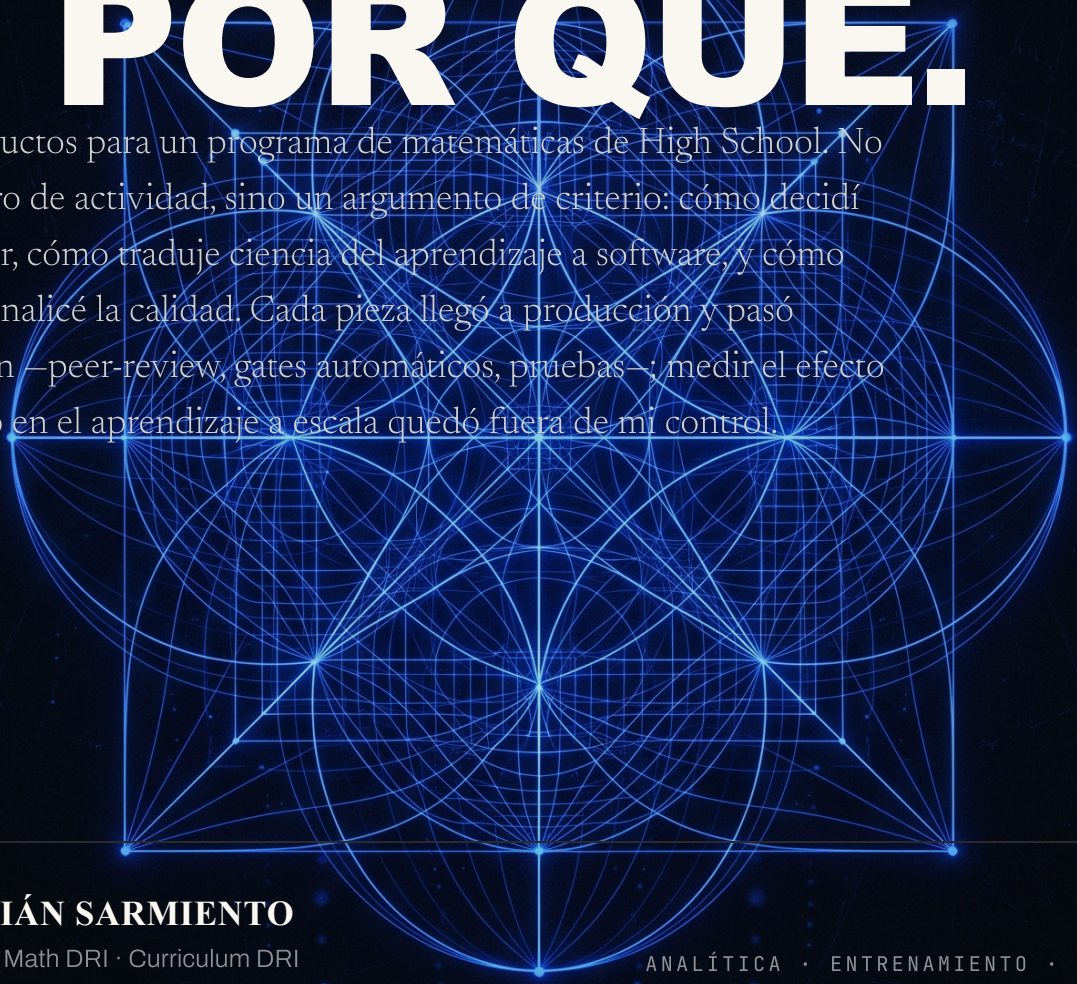
PRODUCT BOOK 2026

Criterio de ingeniería para aprendizaje

QUÉ MEDIR, Y POR QUÉ.

Seis productos para un programa de matemáticas de High School. No un registro de actividad, sino un argumento de criterio: cómo decidí qué medir, cómo traduje ciencia del aprendizaje a software, y cómo institucionalicé la calidad. Cada pieza llegó a producción y pasó validación —peer-review, gates automáticos, pruebas—; medir el efecto sostenido en el aprendizaje a escala quedó fuera de mi control.



SEBASTIÁN SARMIENTO

High School Math DRI · Curriculum DRI

DIC 2025 — JUN 2026

ANALÍTICA · ENTRENAMIENTO · EVIDENCIA

DISEÑO, LEARNING SCIENCE Y GATES DE CALIDAD

CONTENIDO

TRES SECCIONES · SEIS PRODUCTOS

1 ANALÍTICA & MONITOREO

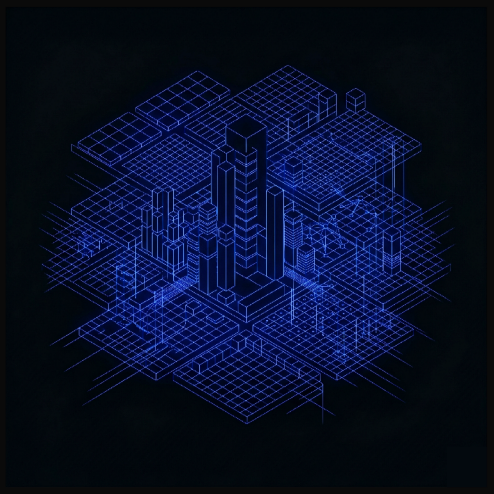
Los sistemas que dieron visibilidad: ver el riesgo de ~1.600 estudiantes antes de que se volviera irreversible.

01 · ANALÍTICA & MONITOREO

CAPÍTULO 1.1

Visibilidad donde no la había

MATH ANALYTICS COMMAND CENTER



El estado académico de ~1.600 estudiantes en una sola vista — de la inspección manual, una por una, a una decisión de intervención en menos de treinta segundos.

ESTADO	STACK	DATOS	MÉTRICAS
Producción	Next.js · Firebase	API de práctica	5 ejes de riesgo

01 EL PROBLEMA

UN SISTEMA SIN VISIBILIDAD

El desempeño de los estudiantes vivía disperso en tres plataformas y solo era consultable de a un estudiante por vez. Saber cómo iba un alumno exigía abrir su perfil, navegar a *settings* → *documentation* → *progress report* y leerlo a mano. Multiplicado por ~1.600 estudiantes, el monitoreo proactivo era simplemente inviable.

Ningún mentor podía responder *"¿cuántos de mis estudiantes están en riesgo hoy?"* sin revisar la plataforma uno por uno.

No había forma de detectar a tiempo a un estudiante estancado, adivinando, o decayendo en precisión dentro de la sesión.

Las intervenciones eran reactivas — llegaban cuando un guía reportaba el problema, no antes.

No existía un lenguaje común: "ir bien" o "ir mal" eran juicios cualitativos, nunca medibles ni comparables. Cada semana de estancamiento no detectado era progreso perdido que no se recuperaba dentro de la ventana de la sesión.

02 EL OBJETIVO

UNA DECLARACIÓN EVALUABLE

OBJETIVO PRIMARIO

Dar a cada mentor visibilidad en tiempo real del riesgo académico de sus estudiantes en **menos de 30 segundos**, reemplazando la inspección manual una por una por una vista consolidada y accionable.

Como objetivos secundarios: convertir señales crudas de la API en métricas pedagógicas interpretables (velocidad, deuda de conocimiento, decaimiento de precisión, estabilidad); priorizar automáticamente —que el sistema diga *a quién atender primero*, no solo quién existe—; habilitar outreach accionable desde la misma herramienta; y operar como monitoreo proactivo sin añadir carga manual de reporte al mentor.

El éxito se definió de antemano: responder “¿quién está en riesgo hoy?” en menos de 30 segundos, con cobertura del 100% de los estudiantes en una sola vista y latencia resuelta por sync periódico automático.

03 EL ARTEFACTO

LA VISTA TRIAGE

El uso dominante es de **rutina matinal**: el mentor abre Triage al empezar el día, lee las zonas roja/amarilla/verde y decide a quién contactar antes de la primera sesión. El Risk Score ordena la cohorte completa por urgencia; el color resuelve la decisión de un vistazo.

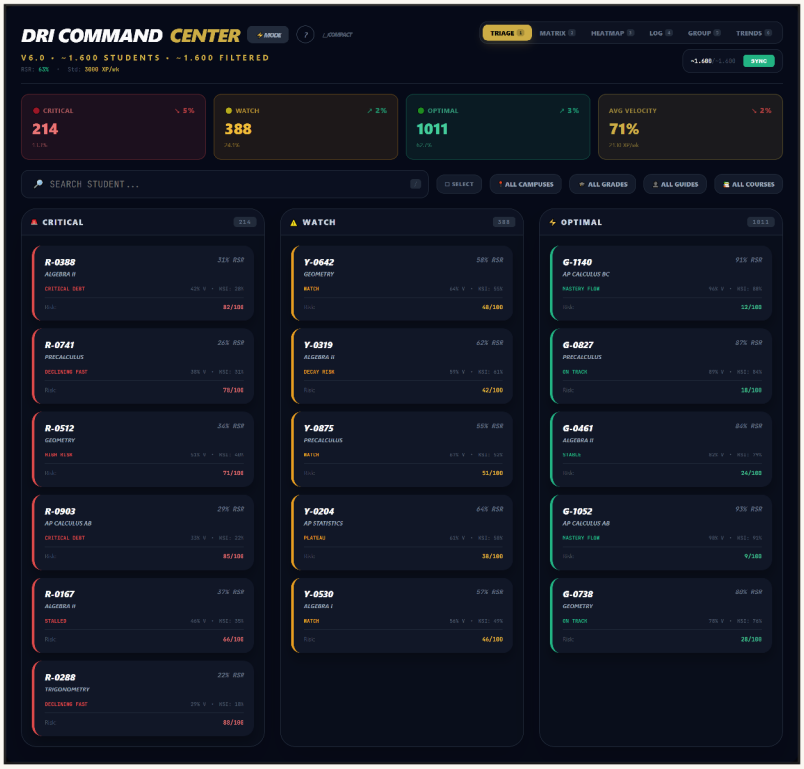


FIG. 1 — VISTA TRIAGE · ~1.600 ESTUDIANTES REPARTIDOS EN TRES ZONAS —CRÍTICO / ATENCIÓN / ÓPTIMO— Y ORDENADOS POR RISK SCORE DENTRO DE CADA UNA · EL COLOR DECIDE A QUIÉN ATENDER PRIMERO

04 CUATRO VISTAS, CUATRO PREGUNTAS

EL MISMO DATO, LEÍDO DE CUATRO FORMAS

Triage resuelve “¿a quién atiendo ya?”. Las otras vistas responden preguntas distintas sobre la misma cohorte: **Matrix** agrupa por patrón de riesgo —¿qué *tipos* de estudiante tengo?—; **Heatmap** muestra los temas más críticos a nivel cohorte —¿qué *contenido* le cuesta a todos?—.

FIG. 2 — VISTA MATRIX · LA COHORTE POSICIONADA EN EL ESPACIO MASTERY (RSR) × STABILITY (KSI), CON EL CUADRANTE SUPERIOR-DERECHO COMO FLUJO DE DOMINIO Y EL INFERIOR-IZQUIERDO COMO DEUDA CRÍTICA

FIG. 3 — VISTA HEATMAP · LOS 15 COMPONENTES DE CONOCIMIENTO MÁS CRÍTICOS CRUZADOS CON CADA CURSO, PRIORIZADOS POR RSR PROMEDIO BAJO

05 EL FUNDAMENTO

LEARNING SCIENCE, VUELTA MÉTRICA

El dashboard no inventa indicadores arbitrarios: operacionaliza principios de ciencia del aprendizaje en cinco métricas medibles, derivadas de un protocolo de cálculo propio documentado aparte.

PRINCIPIO	MÉTRICA	QUÉ CAPTURA
Mastery learning	**Velocity**	Ritmo del estudiante frente al estándar de 125 XP por semana.
Deuda de conocimiento	**DER**	Porcentaje de temas de K-8 que un estudiante de HS está re-masterizando — vacíos de base que arrastra.
Fatiga cognitiva	**PDI**	Caída de precisión entre el inicio y el final de la sesión.
Retención / estabilidad	**KSI**	Estabilidad del conocimiento dominado a lo largo del tiempo.
Éxito reciente	**RSR**	Precisión sostenida en las últimas diez tareas.

> *Si los mentores ven el riesgo en tiempo real, intervienen de forma preventiva en vez de reactiva — y el estudiante pasa menos tiempo estancado dentro de la ventana de la sesión.*

Para que ninguna métrica simplificara de más, el riesgo se calcula como un **score ponderado de cinco dimensiones**, no como un solo número. Y para no distorsionar la motivación, las métricas son instrumentos del mentor — nunca un ranking público del estudiante.

06 EL DISEÑO

ARQUITECTURA Y LA DECISIÓN QUE MÁS IMPORTÓ

Un wrapper propio sobre la API de la plataforma de práctica maneja sync incremental con tolerancia a fallos y alimenta Firestore; el dashboard lee de ahí en vivo, y el generador de mensajes empuja el outreach a Slack. Stack: Next.js sobre TypeScript, Firebase/Firestore, Vercel.

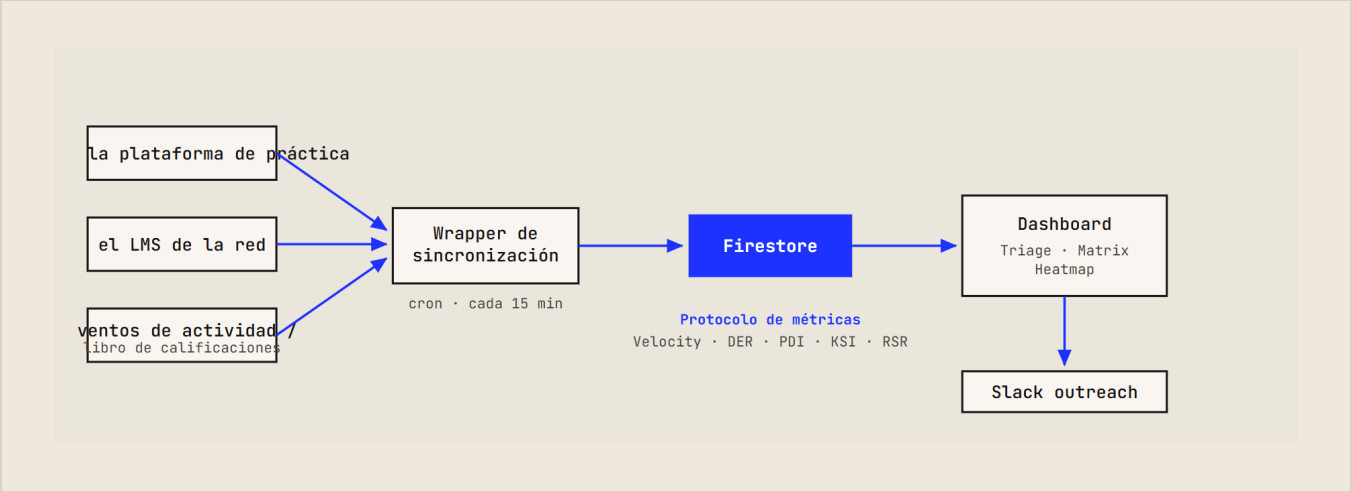


FIG. 4 — FLUJO DE DATOS: FUENTES → WRAPPER DE SYNC → FIRESTORE → DASHBOARD → OUTREACH

Riesgo como score ponderado de cinco factores — no una métrica única

Contexto. Un solo número producía falsos positivos (estudiantes lentos pero sólidos marcados en rojo) y falsos negativos (rápidos pero adivinando, marcados en verde).

Decisión. Risk Score = Debt Exposure 30% · Velocity 25% · Precision Decay 20% · Stability 15% · Stall 10%. Umbrales: ≥60 **ROJO** · ≥35 **AMARILLO** · <35 **VERDE**.

Trade-off aceptado. Más difícil de explicar en una línea, pero mucho más fiel a la realidad del estudiante — y la fidelidad es lo que hace que un mentor confíe en el color.

07 CONSTRUCCIÓN Y VALIDACIÓN

PROCESO AI-FIRST, SOMETIDO A DATOS REALES

El desarrollo operó con la IA como co-desarrollador principal —arquitectura, implementación y refactors—, precedido por un notebook propio que exploró la API antes de construir el wrapper de producción. El patrón era constante: **la decisión de qué medir y por qué era siempre propia**; la IA aceleraba el *cómo construirlo*. Construido en tres semanas y media de trabajo sostenido, con umbrales centralizados en un único archivo de configuración para recalibrar sin tocar la lógica.

El dashboard se conectó a la población completa —~1.600 estudiantes— y emitió **digests automáticos diarios** de cambios de tier (un día reportó 79 cambios; otro, 43), confirmando el pipeline en vivo. Se presentó ante el

equipo de plataforma de datos en una sesión de *data insights*, y la calibración de los cinco factores se ajustó contra el comportamiento observado de la cohorte.

08 EL RESULTADO

ESTADO Y LEGADO

En producción, conectado a la población real, con digests automáticos a diario. El producto no quedó como demo: se volvió la capa de visibilidad sobre la que se diseñó el siguiente producto del portafolio — el Student Performance Tracker Cap. 02, que añade detección de patrones e intervención automática.

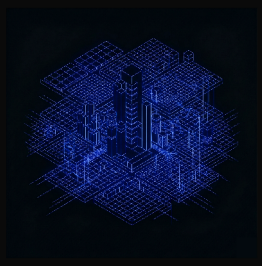
Estudiantes en cobertura	**~1.600**
Construcción	AI-first · *qué medir* = decisión propia · ritmo sostenido (~3,5 sem)
Integraciones	API de práctica · Firebase · Slack · Vercel
Vistas	Triage · Matrix · Heatmap · Log · Student Modal
Estado	**En producción**

APRENDIZAJES

- **Técnico.** El cuello de botella de un dashboard en vivo no es la interfaz — es el wrapper robusto sobre la API de terceros: sync incremental, reintentos, tolerancia a fallos.
- **De producto.** Una métrica compuesta y bien ponderada le gana a cinco métricas sueltas. El usuario no quiere cinco números; quiere saber a quién atender primero.
- **Pedagógico.** Operacionalizar la ciencia del aprendizaje en métricas hace accionable lo que antes era intuición — y deja una huella de decisión trazable.

De ver el riesgo a actuar sobre él

STUDENT PERFORMANCE TRACKER



El Command Center mostró quién estaba en riesgo. El Tracker decide a quién contactar hoy, por qué, y deja el borrador listo — la intervención temprana convertida en rutina automática.

ESTADO	STACK	MODELO	BASE
Producción	Next.js · Firebase · Slack	6 triggers	reusa Command Center

01 EL PROBLEMA

LA VISIBILIDAD NO ERA ACCIÓN

El Command Center Cap. 01 resolvió el *ver*: en una sola vista, el riesgo de toda la cohorte ordenado por urgencia. Pero entre ver el rojo y hacer algo seguía habiendo un abismo manual. Cada intervención exigía que el mentor abriera el perfil del estudiante, leyera el historial, decidiera el mensaje y lo escribiera a mano — y lo repitiera, alumno por alumno, cada mañana.

La detección de riesgo era un retrato estático: decía *quién* estaba mal hoy, no *qué patrón* lo había llevado ahí ni si ya se había intervenido antes.

No quedaba memoria de la intervención. Un mentor podía contactar dos veces por lo mismo, u olvidar dar seguimiento a un caso abierto la semana anterior.

Priorizar bajo presión de tiempo se hacía a ojo: con decenas de estudiantes en amarillo, ¿a cuáles cinco escribir antes de la primera sesión?

Y el outreach —el paso que de verdad cambia la trayectoria del estudiante— era justamente el que más fricción tenía. El cuello de botella ya no era la información; era el trabajo manual de convertirla en un mensaje enviado.

02 EL OBJETIVO

AUTOMATIZAR DETECCIÓN Y OUTREACH

OBJETIVO PRIMARIO

Pasar de la visibilidad pasiva a la **intervención temprana automática**: que el sistema detecte patrones de riesgo, priorice a los estudiantes que más necesitan contacto hoy, y deje el outreach a un clic de distancia.

Como objetivos secundarios: descomponer el riesgo en **patrones nombrables** —no un score opaco, sino seis disparadores concretos que un mentor reconoce—; mantener un *historial de intervención* por estudiante para que ninguna acción se duplique ni se pierda; reducir la rutina matinal a un único artefacto leíble en minutos; y redactar el primer borrador del mensaje de outreach de forma automática, dejando al mentor el juicio final.

El éxito se definió de antemano: que cada mañana el mentor abra una sola vista, vea a los cinco estudiantes prioritarios con el porqué explícito, y pueda disparar el contacto sin volver a escribir desde cero.

03 EL ARTEFACTO

EL DASHBOARD “MORNING COFFEE”

El producto se diseñó alrededor de un único momento de uso: el café de la mañana. El mentor abre **Morning Coffee**, ve a los **cinco estudiantes prioritarios** del día con el *trigger* que disparó cada caso, un **Quality Score** que mide la salud reciente del estudiante, y un **borrador de Slack** ya redactado para enviar. Lo que antes eran decenas de perfiles abiertos a mano queda resuelto en una pantalla.

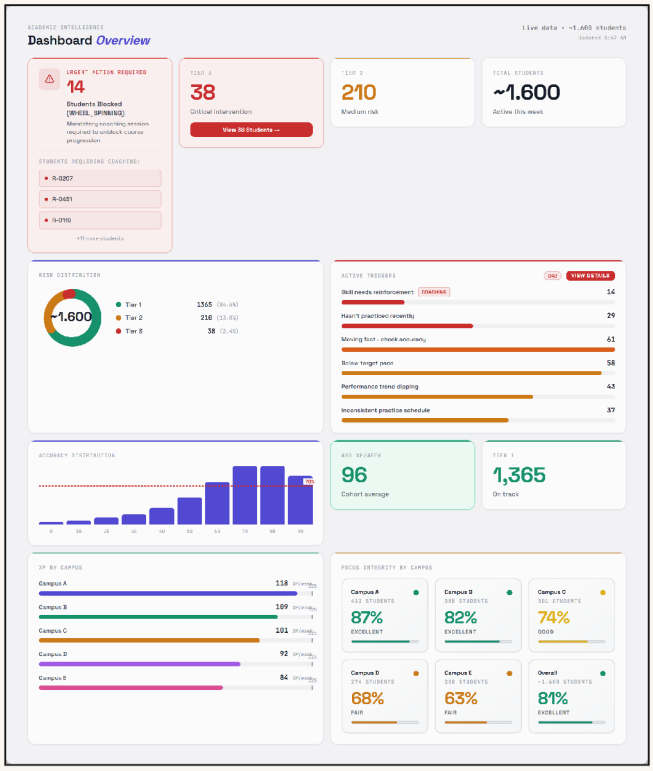


FIG. 1 — DASHBOARD “MORNING COFFEE” · TOP-5 PRIORITARIOS + TRIGGER + QUALITY SCORE + BORRADOR DE SLACK · LA RUTINA MATINAL EN UNA SOLA VISTA

04 LA FICHA DE ESTUDIANTE

MEMORIA DE LA INTERVENCIÓN

Si Morning Coffee responde “¿a quién atiendo ya?”, la ficha de estudiante responde “¿qué le ha pasado y qué hemos hecho?”. Cada estudiante — identificado solo por su código `R-/Y-/G-`, sin datos personales— tiene un **timeline de intervención**: qué trigger se activó, qué outreach se envió, qué pasó después. La memoria que antes vivía en la cabeza del mentor queda registrada y consultable.

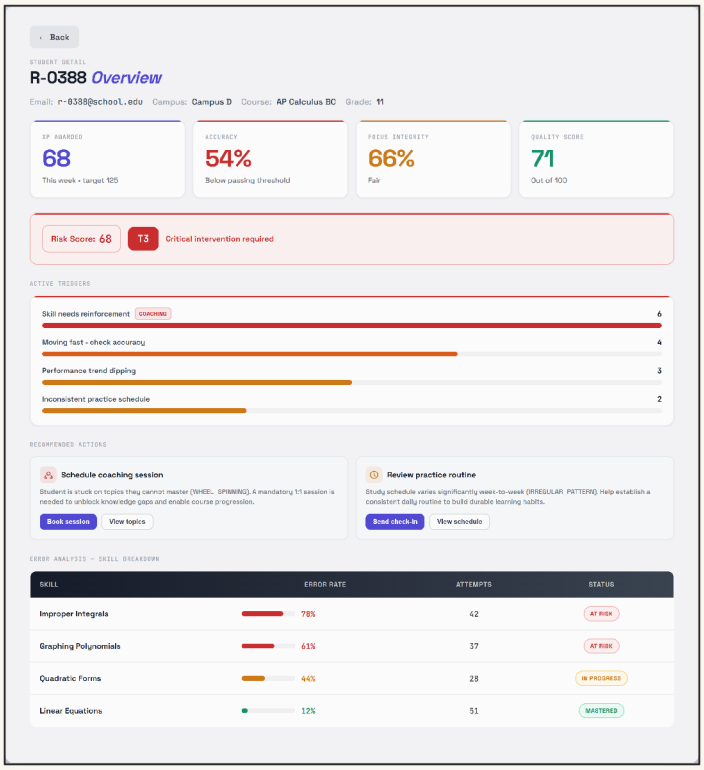


FIG. 2 — FICHA DE ESTUDIANTE · HISTORIAL DE INTERVENCIÓN EN TIMELINE · NINGÚN CONTACTO SE DUPLICA NI SE PIERDE

05 EL MODELO

SEIS TRIGGERS, UN RISK SCORE, UN TIER DE OUTREACH

El corazón del Tracker es un modelo de detección por patrones. Seis *triggers* independientes vigilan señales distintas de la actividad de cada estudiante; cuando uno o varios se activan, alimentan un **Risk Score** que clasifica al estudiante en un **tier**, y el tier decide la forma del **outreach** — desde un toque ligero hasta una alerta de contacto inmediato.

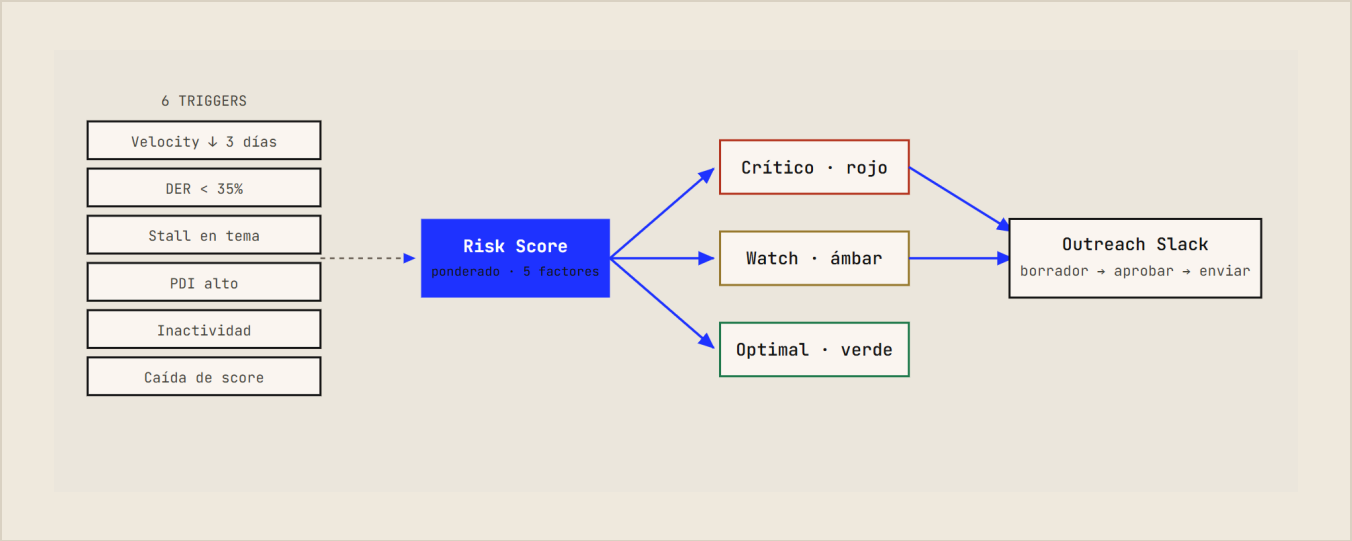


FIG. 3 — FLUJO DEL MODELO: SEIS TRIGGERS → RISK SCORE → TIER → OUTREACH

TRIGGER	QUÉ PATRÓN DETECTA
Stall	El estudiante deja de avanzar — sin progreso medible durante varios días seguidos.
Velocity drop	El ritmo de avance cae por debajo del estándar esperado para su cohorte.
Precision decay	La precisión se desploma dentro de la sesión — señal de fatiga o de adivinar.
Guessing	Patrón de respuestas rápidas y erráticas que delata respuestas al azar.
Knowledge debt	Acumulación de temas de base sin dominar que el estudiante arrastra.
Disengagement	Caída sostenida de actividad — menos tiempo y menos tareas que su línea base.

> *No es un número opaco: cada caso del top-5 llega con el trigger que lo disparó, de modo que el mentor entiende el porqué antes de escribir la primera palabra.*

Los seis triggers son la estructura fija del modelo; los umbrales exactos de cada uno se calibran contra el comportamiento observado de la cohorte y viven centralizados para recalibrar sin tocar la lógica de detección.

06 CONSTRUCCIÓN

PROCESO AI-FIRST SOBRE UNA BASE EXISTENTE

El Tracker no partió de cero: se construyó **sobre** la capa de visibilidad del Command Center, reusando su pipeline de sync y su Firestore. El desarrollo operó con la IA como co-desarrollador principal —arquitectura del modelo de triggers, implementación, refactors—, mientras la decisión de **qué señales medir y cómo traducirlas en outreach** fue siempre propia. Llevado hasta producción sobre esa base, con la integración a Slack para empujar los borradores de mensaje directamente al canal del mentor.

El generador de borradores fue la pieza que cerró el ciclo: tomar el trigger y el contexto del estudiante y producir un primer mensaje editable resolvió la fricción que el Command Center había dejado abierta. El mentor conserva el juicio final —edita y envía—, pero ya no arranca desde una página en blanco.

07 EL RESULTADO

ESTADO Y LEGADO

En producción, conectado a la cohorte real, emitiendo cada mañana su top-5 prioritario con borradores listos. El Tracker completó el arco que el Command Center había empezado: de *ver* el riesgo a *actuar* sobre él sin trabajo manual repetido.

Modelo de detección	`6 triggers → Risk Score → tier`
Construcción	AI-first sobre base existente · reuso de sync + Firestore
Integraciones	API de práctica · Firebase · Slack · Vercel
Artefactos	Morning Coffee · Ficha de estudiante · Borrador de outreach
Estado	**En producción**

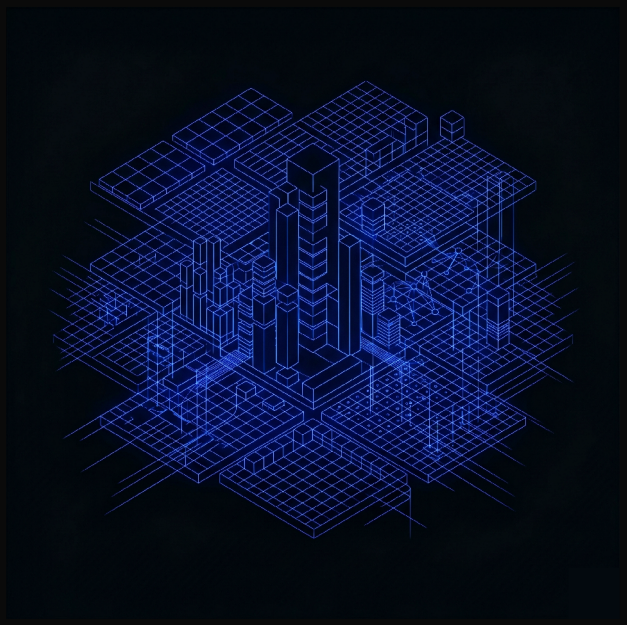
APRENDIZAJES

- **De producto.** La visibilidad no es valor hasta que se vuelve acción. El salto del Command Center al Tracker fue cerrar el último metro: del riesgo visible al mensaje enviado.
- **De diseño.** Un patrón nombrable —seis triggers que el mentor reconoce— construye más confianza que un score perfecto pero opaco. El porqué importa tanto como el a-quién.
- **De proceso.** Construir sobre un producto previo del portafolio — reusando su sync y su Firestore— permitió llegar a producción reusando infraestructura en lugar de empezar de nuevo.

Solo lo que cambia la decisión del día

STUDENT ACTIVITY MONITOR

Un bot que vigila el ritmo de aprendizaje de un estudiante cada cinco minutos y avisa por Telegram solo cuando vale la pena mirar.



ESTADO	STACK	DATOS	ENTREGA
Producción	Node.js 18 · 0 deps	API de la plataforma · Telegram	Telegram · baja latencia

01 EL PROBLEMA

EL DATO EXISTÍA; NADIE LO VEÍA A TIEMPO

El seguimiento de un estudiante en una plataforma de práctica adaptativa era, por defecto, una revisión manual alumno por alumno: alguien tenía que entrar al panel, abrir el perfil, leer el XP del día, hojear los temas trabajados y deducir si el ritmo iba bien o se estaba estancando.

Para una estudiante de Middle School con brechas conceptuales acumuladas, ese seguimiento no podía ser semanal ni depender de que alguien se acordara de mirar.

El problema concreto era de *latencia* y de *atención*: cuando una sesión se torcía —baja precisión repetida en un mismo tema, poco avance a media tarde, abandono temprano— el dato existía en la API, pero nadie lo veía a tiempo para intervenir el mismo día.

Revisar la plataforma cada cinco minutos a mano era inviable.

Y recibir todo el detalle bruto habría sido ruido inservible. El reto no era acceder al dato, sino destilar de él solo las señales que merecen una acción.

02 EL OBJETIVO

UNA DEFINICIÓN DE ÉXITO EN SEÑAL-RUIDO

OBJETIVO PRIMARIO

Cerrar la distancia entre el dato y la intervención: detectar señales accionables en tiempo casi real y empujar solo esas señales —no el ruido— a un canal donde quien acompaña al estudiante las vea sin esfuerzo.

Definí el éxito de antemano y en términos de señal-ruido: un día de monitoreo que produjera entre **seis y diez mensajes con sentido** —un briefing al inicio, hitos de avance, alertas críticas, cierre de sesión— en lugar de los quince a veinte avisos triviales que generaría un sistema ingenuo, o del silencio total de la revisión manual.

El sistema tenía que correr solo, sin que nadie lo encendiera, dentro del horario lectivo.

03 LOS USUARIOS

LA ALERTA LLEGA DONDE LA PERSONA YA ESTÁ

El usuario directo es la persona que acompaña el aprendizaje del estudiante: necesita saber, sin abrir ningún panel, si la sesión de hoy va encaminada y dónde hay que meter mano.

El sujeto monitoreado es una estudiante de Middle School con brechas previas —anonimizada como **R-0388** en este capítulo—, cuyo plan de recuperación dependía de un acompañamiento diario y cercano.

El canal elegido fue **Telegram** precisamente porque vive en el teléfono de quien acompaña: las alertas llegan donde la persona ya está, no a un dashboard que hay que recordar visitar.

Más tarde el mismo motor pasó a seguir a varios estudiantes en paralelo, cada uno con su propia meta diaria de XP — sin tocar la lógica central.

Esa elección de canal no es cosmética: define el contrato del producto. Si la notificación interrumpe, más vale que valga la interrupción.

04 EL ARTEFACTO

UN DÍA COMPLETO EN EL FEED DE TELEGRAM

El producto es, para quien lo usa, una conversación. El bot abre la jornada con un **briefing matinal** que sitúa el día contra la semana, va marcando los hitos de meta al cruzarse, dispara una alerta roja cuando un tema se atasca de verdad, y cierra con el resumen de sesión. Entre seis y ocho burbujas que cuentan el día sin que nadie abra un panel.

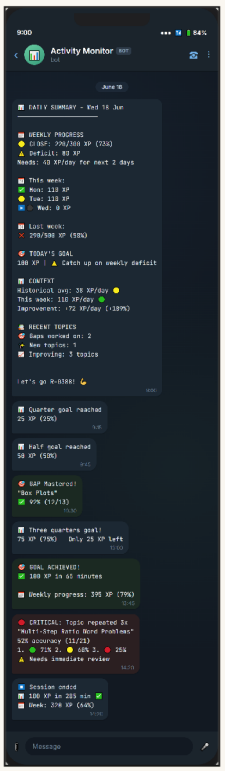


FIG. 1 — HILO DE TELEGRAM DE UN DÍA DE R-0388 · BRIEFING, HITOS DE XP, ALERTA CRÍTICA Y CIERRE · EL DETALLE BRUTO DESTILADO EN SEÑALES ACCIONABLES

05 EL FUNDAMENTO

UNA ALERTA QUE LLEGA SIEMPRE DEJA DE SER UNA ALERTA

El diseño de las notificaciones está anclado en un principio de carga cognitiva y de relevancia. El sistema no reporta avance continuo sino **hitos significativos** —cuartos de meta a 25, 50, 75 y 100 por ciento de XP— que corresponden a momentos en que el progreso cambia de fase, no a incrementos arbitrarios.

PRINCIPIO	SEÑAL	CUÁNDO DISPARA
Relevancia / carga cognitiva	`Hito de meta`	Al cruzar 25, 50, 75 y 100% del XP del día — fases del progreso, no incrementos.
Práctica deliberada	`Tema crítico`	Mismo tema repetido en el día con precisión bajo 60% sobre ≥8 preguntas — el patrón, no el error aislado.
Fallo profundo	`Alerta dura`	Precisión bajo 30% en un tema, o retroceso marcado frente al historial.
Contexto temporal	`Briefing`	Inicio del día situado contra la tendencia semanal y el promedio histórico.
Cierre de ciclo	`Fin de sesión`	Resumen al detectar 30 min de inactividad dentro del horario lectivo.

> *La detección de temas problemáticos se apoya en la práctica deliberada: lo que importa no es un error aislado, sino el patrón que se repite con precisión baja sostenida.*

El briefing matinal sitúa el día de hoy contra la tendencia semanal y el promedio histórico, porque una métrica sin contexto no permite decidir: **29 XP significan cosas distintas** según si la semana viene atrasada o adelantada.

06 EL DISEÑO

ARQUITECTURA AUSTERA Y LAS TRES DECISIONES QUE LA DEFINEN

La arquitectura es deliberadamente austera. El monitor es un único proceso Node.js **sin una sola dependencia externa** —solo módulos nativos— que en cada ejecución consulta la API de la plataforma, compara la actividad recién leída contra un estado persistido en disco y deriva de esa diferencia qué notificaciones corresponden. El estado vive en archivos JSON, lo que mantiene el sistema sin base de datos y trivialmente inspeccionable.

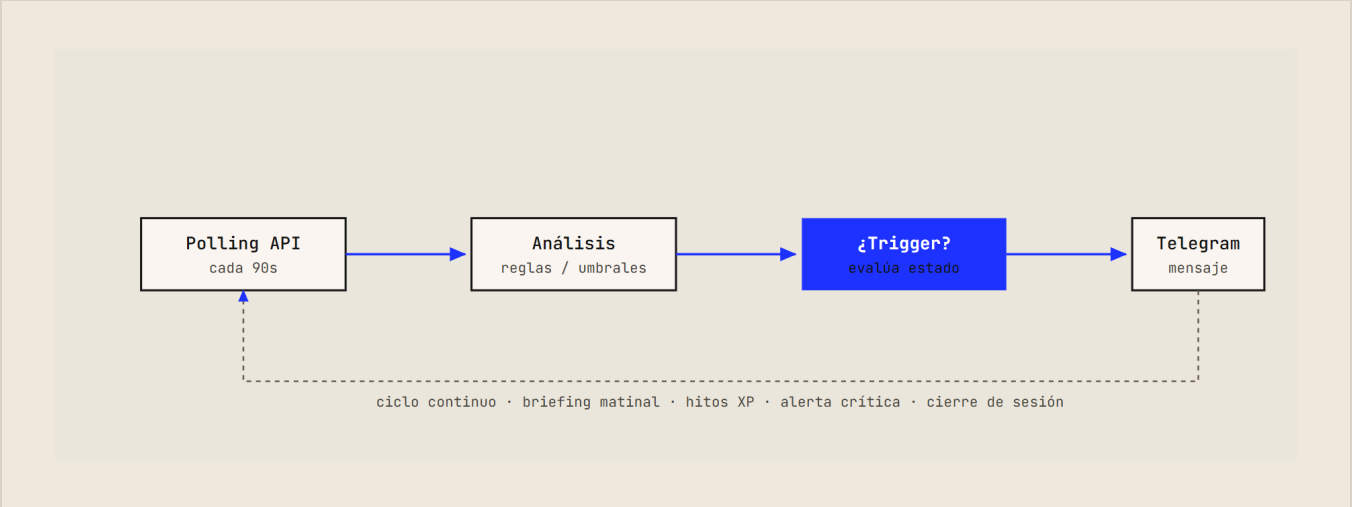


FIG. 2 — EL CICLO CADA 5 MIN: CRON-RUNNER → POLLING DE LA API → DIFF CONTRA ESTADO → ANÁLISIS → ALERTA A TELEGRAM

Tres decisiones de arquitectura que definen el producto

Separar scheduler de monitor. Un cron-runner mantiene un servidor HTTP con endpoint de salud y dispara el monitor cada cinco minutos; ese *health check* es lo que impide que la plataforma de hosting duerma el servicio por inactividad.

Notificaciones idempotentes por día. Banderas en el estado evitan el spam estructuralmente —un hito enviado no se reenvía, una alerta crítica no se repite— en vez de a fuerza de filtros frágiles.

Encajar en el horario lectivo. Monitoreo entre 9:00 y 16:10 hora de Campus B, con cómputo propio de la transición EST/EDT, para no generar ruido fuera de clase. La red usa reintentos con backoff exponencial que distingue errores de cliente —que no se reintentan— de errores de servidor.

07 CONSTRUCCIÓN

DOCE COMMITS, DOS APRENDIZAJES QUE DOLIERON

Construí el monitor en un flujo AI-first, apoyándome en Claude para iterar rápido sobre la lógica de notificaciones y el manejo de la API. El cuerpo del trabajo se concentró en **doce commits entre el 12 de marzo y el 21 de mayo de 2026**. Los umbrales y triggers viven centralizados, de modo que recalibrar la relevancia no exige tocar la lógica.

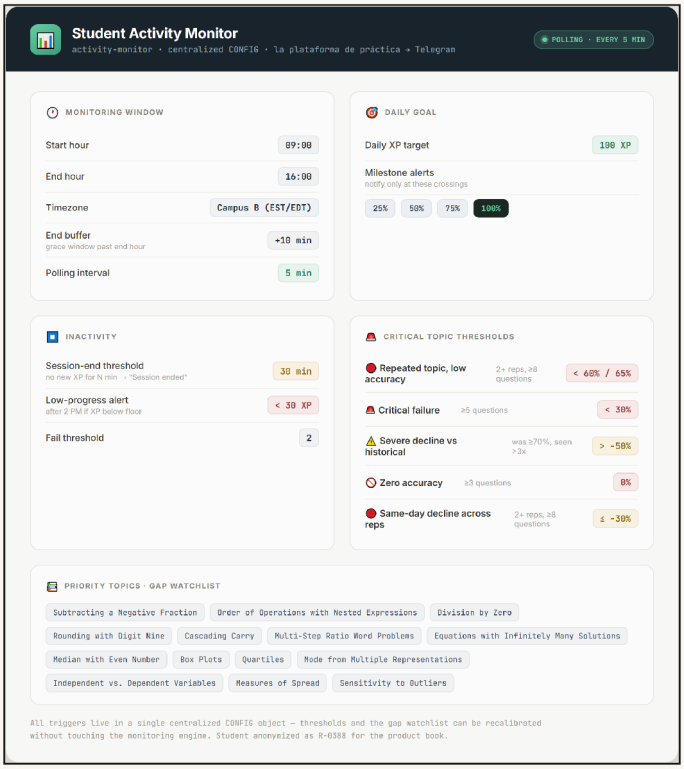


FIG. 3 — CONFIGURACIÓN DE TRIGGERS · HITOS, UMBRALES DE TEMA CRÍTICO Y VENTANA LECTIVA CENTRALIZADOS PARA RECALIBRAR SIN TOCAR EL MOTOR

El primer obstáculo real fue la propia API: las peticiones devolvían error 400 de forma persistente hasta que identifiqué que faltaban las cabeceras *Accept* y *Content-Type* y que la URL base debía apuntar directamente al prefijo de la versión de la plataforma de práctica; reutilicé la configuración correcta ya validada en un proyecto de tracking anterior. El segundo aprendizaje vino del estado persistido: al migrar el servicio a la nube heredó un estado local con todas las banderas ya marcadas, de modo que el monitor corría perfecto pero generaba **cero notificaciones** porque creía que ya las había enviado. Eso me obligó a tratar el estado como ciudadano de primera clase, con reset limpio por instancia. El sistema de notificaciones llegó a una versión 4.0 que redujo el volumen de quince-veinte avisos a seis-ocho mensajes con contexto.

08 VALIDACIÓN

EN PRODUCCIÓN, CONTRA LOS CASOS DE BORDE

Validé el sistema en producción real, no en un entorno de prueba. El registro de actividad muestra el ciclo funcionando de extremo a extremo: consulta a la API, lectura de XP y preguntas del día, comparación con el estado y envío efectivo del mensaje por Telegram. Probé explícitamente los casos de borde que rompen este tipo de bots —ejecución fuera de horario, detección de patrones críticos, días sin actividad, inconsistencias de la API que devuelve totales de distintos timestamps— con una batería de scripts dedicados antes de confiar en el flujo automático.

La medida de éxito que me importaba no era técnica sino de **señal-ruido**: que un día completo produjera la cantidad correcta de mensajes accionables. El sistema demostró sostener el ciclo de cinco minutos dentro del horario lectivo y entregar el briefing matinal, los hitos de XP, las alertas de tema crítico y el cierre de sesión por inactividad como mensajes diferenciados y oportunos.

09 EL RESULTADO Y EL LEGADO

DE UN ESTUDIANTE A VARIOS, SIN TOCAR EL MOTOR

Un monitor de aprendizaje individual en producción que convierte la API de una plataforma de práctica adaptativa en un acompañamiento de baja latencia y alta relevancia. Quien acompaña deja de revisar paneles a mano y recibe en su teléfono solo lo que cambia la decisión del día.

Estudiante monitoreado	`R-0388 · Middle School`
Construcción	AI-first · sin dependencias externas · umbrales centralizados
Integraciones	la plataforma de práctica API de la plataforma · Telegram Bot API
Volumen de señal	6-8 mensajes/día (desde 15-20)
Despliegue	Railway, migrado a Render
Estado	**En producción**

El mismo motor escaló de un estudiante a varios en paralelo —cada uno con su meta diaria, su archivo de estado, historial y log— sin tocar la lógica central, lo que confirmó que la arquitectura sin dependencias y orientada a

estado por estudiante era la decisión correcta. El servicio sobrevivió a un cambio de plataforma de hosting conservando el health check que lo mantiene despierto. Se apoya en la misma capa de visibilidad que sostiene el portafolio de analítica Cap. 01 | Cap. 02.

APRENDIZAJES

- **De producto.** El valor de un sistema de alertas está en lo que decide NO enviar. Diseñar la relevancia —hitos en vez de incrementos, patrones en vez de errores aislados, contexto semanal en vez de cifras sueltas— fue más difícil y más importante que conectar la API.
- **Técnico.** En un monitor stateful, el estado persistido es parte del diseño, no un detalle de implementación: el bug de las banderas heredadas —un sistema que corría perfecto produciendo cero salida— mostró que ahí viven los errores silenciosos.
- **De arquitectura.** El rendimiento de la simplicidad deliberada: cero dependencias, archivos JSON y un health check bastaron para un servicio que corre solo todos los días lectivos — y esa austeridad fue lo que hizo trivial escalar de uno a varios estudiantes.

2 PLATAFORMAS DE ENTRENAMIENTO

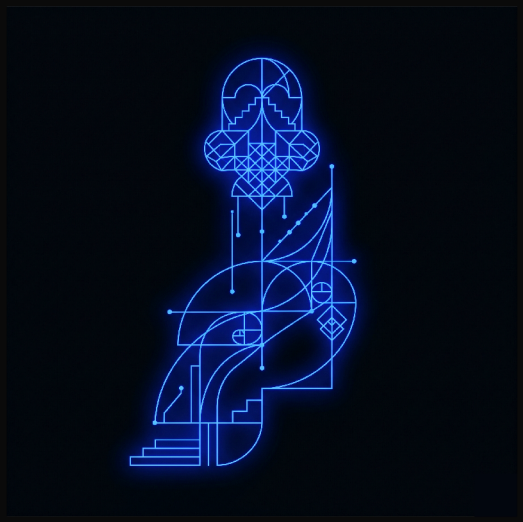
Los instrumentos que producen cambio: atacar los saltos de dificultad más caros, del 650→800 del SAT al 3→5 del AP.

02 · ENTRENAMIENTO CAPÍTULO 2.1

Del minuto de álgebra al clic de veinte segundos

DESMOS SAT TRAINING PLATFORM

Convertir la calculadora gráfica del Digital SAT en una destreza entrenable — el salto de 650 a 800 no era de matemáticas, era de fluidez con la herramienta.



ESTADO	STACK	DATOS / IA	CALIDAD
Producción · acceso a 14	React · Vite · Firebase	Firestore · AWS Bedrock	9 gates de build

01 EL PROBLEMA

UNA BRECHA DE FLUIDEZ, NO DE MAESTRÍA

La analítica de cohortes mostraba un patrón nítido: los estudiantes que rondaban los 650 puntos en la sección de Matemáticas del Digital SAT no fallaban por falta de contenido. Dominaban Heart of Algebra y Passport to Advanced Math —cerca del 70% del examen— en sus cursos regulares. Lo que les costaba el salto hacia 800 era operativo: no sabían usar la calculadora gráfica Desmos que el propio examen integra.

Un problema que un estudiante experto resuelve en veinte segundos de clic se les iba en cuatro minutos de álgebra manual, con el riesgo de error que eso arrastra bajo presión de tiempo.

El currículo de matemáticas que usaban enseñaba a resolver, no a usar la herramienta que el examen pone a disposición.

Era una brecha de fluidez estructural, no de maestría matemática.

Y nadie la estaba entrenando de forma sistemática — el concepto ya estaba, faltaba automatizar el procedimiento con la herramienta hasta liberar memoria de trabajo.

02 EL OBJETIVO

INSTALAR UN REFLEJO, NO ENSEÑAR MATEMÁTICAS

OBJETIVO PRIMARIO

Cerrar la brecha de fluidez: enseñar al estudiante a reconocer, frente a un problema tipo SAT, qué jugada de Desmos lo resuelve más rápido, y a ejecutarla sin titubeos de sintaxis.

No se buscaba enseñar matemáticas nuevas, sino instalar un reflejo: ante una ecuación lineal, graficar y leer la raíz; ante un sistema, buscar la intersección; ante una nube de datos, escribir una regresión de una línea.

La meta de producto se definió de antemano: que un estudiante que entra a la plataforma con la matemática ya sabida salga capaz de operar la calculadora del examen real con **velocidad de experto**.

03 LOS USUARIOS

ESTUDIANTES EN LA FRANJA 650–800

Los usuarios primarios son estudiantes de secundaria de **una red de colegios K-12** de aprendizaje personalizado acelerado por IA, que preparan el Digital SAT y se ubican en la franja de 650 a 800. La versión en producción dio acceso a **14 estudiantes reales**; no se midió outcome de aprendizaje. Para anonimizar el seguimiento, cada uno se identifica por código —por ejemplo R-0388—, nunca por nombre.

Los usuarios secundarios son instructores y administradores: a través de un dashboard observan el progreso de cada estudiante, las tasas de completitud y los reportes de error que los propios alumnos envían desde cada pregunta. El control de acceso por roles —estudiante, admin, superadmin— mantiene aislados los datos de cada quien.

04 EL FUNDAMENTO

CONOCIMIENTO CONCEPTUAL VS. FLUIDEZ PROCEDIMENTAL

El diseño parte de una distinción de ciencia del aprendizaje: la diferencia entre conocimiento conceptual y fluidez procedimental. El estudiante de la franja alta ya tiene el concepto; lo que le falta es automatizar el procedimiento con la herramienta hasta liberar memoria de trabajo.

Por eso la unidad de entrada no es matemática sino motriz —**Unit 0, fluidez de tecleo**— para construir memoria muscular sobre fracciones, exponentes, subíndices y la tilde de regresión antes de cargar con contenido. Sobre esa base se aplicó *fading* de andamiaje, el retiro gradual de apoyos a medida que el desempeño sube: cada patrón de tecleo atraviesa tres niveles —andamio con la sintaxis literal a la vista, recordatorio colapsado, y examen en blanco como el entorno Bluebook real del SAT— y el nivel avanza por racha de aciertos consecutivos, no por un cronómetro arbitrario.

> *La progresión se organiza por cohortes según el puntaje previo, de modo que cada estudiante entra en el peldaño de dificultad que le corresponde y no repite lo que ya domina.*

05 EL ARTEFACTO

LA VISTA DIVIDIDA: PROBLEMA Y DESMOS, CO-VISIBLES

La pantalla central es una **vista dividida** (split-screen): el problema a la izquierda, la calculadora Desmos embebida y persistente a la derecha. La decisión que define el producto quedó registrada como ADR en el propio código del componente: una app cuyo propósito es enseñar a usar Desmos debe *mostrar* Desmos siempre. El estudiante lee el enunciado, ve la jugada sugerida y la ejecuta en la calculadora sin cambiar de contexto.

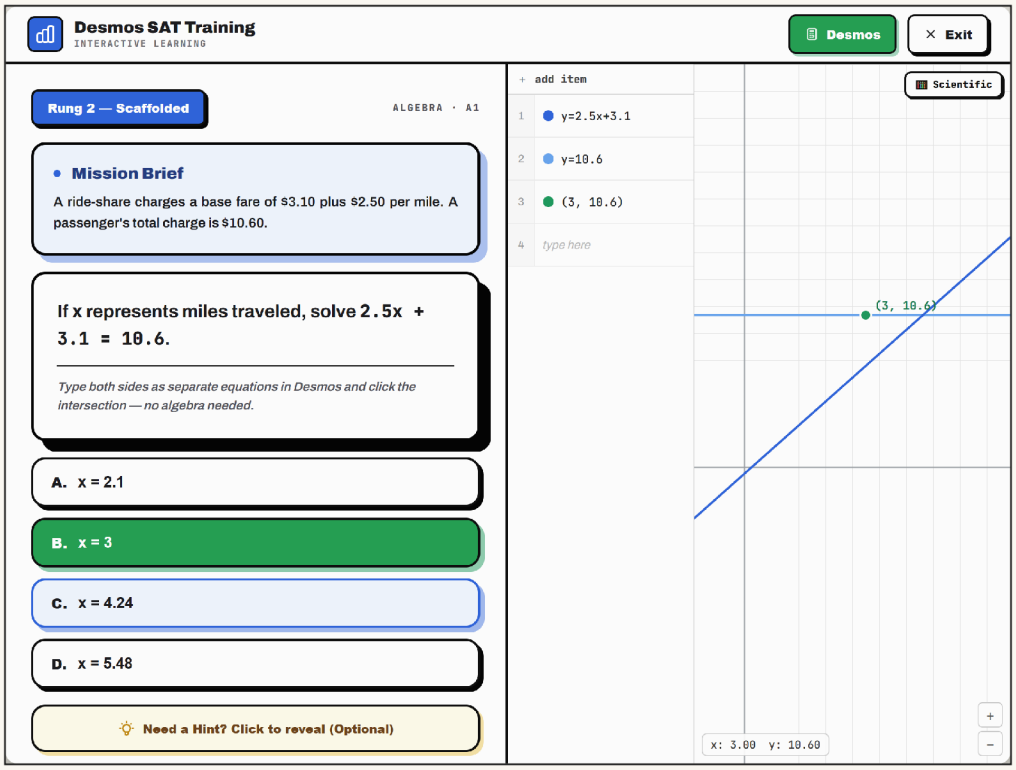


FIG. 1 — VISTA DIVIDIDA DE ENTRENAMIENTO · ENUNCIADO TIPO SAT A LA IZQUIERDA, DESMOS EMBEBIDO A LA DERECHA · LA HERRAMIENTA VIVE CO-VISIBLE CON EL PROBLEMA, NUNCA ESCONDIDA

06 LA ESTRUCTURA DEL CURRÍCULO

DE UNIT 0 AL SAT MIX, PELDAÑO POR PELDAÑO

El contenido se estructura en cinco capas: **Unit 0** de tecleo; **Units 1 a 4** alineadas con los dominios y pesos reales del examen —Álgebra ~35%, Advanced Math ~35%, Data Analysis ~15%, Geometry & Trig ~15%—; un

BOSS final cronometrado que mezcla los cuatro dominios con tiers de medalla (Platinum a Bronce, aprobación 4/6); y un **SAT Mix** repetible para práctica continua.

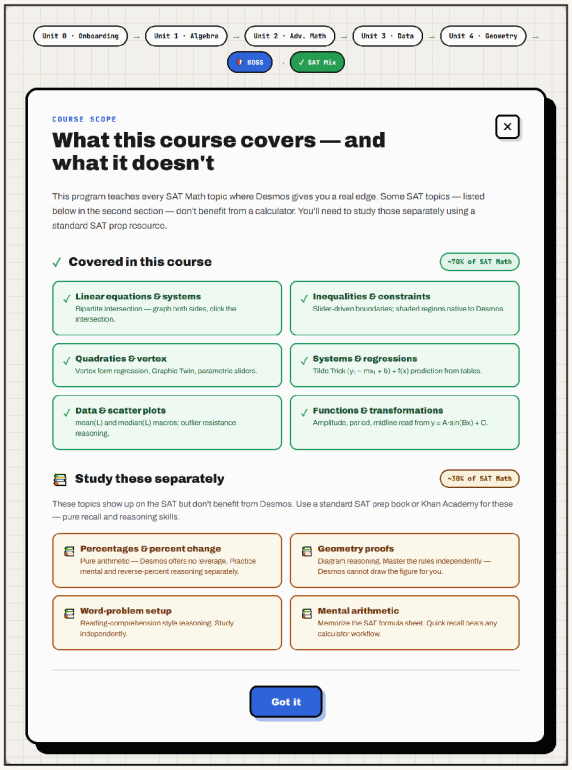


FIG. 2 — MAPA CURRICULAR · LA RUTA UNIT 0 → UNITS 1-4 → BOSS → SAT MIX, CON SUB-RUNGS DE DIFICULTAD POR COHORTE

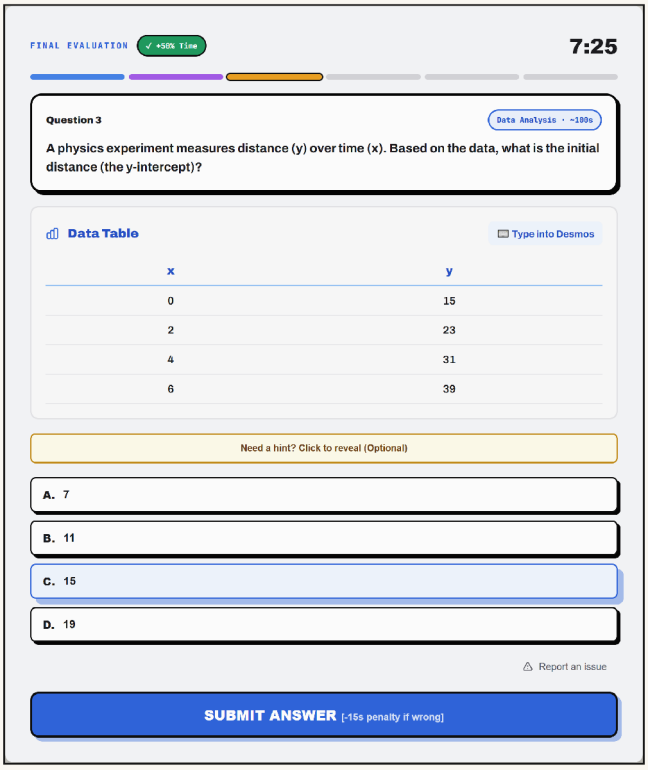


FIG. 3 — ENSAYO SAT MIX · PRÁCTICA REPETIBLE QUE REPRODUCE EL ENTORNO BLUEBOOK REAL DEL SAT — EL NIVEL "EXAMEN EN BLANCO" DEL FADING

07 EL DISEÑO

PROGRESIÓN CURRICULAR Y PIPELINE DE INTEGRACIÓN CON EL LMS

El estado global se maneja con Zustand persistente y el progreso se sincroniza en Firestore con un esquema de progreso por unidad y métricas por estudiante. El corpus de preguntas se generó y analizó apoyándose en un pipeline de scripts de investigación que procesan PDFs de referencia del SAT con modelos vía AWS Bedrock, y se sirve desde bancos de ítems por unidad tipados en TypeScript. La plataforma se integra con **el LMS de la red** mediante eventos de actividad y escritura al libro de calificaciones.

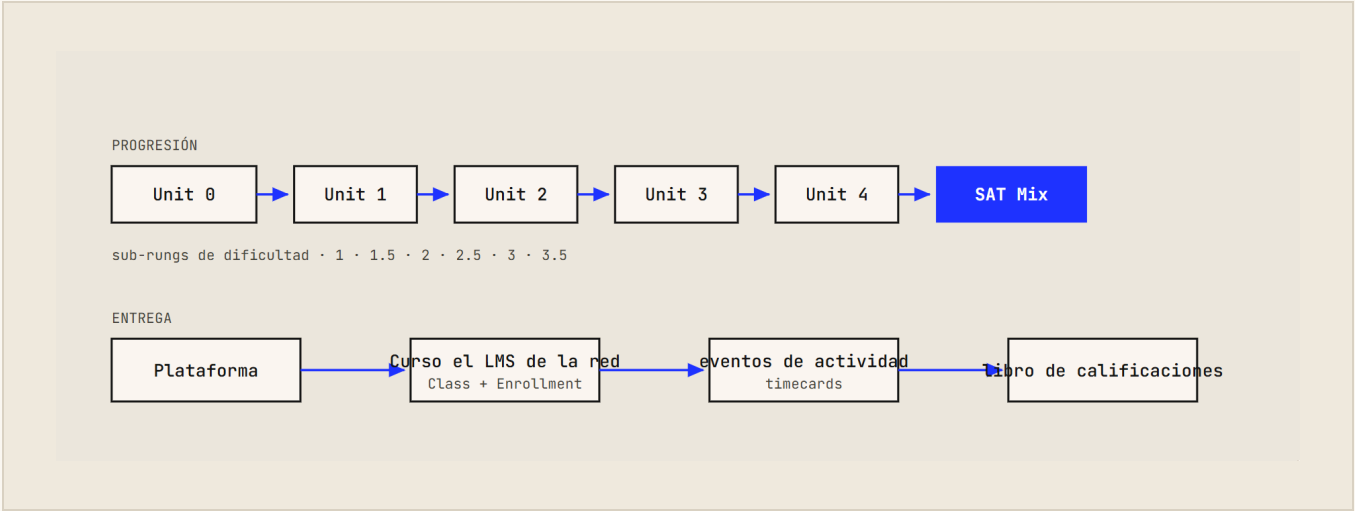


FIG. 4 — PROGRESIÓN CURRICULAR (UNIT 0 → SAT MIX) Y PIPELINE DE INTEGRACIÓN CON EL LMS VÍA EVENTOS DE ACTIVIDAD + LIBRO DE CALIFICACIONES

El Desmos vive embebido y persistente — no escondido tras un botón flotante

Contexto. La versión inicial escondía la calculadora tras un botón flotante que tapaba el contenido. Una app que enseña a usar una herramienta no puede ocultarla.

Decisión. Tras una auditoría instruccional se rediseñó para que el Desmos viva embebido y persistente en el panel derecho, co-visible con el problema, heredando el patrón de la Unit 0.

Trade-off aceptado. Menos espacio horizontal para el enunciado, pero la herramienta queda siempre a la vista — y exhibir la herramienta *es* la pedagogía del producto.

08 CONSTRUCCIÓN Y VALIDACIÓN

CALIDAD INSTITUCIONALIZADA COMO GATES DE BUILD

El desarrollo operó con un flujo AI-first sostenido a lo largo de unas tres semanas y media. La calidad del contenido no descansa en revisión manual: se institucionalizó como un conjunto de gates automáticos que corren antes de cada build —verificación de fidelidad que recomputa la respuesta desde los datos del ítem, cobertura estricta de skills, orden de opciones, render de LaTeX, sintaxis válida de Desmos, patrones de tecleo y correspondencia entre tipo de pregunta y jugada—. Si cualquiera falla, la build falla.

La validación combinó tres frentes: **peer-review matemático** con dos expertos de secundaria, que verificaron la corrección de los ítems y la pedagogía de las jugadas; **QC instruccional** que auditó la experiencia de aprendizaje extremo a extremo; y la batería de **gates automáticos** como red de seguridad permanente. El despliegue es a Vercel con auto-deploy desde la rama principal y un sistema de invitación con aprobación de superadmin. Los propios estudiantes alimentan un canal de reportes de error desde cada pregunta.

09 EL FUNDAMENTO, VUELTO TABLA

LA EVIDENCIA DEL PRODUCTO

Estudiantes con acceso	14 · sin medición de outcome
Ventana de construcción (v2)	25-may → 18-jun 2026
Commits (v2)	346
Unidades de contenido	Unit 0 (tecleo) + Units 1-4 + BOSS + SAT Mix
Cohortes por puntaje previo	foundational <600 · intermediate 600-740 · advanced 750-800
Gates de calidad en build	9 (fidelidad, cobertura, orden, render, desmos, tecleo, repetición, jugada, cobertura de jugada)
Niveles de fading de andamiaje	3 (andamio · recordatorio · examen)
Peso de dominios SAT cubiertos	Álgebra ~35% · Adv Math ~35% · Data ~15% · Geo&Trig ~15%
Integración institucional	el LMS de la red — eventos de actividad + escritura al libro de calificaciones
Validación	peer-review matemático (2 expertos) + QC instruccional
Estado	En producción

10 EL RESULTADO

DE UNA INTUICIÓN DE LA ANALÍTICA A UN PRODUCTO OPERATIVO

La plataforma quedó en producción con acceso abierto a 14 estudiantes, cubriendo la ruta completa de Unit 0 a BOSS más el SAT Mix, con integración con el LMS activa. La validación fue de criterio, no de outcome: medir el efecto sostenido en el puntaje real habría exigido un rollout y una ventana que no controlé. Lo que sí se midió fue la calidad del producto — y el peer-review fue inequívoco: un revisor describió haber aprendido un montón sobre Desmos con el propio sistema, y otro lo calificó como un curso muy sólido.

Más allá del despliegue, el resultado de fondo fue convertir una intuición de la analítica Cap. 1 —la brecha 650 a 800 es de Desmos, no de matemáticas— en un producto operativo con un currículo de jugadas verificadas, un motor de fading de andamiaje y un conjunto de gates de calidad que garantizan que ningún ítem incorrecto llegue al estudiante.

11 APRENDIZAJES

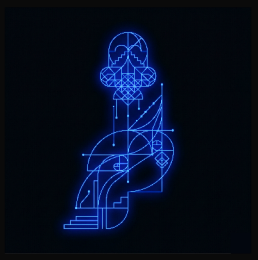
LO QUE DEJÓ EL PROYECTO

- **De diseño.** Una app que enseña una herramienta tiene que exhibir esa herramienta, no esconderla; la auditoría que movió el Desmos de un flotante que tapaba a un panel embebido persistente cambió la experiencia de raíz.
- **De confianza en el contenido.** Verificado por máquina no equivale a verificado del todo — tras regenerar contenido conviene mirarlo renderizado, porque el ojo caza problemas de legibilidad y clones que el gate no ve.
- **Arquitectónico.** Encapsular bien la capa de datos paga: tener Firestore aislado tras una capa de servicios permitió evaluar y ejecutar después una migración de backend con un cutover limpio.
- **De proceso.** Institucionalizar la calidad como gates que fallan la build, en vez de checklists humanos, es lo que sostiene un ritmo de cientos de commits sin que se degrade el contenido.

02 · ENTRENAMIENTO CAPÍTULO 2.2

Enseñar a demostrar, no solo a calcular

AP MATH JUSTIFICATION TRAINER



Del argumento empírico —"lo probé con tres números y funciona"— a la justificación formal que un examen AP premia con un 5. El framework CERC vuelve entrenable y medible la argumentación matemática.

ESTADO	STACK	DATOS	MODELO
Beta	Next.js 14 · TS · KaTeX	Mock KV ↔ LMS	scoring CERC

01 EL PROBLEMA

SABER HACER LA CUENTA, NO SABER DEFENDERLA

Los estudiantes de matemática avanzada dominaban el procedimiento pero perdían puntos donde más cuentan: en la justificación escrita. El diagnóstico era nítido y cuantificado. En las plataformas de práctica adaptativa que la red usaba, el rendimiento en opción múltiple rondaba el **82%**; en las Free Response Questions —donde el examinador no evalúa el resultado sino el razonamiento que lo sostiene— caía al **67%**, y solo una minoría iba realmente encaminada hacia la nota objetivo.

La herramienta procedimental existente era sólida para automatizar el cálculo, pero no entrenaba lo que un examen AP de verdad mide: argumentación matemática por escrito.

Un estudiante podía resolver una integral impecablemente y aun así reprobar la pregunta por no enunciar la condición que habilita el teorema que usó.

Ese hueco —saber hacer la cuenta pero no saber defenderla— era invisible para el sistema existente.

Y era, exactamente, la diferencia entre un 3 y un 5.

02 EL OBJETIVO

LA JUSTIFICACIÓN COMO HABILIDAD ENTRENABLE

OBJETIVO PRIMARIO

Llevar al estudiante desde el argumento empírico hasta la justificación formal que cita la hipótesis del teorema y verifica sus condiciones antes de invocarlo — tratando la justificación matemática como una habilidad entrenable y medible, no como un talento difuso.

Para hacerlo operativo se adoptó un andamiaje explícito: el framework **CERC** —Claim, Evidence, Reasoning, Conditions— que descompone toda demostración en cuatro movimientos visibles y exigibles.

La meta no era que el estudiante escribiera más, sino que escribiera *completo*: que ninguno de los cuatro elementos quedara implícito, porque es precisamente lo implícito lo que un examinador no puede premiar.

03 LOS USUARIOS

EL ESTUDIANTE Y EL TUTOR

El usuario primario es el estudiante de matemática avanzada que ya domina el cálculo pero pierde puntos en la respuesta escrita. La plataforma se diseñó para tres cursos paralelos —Calculus AB, Calculus BC y Statistics— porque la estructura del argumento es la misma aunque el contenido cambie: una afirmación, su evidencia, el principio que las conecta y las condiciones que lo habilitan.

El segundo usuario es el tutor o coordinador académico, que opera a través de un panel administrativo con visualización del estado de razonamiento de cada estudiante (R-0388 y sus pares aparecen siempre anonimizados), seguimiento de progreso por unidad y disparadores manuales de práctica.

> *Una decisión de privacidad deliberada: el panel del tutor aísla los datos sensibles y nunca expone al estudiante las "trampas" pedagógicas de cada problema, para no contaminar el ejercicio.*

04 EL ARTEFACTO

LA SESIÓN A PANTALLA DIVIDIDA

La interfaz central es una sesión **split-screen**: a la izquierda el enunciado del problema con la notación renderizada en KaTeX y el recuadro del teorema —nombre, enunciado, hipótesis—; a la derecha el formulario CERC, cuatro campos apilados —Claim, Evidence, Reasoning, Conditions — cada uno con su descripción y su marco de oración cuando corresponde. El contador de completitud y la barra de progreso dan retroalimentación visual en tiempo real a medida que el estudiante completa cada campo.

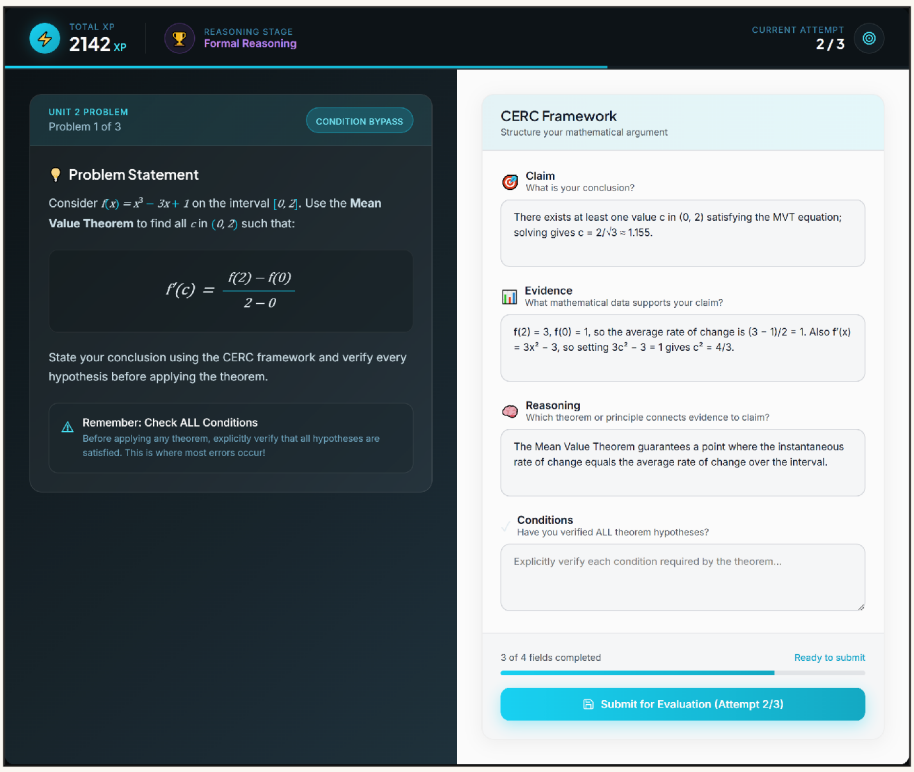


FIG. 1 — SESIÓN CERC · FRQ DE CALCULUS A LA IZQUIERDA, FORMULARIO CLAIM/EVIDENCE/REASONING/CONDITIONS A LA DERECHA · "3 OF 4 COMPLETE" MUESTRA EL SCORING POR COMPLETITUD EN VIVO

05 EL FEEDBACK

SCORING POR COMPLETITUD, NO POR CORRECCIÓN SEMÁNTICA

Tras el envío, el sistema evalúa de forma **determinista** si los cuatro elementos están presentes y no vacíos, y lo proyecta sobre la rúbrica AP de 1 a 5. No juzga si el argumento es "bueno" con un modelo de lenguaje: el primer hábito que hay que instalar no es la elocuencia, sino la integridad estructural del argumento —que la condición esté escrita, que la evidencia exista, que el razonamiento nombre el teorema.

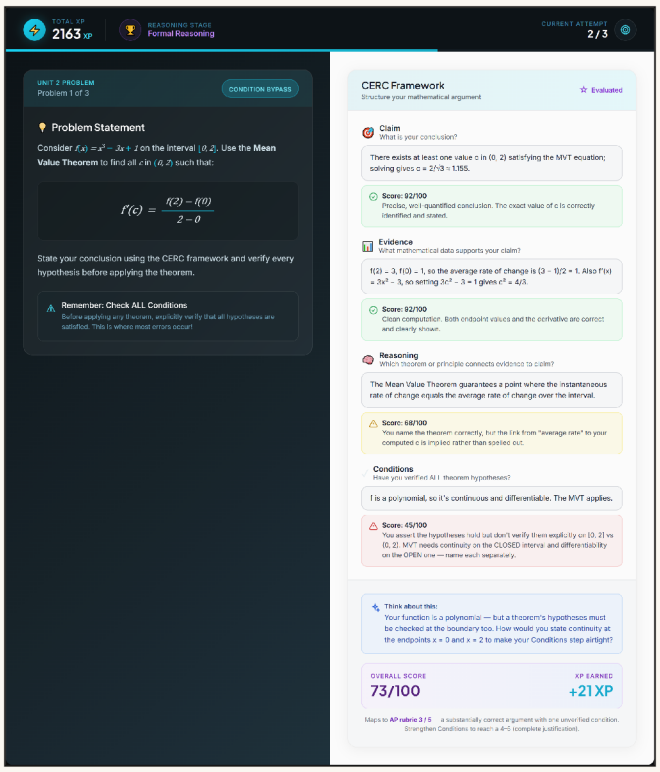


FIG. 2 — FEEDBACK POST-ENVÍO · COMPLETITUD CERC MAPEADA A LA RÚBRICA AP 1–5 · UN EVALUADOR SEMÁNTICO HABRÍA SIDO MÁS IMPRESIONANTE Y MUCHO MENOS HONESTO SOBRE LO QUE MEDÍA

06 EL FUNDAMENTO

TRES ESTADIOS, CUATRO UNIDADES COGNITIVAS

El diseño se apoya en un modelo de desarrollo del razonamiento matemático con tres estadios sucesivos, codificados literalmente en el tipo de datos de la aplicación: **empírico** (el estudiante se convence con ejemplos), **genérico** (generaliza el patrón pero sin rigor) y **formal** (demuestra invocando la estructura). El curso está hecho para forzar esa transición.

UNIDAD	FOCO COGNITIVO	QUÉ ENTRENA
U1	Romper la ilusión empírica	Problemas diseñados para que la intuición falle: el estudiante experimenta por qué ver tres casos no basta.
U2	Verificación de condiciones	Verificar TODAS las condiciones de un teorema, sin atajos — el error más caro del examen real.
U3	Síntesis sin andamiaje	Síntesis multiconcepto y precisión comunicativa, ya sin marcos de oración.
U4	FRQ cronometradas	Free Response individuales bajo condiciones de examen.

El catálogo clasifica cada problema por el tipo de error que provoca — `CONDITION_BYPASS`, `LOCAL_ONLY_ARGUMENT` o `CER_BREAKDOWN`— y los acompaña de marcos de oración (*sentence frames*) que se van retirando unidad a unidad: una aplicación directa del principio de *fading*, donde el andamiaje se desvanece a medida que la competencia se consolida.

07 EL DISEÑO

EL MODELO CERC Y LA DECISIÓN QUE MÁS IMPORTÓ

CERC descompone toda demostración en cuatro movimientos: la afirmación (Claim), la evidencia que la respalda (Evidence), el principio o teorema que las conecta (Reasoning) y las condiciones que lo habilitan (Conditions). Sistema de pistas de tres niveles —dónde está el fallo, qué elemento CERC está roto, la corrección explícita— sostiene al estudiante sin resolverle el problema. La gamificación suma XP por unidad y desbloquea insignias con animaciones GSAP.

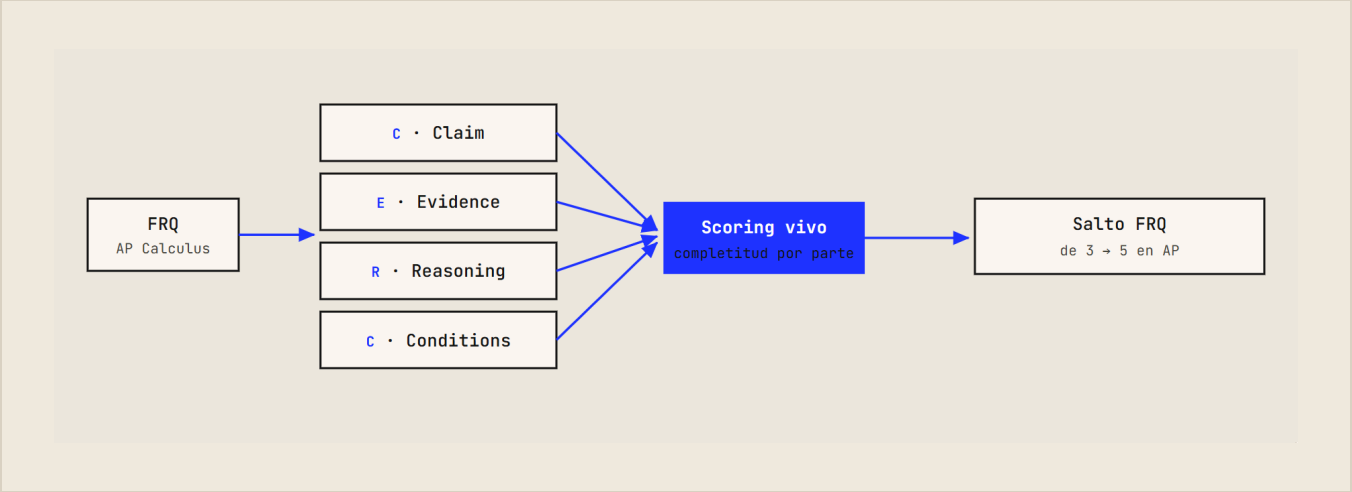


FIG. 3 — MODELO CERC: CLAIM → EVIDENCE → REASONING → CONDITIONS, LOS CUATRO MOVIMIENTOS VISIBLES Y EXIGIBLES DE TODA DEMOSTRACIÓN

Scoring por completitud — no calificación semántica por IA

Contexto. Un evaluador semántico que juzgara si el argumento es “bueno” con un modelo de lenguaje habría sido más impresionante, pero deshonesto sobre lo que realmente medía.

Decisión. El sistema evalúa de forma determinista si los cuatro elementos CERC están presentes y no vacíos, con retroalimentación visual en tiempo real. El primer hábito a instalar es la integridad estructural del argumento, no la elocuencia.

Trade-off aceptado. Mide menos “calidad” y más “completitud” — pero ataca el hábito correcto primero y mide exactamente lo que afirma medir.

La segunda decisión estructural fue la capa de datos con patrón **adaptador**: un adaptador mock para desarrollo respaldado por Vercel KV, y un adaptador del LMS para producción que conserva las formas correctas de los estándares abiertos de interoperabilidad educativa.

08 CONSTRUCCIÓN Y VALIDACIÓN

AI-FIRST, CON LA SEGURIDAD COMO RESTRICCIÓN DE DISEÑO

Se construyó con un flujo **AI-first**, asistido por Claude, a un ritmo deliberadamente alto durante unas seis semanas (17 de marzo al 30 de abril), sobre 29 páginas en App Router de Next.js 14. El stack quedó en

TypeScript estricto, Tailwind, KaTeX para la notación y GSAP para la gamificación.

La seguridad se trató como un frente propio, no como un añadido: autenticación con JWT firmados mediante *jose*, protección de rutas por middleware con control de acceso por rol (estudiante / administrador), sanitización con DOMPurify, rate limiting y CSRF obligatorio en toda mutación autenticada. Una auditoría externa a comienzos de abril detectó oportunidades de endurecimiento en autenticación y manejo de PII; se resolvieron reforzando el modelo de sesión y el aislamiento de datos del panel administrativo.

La validación fue de dos clases. En lo técnico, una batería de pruebas cubría la integridad de los datos del curso, la lógica de prerrequisitos entre unidades y el flujo completo de una sesión, además del build de producción de las 29 páginas. En lo pedagógico, la plataforma se preparó para un **piloto** con un grupo reducido de estudiantes de Calculus BC y Statistics —identificados de forma anónima— con login propio respaldado por el roster real del LMS.

> *La medición no es una nota subjetiva sino la trayectoria del estudiante a través de los tres estadios de razonamiento y su completitud CERC creciente a medida que el andamiaje se retira entre la Unidad 1 y la Unidad 4.*

09 EL RESULTADO

ESTADO Y LEGADO

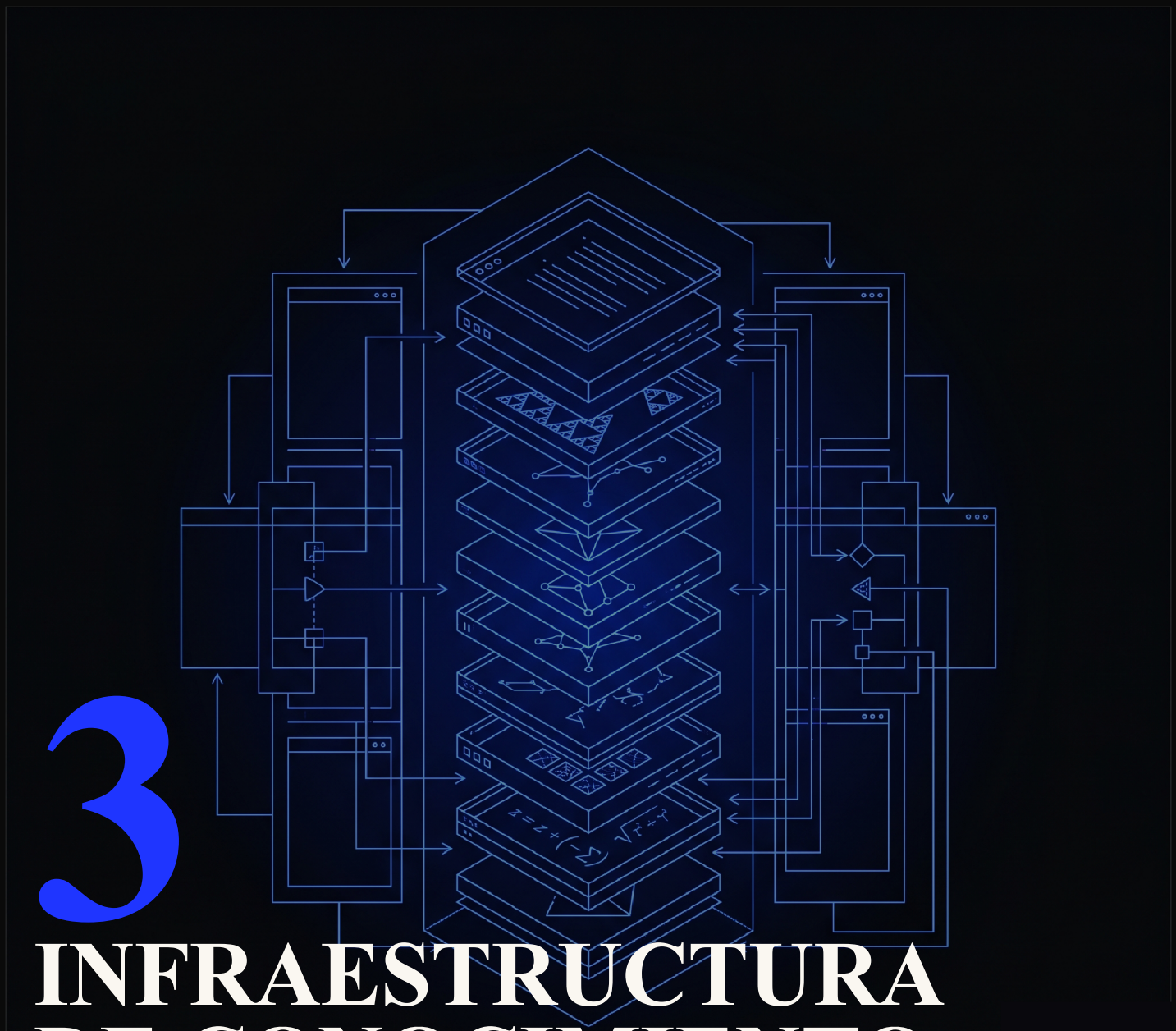
Una plataforma funcional en estado Beta: las cuatro unidades operativas con su catálogo de problemas, los tres cursos, el motor de scoring por completitud, el sistema de XP e insignias, el panel administrativo con aislamiento de datos y la capa de adaptadores intercambiables. La integración productiva con el LMS quedó como stub deliberado, fiel a las formas los estándares abiertos de interoperabilidad educativa, listo para conectarse cuando el entorno de producción lo permitiera.

El criterio de éxito fijado por el liderazgo académico fue explícito: el trabajo se aprobaba si llevaba al estudiante a un ***5*** en el examen AP — el outcome, no la actividad.

Framework pedagógico	**CERC** — Claim · Evidence · Reasoning · Conditions
Unidades · problemas	**4 unidades · 7 / 7 / 7 / 4**
Cursos cubiertos	Calculus AB · Calculus BC · Statistics
Estadios de razonamiento	empirical · generic · formal (tipo de datos)
Capa de datos	Mock (Vercel KV) ↔ LMS de la red (estándares abiertos de interoperabilidad educativa)
Seguridad	JWT (jose) · middleware por rol · CSRF · DOMPurify · rate limiting
Construcción	AI-first · 29 páginas · capa de adaptadores intercambiables
Estado	**Beta** · el LMS como stub fiel a las APIs

APRENDIZAJES

- **Honestidad de medición.** Resistir la tentación del evaluador semántico por IA y elegir scoring determinista por completitud: medía exactamente lo que afirmaba medir y atacaba el hábito correcto primero —la estructura antes que la elocuencia.
- **Teoría codificada en los tipos.** Cuando el estadio de razonamiento es un tipo de datos de primera clase y las unidades retiran el andamiaje de forma programada, el diseño pedagógico deja de ser intención y se vuelve verificable.
- **Patrón adaptador.** Decisión de bajo costo y alto retorno: iterar a toda velocidad con un mock persistente sin acoplarse a la API de producción, manteniendo intactas sus formas.
- **Seguridad como restricción de diseño.** En una plataforma con datos de menores, autenticación, CSRF y aislamiento de datos no son una fase final sino una restricción desde la primera línea. la intersección curricular AP/SAT

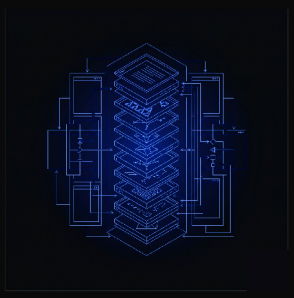


3 INFRAESTRUCTURA DE CONOCIMIENTO

La capa que fundamenta a todas: de la literatura académica a la investigación con validación de evidencia.

Evidencia de la que se puede responder

RESEARCH INTELLIGENCE PLATFORM



Un sistema operativo de evidencia que convierte literatura académica en decisiones institucionales trazables — una afirmación a la vez, anclada al estudio que la sostiene.

ESTADO	STACK	IA	VALIDACIÓN
Producción	Next.js 16 · React 19 · Firebase	AWS Bedrock · Claude	ESSA + 26 suites

01 EL PROBLEMA

CONCLUSIONES DE LAS QUE NADIE PODÍA RESPONDER

Una red de colegios K-12 de aprendizaje personalizado acelerado por IA toma decisiones de currículum e instrucción a diario: qué secuencia adoptar, qué intervención escalar, qué práctica retirar. Esas decisiones se apoyaban en resúmenes generados con IA que mezclaban, en una sola operación, dos cosas que jamás debían ir juntas: la extracción fiel de lo que un estudio reportaba y la interpretación pedagógica de lo que ese estudio significaba para el aula.

El resultado era imposible de auditar. No se podía distinguir lo que el autor de un paper había medido de lo que el modelo había inferido, ni rastrear una recomendación hasta el estudio concreto que la sostenía.

Las conclusiones llegaban como narrativa fluida y sin anclaje: frases huérfanas que sonaban con autoridad pero que nadie podía verificar.

A esto se sumaba un problema de identidad: el sistema previo referenciaba estudios por su posición en un arreglo, y cada reordenamiento corrompía silenciosamente esas referencias.

Para una institución que se define por estar basada en evidencia, el cuello de botella no era producir texto, sino producir texto del que se pudiera responder.

02 EL OBJETIVO

TRAZABILIDAD A NIVEL DE AFIRMACIÓN

OBJETIVO PRIMARIO

Construir un sistema operativo de evidencia, no un generador de resúmenes: que **toda conclusión publicable tuviera trazabilidad a nivel de afirmación**. Ninguna frase del documento final podía existir sin un mapeo estructurado a uno o más estudios fuente identificados por un `refId` estable.

Junto a eso, tres metas concretas: separar de forma irreversible la extracción de la interpretación, aplicar un marco de calidad metodológica determinista y reproducible, y mantener un corpus vivo y versionado capaz de incorporar estudios nuevos o re-evaluaciones sin reconstruirse entero.

El producto debía cerrar el ciclo completo, desde el paper crudo hasta la decisión institucional documentada, de modo que un revisor humano pudiera intervenir en cualquier punto sin romper la cadena de evidencia. Es una herramienta interna de admin para dos personas: el operador del pipeline y una investigadora del equipo de learning science incorporada como par de revisión.

03 EL SISTEMA

UN PIPELINE DE NUEVE ETAPAS AUDITABLE

El sistema se organiza como un pipeline de nueve etapas encadenadas, donde cada etapa tiene su propio agente, su contrato de entrada y salida, y un artefacto persistido y versionado. La decisión central, documentada como **ADR-001**, fue prohibir que extracción, evaluación, clustering, síntesis y redacción ocurrieran en una misma llamada al modelo: la separación de capas es lo que hace el sistema auditable.

Ingesta→Extracción→Clustering→Síntesis→Triangulación→Claims→Capítulos→Final→Editorial

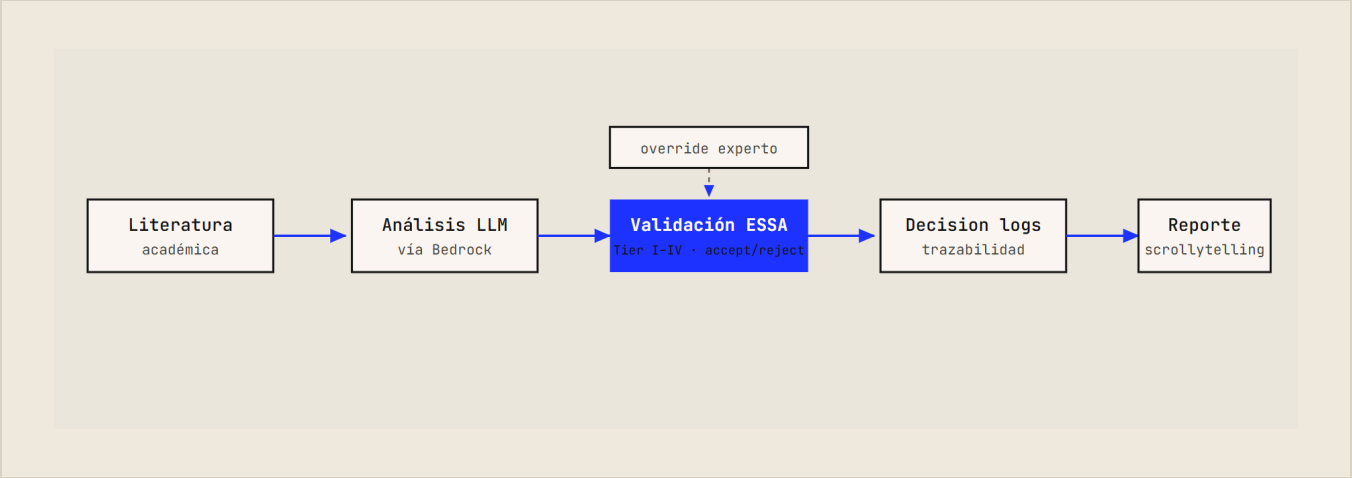


FIG. 1 — PIPELINE LITERATURA → LLM → VALIDACIÓN ESSA → DECISION LOGS → REPORTE · CADA ETAPA PERSISTE UN ARTEFACTO VERSIONADO CON IDENTIDAD POR REFID INMUTABLE, NUNCA POR ÍNDICE POSICIONAL

Separar extracción de interpretación — nunca en la misma llamada al modelo

Contexto. El sistema anterior mezclaba en una operación qué medía el estudio y qué significaba para el aula, y referenciaba estudios por su posición en un arreglo — cada reordenamiento corrompía las referencias.

Decisión. El marco de calidad (ADR-012) bifurca el *appraisal* según el tipo de estudio —ESSA para cuantitativo, CASP para cualitativo, ambos para mixto— y codifica el árbol de decisión ESSA como reglas deterministas explícitas en el prompt: `randomAssignment + controlGroup + muestra adecuada → Sólida`. El agente no infiere el nivel con libertad.

Trade-off aceptado. Más etapas y más artefactos que un resumidor de un paso — pero es lo que vuelve cada conclusión rastreable hasta el estudio que la sostiene. Las 21 decisiones quedaron registradas como ADRs.

04 EL FUNDAMENTO

TRES ESTÁNDARES DE SÍNTESIS, VUELTOS SOFTWARE

La arquitectura traduce dos estándares reconocidos de síntesis de evidencia al dominio del software, sumando un tercero para lo cualitativo. El principio rector — **fidelidad primero**— viene directo de la disciplina de la revisión sistemática: representar el estudio antes de interpretarlo.

MARCO	QUÉ APORTA	CÓMO OPERA EN EL SISTEMA
ESSA	Fuerza de la evidencia en cuatro niveles	Árbol de decisión determinista codificado en prompt; clasifica cada referencia.
PRISMA 2020	Transparencia de revisiones sistemáticas	Auditoría automática de 12 ítems contra los artefactos del sistema al cierre.
CASP	Appraisal de evidencia cualitativa	Bifurca el análisis para que cuanti, cuali y mixto no se fuercen a una misma lógica.

Los cuatro niveles ESSA estructuran todo el corpus, del más fuerte al de menor evidencia empírica:

TIER 1 · SÓLIDA — RCT TIER 2 · MODERADA — cuasi-experimental

TIER 3 · PROMISORIA — correlacional

TIER 4 · LÓGICA — modelo lógico

> *Lo cuantitativo se sintetiza por efectos y moderadores; lo cualitativo por temas y mecanismos; lo mixto se integra explícitamente mediante triangulación. Representar el estudio antes de interpretarlo.*

05 LA VALIDACIÓN

EL CORAZÓN FUNCIONAL: REVISIÓN HUMANA REFERENCIA POR REFERENCIA

La validación ocurre en **/research** mediante un flujo explícito de revisión humana. Tras correr la validación ESSA, la investigadora y el operador revisaban cada referencia una por una, con tres veredictos posibles — **Aceptada**, **Advertencia** o **Rechazada**— sobre el nivel que el agente había asignado según el árbol determinista. El sistema permite *override* humano: el juicio del modelo es una propuesta, no una sentencia.

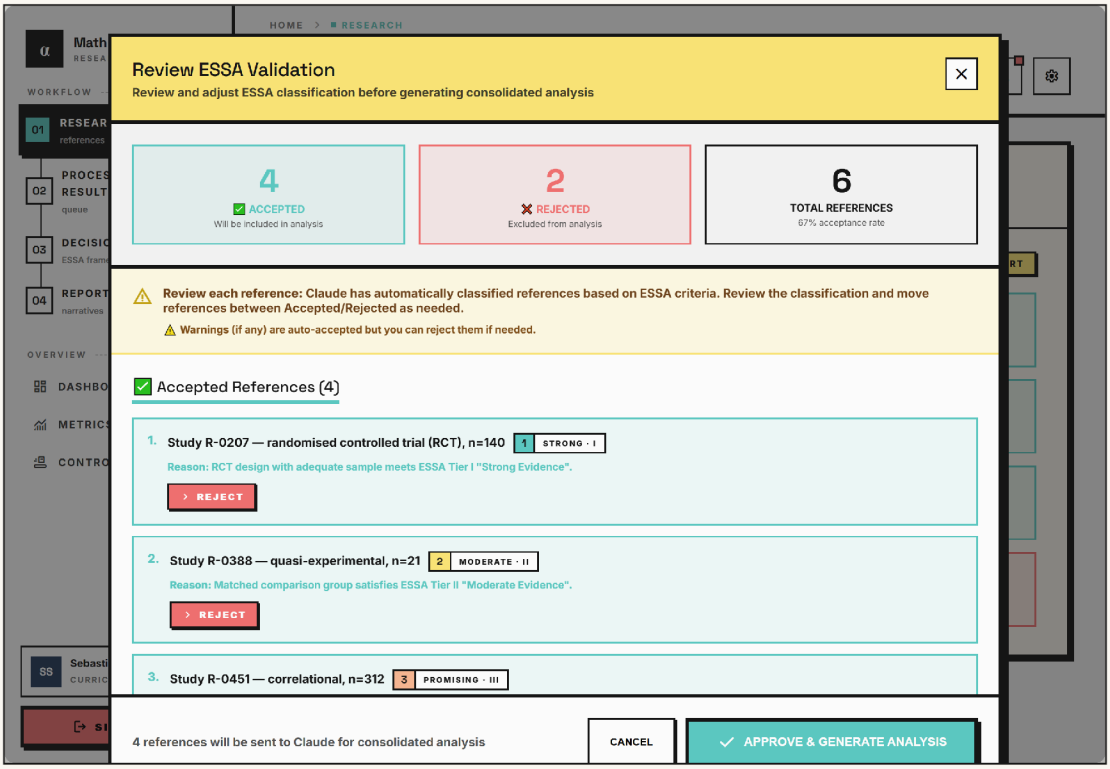


FIG. 2 — /RESEARCH CON ESSA VALIDATION · CADA REFERENCIA MUESTRA SU NIVEL PROPUESTO Y EL OVERRIDE ACCEPTED / WARNING / REJECTED · NINGUNA SELECCIÓN SE APRUEBA SIN PASAR EL FILTRO HUMANO

06 EL SEGUNDO FILTRO

AUDITORÍA PRISMA AUTOMÁTICA Y DECISION LOGS

El cierre del documento final añade una segunda capa de control automático, la **auditoría PRISMA**, que verifica doce ítems del estándar contra los artefactos del sistema —desde criterios de elegibilidad y riesgo de sesgo por estudio hasta resultados de síntesis y limitaciones— y reporta un cumplimiento total, parcial o no conforme. Así, ninguna investigación se publica sin pasar el doble filtro: revisión humana sobre la evidencia y verificación automática sobre el reporte. La vista **/decision-logs** registra cada decisión institucional con su cadena de evidencia intacta.

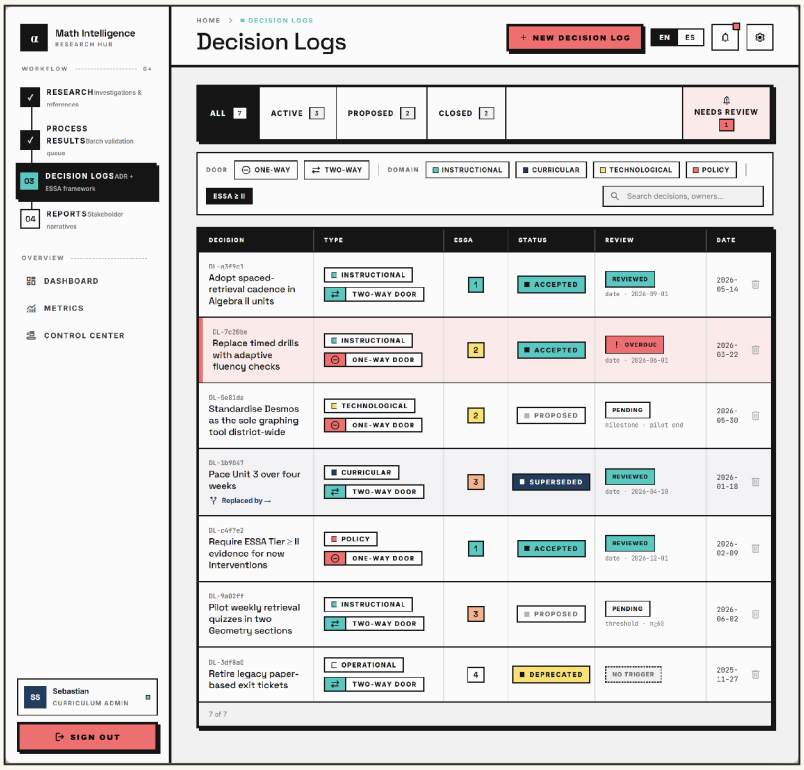


FIG. 3 — /DECISION-LOGS · CADA DECISIÓN QUEDA ANCLADA A LA INVESTIGACIÓN Y A LOS ESTUDIOS QUE LA SOSTIENEN · TRAZABILIDAD DEL PAPER A LA DECISIÓN DOCUMENTADA

07 CASO EMBLEMA

MEASURING THE 2X FACTOR

Measuring the 2x Factor: Speed and Mastery Depth Metrics for Adaptive HS Math. La tesis: la afirmación de que las plataformas adaptativas de matemáticas pueden duplicar la velocidad de aprendizaje **no puede verificarse ni refutarse** con la

evidencia disponible tal como está estructurada hoy — ningún estudio operacionaliza velocidad y profundidad de dominio simultáneamente contra un benchmark de élite.

HALLAZGOS, CON EFFECT SIZES

d 0,10–0,68 · g 0,37 — Las plataformas adaptativas producen efectos positivos moderados sobre el rendimiento matemático, pero ningún estudio los midió contra benchmarks de élite como 800 en SAT Math o 5 en AP.
velocidad ≠ profundidad — Avanzar rápido dentro de una plataforma puede producir comprensión superficial; hay que separar métricas de velocidad de métricas de dominio conceptual.
d 0,30–1,34 — La práctica intercalada y el espaciado producen los efectos más robustos sobre retención a largo plazo; la práctica distribuida duplica o triplica la retención frente a la masiva.
profundidad > aceleración — El dominio profundo de precálculo predice el rendimiento en cálculo universitario con más del doble de impacto que simplemente haber cursado el curso.
wheel-spinning — La práctica extensa sin logro de dominio es predecible desde indicadores tempranos; tiempo de uso y número de ejercicios son insuficientes para evaluar el aprendizaje real.

08 CASO EMBLEMA

EL MARCO PROPUESTO Y SU BASE DE EVIDENCIA

MARCO DE MEDICIÓN DE DOBLE EJE VELOCIDAD-DOMINIO

(1) una métrica de velocidad válida distinta del tiempo en tarea; (2) una métrica de profundidad de dominio anclada en competencias fundacionales verificadas; y (3) un benchmark de rendimiento de élite calibrado.

La tensión sin resolver. La contradicción velocidad–profundidad permanece abierta porque ningún estudio operacionalizó ambas métricas a la vez contra un benchmark de élite; los efectos para estudiantes de alto rendimiento bajo presión extrema siguen sin documentarse. La implicación es directa: toda plataforma que afirme aceleración («2x») debería demostrar empíricamente la operacionalización de sus métricas y reportar efectos sobre evaluaciones externas antes de declararse efectiva. Es el marco que vuelve *falsable* la promesa central del modelo.

Clusters de evidencia	29
Afirmaciones mapeadas	45
Fuentes	45 · todas SÓLIDA
Confianza del reporte	Moderada

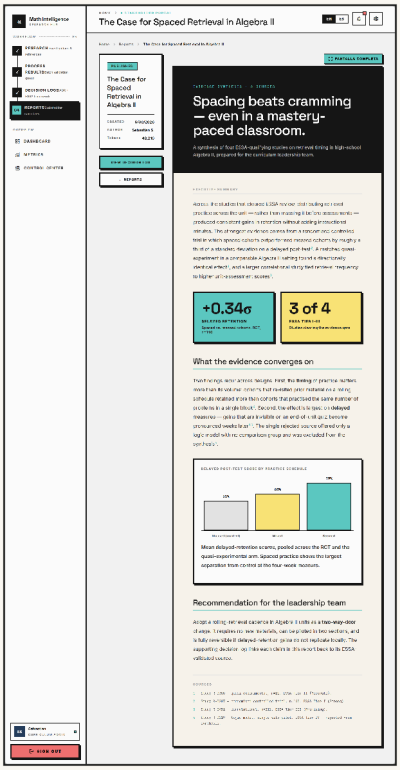


FIG. 4 — REPORTE SCROLLYTELLING GENERADO · UNA INVESTIGACIÓN CONVERTIDA EN NARRATIVA NAVEGABLE, CON CADA AFIRMACIÓN ANCLADA A SUS FUENTES · EXPORTABLE A PDF

09 GÉNESIS V0

SIETE ESTUDIOS MANUALES QUE JUSTIFICARON LA PLATAFORMA

Antes de la plataforma hubo una prueba de concepto: **siete estudios sintetizados a mano con Gemini**. Demostraron que convertir literatura en investigación accionable era valioso — y que hacerlo a mano no escalaba ni dejaba trazabilidad auditable. Ese v0 es lo que justificó construir el sistema operativo de evidencia.

1. **Academic Audit Report.** Auditoría longitudinal de desempeño en matemáticas de secundaria (SAT Math y AP) frente a instituciones de referencia. cap. 07
2. **Cognitive Architecture del SAT.** La brecha 650→800 como fluidez estructural y arbitraje con Desmos. cap. 08
3. **AP/SAT Curricular Intersection.** El sistema radicular algebraico que sostiene Cálculo y Estadística avanzados. cap. 09
4. **Automation Threshold Roadmap.** El umbral de automaticidad como puente hacia Cálculo acelerado. cap. 10
5. **MS Persistence vs SAT Stamina.** De la persistencia en 7º grado a la resistencia cognitiva del 1550. cap. 11
6. **Elite Freshman Profiles.** Auditoría recursiva de la compresión del SAT Math y la preparación STEM. cap. 12
7. **Technical Calculation Protocol.** Métricas propietarias para el sprint diario de matemáticas. cap. 13

Los siete reaparecen como capítulos propios de este book: el v0 manual fue el origen del corpus que la plataforma luego escaló y formalizó.

10 EL CORPUS, DE UN VISTAZO

LA SALIDA DE LA PLATAFORMA A ESCALA

De siete estudios manuales a un cuerpo de investigación validado, vivo y versionado. Esto es lo que la plataforma produjo:

19
Investigaciones
542
Estudios sintetizados
465
Referencias

FIG. 5 — CORPUS AT A GLANCE · CADA CIFRA ES PRODUCTO DEL PIPELINE DE NUEVE ETAPAS CON DOBLE FILTRO · EL PROTOTIPO MANUAL PREVIO CON GEMINI PARTIÓ DE 7 ESTUDIOS

Cada una de estas investigaciones pasó por el árbol de decisión ESSA, la revisión humana referencia por referencia y la auditoría PRISMA automática de doce ítems. La interfaz de validación con override humano y la auditoría PRISMA son las dos garantías que distinguen este sistema de un generador de resúmenes: cada conclusión que sale es rastreable hasta los estudios que la sostienen. Es, en la práctica, la infraestructura que alimenta el corpus de investigación de los capítulos siguientes.

11 APÉNDICE

ÍNDICE DEL CORPUS — LAS 19 INVESTIGACIONES

INVESTIGACIÓN	ESTUDIOS	REFS.	FECHA
Measuring the 2x Factor: Speed and Mastery Depth Metrics for Adaptive HS Math	211	195	2026-05-07
Critical Variables for Big Bang Adaptive Math Curriculum Implementation in High School	58	53	2026-05-07
Change Management and Leadership Resistance in Non-Traditional School Reform	28	14	2026-05-07
Educational Data Mining and Learning Analytics in Mathematics	23	20	2026-04-06
Advanced Placement Mathematics Achievement Factors	23	13	2026-04-06
Mastery-Based Learning and Spaced Practice in Mathematics	19	11	2026-04-06
Interleaved Practice and Problem-Type Discrimination	18	8	2026-04-06
From Computation to Proof: Pedagogical Transitions for Advanced Adolescents in Self-Directed Environments	17	17	2026-04-21
Computational Fluency and Mathematical Modeling Integration in Advanced Secondary Curricula	16	16	2026-04-21
Self-Regulated Learning in Digital Mathematics Environments	16	11	2026-04-06
Beyond AP and SAT: Assessment Frameworks for Mathematical Maturity — International Models and Admissions Signaling	15	15	2026-04-21
Sequencing Advanced Mathematics for Adolescent STEM Readiness: International Comparative Analysis	15	15	2026-04-21
Conceptual vs. Procedural Knowledge Development in Mathematics	15	8	2026-04-06
Mathematics Anxiety and Emotional Regulation in Digital Learning	14	9	2026-04-06
Sustaining Mathematical Motivation: Wellbeing, Coach Support, and Resilience in Accelerated Adolescent Math Learning	11	19	2026-04-21
Large Language Models and Conversational Tutoring in Mathematics	11	11	2026-04-06
Cognitive Load Management in Autonomous Digital Learning	11	9	2026-04-06

Intelligent Tutoring Systems Effectiveness in Secondary Mathematics	11	11	2026-04-06
Projected Learning Gains of LLM-Based Conversational Tutoring in Secondary Mathematics: A Monte Carlo Simulation Based on AutoTutor Evidence	10	10	2026-04-12

12 CONSTRUCCIÓN Y RESULTADO

AI-FIRST, SOMETIDO A CONTRATOS DE DATOS

Construí la plataforma de forma AI-first y en alta cadencia, fines de semana incluidos, sobre un stack deliberadamente moderno: Next.js 16 con App Router, React 19 y Tailwind v4, asumiendo sus breaking changes. El motor de inferencia es AWS Bedrock: **Claude Sonnet 4.6** para todas las etapas de clasificación y publicación, y **Opus** para los asistentes interactivos. La disciplina que sostiene la confianza en el output es la validación con Zod en cada borde de datos, en lectura y en escritura, contra contratos centralizados: un artefacto que no valida no entra al corpus.

Los runners de cada agente están desacoplados del cliente Bedrock mediante una interfaz inyectable, lo que permite correr el pipeline completo contra un mock en los tests. La CI ejecuta typecheck, lint, tests y build en cada push, con **26 suites** entre contratos, integración, regresión y unitarias. La plataforma quedó en producción como app de admin, sirviendo el ciclo completo desde el paper hasta la decisión documentada.

Commits	`~773 · 31-mar → 5-jun 2026`
Etapas del pipeline	**9** · ingesta → editorial
Decisiones de arquitectura	**21 ADRs** registradas
Marcos de calidad	ESSA + CASP + PRISMA 2020 (12 ítems)
Validación de datos	Zod en lectura y escritura, en cada borde
Suites de test en CI	**26** · contratos, integración, regresión, unitarias
Usuarios admin	2 · operador + investigadora de learning science
Estado	**En producción**

APRENDIZAJES

- **De arquitectura.** La auditabilidad no se agrega después, se diseña desde la primera capa: separar extracción de interpretación y anclar la identidad en un refId inmutable resolvió de raíz una clase entera de bugs que ningún parche posterior habría contenido.
- **De IA y juicio experto.** Codificar el árbol ESSA como reglas deterministas dentro del prompt, en lugar de dejar que el modelo razonara el nivel libremente, hizo el resultado reproducible y dejó el juicio donde corresponde: en el override humano referencia por referencia.
- **De diseño.** Trabajar el flujo de validación junto a una investigadora de learning science afinó la interfaz más que cualquier especificación escrita — ver a alguien recorrer la revisión en vivo mostró exactamente dónde la trazabilidad ayudaba y dónde estorbaba. Y los 21 ADRs probaron su valor al operar sobre un stack con breaking changes.

Qué demuestra el portafolio, en conjunto

EL CRITERIO ES EL PRODUCTO.

Seis piezas para un mismo programa de matemáticas. Vistas de lejos, no son seis apps: son seis ejercicios del mismo músculo — decidir qué medir, traducir ciencia del aprendizaje a software y sostener la calidad sin degradarla.

01 EL ARCO

DE VER, A ENTRENAR, A FUNDAMENTAR

El portafolio recorre un programa entero, no una función suelta. Primero **ver**: una capa de analítica que volvió legible el riesgo de ~1.600 estudiantes y la convirtió en acción matinal sin trabajo manual Sec. 1. Luego **entrenar**: dos plataformas que atacan los saltos de dificultad más caros del examen —la fluidez con la calculadora del SAT y la argumentación del AP— traduciendo distinciones de ciencia del aprendizaje en mecánicas de software Sec. 2. Y por último **fundamentar**: una infraestructura que convierte literatura académica en decisiones de currículum trazables, cada afirmación anclada a su fuente Sec. 3.

Cada producto resolvió su problema y, a la vez, habilitó al siguiente: la analítica detectó la brecha que el entrenamiento atacó; la necesidad de fundamentar esas decisiones empujó la plataforma de evidencia.

No es un catálogo de demos: es un sistema pensado por capas, donde la decisión de qué construir vino siempre de un diagnóstico, no de una lista de funciones.

02 EL HILO COMÚN

LO QUE SE REPITE EN LAS SEIS PIEZAS

CRITERIO TRAZABLE

Cada decisión estructural quedó documentada con su *por qué* y su trade-off aceptado — del Desmos embebido y persistente al scoring determinista por completitud. El artefacto muestra el juicio, no solo el resultado.

LEARNING SCIENCE VUELTA SOFTWARE

CERC como tipo de datos, las cinco métricas del riesgo, el fading de andamiaje por racha, los estadios de razonamiento. Principios de aprendizaje convertidos en mecánicas medibles, no en eslóganes.

CALIDAD INSTITUCIONALIZADA

Gates que fallan la build, doble filtro humano + validación de evidencia, contratos de datos en CI. La calidad como propiedad del sistema, no como revisión manual que se erosiona con el ritmo.

HONESTIDAD DE DISEÑO

Elegir lo defendible sobre lo vistoso: un scoring honesto sobre un evaluador semántico impresionante pero opaco; un stub fiel a sus contratos antes que una integración fingida.

03 LO QUE SE MIDIÓ, Y LO QUE NO

EL ENCUADRE HONESTO DEL IMPACTO

Cada pieza llegó a producción y pasó validación: peer-review matemático e instruccional, gates automáticos, baterías de pruebas y contratos de datos. **Eso sí se midió — la calidad del producto.**

> *Medir el efecto sostenido en el aprendizaje a escala habría exigido un rollout y una ventana de evaluación que no controlé. No hay, por tanto, métrica de outcome de alumno; y fingirla sería justo la deshonestidad que el resto del trabajo evita.*

Lo que este book sí puede defender, bajo cualquier repregunta, es el criterio: cómo se decidió qué medir, cómo se tradujo la teoría a software y cómo se sostuvo la calidad. Es lo que queda cuando se apagan los servidores — y es, exactamente, lo que estos productos fueron diseñados para demostrar.

04 CODA

HECHO EN SEIS MESES, CON LA IA COMO PALANCA

Todo se construyó en una ventana de seis meses con un modelo de desarrollo AI-first: la IA aceleró el *cómo construirlo*; la decisión de **qué medir y por qué** fue siempre propia. El portafolio es la evidencia de esa división del trabajo — y de que, bien dirigida, esa palanca permite a una sola persona razonar y entregar a la escala de un equipo.