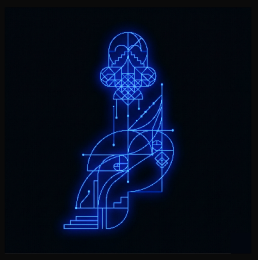
02 · TRAINING

CHAPTER 2.2

Teaching how to prove, not just how to compute

AP MATH JUSTIFICATION TRAINER



From the empirical argument —"I tried it with three numbers and it works"— to the formal justification that an AP exam rewards with a 5. The CERC framework makes mathematical argumentation trainable and measurable.

STATUS	STACK	DATA	MODEL
Beta	Next.js 14 · TS · KaTeX	Mock KV ↔ LMS	CERC scoring

01 THE PROBLEM

KNOWING HOW TO DO THE MATH, NOT HOW TO DEFEND IT

Advanced math students had mastered the procedure but were losing points where they count most: in the written justification. The diagnosis was clear and quantified. On the adaptive practice platforms the network used, multiple-choice performance hovered around **82%**; on the Free Response Questions –where the examiner grades not the result but the reasoning that supports it– it dropped to **67%**, and only a minority were truly on track toward the target score.

The existing procedural tool was solid for automating computation, but it did not train what a real AP exam measures: written mathematical argumentation.

A student could solve an integral flawlessly and still fail the question for not stating the condition that enables the theorem they used.

That gap —knowing how to do the math but not how to defend it— was invisible to the existing system.

And it was, exactly, the difference between a 3 and a 5.

02 THE GOAL

JUSTIFICATION AS A TRAINABLE SKILL

PRIMARY GOAL

Move the student from the empirical argument to the formal justification that cites the theorem's hypothesis and verifies its conditions before invoking it — treating mathematical justification as a trainable, measurable skill, not a diffuse talent.

To make this operational, an explicit scaffold was adopted: the **CERC** framework —Claim, Evidence, Reasoning, Conditions— which breaks every proof down into four visible, required moves.

The goal was not for the student to write more, but to write *complete*: that none of the four elements stay implicit, because it is precisely the implicit that an examiner cannot reward.

03 THE USERS

THE STUDENT AND THE TUTOR

The primary user is the advanced math student who already masters the calculation but loses points on the written answer. The platform was designed for three parallel courses —Calculus AB, Calculus BC and Statistics— because the structure of the argument is the same even when the content changes: a claim, its evidence, the principle that connects them, and the conditions that enable it.

The second user is the tutor or academic coordinator, who operates through an admin panel with visualization of each student's reasoning state (R-0388 and its peers always appear anonymized), per-unit progress tracking, and manual practice triggers.

> *A deliberate privacy decision: the tutor panel isolates sensitive data and never exposes to the student the pedagogical "traps" of each problem, so as not to contaminate the exercise.*

04 THE ARTIFACT

THE SPLIT-SCREEN SESSION

The central interface is a **split-screen** session: on the left, the problem statement with the notation rendered in KaTeX and the theorem box — name, statement, hypotheses—; on the right, the CERC form, four stacked fields —Claim, Evidence, Reasoning, Conditions— each with its description and its sentence frame where applicable. The completeness counter and the progress bar give real-time visual feedback as the student fills in each field.

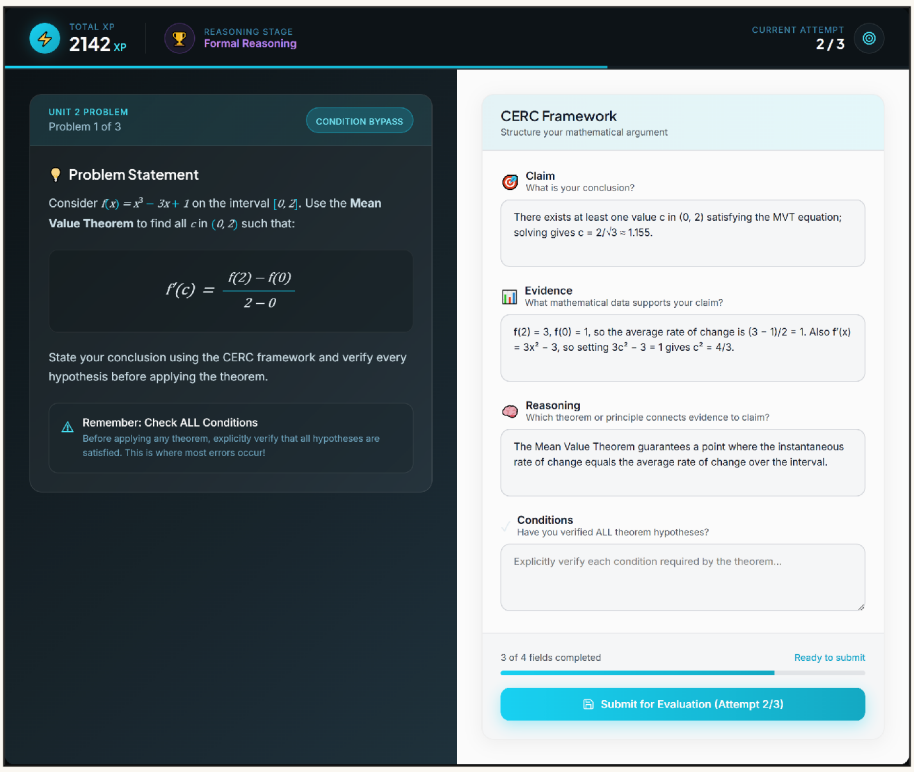


FIG. 1 — CERC SESSION · CALCULUS FRQ ON THE LEFT, CLAIM/EVIDENCE/REASONING/CONDITIONS FORM ON THE RIGHT · "3 OF 4 COMPLETE" SHOWS THE LIVE COMPLETENESS SCORING

05 THE FEEDBACK

COMPLETENESS SCORING, NOT SEMANTIC CORRECTNESS

After submission, the system evaluates **deterministically** whether the four elements are present and non-empty, and projects it onto the AP 1-to-5 rubric. It does not judge whether the argument is "good" with a language model: the first habit to install is not eloquence, but the structural integrity of the argument —that the condition is written, that the evidence exists, that the reasoning names the theorem.

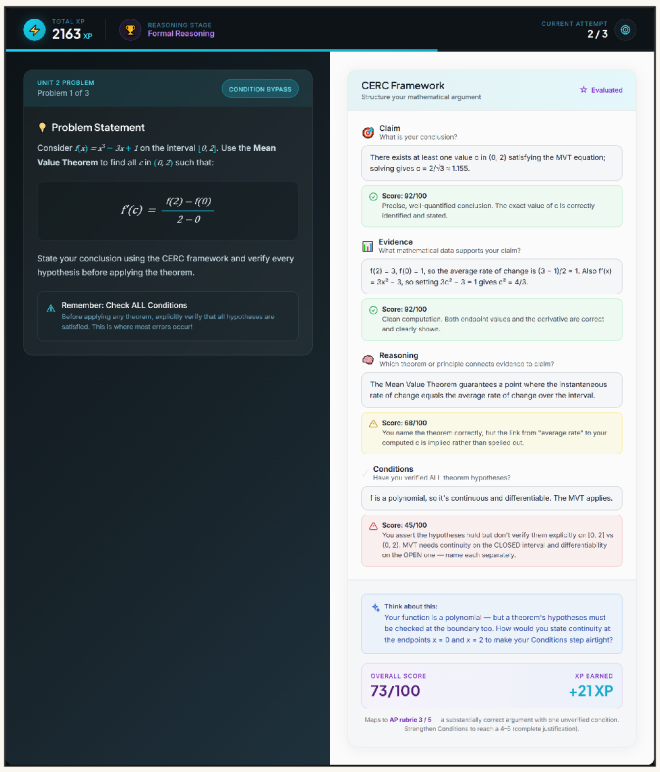


FIG. 2 — POST-SUBMISSION FEEDBACK · CERC COMPLETENESS MAPPED TO THE AP 1–5 RUBRIC · A SEMANTIC EVALUATOR WOULD HAVE BEEN MORE IMPRESSIVE AND FAR LESS HONEST ABOUT WHAT IT MEASURED

06 THE FOUNDATION

THREE STAGES, FOUR COGNITIVE UNITS

The design rests on a model of mathematical reasoning development with three successive stages, coded literally into the application's data type: **empirical** (the student is convinced by examples), **generic** (generalizes the pattern but without rigor), and **formal** (proves by invoking the structure). The course is built to force that transition.

UNIT	COGNITIVE FOCUS	WHAT IT TRAINS
U1	Breaking the empirical illusion	Problems designed so that intuition fails: the student experiences why seeing three cases is not enough.
U2	Condition verification	Verifying ALL of a theorem's conditions, with no shortcuts — the most expensive mistake on the real exam.
U3	Synthesis without scaffolding	Multi-concept synthesis and communicative precision, now without sentence frames.
U4	Timed FRQs	Individual Free Response under exam conditions.

The catalog classifies each problem by the type of error it provokes — `CONDITION_BYPASS`, `LOCAL_ONLY_ARGUMENT` or `CER_BREAKDOWN`— and pairs them with sentence frames (*sentence frames*) that are withdrawn unit by unit: a direct application of the *fading* principle, where the scaffolding fades away as competence consolidates.

07 THE DESIGN

THE CERC MODEL AND THE DECISION THAT MATTERED MOST

CERC breaks every proof down into four moves: the assertion (Claim), the evidence that backs it (Evidence), the principle or theorem that connects them (Reasoning), and the conditions that enable it (Conditions). A three-level hint system —where the flaw is, which CERC element is broken, the explicit correction— supports the student without solving the problem for them. Gamification adds XP per unit and unlocks badges with GSAP animations.

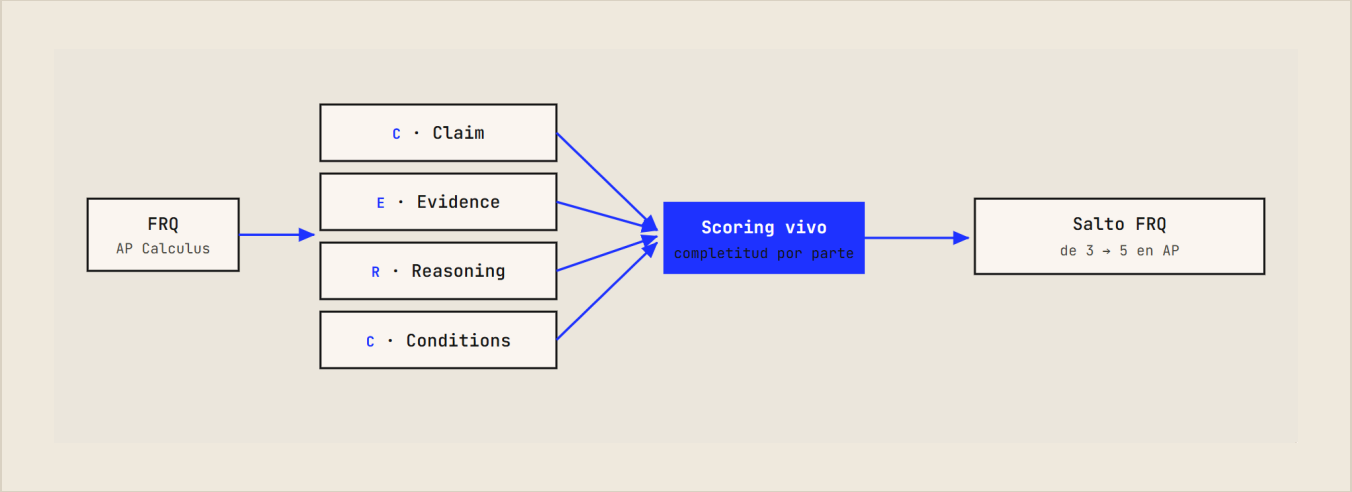


FIG. 3 — CERC MODEL: CLAIM → EVIDENCE → REASONING → CONDITIONS, THE FOUR VISIBLE AND REQUIRED MOVES OF EVERY PROOF

Completeness scoring — not semantic AI grading

Context. A semantic evaluator that judged whether the argument is “good” with a language model would have been more impressive, but dishonest about what it actually measured.

Decision. The system evaluates deterministically whether the four CERC elements are present and non-empty, with real-time visual feedback. The first habit to install is the structural integrity of the argument, not eloquence.

Accepted trade-off. It measures less “quality” and more “completeness” — but it attacks the right habit first and measures exactly what it claims to measure.

The second structural decision was the data layer with an **adapter** pattern: a mock adapter for development backed by Vercel KV, and an LMS adapter for production that preserves the correct shapes of the open educational interoperability standards.

08 BUILD AND VALIDATION

AI-FIRST, WITH SECURITY AS A DESIGN CONSTRAINT

It was built with an **AI-first** flow, assisted by Claude, at a deliberately high pace over about six weeks (March 17 to April 30), across 29 pages in Next.js 14's App Router. The stack settled on strict TypeScript, Tailwind, KaTeX for the notation and GSAP for the gamification.

Security was treated as its own front, not as an add-on: authentication with JWTs signed via *jose*, route protection by middleware with role-based access control (student / administrator), sanitization with DOMPurify, rate limiting and mandatory CSRF on every authenticated mutation. An external audit in early April detected hardening opportunities in authentication and PII handling; they were resolved by reinforcing the session model and the data isolation of the admin panel.

Validation was of two kinds. On the technical side, a test battery covered the integrity of the course data, the prerequisite logic between units, and the full flow of a session, plus the production build of all 29 pages. On the pedagogical side, the platform was prepared for a **pilot** with a small group of Calculus BC and Statistics students —anonymously identified— with their own login backed by the network's LMS real roster.

> *The measurement is not a subjective grade but the student's trajectory through the three reasoning stages and their growing CERC completeness as the scaffolding is withdrawn between Unit 1 and Unit 4.*

09 THE RESULT

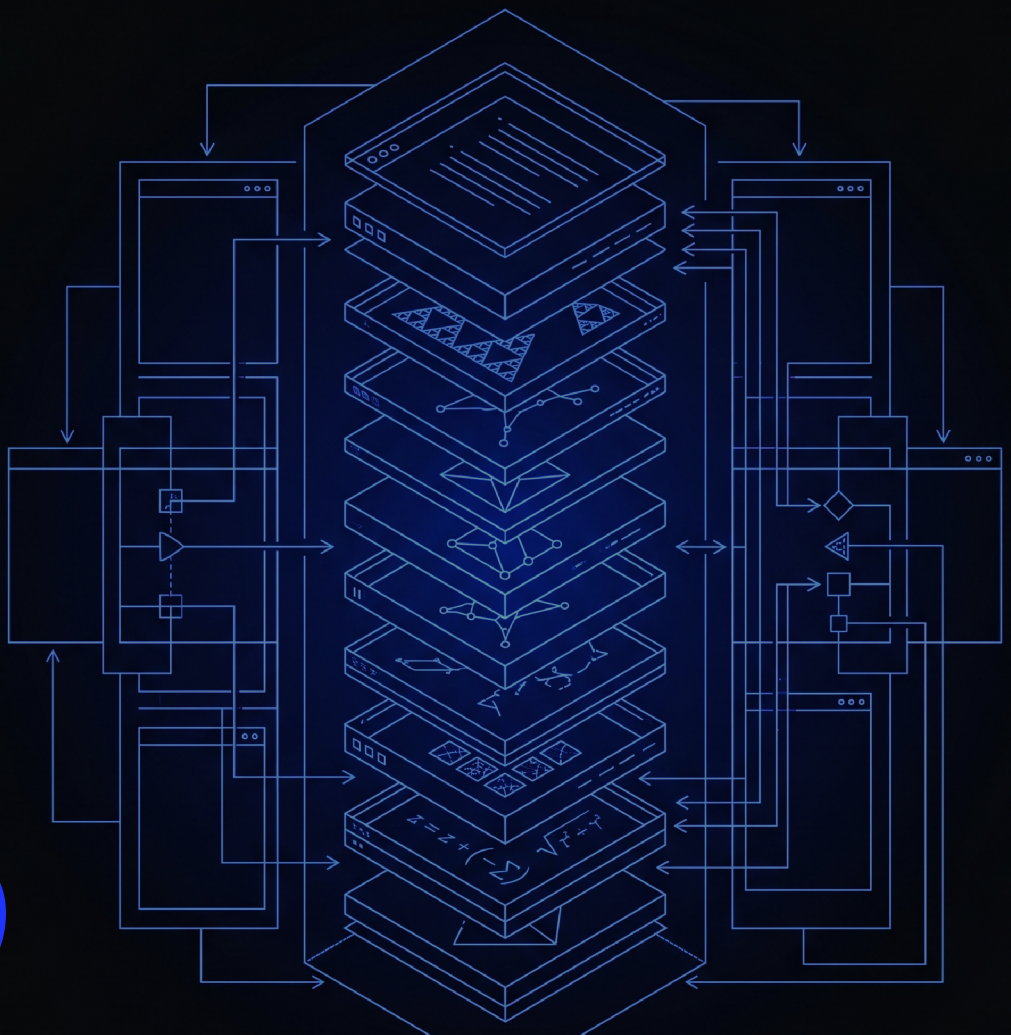
STATUS AND LEGACY

A functional platform in Beta status: the four units operational with their problem catalog, the three courses, the completeness scoring engine, the XP and badge system, the admin panel with data isolation, and the layer of interchangeable adapters. The production integration with the LMS was left as a deliberate stub, faithful to the shapes of the open educational interoperability standards, ready to connect when the production environment allowed it. The success criterion set by academic leadership was explicit: the work was approved if it brought the student to a **5** on the AP exam — the outcome, not the activity.

Pedagogical framework	**CERC** — Claim · Evidence · Reasoning · Conditions
Units · problems	**4 units · 7 / 7 / 7 / 4**
Courses covered	Calculus AB · Calculus BC · Statistics
Reasoning stages	empirical · generic · formal (data type)
Data layer	Mock (Vercel KV) ↔ the network's LMS (open educational interoperability standards)
Security	JWT (jose) · role-based middleware · CSRF · DOMPurify · rate limiting
Build	AI-first · 29 pages · interchangeable adapter layer
Status	**Beta** · the LMS as a stub faithful to the APIs

LESSONS LEARNED

- **Measurement honesty.** Resisting the temptation of the semantic AI evaluator and choosing deterministic completeness scoring: it measured exactly what it claimed to measure and attacked the right habit first —structure before eloquence.
- **Theory encoded in the types.** When the reasoning stage is a first-class data type and the units withdraw the scaffolding on a schedule, the pedagogical design stops being intention and becomes verifiable.
- **Adapter pattern.** A low-cost, high-return decision: iterating at full speed with a persistent mock without coupling to the production API, keeping its shapes intact.
- **Security as a design constraint.** On a platform with minors' data, authentication, CSRF and data isolation are not a final phase but a constraint from the first line. the AP/SAT curricular intersection



3 KNOWLEDGE INFRASTRUCTURE

The layer that grounds them all: from academic literature to research with evidence validation.