Research Article

Instrumented Correction Windows in Moodle: A Formative Intervention in Seven University Course Instances and the Configuration Contradictions That Make It Fail Silently

Julian Andrés Castaño Espinosa[1]

1. School of Engineering, Corporación Unificada Nacional de Educación Superior, Bogotá, Colombia

A low grade communicated after the deadline has closed is not feedback: it is a verdict. This experience report describes the design, instrumentation and deployment of a correction cycle with an open window in seven Moodle course instances belonging to five undergraduate and graduate courses, with 279 enrolled students, during the 2026-2 term at a Colombian higher education institution. The intervention reverses the usual order of grade closing: instead of grading and publishing, it first measures who is below the institutional passing threshold, checks live whether the platform still allows the submission to be replaced, and only then issues a report per course that tells each student what to correct and how many hours are left.

The contribution is not the idea of second chances, which is old, but the observation that in Moodle the correction window depends on four settings that can contradict each other without producing any visible error. We document those contradictions —duedate versus cutoffdate, a maxattempts of 1 that makes any attemptreopenmethod inert, an untilpass without a declared gradepass that has no criterion to evaluate, and an empty gradebook cell that does not equal zero— and we describe the five tools built to operate them: per-course correction reports, individual extensions on quizzes and on assignments, raising the attempt limit, and closing a graded forum including those who did not participate.

We also report three design decisions that are ethical rather than technical: separating the teacher's record from the text the student reads, granting time to one person without reopening the activity for the whole group, and not stating who passes when the platform declares no passing grade. We state

 doi.org/10.32388/I9AN4Z

explicitly what this work does not provide: there is no control condition, the effect on learning is not measured, and the results of the intervention are not reported as evidence of effectiveness.

Corresponding author: Julián Andrés Castaño Espinosa, julian_castanoe@cun.edu.co

1. Introduction

There is a moment, at the end of every academic term, when feedback stops being formative and becomes a record. It is the moment the deadline closes. Before it, saying "your theoretical framework does not support your objectives" is an actionable instruction; after it, it is an explanation of why the grade is what it is. The text may be identical; its pedagogical function is not.

The literature on formative feedback has insisted for decades that its value depends on the student being able to **act** on it[1][2][3][4][5]. What this report contributes is not that idea, but an operational observation that conditions it and is rarely written down: in a Moodle course, knowing whether the student can still act **is not a trivial question, and the platform does not answer it directly**. It depends on the interaction of several parameters that an instructor configures at different moments, sometimes months apart, and whose inconsistent combinations the platform accepts without complaint.

The result is a kind of failure that is particularly hard to detect: the course *looks* open —the activity is visible, the due date has not passed yet— and yet the student cannot upload a new version. Nobody receives an error. The instructor believes an opportunity was given; the student finds a closed door with no sign on it.

This paper describes how that check was instrumented and what was built around it.

1.1. Context

The setting is seven Moodle course instances at a Colombian higher education institution, corresponding to five courses: one graduate course (Project I of a specialization programme in artificial intelligence) and four undergraduate courses (Research in Science and Technology, Creativity and Innovative Thinking, and Degree Project 2 and 3). One of the courses is taught to three simultaneous groups with their own calendars, which is why seven course instances correspond to five courses.

Course	Groups	Enrolled
Research in Science and Technology	1	20
Creativity and Innovative Thinking	1	50
Degree Project 2	1	50
Degree Project 3	3	107
Project I (specialization)	1	52
Total	**7**	**279**

The installation is Moodle 4.5, and the grading scale is 0.00 to 5.00 per item. The institutional passing threshold is 3.0: the official instructor report form for the term defines as passing a grade equal to or higher than 3.0. That threshold lives outside the platform and outside the working repository: **none of the seven course instances has a `gradepass` configured** —the items' grade to pass is 0.00— and no versioned source in the repository declares it, which is the observation from which Castaño Espinosa[6] starts. That gap between the institutional form and the platform's configuration has mechanical consequences, which are described in §3.3.

1.2. What is reported and what is not

This is an **experience report** by a single instructor at a single institution. There is no control condition, there is no random assignment and the effect on learning is not measured. The figures that appear are descriptive: how many course instances, how many enrolled students, what was configured and what was observed when configuring it.

We say this at the beginning and not at the end because the temptation in this genre is to present a well-intentioned intervention as if it were a result. It is not. What can be stated with evidence is what the platform does and does not do under certain configurations, which is verifiable and reproducible by anyone who has a Moodle in front of them.

1.3. Relationship with other manuscripts by the author

Two works by the same author, a published preprint and a manuscript not yet published, deal with tools that also appear here. *Escritura segura sobre un Moodle institucional*[7] analyses the safety property of the writes —simulation by default, explicit confirmation, re-reading after writing— of the extension, attempt-limit and forum-closing tools listed in §2.1. *Auditoría de solo lectura del cierre de notas en un Moodle institucional*[6] describes the audit of the gradebook and summarises in one section the correction report tool, the separation of the two records and the refusal to state who passes.

This manuscript does not repeat the confirmation architecture, treated in the first, nor the audit of grade closing, treated in the second. What is specific to it is: (1) the pedagogical cycle of correction with an open window and the order of its four steps (§2); (2) the complete inventory of the four configuration contradictions (§3), of which the second work only mentions the combination of reopening with a maximum of one attempt and the absence of a passing grade; and (3) the decisions on individual extensions and on per-course reports (§4.1 and §4.2). The separation of records (§4.3) and the lack of gradepass (§3.3) coincide with the second work; here they are treated for their effect on the correction window and not on grade closing.

2. The intervention

The cycle has four steps and their order is not negotiable.

First, measure without publishing. Submissions are downloaded and checked against the criteria guide that the student already has in hand —the same document with which the work was requested, not a different internal rubric— and a **proposed grade** is produced with its justification. None of this touches the course. The distinction between measuring and publishing is the backbone of the design: a proposed grade can be discussed, corrected and discarded; a published one has already changed the conversation.

Second, check the clock. Before telling anyone to make corrections, it is checked live whether the activity still accepts a new version. This is the part that the platform does not answer directly, and it is the subject of §3.

Third, issue a report per course. Not per student. The decision to group by course instead of generating one document per person was made after trying the opposite, and it is explained in §4.2. The report opens with the final deadline and the hours remaining, lists those below the threshold with what they

must correct, **also includes those who have not submitted anything**, and gives the exact link to the activity in the course.

Fourth, open the door to those who need it. An individual extension, or one more attempt, depending on what is closed. This is done person by person and never by reopening the activity for the whole group, for reasons that are both of fairness and of traceability (§4.1).

2.1. The tools

Five programs, built between 18 and 21 September 2026, all with the same safety architecture: **without an explicit confirmation flag they do not send a single write request**; they simulate and show the plan.

Tool	What it does	Writes to the course
Correction report	Who is below the threshold, what they must correct, how many hours are left, who has not submitted	No
Quiz extension	Individual deadline through a *user override*, without touching the group's dates	Yes, with confirmation
Assignment extension	Individual extension; it also reads the grading deadline and warns if it is exceeded	Yes, with confirmation
Attempt limit	Raises maxattempts; re-reads the form after writing and compares the 24 date fields	Yes, with confirmation
Graded forum closing	Grades those who participated and leaves at zero those who did not	Yes, with confirmation

The fifth deserves a note. A graded peer-assessment forum leaves, when it closes, two distinct populations: those who participated and received a rating, and those who did not participate and **have no cell**. Those two are not the same, but the platform treats them alike unless someone intervenes, with the consequence described in §3.4.

3. Four contradictions that produce no error

This is the central contribution of the paper. Each of these four situations leaves the course in a state that *looks* correct and is not.

3.1. duedate is not the close

In a Moodle assignment, duedate is the date after which a submission is marked as late. cutoffdate is the date after which **submissions are no longer accepted**. They are two different fields and they can be far apart in time.

An instructor who looks at the course and sees that the due date passed three days ago may conclude that nothing can be done, when in fact the window remains open until cutoffdate, which may be weeks later. And if cutoffdate is not enabled, the assignment keeps accepting submissions, marked as late, with no time limit. What the platform does prevent is the opposite case: in Moodle 4.5 the assignment form refuses to save a cutoffdate earlier than the duedate ("Cut-off date cannot be earlier than the due date"; [8] [9]). The confusion, therefore, is not in the platform but in the reading: the label "due date" is read as the close, and it is not.

The report tool computes the remaining hours **against `cutoffdate`**, not against duedate. If the assignment has no cut-off date, it takes duedate as the close, which is a conservative reading: without cutoffdate, the platform would keep accepting late submissions after that date.

3.2. A single attempt makes reopening inert

In an assignment, attemptreopenmethod controls how additional attempts are granted. In Moodle 4.5 the form offers three options: "Manually" (manual), "Automatically" (automatic) and "Automatically until pass" (untilpass); the "Never" option (none) was deprecated in that version[8]. It looks like the decisive setting, and it is not.

If maxattempts is 1, **no value of `attemptreopenmethod` reopens anything** (the setting's own help says that, once the maximum has been reached, the submission can no longer be reopened; [8]). In Moodle 4.5 the settings form hides the reopen method when the maximum is 1[9], but the saved value is still there: the configuration can keep saying "until pass" with a single attempt already used. There is no error message or warning on saving, and since the field is hidden, the panel does not show the contradiction

either. The report tool detects it because it reads the value of the form and not what is displayed on screen.

In Moodle 4.5 this combination does not even require an oversight: it is the factory configuration. The site defaults are a maximum of 1 attempt and the "Automatically until pass" method[10], so that, unless the administrator changes them, every new assignment is born with "until pass" saved and hidden. Raising the limit is not enough on its own either: the "Allow another attempt" action in the grading panel only appears when the method is "Manually" and the student has attempts left[11]. To grant another attempt person by person, the limit has to be raised and the method changed to "Manually"; the alternative is to declare a grade to pass so that "until pass" has a criterion (§3.3).

3.3. untilpass without a passing grade has no criterion

attemptreopenmethod = untilpass means "reopen until the student passes". Passing, for Moodle, is reaching the item's gradepass[8].

In the seven course instances studied **there is no `gradepass` configured**: the items' grade to pass is 0.00. The 3.0 threshold exists in the institutional closing form, but not in the configuration of the gradebook. "Until pass" is then left with no criterion to evaluate, and the code confirms it: the assignment only reopens an attempt if the grade is judged "not passed", and that judgment returns a null value when the grade to pass equals the minimum grade, so that with 0.00 **`untilpass` reopens no attempt**[12][13].

This has a consequence that goes beyond the technical and that should be stated carefully: **if the platform does not declare a passing grade, no tool that reads the platform can state who passes and who fails.** It can say who is below a threshold that the instructor passes to it as a parameter —which is what the correction report does, with the 3.0 of the institutional form— but that is a measurement against an external criterion, not a reading of the course. The difference matters when the recipient of the report is a person.

3.4. An empty cell is not a zero

When a gradebook item is left without a grade, Moodle **does not count it as zero**: it excludes it and renormalises the rest. The course total is computed over what is there. This is the behaviour of the "Exclude empty grades" setting, enabled by default[14][15].

The practical effect is that a student who did not take part in the peer-assessment forum, and therefore has no cell, can show a higher course total than a classmate who did take part and received a low rating. The platform is not lying: it is computing an average over the available items. But the number that comes out does not mean what the instructor thinks it means.

That is why the forum-closing tool does two things in a specific order: first it grades those who participated, and then it fills with zero the cells that were left empty. Reversing the order would overwrite legitimate ratings.

This phenomenon interacts with the previous one in an uncomfortable way: since there is no gradepass, and since empty cells renormalise, the "course total" that the course displays during the term is a provisional number whose interpretation changes every time a new item is graded. Showing it to the student as if it were their grade is a predictable source of conflict.

4. Three decisions that are not technical

4.1. Individual extension, never reopening for the group

When a student asks for time for a justified reason, the quickest way out is to move the activity's closing date. It is also the worst.

Moving the group's date changes the conditions of those who already submitted on time, erases the distinction between on-time and late submission for everyone, and leaves no trace of who was granted what. A *user override* does the opposite: it affects a single person, it is recorded with its date, and the general conditions remain intact.

The two extension tools —one for quizzes, the other for assignments— work exclusively this way. They refuse to act if the name they receive matches more than one person on the roster, because an ambiguous match resolved for convenience gives time to someone who did not ask for it.

The assignment tool also performs a check that the platform does not require: it reads the instructor's **grading deadline** and warns if the extension granted goes beyond it. Giving until the 28th when grading must be closed on the 25th is not granting an opportunity; it is manufacturing a breach.

4.2. One document per course, not per student

The first version of the report generated one document per student. That was the intuitive choice: everyone receives their own.

It was changed to one document per course with a section per student. The reasons were practical and they held: an instructor with seven groups does not open 279 files; the message that accompanies the report is the same for all the courses and only the attachment changes; and grouping allows the report to include as well, in the same view, **those who have not submitted anything**, who under a per-student scheme simply generate no file and therefore drop off the radar.

That last consequence is the important one. The per-student design made exactly the highest-risk population invisible.

4.3. Two records that are never mixed

Everything the system produces exists in two forms that are never confused: the **instructor's record** — evidence, weights, internal identifiers of the activity, what is still to be decided— and the **text for the student**, which answers what concerns them and nothing else, in the second person, without percentages or the internal mechanics of the course.

The reason is not aesthetic. The scaffolding with which a decision is made does not help the person who receives the decision: it overwhelms them and moves their focus away from what they have to do. A student who reads "your item weighs 30% and your proposed grade is 2.4 on criterion 3 of the rubric" does not know better what to correct than one who reads "your theoretical framework does not support the objectives you set; revise specific objective 2".

A minor but real consequence of this separation: the text for the student **does not infer the person's gender from their name**. It is worded so that it does not need to.

5. What it does not solve

It is worth being explicit, because an experience report without declared limits is advertising.

It does not measure learning. We do not know whether those who made corrections within the window learned more than those who did not. We know that they had the opportunity to do so and that before they did not have it in a systematic way.

It does not replace the instructor's judgment. The grades that the cycle produces are proposals. The decision to publish them, modify them or discard them belongs to the instructor, and no tool makes it for them. This is implemented, not only stated: without explicit confirmation, the programs do not write.

It does not correct the cause. That a correction window is needed at the end of the term is, in part, a symptom that intermediate feedback did not arrive on time or was not actionable. The intervention mitigates; it does not fix the assessment design that makes it necessary.

It is not generalizable without checking. The four contradictions in §3 were observed in a specific Moodle 4.5 installation and contrasted with the source code of that version. The field names are standard, but the behaviour changes between versions —in 4.1, for example, the reopen method was not hidden with a single attempt and the "Never" option was still in force— and between institutional configurations. The way to use this work is not to trust its conclusions: it is to repeat the checks in one's own course.

The scope is one instructor and one institution. Everything else is a hypothesis.

6. Conclusion

The useful question is not whether correction opportunities should be given, which they evidently should, but whether the course is really configured to admit them at the moment they are announced. This paper argues that the second question cannot be answered by looking at the interface, because Moodle accepts without complaint combinations of settings that contradict each other, and that therefore **the check has to be instrumented and run live, right before promising anything**.

What is contributed is that check, the set of tools around it, and the inventory of the four contradictions that make it necessary. What is not contributed is evidence that this improves learning; that is the next phase, and it requires a design that was not available here.

One observation remains that goes beyond the technical scope. Three of the four contradictions described —the close that is not the close, the reopening that does not reopen, the pass without a criterion— have the same form: **the platform offers a setting whose name promises a semantics that its implementation does not guarantee**. The instructor configures what the name suggests and gets something else, without warning. If anything should change upstream, it is that.

doi.org/10.32388/I9AN4Z

Notes

A preliminary version of this work was presented as a conference paper (code P001) at the 2026 International Congress "Arquitecturas del Sentido" (UNIMINUTO, online, 12 November 2026).

Statements and Declarations

Funding

No funding was received for this work.

Potential Competing Interests

The author declares no conflict of interest. The author is the instructor of the courses described and developed the tools that are reported.

Ethics

No people took part as research subjects. No name, identity document or student identifier is reproduced.

Data Availability

The research data are contained in the manuscript itself. No student identity data are reported: all figures are aggregates by group. The code of the tools is in the author's working repository, which is private because it contains the configuration of institutional courses in operation.

Author Contributions

J.A.C. was the sole author and is responsible for all aspects of the manuscript.

Use of Generative AI

The commits that created the five tools described, between 18 and 21 September 2026, carry the co-authorship mark of a language model (Claude, by Anthropic), which assisted the author in writing the code. The revision of the manuscript, the translation of the abstract and the verification of the references were done with the assistance of the same type of tool. This English version was translated with the assistance of Claude (Anthropic) from the author's Spanish original, which the author reviewed. The

author reviewed the claims in that original and is responsible for the content. No reference was included without verifying that it corresponds to an existing work.

References

1. ^*Sadler DR (1989). "Formative Assessment and the Design of Instructional Systems." Instr Sci. **18**(2):119–144. doi:10.1007/BF00117714.*
2. ^*Black P, Wiliam D (1998). "Assessment and Classroom Learning." Assess Educ Princ Policy Pract. **5**(1):7–74. doi:10.1080/0969595980050102.*
3. ^*Nicol DJ, Macfarlane-Dick D (2006). "Formative Assessment and Self-Regulated Learning: A Model and Seven Principles of Good Feedback Practice." Stud High Educ. **31**(2):199–218. doi:10.1080/03075070600572090.*
4. ^*Hattie J, Timperley H (2007). "The Power of Feedback." Rev Educ Res. **77**(1):81–112. doi:10.3102/003465430298487.*
5. ^*Carless D, Boud D (2018). "The Development of Student Feedback Literacy: Enabling Uptake of Feedback." Assess Eval High Educ. **43**(8):1315–1325. doi:10.1080/02602938.2018.1463354.*
6. a, b *Castaño Espinosa JA (2026). "Auditoría de solo lectura del cierre de notas en un Moodle institucional: contrastar el libro de calificaciones con una fuente de verdad versionada" [Read-Only Audit of Grade Closing on an Institutional Moodle: Checking the Gradebook Against a Versioned Source of Truth]. Corporación Unificada Nacional de Educación Superior.*
7. ^*Castaño Espinosa JA (2026). "Escritura segura sobre un Moodle institucional: simulación por defecto y confirmación explícita en herramientas docentes" [Safe Writing on an Institutional Moodle: Simulation by Default and Explicit Confirmation in Teaching Tools]. Zenodo. doi:10.5281/zenodo.23481348.*
8. a, b, c, d *Moodle HQ. "mod/assign/lang/en/assign.php (MOODLE_405_STABLE branch, Moodle 4.5)." GitHub. https://github.com/moodle/moodle/blob/MOODLE_405_STABLE/mod/assign/lang/en/assign.php.*
9. a, b *Moodle HQ. "mod/assign/mod_form.php (MOODLE_405_STABLE branch, Moodle 4.5)." GitHub. https://github.com/moodle/moodle/blob/MOODLE_405_STABLE/mod/assign/mod_form.php.*
10. ^*Moodle HQ. "mod/assign/settings.php (MOODLE_405_STABLE branch, Moodle 4.5)." GitHub. https://github.com/moodle/moodle/blob/MOODLE_405_STABLE/mod/assign/settings.php.*
11. ^*Moodle HQ. "mod/assign/gradingtable.php (MOODLE_405_STABLE branch, Moodle 4.5)." GitHub. https://github.com/moodle/moodle/blob/MOODLE_405_STABLE/mod/assign/gradingtable.php.*

12. ^Moodle HQ. "lib/grade/grade_grade.php (MOODLE_405_STABLE branch, Moodle 4.5)." GitHub. https://github.com/moodle/moodle/blob/MOODLE_405_STABLE/lib/grade/grade_grade.php.
13. ^Moodle HQ. "mod/assign/locallib.php (MOODLE_405_STABLE branch, Moodle 4.5)." GitHub. https://github.com/moodle/moodle/blob/MOODLE_405_STABLE/mod/assign/locallib.php.
14. ^Moodle HQ. "admin/settings/grades.php (MOODLE_405_STABLE branch, Moodle 4.5)." GitHub. https://github.com/moodle/moodle/blob/MOODLE_405_STABLE/admin/settings/grades.php.
15. ^Moodle HQ. "lang/en/grades.php (MOODLE_405_STABLE branch, Moodle 4.5)." GitHub. https://github.com/moodle/moodle/blob/MOODLE_405_STABLE/lang/en/grades.php.

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.