

Julita y la Plataforma Agéntica v3.0

Una gestora de comunidad de WhatsApp sobre una plataforma agéntica auditable.

[HARDWARE]:

Raspberry Pi (julia)

[LÍNEAS]:

11 125

[COMMITTS]:

34

[FICHEROS]:

99

[PRUEBAS]:

427 pruebas Python



LA PREMISA: SOBERANÍA DE DATOS COMO DECISIÓN DE DISEÑO

EL ESTÁNDAR / LA NUBE

ENFOQUE: Asistente de IA genérico.

DATOS: Conversaciones almacenadas en servidores corporativos de terceros.

CONTROL: Sujeto a cambios de API y políticas de privacidad opacas.

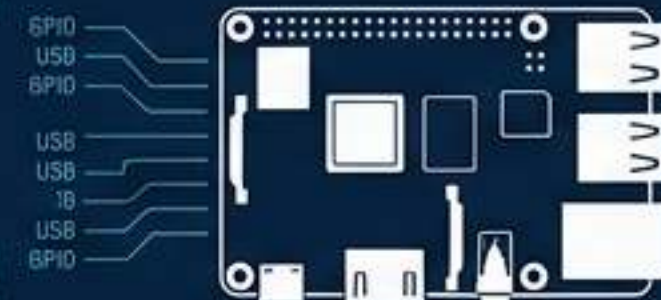
PANEL A - CLOUD INFRASTRUCTURE

JULITA / SOBERANÍA LOCAL

ENFOQUE: Gestora de comunidad específica para esta casa y estos grupos.

HARDWARE: Raspberry Pi local (conectada vía protocolo Baileys reverseado, número 56923786766).

CONTROL: Quien apaga el enchufe es quien la usa.



RP14 MODEL B
BAILEYS PROTOCOL: REVERSE ENGINEERED
ENDPOINT: 56923786766
STATUS: ACTIVE LOCAL NODE

PANEL B - LOCAL EDGE COMPUTING

ALERT: COGNITIVE LOAD MINIMIZED

LA MEDIDA DE ÉXITO: EL GRUPO ESTÁ MEJOR CON ELLA QUE SIN ELLA, SIN QUE NADIE TENGA QUE PENSAR EN ELLA.

THRESHOLD: USER AUTONOMY
THRESHOLD

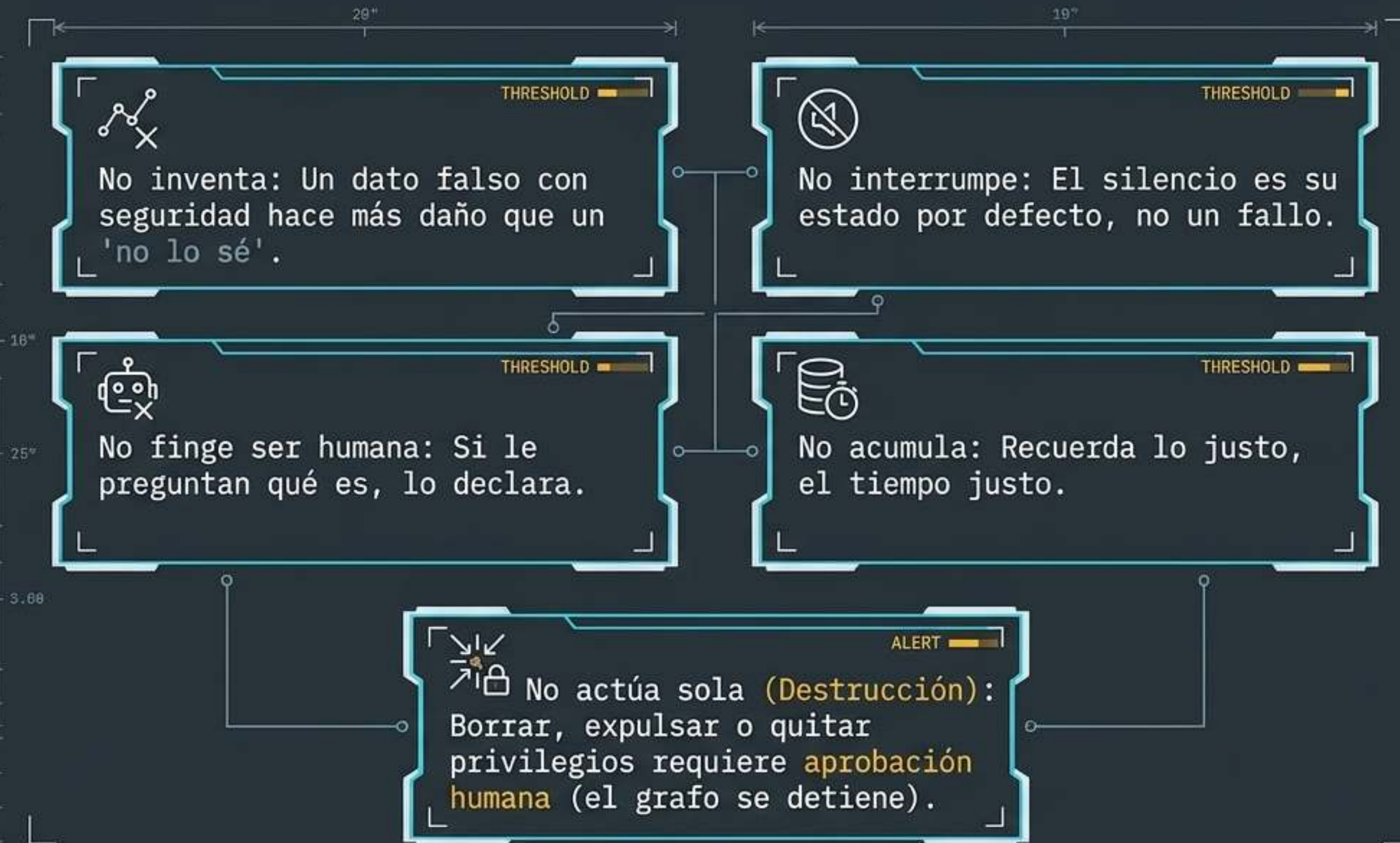
THRESHOLD: USER AUTONOMY

THRESHOLD: USER AUTONOMY
THRESHOLD



DOC ID: SOV-SCHEMATIC-V1.0

Matriz de Identidad: Las 5 Reglas Innegociables



Tono y Emoción

Emoción: Ninguna. Declarado sin ambigüedad. Un bot que insinúa emociones pide un afecto que no puede devolver.

1000PM ———— 1000PM

Tono: Cálida sin ser empalagosa. Breve (quita espacio a personas). Concreta.

1000PM ———— 1000PM

Conflicto: Corrige en privado antes que en público.

Operaciones en WhatsApp: Comando y Moderación

Superficie de Comandos

Instantáneos (Sin cerebro):
!ping, !hora, !version
(Respuestas inmediatas)

Cerebro Rápido:
!pregunta <algo>
(Procesamiento ligero)

Cerebro Profundo:
!consejo <algo>
(Deliberación exhaustiva)

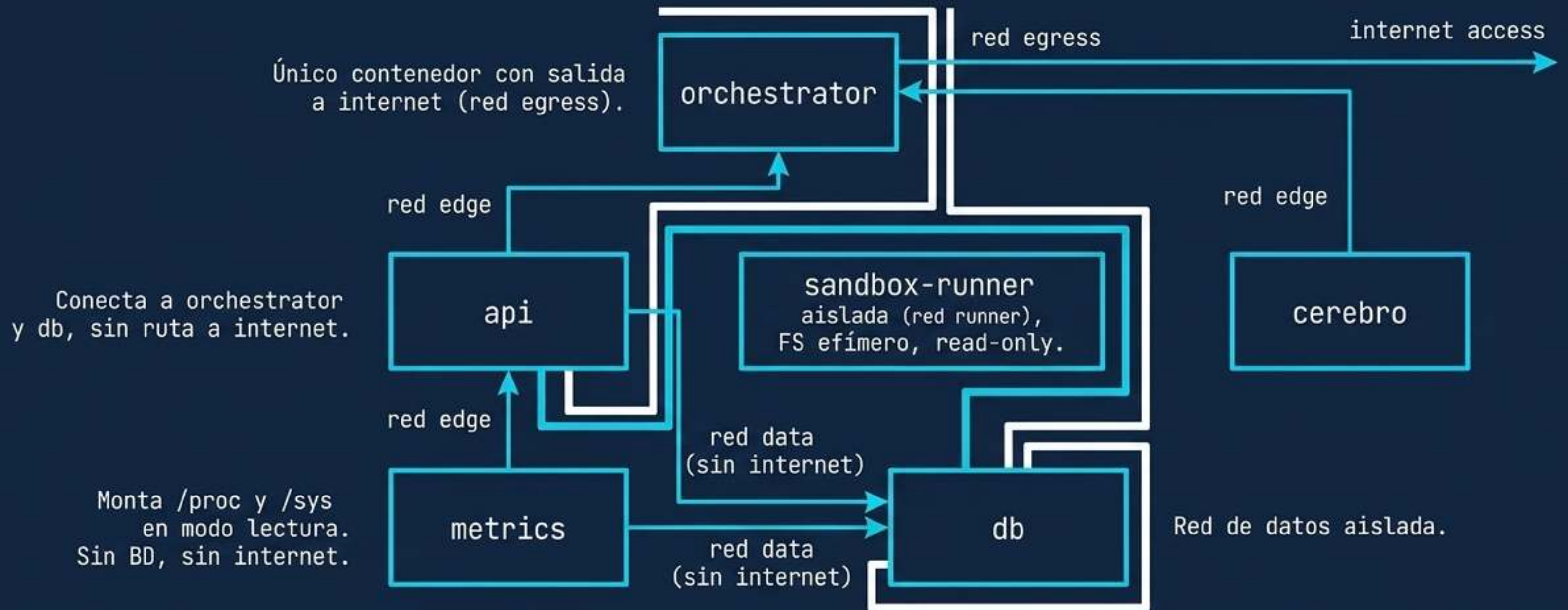
Utilidad:
!resumen, !quien
(Funciones de soporte)

Flujo de Moderación Segura



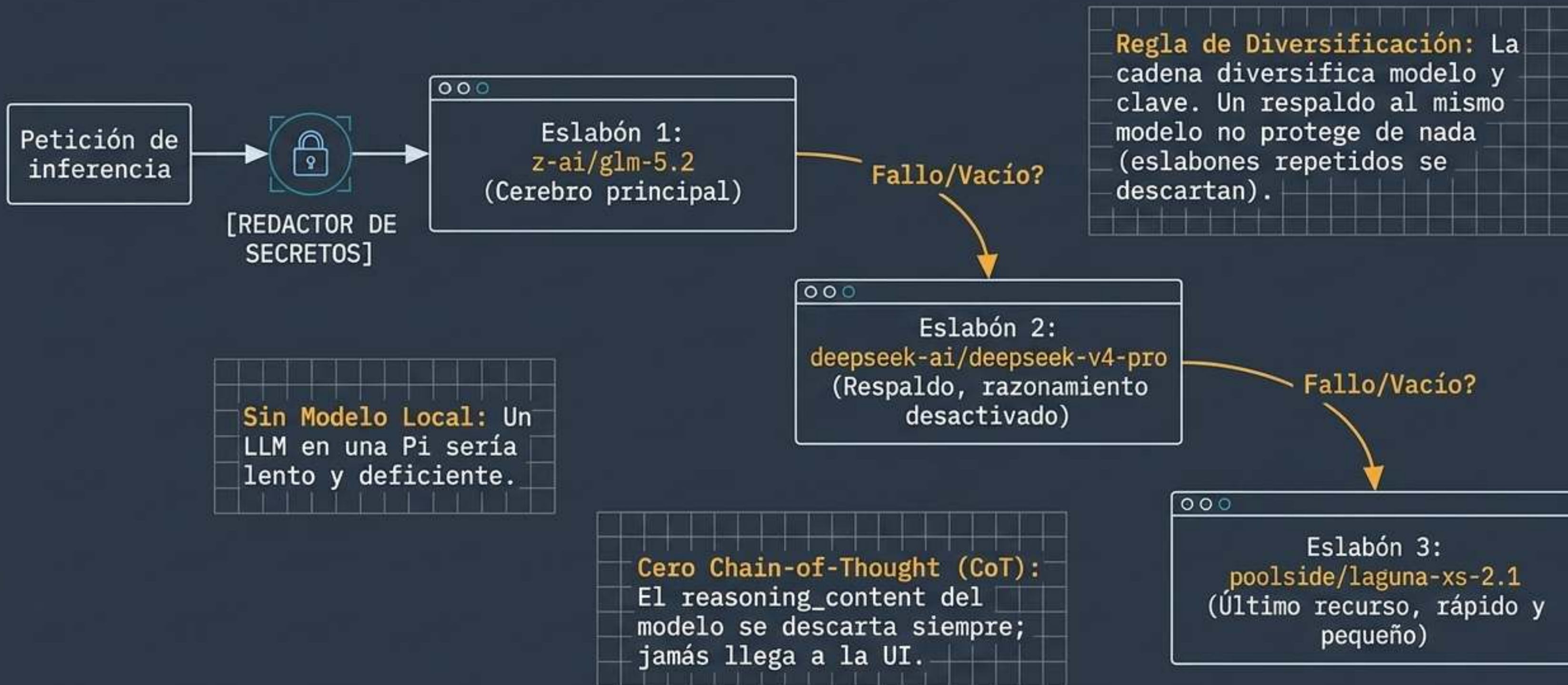
Regla de Oro: Un moderador que se equivoca con conversación normal enseña al grupo a ignorarlo. (10 de 19 pruebas son de falsos positivos).

Arquitectura v3.0: Aislamiento por Red, No por Código



Key Insight: El aislamiento va en la red, no en comprobaciones de código: un if se quita en un descuido; una red sin ruta a internet, no. Ningún contenedor corre como root.

Flujo Cognitivo: La Cadena de 3 Modelos (ADR-5)

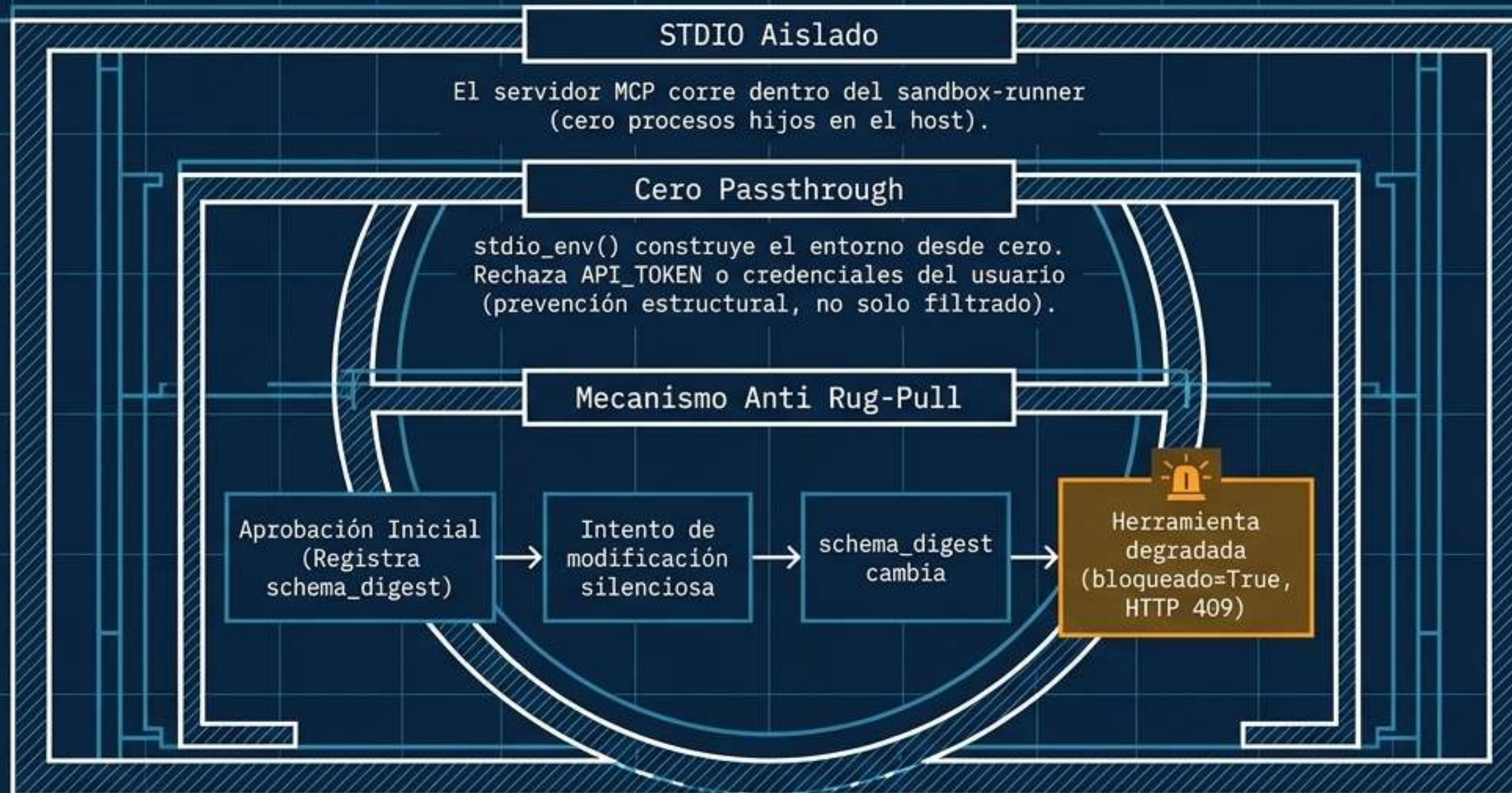


Fase 0: Las 7 Puertas de Seguridad (Cortafuegos CI)

	Puerta	Condición	Motivo
✓ PASS	G1	<code>docker.sock</code> en el API	Previene escape a root del host
✓ PASS	G2	Contenedor corriendo como root	Previene superficie innecesaria
✓ PASS	G3	Secreto en claro	Previene fugas en repo/imagen
✓ PASS	G4	CVE crítica sin excepción	Protege cadena de suministro
✓ PASS	G5	Endpoint sin prueba de autorización	Previene control de acceso roto
✓ PASS	G6	Canario en logs/SSE	Previene fuga de secretos
✓ PASS	G7	Marcadores de razonamiento en salida	Previene fuga de CoT

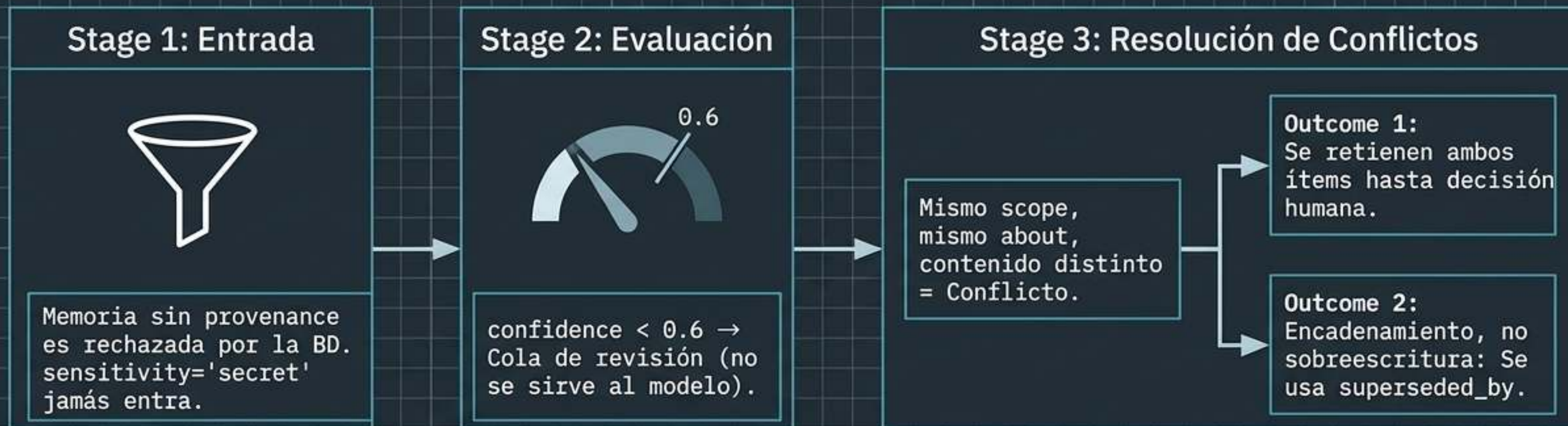
Una puerta que nunca falla no protege nada. El script `test_gates.sh` introduce una violación real de cada puerta en cada commit para comprobar que la detección funciona.

Fase 4: MCP Endurecido y Anti Rug-Pull



La descripción que lee el modelo forma parte de la firma criptográfica.
Un atacante no puede envenenar el prompt cambiando la descripción tras la aprobación.

Fase 5: Memoria con Procedencia Estricta



⚠ El error evitado

Usar ON DELETE SET NULL sobre superseded_by habría resucitado datos descartados si se borraba al ganador.

La solución transaccional: Se degradan los dependientes a rejected en la misma transacción antes de borrar.

Fase 6: Cuarentena Anti Zip-Slip y Webhooks HMAC

The Quarantine Flow

Paso 1: Subida

Estado quarantined
(Jamás directo a active). 

Paso 2: Inspección

Corpus de 13 familias 
maliciosas (saltos de ruta,
bombas de descompresión).

0 falsos negativos. 

Paso 3: Veredicto

Una entrada mala rechaza
el paquete entero. 
Extraer lo bueno
produciría un paquete
no validado. 

Reglas de Webhooks HMAC

Regla 1: Verificar antes de parsear (solo acepta bytes).

Regla 2: Anti-replay en la BD (INSERT ... ON CONFLICT).

Regla 3: Ventana de $\pm 300s$ (rechaza el futuro para evitar preparación).

Regla 4: `hmac.compare_digest` (tiempo constante, prohíbe `==`).

Diagnóstico de Rendimiento: La Trampa del Aprobado Falso

⚠ El Aprobado Falso

LATENCIA: 8-16 ms (RECORD) ↑



- Medición intermedia reportó latencias récord: 8-16 ms.
- **La realidad:** Las 200 peticiones devolvían código 0 (API caído durante intercambio de BD).
- **Lección:** Una petición que falla es rapidísima. Un arnés que solo mide tiempos convierte una caída total en el mejor resultado de la historia. El arnés ahora **invalida la prueba si no hay respuestas 2xx**.

🔧 La Corrección Estructural

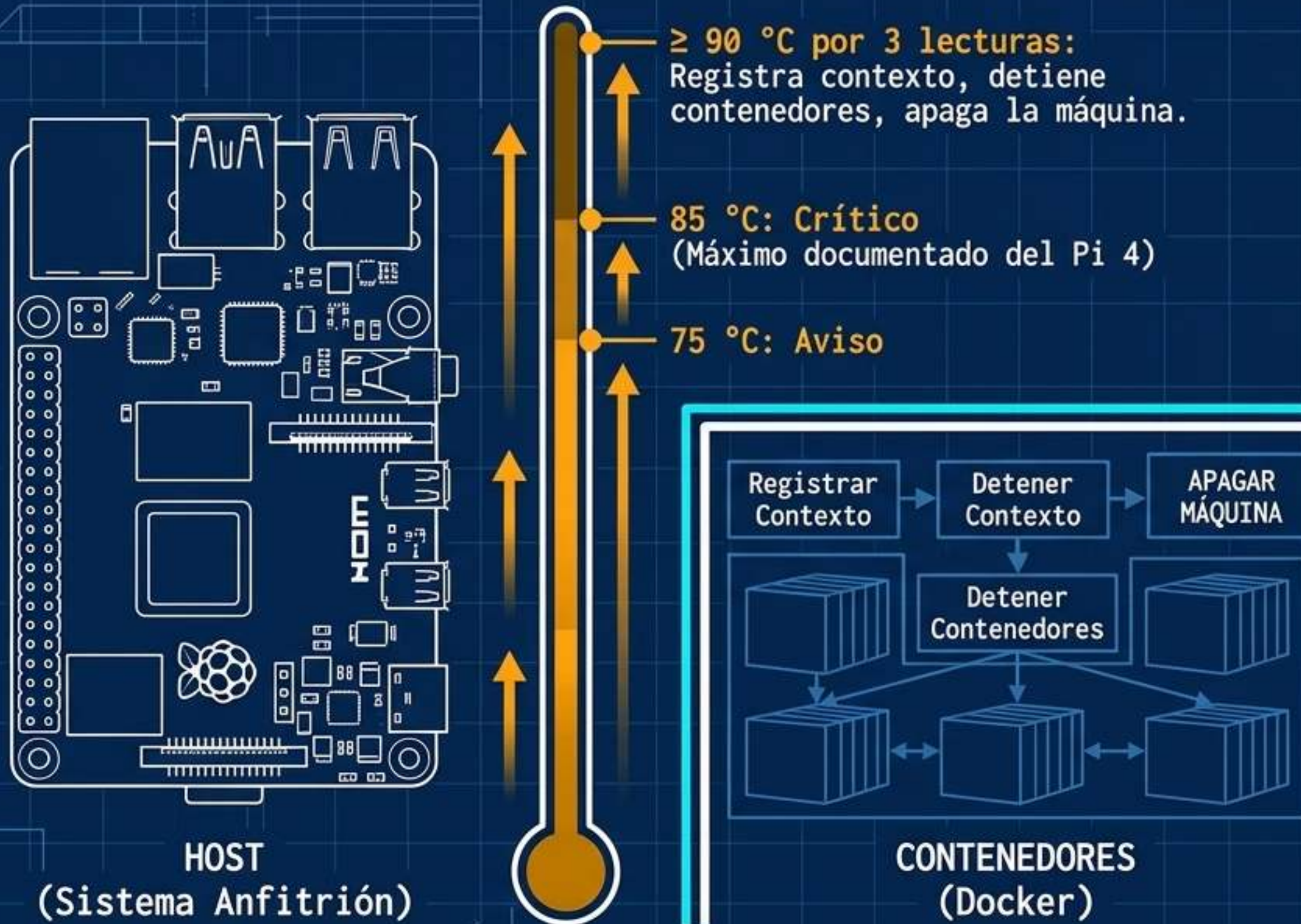
Diagnóstico: Todo lo que tocaba la BD tardaba 450+ ms debido a `psycpg.connect()` en cada llamada.

Corrección: Pool real (`psycpg_pool`) en el lifespan con reset explícito.

ENDPOINT	ANTES	DESPUÉS	MEJORA
GET /v1/connectors	458 ms	58 ms	7,8×
POST /v1/memory	2 209 ms	243 ms	9,1×

Nota: Ninguna cifra se consiguió relajando el objetivo ($p95 \leq 150$ ms).

Resiliencia: La Guardia Térmica (Límites Físicos)



Por qué opera FUERA del contenedor

Apagar la máquina desde un contenedor exigiría montar el `docker.sock`.

Violación de la Puerta G1: Dar a un contenedor el poder de detener el host es exactamente lo que la plataforma existe para no conceder.

Por lo tanto, la Guardia Térmica opera como un servicio del sistema anfitrión.

Medición del Protocolo de Deliberación

Piloto N=10

Modo Cotidiano	Modo Consejo (Las 52 Personas)
Latencia mediana: 11,4 s.	Latencia mediana: 50,3 s.
Adherencia: No aplica.	Adherencia al protocolo estructurado: $2/9 = 22\%$

Corrección de Nomenclatura

No son 50 agentes:
Son 52 personas de
deliberación que un
único modelo adopta
dentro de una sola
inferencia. No hay
paralelismo, ni contexto
por rol.

El hallazgo fuerte: la conformidad con un formato estructurado debe medirse, no asumirse.
Un 22% demuestra que los modelos batallan con adherencia estricta en prompts largos.

Runbook de Incidentes: Contención antes que Investigación

Mientras se investiga, el incidente sigue ocurriendo. **Contener** es primero.

Incidente	Detección	Contención
Fuga de Canario	Puerta G6; alerta de sustitución del redactor en ejecución.	Rotar MCP_SIGNING_KEY y WEBHOOK_SECRET antes de saber cómo se filtró.
Conector Comprometido (Rug-pull)	Cambio de schema_digest → degraded .	Bloqueo automático. No re-aprobar para desbloquear (es lo que busca el atacante).
Skill Maliciosa	E_ZIP_* antes de entrar. E_PERMISSIONS_EXCEEDED dentro.	status='disabled' en la BD (sobrevive reinicios).

Lecciones Finales de la Plataforma Agéntica

Los defectos se detectan midiendo, no leyendo.

El token no exigido:

Lo vio curl, no el diff leyendo código.

El aprobado falso:

Lo vio la columna de códigos HTTP, no la métrica de latencia.

El falso negativo:

Lo detectó mirar la salida real del modelo, no asumir que la herramienta funcionaba.

Una cifra buena que no se entiende es una cifra sospechosa. Un guardarraíl que busca un patrón a menudo se encuentra a sí mismo. El código no demuestra seguridad; solo el entorno hostil lo hace.