Can self-play remember its past?

Un experimento sobre opponent pools — y qué me enseñó sobre poner a prueba mis propias ideas

Pedro Pérez Blanco

Self-play can forget

El agente entrena contra su yo más reciente… y solo contra él.

Aprende a batir
la estrategia de ayer

Olvida cómo defenderse
de la de anteayer

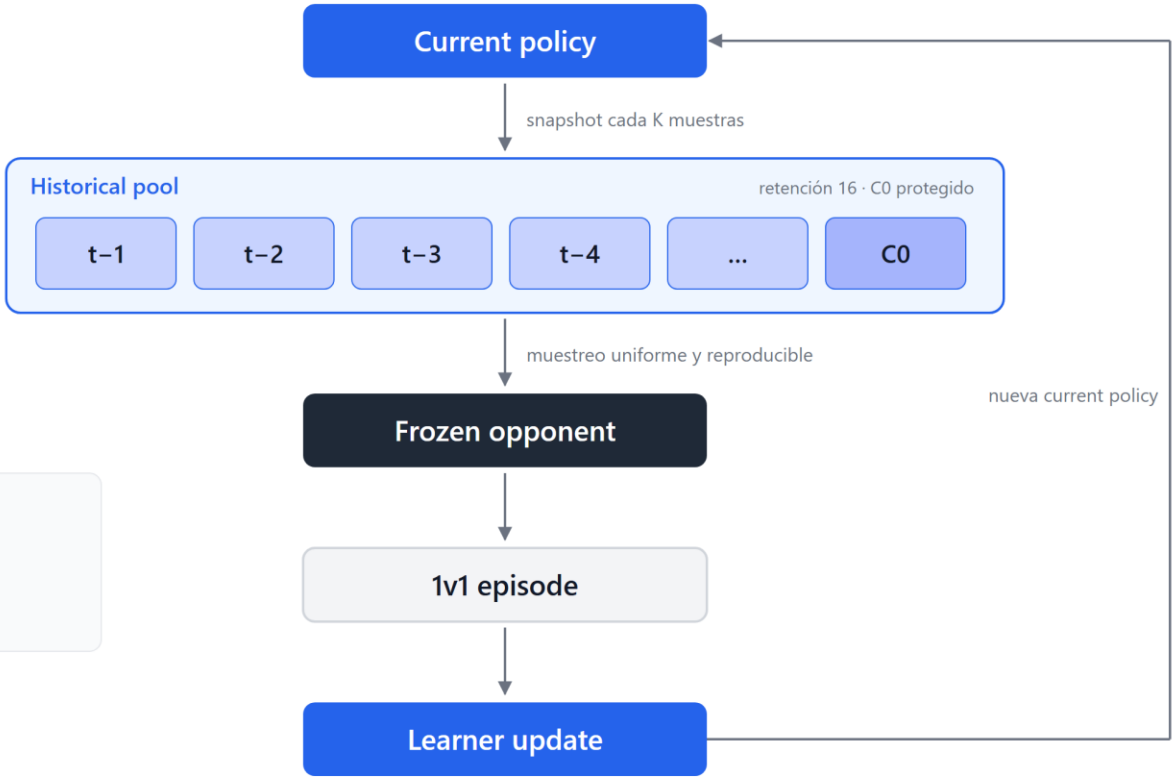
Vuelve a una versión
de la que ya había escapado

Mejorar contra el rival de hoy no significa mejorar en general.

Give it a memory

La idea: entrenar contra tu propio pasado

En vez de una única copia congelada, un pool de versiones históricas



Control (brazo A)

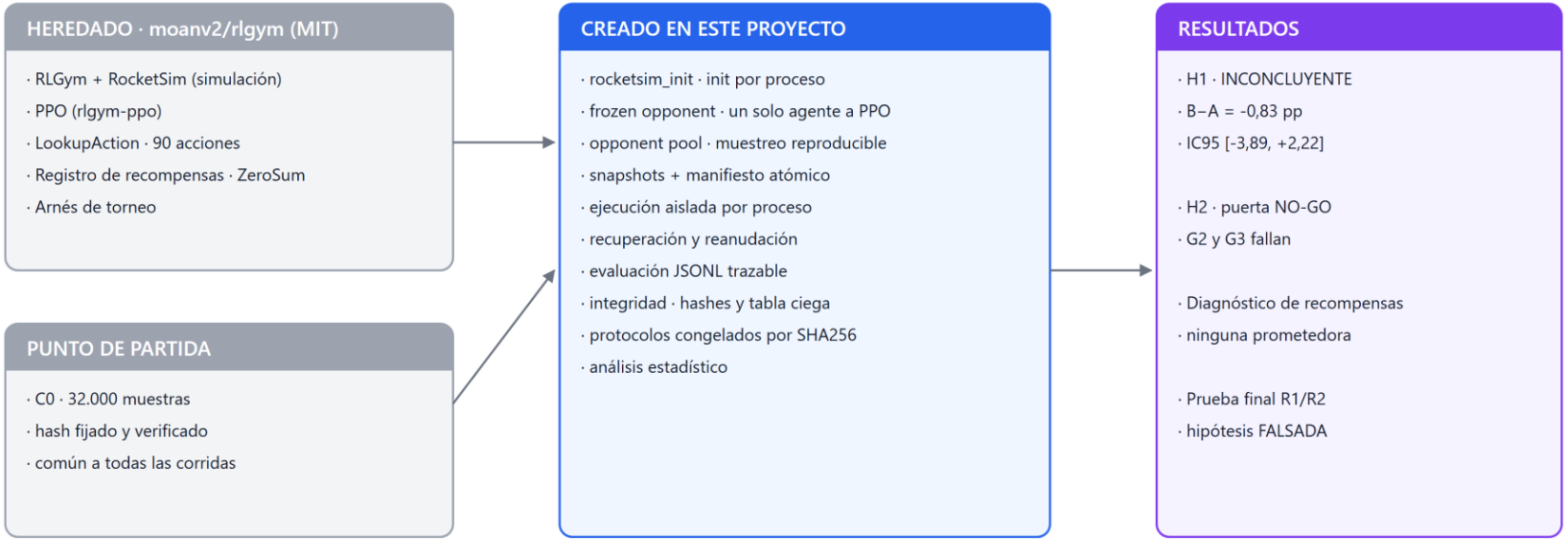
retención 1: el pool guarda solo la copia más reciente

Mismo código, misma clase.

What I built on top

Arquitectura: qué construí encima de qué

En gris el proyecto heredado · en azul la capa experimental propia · en violeta los resultados



5.401 líneas propias en 24 archivos

Ningún archivo heredado fue modificado · verificado con git status

0

archivos heredados modificados

Fuente: docs/CONTRIBUTIONS.md

The experiment, frozen before running

BRAZO A · control

Un rival congelado
Retención 1

504.000 muestras

BRAZO B · tratamiento

Pool de 16 instantáneas
Retención 16, C0 protegido

504.000 muestras

Mismo punto de partida · mismas semillas emparejadas · mismo presupuesto al dígito

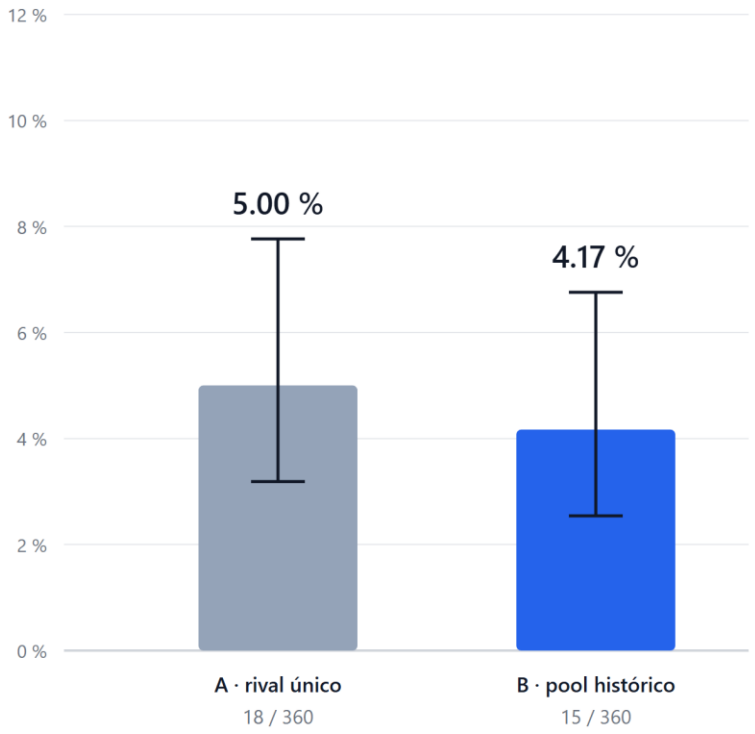
Criterio de decisión congelado por SHA256 antes de entrenar: 8ae06382…

The result

H1: entrenar contra un pool de versiones pasadas

Tasa de victoria frente a tres rivales reservados · 720 partidas

Tasa de victoria, con intervalo de Wilson 95 %



Diferencia B − A, con IC bootstrap 95 %

INCONCLUSIVE

El intervalo cruza el cero: no hay efecto detectable.
Cambiar la semilla movía el resultado más que cambiar el método.

Fuente: results/h1/analisis-h1.json · 3 semillas × 2 brazos × 3 rivales × 40 partidas

Why it couldn't tell

5 %

ganaban ambos brazos

rivales entrenados
4.000 veces más

1,55 %

se movió la política

en 504.000 muestras

0,0074

distancia entre «rivales distintos»

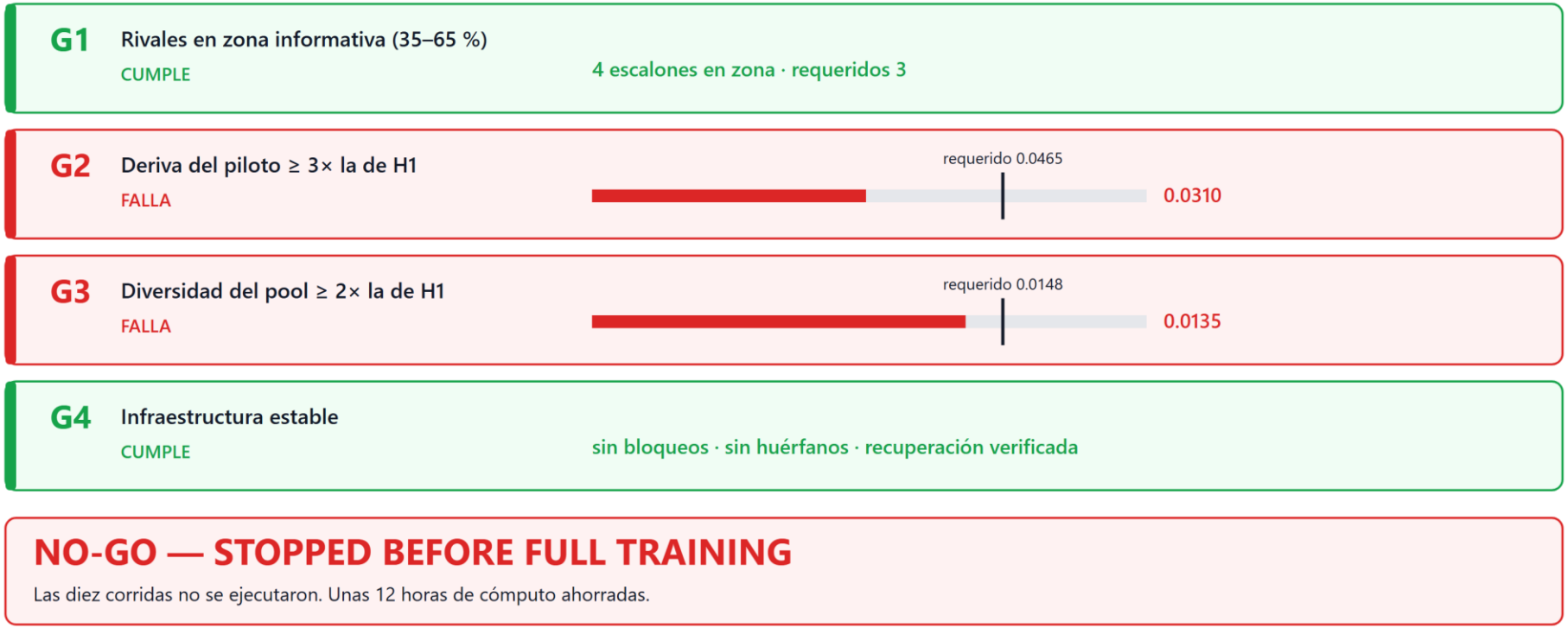
todas dentro
del 1,5 %

El tratamiento apenas se diferenciaba del control.

H2: stopped before full training

La puerta de H2: umbrales escritos antes de gastar el cómputo

Se evalúa antes de entrenar. Si falla alguno, el experimento no se ejecuta.

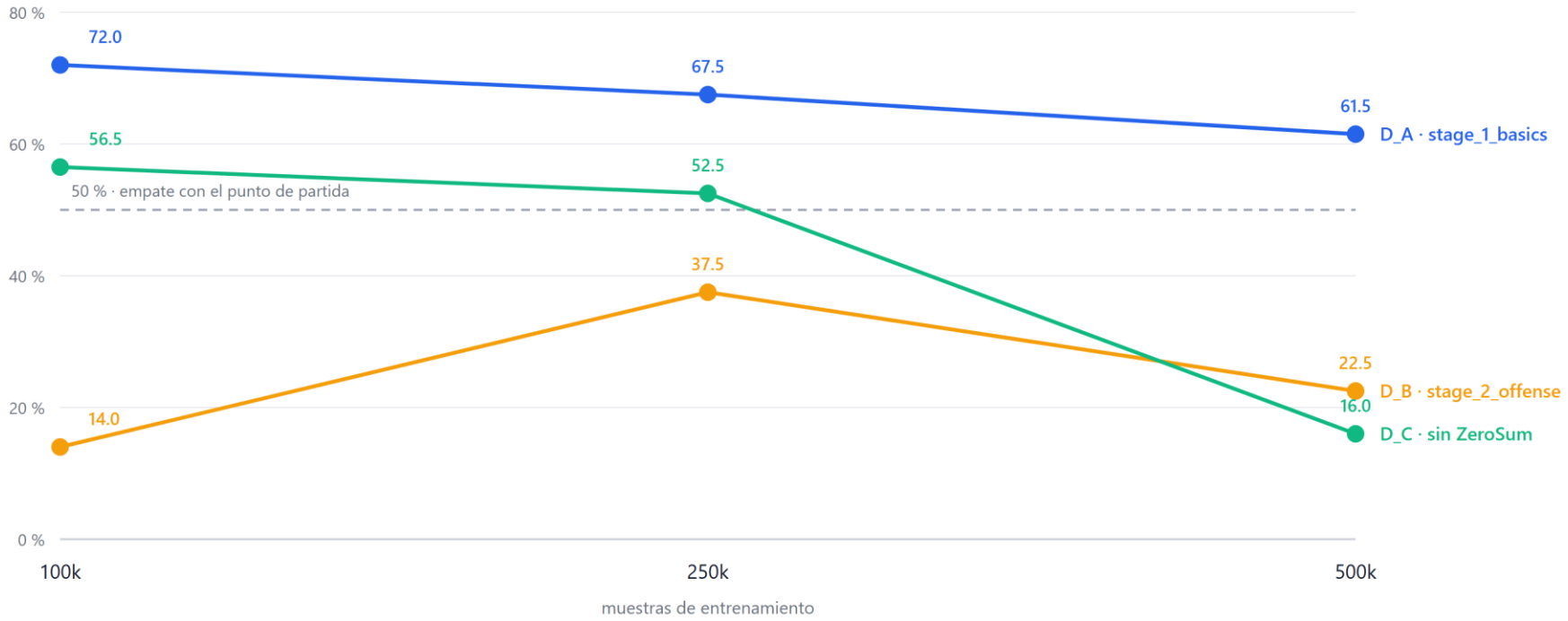


Fuente: results/h2/puerta.json

Then something worse showed up

Tres recompensas, ninguna con mejora sostenida

Tasa de victoria contra C0 · 200 partidas por punto

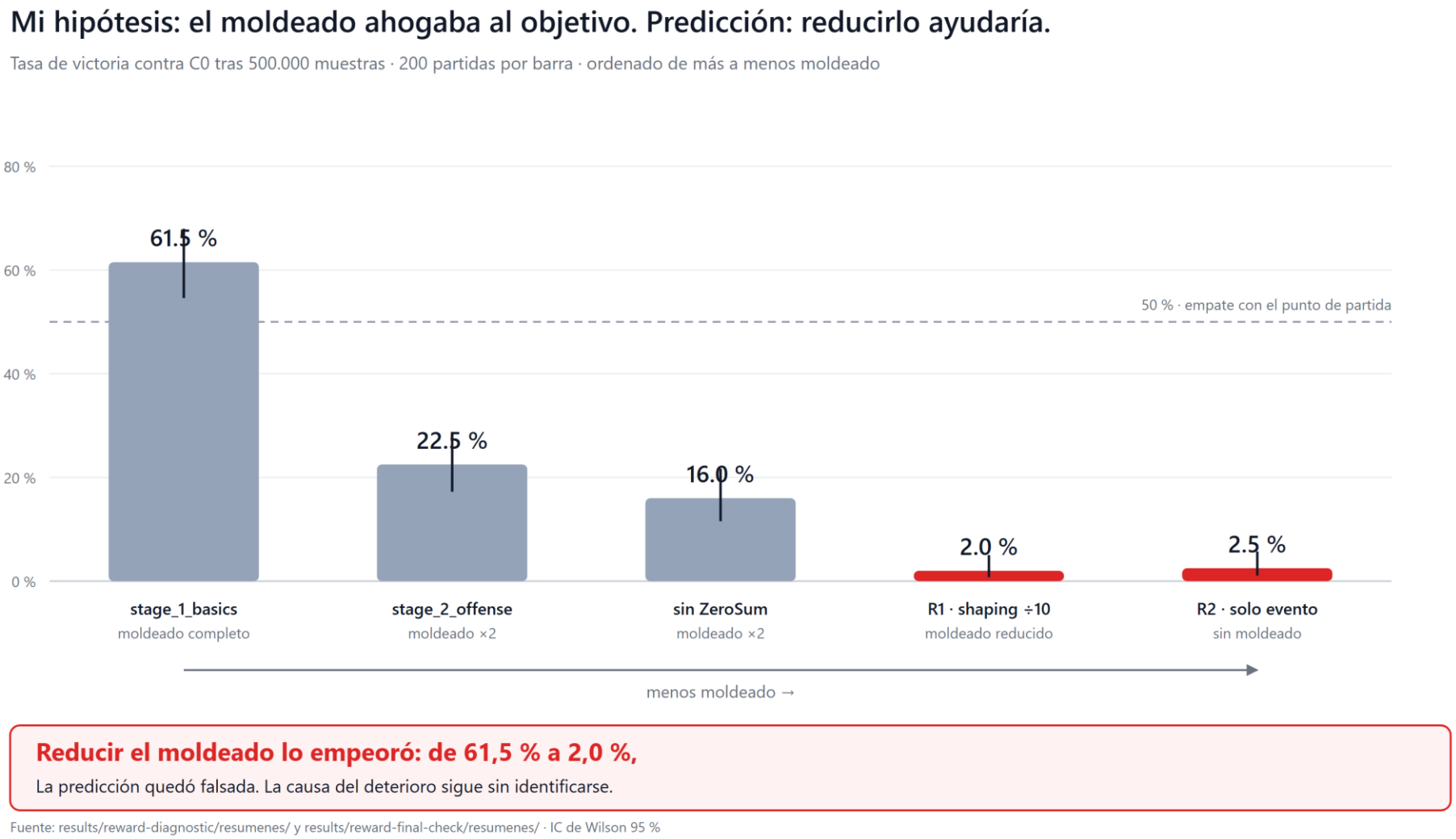


Ninguna mejora de forma sostenida al aumentar el presupuesto.

Si entrenar más empeora, comparar dos métodos por tasa de victoria compara dos formas de empeorar.

Fuente: results/reward-diagnostic/resumenes/

I had an explanation. It was wrong.



What this actually demonstrates

Engineering	aislamiento de procesos, recuperación de interbloqueos, integridad por hashes
Experiment design	criterios preinscritos, brazos emparejados, puertas de decisión
Reproducibility	protocolos con SHA256, semillas fijadas, evaluación sin duplicados
Statistics	bootstrap, intervalos de Wilson, y detectar que mi unidad experimental era incorrecta
Falsification	formular una hipótesis propia y aceptar que los datos la tumben

Descarté una configuración por 0,5 pp · detuve un experimento por 0,0155 de deriva

No proof that opponent pools help — pero un sistema capaz de decirlo

H1 INCONCLUYENTE · H2 NO-GO · hipótesis FALSADA
La causa del deterioro sigue sin identificarse

github.com/ppblanco · informe completo en docs/PROJECT_STORY.md