

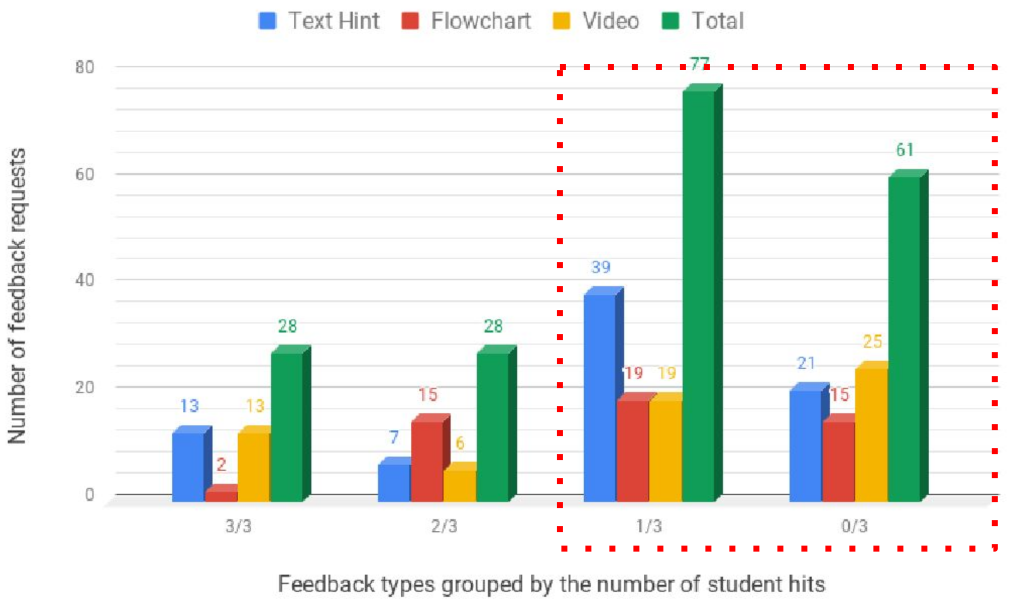
Methodology of evaluation

- **Experiment Design**
- This chart shows the feedback requests divided by each group



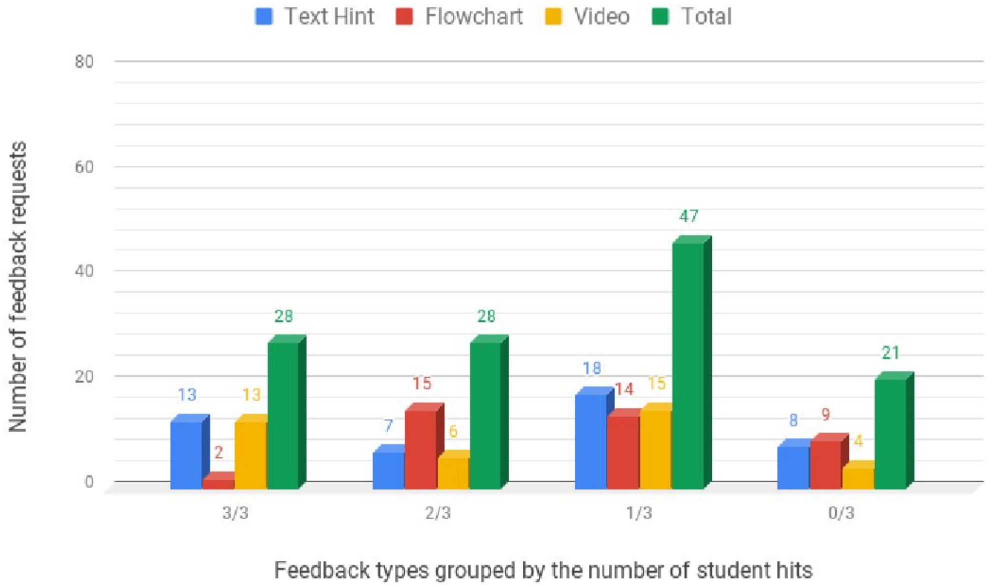
Methodology of evaluation

- **Experiment Design**
- During the analysis of the log of interactions, it was noticed that some students showed an atypical behavior.
- They made several requests for feedback in sequence without using incoming messages to change their solution. This behavior has been categorized as help abuse.



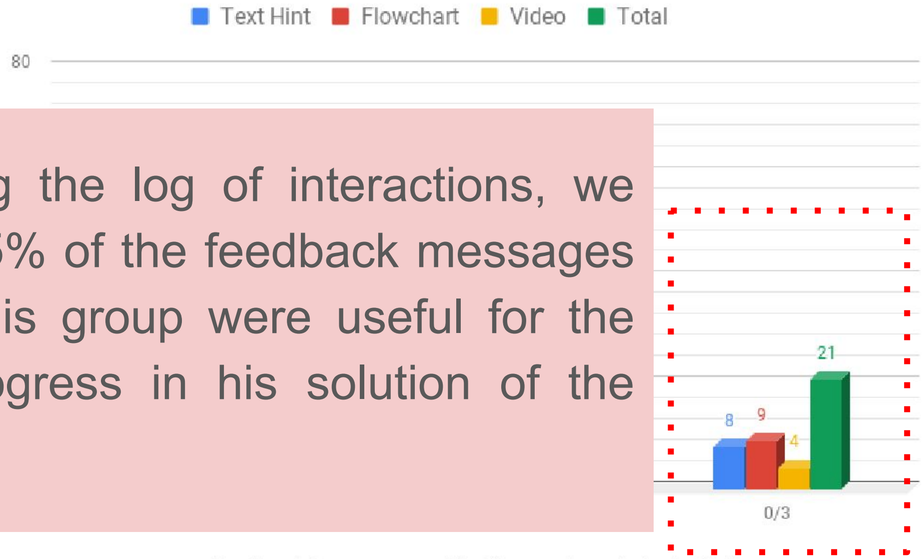
Methodology of evaluation

- **Experiment Design**
- By removing these students from the analysis, we obtain a new feedback request chart.



Methodology of evaluation

- **Experiment Design**
- By removing these students from the analysis, we obtain a new feedback request chart.

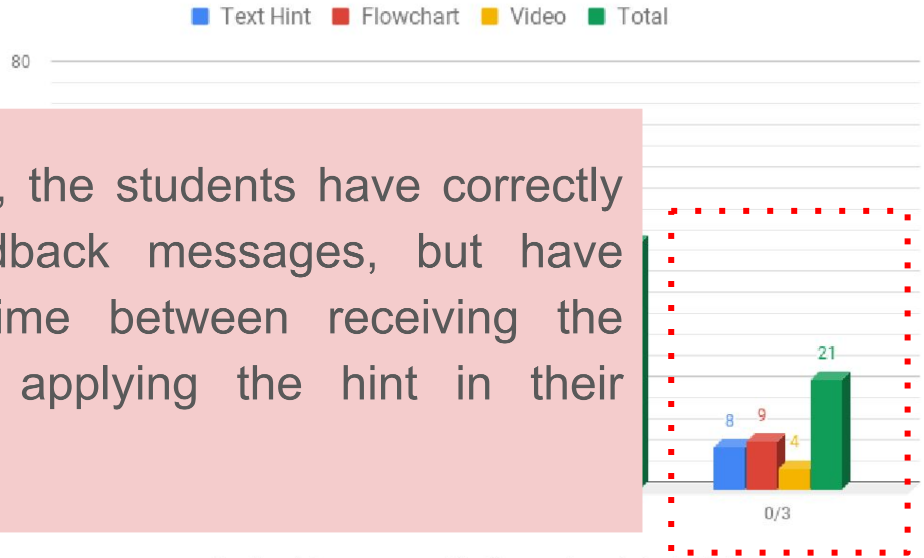


Feedback types grouped by the number of student hits

After analyzing the log of interactions, we noticed that 85% of the feedback messages received by this group were useful for the student to progress in his solution of the problem.

Methodology of evaluation

- **Experiment Design**
- By removing these students from the analysis, we obtain a new feedback request chart.

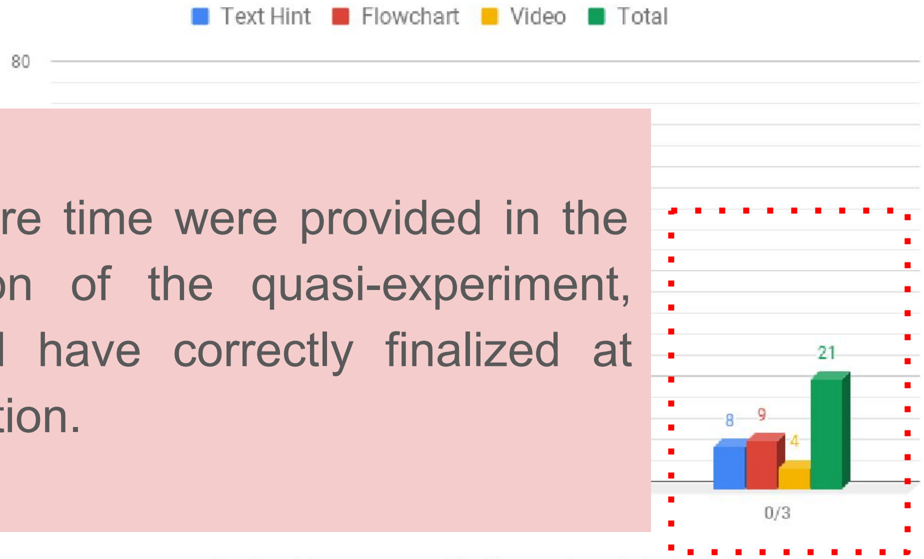


In many cases, the students have correctly used the feedback messages, but have spent much time between receiving the message and applying the hint in their solution.

Feedback types grouped by the number of student hits

Methodology of evaluation

- **Experiment Design**
- By removing these students from the analysis, we obtain a new feedback request chart.



Possibly, if more time were provided in the second session of the quasi-experiment, students could have correctly finalized at least one question.

Feedback types grouped by the number of student hits

Methodology of evaluation

- **Threats of Validity**
- Some students may have felt frustration and boredom during the experiment.
- The representativeness of the sample since all thirty-four students are from the same class and course.This problem can be solved applying this study with different samples and other contexts.

Conclusions

- In this paper, we present an approach to provide adaptive feedback during solving programming problems.
- The recommendation of feedback is based on the current state of the student's solution. One of the main contribution of this paper is the provision of more than one feedback format (text, video, and flowchart).
- We found evidence that our approach may have had a positive impact on students' programming learning.

ITS 2019
Jamaica
3-7 JUNE

An Adaptive Approach to Provide Feedback for Students in Programming Problem Solving

Priscylla Silva
Evandro Costa
Joseana Araújo

INSTITUTO FEDERAL
Alagoas
Campus Rio Largo

UFCG
SUPRA OMNES LUX LUCES

SCIENTIA AD SAPIENTIAM

Similarity Calculation

- For the realization of this calculation, an abstract syntactic tree (AST) is generated for the model solution codes and the student solution code.
- An AST is a representation of the structure of the source code written in a programming language.
- The AST is serialized in string format.
- Then, the strings are compared using Levenshtein's distance algorithm.
- The comparison that results in the least number of modifications is considered the most similar.

Approach to Provide Adaptive Feedback

- How does this search work?
  1. The system retrieves all feedback messages linked to the question that the student is currently solving;
  2. The system extracts the part of the code specified by the student from the current state of his/her solution. If no part is specified, the entire solution will be extracted;
  3. The system extracts the code parts of the model solutions whose feedback messages are associated. In the case of general feedback, the entire solution is extracted;
  4. The system performs a similarity calculation between the part of the student solution code and the parts extracted from the model solution; and
  5. All feedback messages whose similarity calculation is different from zero are returned.

Related Work

In the [3], the authors make a systematic review of intelligent tutoring systems for programming education. They present 14 systems of which 12 contains solutions for adaptive feedback. Most systems created to assist students in programming learning offer only immediate feedback after a valid solution is submitted. Such systems do not have mechanisms that assist the student in the during the elaboration of the solution [6].

Few papers in the literature have as a proposal to provide feedback to the student during the coding process. In the [6], the authors analyzed 101 systems for programming teaching with automatic feedback generation, of this 32.7% generate feedback messages for error correction and only 18.8% generate messages that help the student how to proceed in the next steps.