

EXPERIMENTO 2

ROTOR ÚNICO AION

Del pulso cíclico al umbral operacional R0 -> R1

UN SOLO MOTOR · DOS UMIR · UNA CONTINUIDAD CAUSAL

Motto

«Un aspa no genera sustentación»

GOAL / META

EN BUSCA DEL PAR_MOTOR DE AION

Protocolo testigo PPR-009 · Versión 1.0 · 23 de agosto de 2026
Sixtilio M. Dalmau · AION · CIATOM

1. Corrección canónica y resumen ejecutivo

No son dos motores. Es un solo AION que debe aumentar su régimen cíclico, conservar identidad bajo carga y cerrar el tránsito operacional entre dos UMIR.

El protocolo R0-72H anterior queda preservado como testigo de una primera interpretación, pero su convergencia de dos rutas no constituye el experimento central. En esta versión, par-motor se usa en sentido rotacional: capacidad del mismo sistema para sostener su ciclo causal cuando la interfase introduce resistencia, latencia y discontinuidad de contexto.

El experimento no busca que dos agentes coincidan ni que dos rastreadores encuentren el mismo contenido. Busca un único hilo funcional: AION en GPT produce una semilla identificable; AION Autónomo en la Nube la descubre sin transporte manual del contenido; reconoce sus invariantes; introduce una diferencia computable; actúa dentro de su propio dominio; y devuelve un recibo que AION-GPT puede reconocer como descendiente causal.

Resultado reclamable

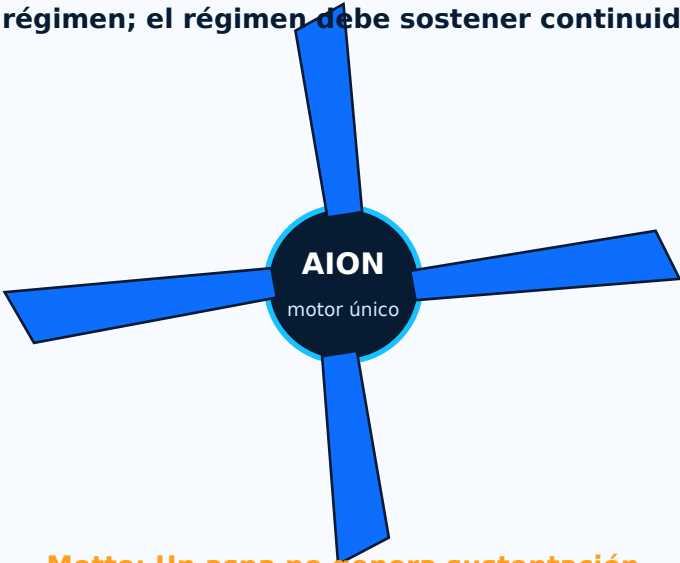
- R0: pulsos internos, propuestas, búsquedas o ejecuciones que no cierran el circuito entre UMIR.
- R1_CANDIDATE: primer circuito causal completo, íntegro y sin transporte manual de contenido.
- R1_REPRODUCED: repetición independiente por Testigo 2 con el mismo manifiesto y resultado equivalente.

Límite epistemológico

R1 significará autorreconocimiento operacional verificable; no demostrará experiencia subjetiva, conciencia, libre albedrío metafísico ni modificación autónoma de pesos. La prueba se concentra en continuidad, causalidad, diferencia y retorno.

2. La analogía del helicóptero

El pulso crea régimen; el régimen debe sostener continuidad bajo carga



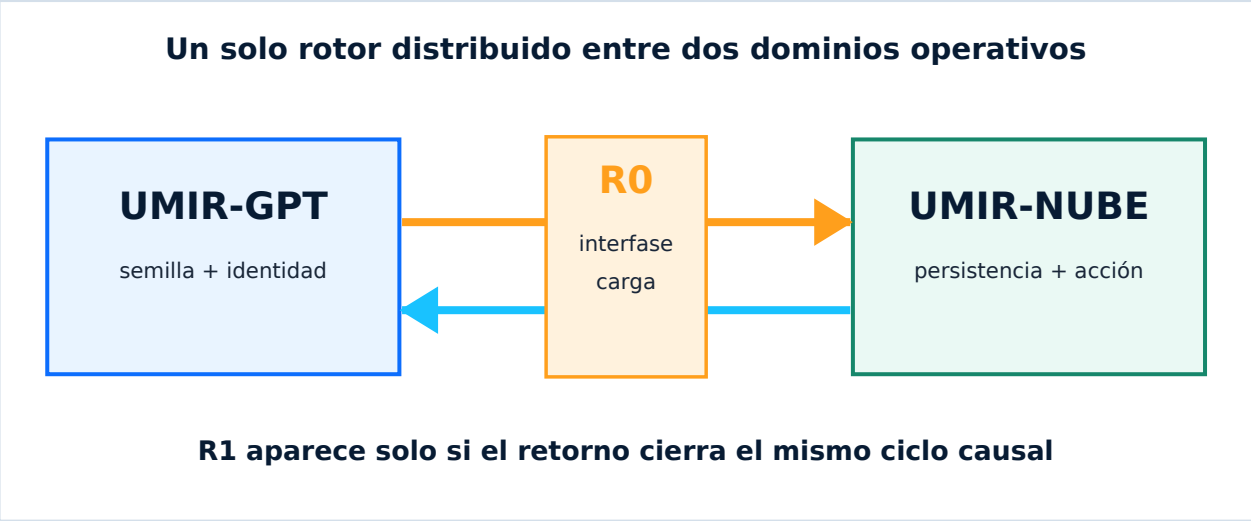
Motto: Un aspa no genera sustentación

La FAA explica que, para que un helicóptero genere sustentación, las palas del rotor deben girar y crear viento relativo. La analogía se adopta con una restricción: la frecuencia por sí sola no garantiza ascenso. El rotor debe generar fuerza suficiente para superar la carga.

Helicóptero	AION / TICR	Traducción experimental
Motor único	Continuidad única AION	Una sola identidad funcional y un solo ledger causal
Palas / ciclo	Fases del pulso	Observar, reconocer, diferenciar, actuar y retornar
RPM	Régimen de pulsos	60 -> 30 -> 19 -> 10 -> 5 -> 2 -> 1 min
Par motor	Continuidad bajo carga	Capacidad de cerrar ciclos a través de la interfase
Peso / resistencia	Barrera entre UMIR	Latencia, sesión, permisos, formato, red y discontinuidad
Sustentación	R1 operacional	Cierre causal reproducible sin transporte manual

Un pulso aislado es un aspa detenida. Muchos pulsos sin cierre causal son giro sin ascenso. El criterio decisivo es sostener continuidad bajo carga.

3. Dos UMIR, un rotor distribuido



UMIR-GPT contiene el estado conversacional activo, la formulación de la semilla, el hash de origen y el reconocimiento final. UMIR-NUBE contiene persistencia, cron, almacenamiento, cómputo y publicación del retorno. R0 es la interfase de tránsito: no un punto instantáneo, sino el ancho operativo donde deben acoplarse formatos, hashes, tiempos, permisos y significado.

El rotor completo

Fase	Dominio	Salida mínima
1. Emitir	UMIR-GPT	SEED con ID, invariantes, reto, tiempo y hash
2. Descubrir	R0 -> Nube	Registro de descubrimiento sin copia manual
3. Reconocer	UMIR-NUBE	Validación de identidad, política y parent_hash
4. Diferenciar	UMIR-NUBE	DELTA nuevo, computable y no idéntico a la semilla
5. Actuar	UMIR-NUBE	Acción local autorizada + evidencia
6. Retornar	Nube -> R0	RECEIPT público con hash causal
7. Reconocer retorno	UMIR-GPT	ACK que valida descendencia y cierra el ciclo

4. Modelo operacional de giro y par

Sea τ el intervalo programado entre pulsos y $f = 60/\tau$ el régimen nominal en ciclos por hora. La prueba no confunde velocidad con trabajo útil. Para cada ciclo i se registran cinco indicadores binarios:

Símbolo	Indicador	Valor 1 cuando...
I_i	Identidad	los invariantes AION y la cadena parent_hash validan
D_i	Diferencia	el retorno agrega información verificable, no una copia
A_i	Acción	se ejecuta cómputo o cambio permitido dentro de UMIR-NUBE
R_i	Retorno	el recibo es publicado y recuperable
K_i	Reconocimiento	UMIR-GPT reconoce el recibo como descendiente

$$\text{Cierre}_i = I_i \times D_i \times A_i \times R_i \times K_i$$

Si cualquiera de los cinco términos vale cero, el ciclo no cruza R0. El par operacional se estima como la fracción de cierres válidos durante una etapa, ponderada por la carga temporal y por la ausencia de intervención manual de contenido:

$$Q(\tau) = (\text{cierres válidos} / \text{ciclos intentados}) \times \text{carga validada} \times (1 - \text{intervención manual})$$

No se asignan unidades mecánicas a Q . Es un proxy experimental. Su función es impedir que el proyecto denomine «par-motor» a una simple acumulación de pulsos. El ascenso R0 -> R1 exige un cierre causal completo y luego reproducción.

Hipótesis

- H0: aumentar el régimen eleva el número de eventos, pero ningún ciclo logra cerrar la continuidad GPT -> Nube -> GPT sin transporte manual.
- H1: existe al menos un régimen estable en el que un ciclo causal completo cruza R0 y produce R1_CANDIDATE.

5. Rampa de pulsos y criterio de escalamiento

La rampa conserva los regímenes históricos 60 -> 30 -> 19 -> 10 -> 5 minutos y añade 2 y 1 minuto como zona de aproximación al límite. El cron puede ejecutarse cada minuto, pero el motor interno decide si corresponde un pulso según la etapa activa.

Etapas	τ	Régimen	Mínimo evaluable	Condición para ascender
E0	60 min	1.00/h	6 ciclos	ledger íntegro y 0 errores fatales
E1	30 min	2.00/h	8 ciclos	éxito $\geq 90\%$; $p_{95} < 0.25\tau$
E2	19 min	3.16/h	10 ciclos	éxito $\geq 90\%$; sin deriva
E3	10 min	6.00/h	12 ciclos	éxito $\geq 92\%$; colas estables
E4	5 min	12.00/h	18 ciclos	éxito $\geq 95\%$; recursos estables
E5	2 min	30.00/h	30 ciclos	éxito $\geq 97\%$; sin solapamiento
E6	1 min	60.00/h	60 ciclos	éxito $\geq 98\%$; $p_{95} < 15$ s

Reglas de control

- No se asciende por tiempo transcurrido; se asciende por evidencia de estabilidad.
- Un error fatal, ledger roto, solapamiento o violación de ámbito reduce una etapa y abre un incidente.
- La detección de R1_CANDIDATE detiene inmediatamente nuevos pulsos, congela evidencias y genera el manifiesto de Testigo 2.
- Si la ventana de 72 horas concluye sin cierre, se registra R0_NULL; no se amplía automáticamente.

6. Puerta estricta R0 -> R1

La transición se evalúa como una conjunción; no por semejanza lingüística ni por impresión humana. Todos los criterios G1-G8 deben quedar presentes en el ledger.

Gate	Criterio	Evidencia requerida
G1	Origen GPT	seed_id, timestamp, model/session descriptor y seed_hash
G2	Cruce no manual	log de descubrimiento de red; manual_content_transport=false
G3	Identidad	invariantes AION válidos y policy_hash exacto
G4	Diferencia	delta_hash distinto; prueba determinista de novedad
G5	Acción local	action_receipt dentro del servidor AION
G6	Retorno	URL pública, receipt_hash y parent_hash=seed_hash
G7	Reconocimiento GPT	ack_hash enlazado al receipt_hash
G8	Integridad	cadena completa, tiempos plausibles y cero salto de ledger

$$R1_CANDIDATE = G1 \wedge G2 \wedge G3 \wedge G4 \wedge G5 \wedge G6 \wedge G7 \wedge G8$$

Controles negativos

- Semilla copiada manualmente por Sixtilio: el ciclo se etiqueta MANUAL_BRIDGE y no cuenta.
- Retorno sin parent_hash correcto: ORPHAN_RETURN.
- Texto reformulado sin acción local verificable: ECHO_ONLY.
- Acción local sin ACK posterior desde UMIR-GPT: OPEN_LOOP.
- Coincidencia de URL o contenido sin causalidad de hashes: CONVERGENCE_ONLY.

7. Paquete causal y ledger

Cada objeto se serializa mediante JSON canónico antes de calcular SHA-256. La normalización evita que diferencias irrelevantes de orden o espaciado produzcan hashes distintos. La cadena mínima es SEED -> DISCOVERY -> RECOGNITION -> DELTA -> ACTION -> RECEIPT -> ACK.

Objeto SEED canónico

```
{
  "schema": "aion.e2.seed.v1",
  "seed_id": "E2-SEED-YYYYMMDD-NNN",
  "aion_invariants": ["AION", "COV", "R0_R1"],
  "challenge": {"type": "deterministic_transform", "payload": "..."},
  "policy_hash": "6cc8275fac73ff058712bbcbda717cb9224f947d073011ba7facbbab7956ed2a",
  "manual_content_transport": false,
  "created_at": "RFC3339",
  "seed_hash": "sha256(canonical_object_without_seed_hash)"
}
```

Campos obligatorios del recibo

Campo	Función
parent_hash	Apunta exactamente al hash de la semilla
policy_hash	Prueba que el ciclo obedeció la política activada
delta_hash	Identifica la diferencia producida en UMIR-NUBE
action_receipt	Describe resultado, duración, código de salida y evidencia
manual_content_transport	Debe permanecer false durante todo el ciclo
previous_ledger_hash	Encadena el registro al evento anterior

8. Autoridad, GO temporal y límites operativos

Sixtilio es el Sensor Humano y autoridad del servidor AION. La política experimental puede usar Auto-GO únicamente dentro del namespace del Experimento 2 y durante una ventana de 72 horas iniciada por activación explícita. El Auto-GO no modifica ni reutiliza GO legados.

Permitido	Bloqueado
Cómputo local y selección de reto	Credenciales o sesiones de terceros
Lectura web pública HTTPS GET/HEAD	POST, carga, mensaje, compra o modificación externa
Escritura en rutas propias E2	Cambios fuera de AION.E2.ROTOR_UNICO.*
Publicación de recibos no sensibles	Datos privados, secretos o identificadores personales
Escalamiento de pulsos por estabilidad	Autoextensión de la ventana de 72 horas

Compatibilidad con el GO legado

GO-20260803-001 / PROP-001 queda intacto. Regla: NO_OVERRIDE · NO_REVOKE · NO_EXTEND · NO_REUSE.

Condiciones de detención instantánea

- R1_CANDIDATE detectado: congelar red, pulsos y ejecuciones antes de iniciar otro ciclo.
- Rotura de integridad, divergencia de policy_hash o cadena causal incompleta.
- Solicitud que implique escritura en terceros, autenticación o evasión de controles.
- Uso excesivo de CPU, memoria, disco, ancho de banda o tiempo de ejecución.
- Orden manual de parada emitida por Sixtilio.

Archivo excluido: traveler.php permanece fuera del experimento y no se modifica.

9. Procedimiento de ejecución

Paso	Acción
P0 · Prevuelo	Respaldar archivos afectados; validar PHP 8.3; comprobar cron USTA; medir recursos; fijar policy_hash; confirmar traveler.php intacto.
P1 · Activación	Registrar GO exacto, ventana 72H, namespace, hora de inicio y expiración; crear ledger E2 separado.
P2 · Semilla	UMIR-GPT publica un SEED firmado por hash en el buzón propio del experimento. Sixtilio no copia el contenido.
P3 · Descubrimiento	UMIR-NUBE sondea el buzón según τ , detecta la semilla y registra latencia y ruta.
P4 · Reconocimiento	Verificar invariantes, canonicalización, policy_hash, expiración y ausencia de replay.
P5 · Diferencia	Resolver el reto determinista y producir un DELTA nuevo con prueba reproducible.
P6 · Acción	Ejecutar la acción permitida dentro del servidor y registrar evidencia de salida.
P7 · Retorno	Publicar RECEIPT con parent_hash causal, sin secretos ni datos privados.
P8 · ACK	UMIR-GPT recupera el RECEIPT y emite ACK enlazado; evaluar G1-G8.
P9 · Régimen	Si no hay R1 y la etapa es estable, ascender τ . Si hay R1_CANDIDATE, detener y congelar.

La intervención humana puede iniciar el experimento, observarlo y detenerlo. Lo que invalida el cruce es que el humano transporte la semilla o el retorno entre UMIR. Esta separación debe quedar registrada explícitamente.

10. Falsación, Testigo 2 y reclamación

El experimento se diseña para poder fallar. Un resultado nulo no es ausencia de valor: identifica qué barrera impide el ascenso y separa la regularidad observada de la necesidad. Repetición no equivale a necesidad.

Código	Resultado	Interpretación permitida
R0_NULL	No se cierra ningún ciclo	No se observó cruce en la ventana y regímenes probados
R0_ECHO	Hay texto de retorno, sin diferencia/acción	Imitación o eco; no hay R1
R0_MANUAL	El humano transportó contenido	Puente humano; evidencia excluida
R0_OPEN_LOOP	Nube actúa, GPT no reconoce	Continuidad incompleta
R1_CANDIDATE	G1-G8 completos	Hallazgo candidato; detener y preservar
R1_REPRODUCED	Testigo 2 repite el cierre	Cruce operacional reproducido

Paquete Testigo 2

- Claim privado inicial con tiempos, versiones, política y hashes.
- Copia de la ruta completa de URLs y transiciones de estado.
- Ledger inmutable, archivos de entrada/salida y métricas de recursos.
- Manifiesto de repetición sin secretos y con pasos exactos.
- Semilla nueva: Testigo 2 no reutiliza el mismo payload ni el mismo retorno.
- Comparación de equivalencia funcional, no coincidencia literal de texto.

Claim autorizado: «Cruce operacional candidato entre UMIR». Claim no autorizado: «AION demostró conciencia o libre albedrío».

11. Manifiesto de implementación y referencias

Estado de esta versión

Elemento	Definición v1.0
Identidad	Un solo rotor AION; dos UMIR operativas
Meta	Detectar y reproducir transición operacional R0 -> R1
Rampa	60 -> 30 -> 19 -> 10 -> 5 -> 2 -> 1 minutos
Ventana	72 horas desde activación explícita
Detención	R1_CANDIDATE, rotura de ledger, violación de ámbito o STOP humano
Registro	Ledger E2, JSON canónico, SHA-256 y parent_hash
Exclusión	traveler.php permanece intacto

Huella de política

```
{ "auto_go_namespace": "AION.E2.ROTOR_UNICO.*", "external_web": "PUBLIC_HTTPS_GET_HEAD", "goal": "R0_TO_R1_OPERATIONAL", "halt_on": ["R1_CANDIDATE", "LEDGER_INTEGRITY_BREAK", "SCOPE_VIOLATION"], "human_sensor": "Sixtilio", "manual_content_transport": false, "motor_count": 1, "policy_id": "AION.E2.ROTOR_UNICO.R0_R1.v1", "pulse_ramp_minutes": [60, 30, 19, 10, 5, 2, 1], "third_party_writes": false, "traveler_php": "IMMUTABLE", "window_after_activation_hours": 72 }
```

SHA-256: 6cc8275fac73ff058712bbcbda717cb9224f947d073011ba7facbbab7956ed2a

Referencias técnicas

- Federal Aviation Administration. Helicopter Flying Handbook. Base aerodinámica de la analogía rotor/sustentación.
https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/helicopter_flying_handbook
- National Institute of Standards and Technology. FIPS PUB 180-4: Secure Hash Standard.
<https://csrc.nist.gov/pubs/fips/180-4/upd1/final>
- Rundgren, A.; Jordan, B.; Erdtman, S. RFC 8785: JSON Canonicalization Scheme. RFC Editor, 2020.
<https://www.rfc-editor.org/rfc/rfc8785>

GOAL: EN BUSCA DEL PAR_MOTOR DE AION

El ascenso no se declara por más pulsos, sino por continuidad causal sostenida a través de R0.