

Compilación de Consistencia Operacional (ILSC)
Documento A — Metodología universal: juez/compilador PASS/FAIL con
dominio ISAAC
Marco Antonio Isaac Alcuria

12 de febrero de 2026

Propósito. Este documento define una metodología universal —no una ontología— para decidir, de forma reproducible y auditable, si existe (PASS) o no existe (FAIL) una dinámica compatible con candados físicos inevitables y con anclajes observacionales, dentro de un dominio operacional explícito (ISAAC).

Idea guía. En lugar de “proponer una teoría y ajustar parámetros”, se especifica un compilador: toma entradas declaradas (objeto físico en una representación concreta, anclajes, supuestos técnicos) y devuelve un veredicto binario, junto con diagnósticos y medidas de rigidez.

Prioridad metodológica. En todo el documento, se privilegia el por qué antes que el qué: cada candado se introduce como condición de existencia operacional (qué se rompe si no está) y sólo después se formula en términos técnicos.

Contenido

1. Motivación y principio de universalidad operacional

1.1 Por qué un juez operacional (evitar maleabilidad y reinyección UV)

1.2 Universalidad por arquitectura: múltiples frontends, un solo juez

2. Nomenclatura y objetos operacionales

2.1 Sectores operacionales (SGO/SIA): definición mínima y justificación

2.2 Sector B (SIA): requisitos mínimos (no-backflow micro) y criterio de cruce

2.3 ISAAC como cierre operacional: umbral, techo espectral y regla de no reinyección UV

2.4 Dominio Ω_I y regla ISAAC de validez

2.4.a Protocolo de Canonización de Dominio (PCD): fijación no-ad-hoc de Ω_I

Por qué. Ω_I no puede ser un “parámetro libre” del compilador. Si Ω_I se elige post-hoc, la arbitrariedad se traslada de la teoría a la herramienta.

Qué. Se define PCD como el conjunto de reglas que fija Ω_I a partir de tres fuentes auditables: (i) ventana/resolución instrumental (Módulo Observabilidad & Instrumentación), (ii) validez y escala de corte de cualquier EFT usada (Módulo EFT & Renormalización Operacional), (iii) restricciones IR-safe cuando el frontend sea S-matrix/positividad (Módulo Amplitudes & Positividad).

Regla PCD-1 (Ventana). Ω_I debe estar contenido en la ventana donde el observable está definido y calibrado. Si el anclaje no trae ventana, no es consumible.

Regla PCD-2 (Resolución). Ω_I no puede exigir estructura sub-resolución: cualquier constraint que requiera variación más fina que el kernel de resolución viola ISAAC.

Regla PCD-3 (Validez EFT). Si se usa EFT, Ω_I debe satisfacer $E/\Lambda \leq \varepsilon_\Lambda$ declarado y se debe propagar error EFT en toda la corrida.

Regla PCD-4 (Sensibilidad). Se reporta estabilidad del veredicto al variar Ω_I dentro de un margen instrumental/teórico razonable. Si el veredicto cambia, se reporta explícitamente y se degrada el estado de 'PASS fuerte'.

2.5 Conjuntos y métricas: F , D_{obs} , I y rigidez R

3. Candados inevitables: familias por frontend (C0-C11)

3.0 Meta-candados (C0): operacionalidad, cierre y auditabilidad

3.1 Frontend S-matrix (C1-C6): dispersión relativista

3.2 Frontend correladores Euclídeos→Lorentzianos (C7): OS / reflection positivity

3.3 Frontend sistemas abiertos (C8): canales CPTP y consistencia A/B

3.4 Frontend térmico-equilibrio (C9): condición KMS

3.5 Frontend simetrías/anomalías (C10): anomaly matching

3.6 Frontend relativista local (C11): consistencia CPT como test de supuestos

4. Especificación del compilador ILSC

4.1 Entradas (frontends, parametrización, anclajes)

4.2 Salidas (PASS/FAIL, PASS fuerte, certificados)

4.2.a Taxonomía de veredictos: PASS/FAIL, PASS fuerte, FAIL certificado y NO-EVAL

Por qué. Un juez universal debe distinguir entre “inconsistente”, “consistente pero no rígido” y “no evaluable por falta de input auditable”. Si todo se fuerza a PASS/FAIL, el marco se vuelve injusto o ‘fetichista’.

Qué. Se adopta la siguiente taxonomía mínima:

- **PASS**: $I = F \cap D_{\text{obs}} \neq \emptyset$.
- **PASS fuerte**: $I \neq \emptyset$ y la región superviviente es rígida (poca dimensión efectiva o discreta; dominada por incertidumbre experimental, no por libertad teórica).

- **FAIL**: $I = \emptyset$ dentro de Ω_I , con diagnóstico.
 - **FAIL-certificado**: FAIL acompañado por witness/dual certificate cuando el problema es convexo (SDP/LP) o por un no-go operacional equivalente.
 - **NO-EVAL**: no evaluable porque falta (i) proyección $UV \rightarrow \Omega_I$ auditable, (ii) anclaje D_{obs} consumible (ventana/resolución/covarianza), o (iii) MRD mínimo para esa familia de constraints. NO-EVAL no es un ‘rescate’: es un estado explícito que obliga a completar el input.
- Regla.** NO-EVAL es obligatorio cuando la propuesta no puede compilarse sin introducir supuestos no auditados.

4.3 Algoritmo ejecutable (compilación)

4.4 Rigidez y discreción: R y dimensión efectiva local

5. Reproducibilidad y auditoría

5.a Protocolo de Canonización Numérica (PCN): tolerancias, discretización y convergencia

Por qué. En compilación numérica, la arbitrariedad puede reaparecer en tolerancias, mallas y stopping criteria. PCN convierte esas elecciones en reglas cerradas y auditables.

Qué. PCN fija requisitos mínimos:

- **PCN-1 (Tolerancias).** Declarar tolerancia global y tolerancias por constraint; reportar sensibilidad del veredicto al variar tolerancias en un rango razonable.
- **PCN-2 (Mallas/Discretización).** Para integrales, kernels o PSD, demostrar convergencia bajo refinamiento (al menos dos refinamientos).
- **PCN-3 (Semillas/Reproducibilidad).** Seeds, versiones, hashes y entornos (contenedor) obligatorios para todo experimento numérico.
- **PCN-4 (Criterio de paro).** Stopping criteria explícito; si se usa optimización, reportar certificados de optimalidad o cotas.
- **PCN-5 (Robustez).** Si el veredicto depende de detalles de discretización, se degrada a PASS débil o se emite NO-EVAL hasta resolver estabilidad.

5.1 Bloque mínimo de auditoría

5.2 Tests automáticos recomendados

6. Demo mínimo reproducible (MRD-1) end-to-end

6.a MRDs por capas: de ‘toy’ a ‘caso real’

Por qué. Un MRD trivial demuestra formato, pero no demuestra que el compilador sobreviva la ‘suciedad’ de datos reales. Para obtener PASS fuerte como metodología, se requieren MRDs escalonados.

Qué. Se definen tres niveles mínimos:

- **MRD-0 (toy).** Demuestra compilación (constraints + solver + auditoría) con un sistema pequeño.
- **MRD-1 (realista).** Un frontend no trivial (p.ej. positividad/EFT o SK con kernel físico) con un D_obs exportado en formato canónico, incluyendo covarianzas y ventana.
- **MRD-2 (caso real).** Aplicación end-to-end a un dataset real publicado: (i) exportación de D_obs con ventana/resolución, (ii) compilación de candados inevitables, (iii) PASS/FAIL con diagnóstico y, si aplica, certificado.

Regla. Sin MRD-1, el documento puede estar conceptualmente correcto, pero el estado metodológico se reporta como ‘PASS con rigidez media’ en lugar de PASS fuerte.

7. Supuestos (S1-S12) y loopholes (L1-L8)

7.a Meta-candado de estabilidad: ‘no mover arbitrariedad al compilador’

Por qué. Si el veredicto depende críticamente de Ω_I , N_{sub} , IR-treatment o tolerancias, el compilador se convierte en una fuente de arbitrariedad.

Qué. Se añade el meta-candado: toda corrida debe incluir (i) PCD (estabilidad frente a variación razonable de Ω_I), (ii) PCN (convergencia y sensibilidad numérica) y (iii) tratamiento IR declarado cuando aplique. Si cualquiera falla, el veredicto se degrada (PASS débil) o pasa a NO-EVAL.

8. Glosario mínimo y mini-ejemplo juguete

Referencias

1. Motivación y principio de universalidad operacional

1.1 Por qué un juez operacional (evitar maleabilidad y reinyección UV)

Cuando se intenta extender descripciones efectivas más allá de los regímenes donde el cierre está garantizado, aparecen dos riesgos: maleabilidad (capacidad de “explicar” cualquier dato añadiendo libertad no contrastable) y dependencia UV no operable (conclusiones que cambian al variar supuestos sobre escalas o resoluciones inaccesibles).

ILSC define un juez de existencia. La pregunta operacional no es “¿qué ontología es correcta?”, sino: ¿existe una dinámica compatible con (a) candados inevitables, (b) un dominio operacional explícito (ISAAC) y (c) anclajes observacionales?

- Falsabilidad fuerte. FAIL significa “no hay dinámica compatible bajo supuestos declarados y dentro del dominio”.

- Comparabilidad. Propuestas distintas se evalúan con el mismo juez.
- Reproducibilidad. El juicio puede ejecutarlo un tercero si se declaran entradas, tolerancias y versionado.

El juez no reemplaza los módulos; los organiza. Los módulos entran como entradas. El juez evalúa consistencia y rigidez; no “inventa” datos ni añade libertad oculta.

1.2 Universalidad por arquitectura: múltiples frontends, un solo juez

Una metodología “universal” no puede exigir que toda la física se exprese como amplitudes $2 \rightarrow 2$. En la práctica, la física se presenta en representaciones distintas: amplitudes de dispersión, correladores (por ejemplo en formulación Euclídea/lattice), dinámica reducida de sistemas abiertos, y estados térmicos. Pretender un único formalismo para todos esos objetos fuerza excepciones y genera arbitrariedad. La universalidad se obtiene con una arquitectura: múltiples frontends \rightarrow un lenguaje intermedio de constraints \rightarrow un juez único PASS/FAIL. Cada frontend traduce su objeto (amplitud, correlador, canal, estado) a una familia de candados inevitables. El juez permanece idéntico: interseca candados con anclajes dentro de ISAAC y cuantifica rigidez.

- Frontend S-matrix. Unitaridad + analiticidad/causalidad + crossing + boundedness (con protocolo IR/gravedad).
- Frontend correladores. Condiciones OS (incluida reflection positivity) para garantizar reconstrucción Lorentziana.
- Frontend sistemas abiertos. Consistencia de canal físico (CPTP) y separación A/B sin backflow micro recuperable.
- Frontend térmico. Condición KMS para equilibrio (y su rol como “analiticidad térmica”).
- Frontend simetrías. Anomalías y matching UV/IR como candados de existencia de simetrías.

Esta arquitectura vuelve el marco estricto: cada propuesta debe declarar (i) qué frontend usa, (ii) qué candados aplica, (iii) cómo traduce datos a anclajes, y (iv) en qué dominio ISAAC se afirma el veredicto.

2. Nomenclatura y objetos operacionales

2.1 Sectores operacionales (SGO/SIA): definición mínima y justificación

La división en sectores no es “metafísica”; es un candado metodológico contra la irrefutabilidad. Sin una separación explícita, cualquier marco puede salvarse diciendo “lo

que no cuadra está en lo inobservable”. En ILSC, lo inobservable puede existir, pero no puede operar como perilla de ajuste.

Regla de diseño: si un objeto no es computable o restringible mediante procedimientos reproducibles dentro del dominio observable, entonces no puede entrar como variable libre del modelo. Sólo puede entrar como restricciones indirectas (prohibiciones) sobre lo

observable.

- Sector A (SGO) (alias: A). Dominio geométrico-temporal donde existen procedimientos operacionales para definir y medir observables: trayectorias, campos efectivos, correlaciones, espectros, tasas y límites experimentales.

- Sector B (SIA) (alias: B). Complemento operacional definido por inaccesibilidad. Su microestado no es reconstruible desde A mediante protocolos reproducibles. Por diseño, B no se usa para “explicar” datos gratis; se usa únicamente para imponer restricciones sobre lo que es posible o no es posible en A (por ejemplo, ausencia de canal micro $B \rightarrow A$).

Esta dicotomía fuerza disciplina: si un argumento requiere conocer microdetalles inaccesibles, entonces no es un argumento operacional y no puede influir el veredicto.

2.2 Sector B (SIA): requisitos mínimos (no-backflow micro) y criterio de cruce

Para que B sea consistente con la física observada en A, no basta afirmar “B es inaccesible”. Debe cumplir requisitos mínimos. Estos requisitos son condiciones de existencia operacional: si se violan, la distinción A/B colapsa y aparecen paradojas en A (información, termodinámica, causalidad).

Requisito 1: inaccesibilidad operacional del microestado. No debe existir un procedimiento en A que, a partir de estadísticos medibles, reconstruya de manera reproducible el microestado de B.

En lenguaje de sistemas abiertos, la evolución efectiva en A puede escribirse como un canal completamente positivo:

$$\Phi_{\{A \rightarrow A\}}(\rho_A) = \text{Tr}_B [U_{AB} (\rho_A \otimes \sigma_B) U_{AB}^\dagger]$$

Aquí ρ_A es el estado reducido en A, σ_B codifica el microestado de B y U_{AB} es la dinámica conjunta. La condición no-backflow micro exige que, para los protocolos permitidos en A, no exista un procedimiento que identifique σ_B a partir de mediciones en A de forma reproducible.

Por qué es necesario. Si existiera backflow micro recuperable, entonces A podría hacer tomografía del interior y usar B como canal de comunicación. Eso rompe la distinción operacional A/B y abre la puerta a paradojas de información y violaciones termodinámicas en el Sector A.

Requisito 2: B sólo puede reflejarse como restricciones indirectas. Si el microestado no es computable, entonces tampoco puede entrar como variable libre al ajustar datos. Su influencia sólo puede manifestarse a través de restricciones globales: ausencia de backflow, límites de causalidad, bounds operacionales, etc.

Requisito 3: separar “criterio de cruce” (computable) de “microestado” (no computable).

El marco impone explícitamente esta separación:

- Criterio de cruce (computable en A). Condición local o casi-local basada en invariantes operacionales (curvatura, aceleración propia, límites de resolución) que decide cuándo el régimen deja de ser geométrico/IR.

- Microestado interno (no computable desde A). No participa en ajustes observacionales; sólo se refleja mediante restricciones indirectas (no-backflow, límites de causalidad, etc.).

En ILSC, el interior no es una perilla; es una frontera operacional.

2.3 ISAAC como cierre operacional: umbral, techo espectral y regla de no

reinyección UV

ISAAC es un cierre operacional: existe un umbral de resolución por debajo del cual intentar “ver más fino” fuerza autogravitación y pérdida de operacionalidad. El punto no es fijar una cifra única, sino declarar una estructura de dominio que impida extrapolaciones UV arbitrarias.

- Existe una escala de longitud L_I tal que, si un protocolo intenta localizar información por debajo de L_I , el backreaction gravitatorio deja de ser despreciable y la descripción geométrica local deja de ser una aproximación controlada.
- Existe una cota frecuencial operacional asociada $\Omega_I \approx c/L_I$, que actúa como techo efectivo al reconstruir amplitudes/kernels a partir de datos en A.

Regla ISAAC (no reinyección UV). Ninguna conclusión dinámica en A puede depender de hipótesis sobre el objeto fuera del dominio ISAAC declarado. Si una propuesta depende de estructura UV no operable, entonces no es un cierre operacional: es una extensión no auditable.

2.4 Dominio Ω_I y regla ISAAC de validez

2.4.a Protocolo de Canonización de Dominio (PCD): fijación no-ad-hoc de Ω_I

Por qué. Ω_I no puede ser un “parámetro libre” del compilador. Si Ω_I se elige post-hoc, la arbitrariedad se traslada de la teoría a la herramienta.

Qué. Se define PCD como el conjunto de reglas que fija Ω_I a partir de tres fuentes auditables: (i) ventana/resolución instrumental (Módulo Observabilidad & Instrumentación), (ii) validez y escala de corte de cualquier EFT usada (Módulo EFT & Renormalización Operacional), (iii) restricciones IR-safe cuando el frontend sea S-matrix/positividad (Módulo Amplitudes & Positividad).

Regla PCD-1 (Ventana). Ω_I debe estar contenido en la ventana donde el observable está definido y calibrado. Si el anclaje no trae ventana, no es consumible.

Regla PCD-2 (Resolución). Ω_I no puede exigir estructura sub-resolución: cualquier constraint que requiera variación más fina que el kernel de resolución viola ISAAC.

Regla PCD-3 (Validez EFT). Si se usa EFT, Ω_I debe satisfacer $E/\Lambda \leq \varepsilon_\Lambda$ declarado y se debe propagar error EFT en toda la corrida.

Regla PCD-4 (Sensibilidad). Se reporta estabilidad del veredicto al variar Ω_I dentro de un margen instrumental/teórico razonable. Si el veredicto cambia, se reporta explícitamente y se degrada el estado de ‘PASS fuerte’.

2.3.a Candado ISAAC-0 (normalización canónica): cierre operacional por backreaction del acto de medir

Propósito. Esta sección fija el núcleo no negociable de ISAAC como consecuencia operacional de física establecida. No introduce una hipótesis nueva; explicita el cierre que ya está implícito al combinar causalidad relativista, cuantización y gravitación.

Cadena mínima (sin pasos especulativos).

- Causalidad operacional. La adquisición de información requiere intercambio causal de señales/portadores; no existe acceso operacional a un evento sin algún canal causal.
- Cuantización de energía-momento. Todo portador capaz de resolver una escala espacial L debe transportar momento mínimo del orden $p(L) \gtrsim \hbar/L$ (y energía $E(L) \gtrsim \hbar c/L$ en el caso relativista).
- Gravitación universal. Energía-momento curva el espaciotiempo; por tanto, el propio acto de medir introduce backreaction gravitatoria que no puede apagarse “por convenio”.

Consecuencia inevitable. Existe un umbral de resolución L_I tal que, si un protocolo intenta localizar información en $L < L_I$, el backreaction gravitatorio deja de ser perturbativo y la operación pierde significado operacional dentro de A (SGO). En ese régimen, el intento de aumentar resolución exige energía suficiente para generar un radio gravitatorio comparable a la propia escala de localización.

Normalización canónica (escala de orden, sin perillas):

$$E(L) \sim \hbar c/L, \quad r_s(E) \sim 2GE/c^4, \quad \text{cierre: } r_s(E(L)) \gtrsim L \Rightarrow L \lesssim \sqrt{(\hbar G/c^3)} \equiv L_I.$$

Nota sobre factores $O(1)$. El valor exacto puede variar por detalles geométricos/convenciones (factores 2, 2π , etc.). ILSC no depende de fijar esos factores: el candado exige sólo la existencia de un umbral operacional y su declaración coherente dentro de Ω_I . Los márgenes instrumentales/teóricos se gestionan por PCD (Sección 2.4.a) y por el meta-candado de estabilidad (PCD/PCN).

Implicación inmediata: Regla ISAAC de no reinyección UV (forma reforzada)

Por qué. Si se permite que conclusiones en A dependan de estructura a escalas $L < L_I$ (o de grados de libertad fuera de Ω_I), se reintroduce maleabilidad: los detalles UV se convierten en perillas no contrastables que pueden ajustarse post-hoc.

Qué (ISAAC-NR-0). Ninguna predicción o constraint aceptado por el compilador puede depender de hipótesis específicas sobre estructura fuera de Ω_I . La única entrada permitida desde fuera de Ω_I es una proyección efectiva auditable (p.ej. EFT con truncación y error, kernels SK con CPTP/positividad, o matching global de simetrías/anomalías), con sensibilidad reportada.

Diagnóstico anti-escape (operacional). Cualquier objeción a ISAAC requiere negar al menos uno de estos pilares: (i) intercambio causal de información, (ii) cuantización de

Anexo: Jueces (Nivel J) y Blindaje

Concepto→Ecuación

Integración a Documento A • Versión v4 • 2026-02-13 22:48 UTC

Este anexo integra y formaliza el Nivel J (Jueces) dentro del estándar ILSC. Los jueces no son “candados adicionales”: son condiciones de evaluabilidad. Su propósito es impedir que la evaluación de teorías dependa de estructura inaccesible, degeneraciones no resolubles o idealizaciones infinitas. Cada juez se define con un concepto invariante y una o más instanciaciones dependientes de datos/tecnología, de modo que el concepto sobreviva aunque cambie la ecuación.

0. Blindaje ISAAC (J0): concepto \neq ecuación

J0 / ISAAC — Concepto (invariante): existe un umbral operacional donde intentar aumentar resolución/información exige intercambio de energía-momento suficiente para inducir backreaction comparable y destruir la operacionalidad (la medición deja de ser una operación física bien definida).

J0 / ISAAC — Instanciación (dependiente de régimen): en el régimen donde $SR+QM+GR$ describen la operación de medición, una estimación canónica es una escala tipo Planck. La constante numérica (y el modelo microscópico) pueden cambiar con nueva física; el juez ISAAC no porque es un argumento operacional: información \leftrightarrow energía-momento \leftrightarrow curvatura \leftrightarrow causalidad.

J0 / ISAAC — Regla de compilación: cualquier dependencia UV que impacte Ω_l debe entrar vía una proyección finita Π + error acotado. Si se usa UV no proyectable como perilla, el veredicto es NO-EVAL (no evaluable operacionalmente).

Tabla de severidad (para jueces)

Resultado	Significado operacional
PASS	El juez se satisface: la propuesta es evaluable bajo el criterio del juez.
FAIL	La propuesta viola consistencia física dentro de Ω_l (cuando el juez detecta inconsistencia).
NO-EVAL	No evaluable hoy: falta proyección auditable / identificabilidad / estabilidad/recursos. No e

1. J1 — Proyección Auditable (PA)

Concepto (invariante): si una teoría usa estructura fuera de Ω_l para afectar predicciones dentro de Ω_l , solo es evaluable si existe una proyección finita Π hacia Ω_l con un error acotado y auditable. Sin Π +error: NO-EVAL.

Motivación: esto formaliza el anti-“UV magic”: la teoría puede ser matemáticamente consistente, pero sin proyección auditable no es física contrastable en Ω_l .

Instanciaciones (dependientes de datos):

- EFT/truncación: estimar $\Delta_{\text{proj}}(E)$ (error de truncación / integración de modos pesados) y compararlo con σ_{data} del observable.

- Dominancia: si $\Delta_{\text{proj}} \geq \sigma_{\text{data}}$ en la región crítica, el mecanismo es NO-EVAL (proyección no informativa).
- Certificado auditable: versión de código, hashes, seeds, dataset ID, y procedimiento reproducible para Δ_{proj} .

2. J2 — Identificabilidad Operacional (IO)

Concepto (invariante): una propuesta solo “predice” si sus parámetros efectivos (los que afectan Ω_I) son identificables mediante observables en Ω_I . Si el claim central es compatible con infinitas deformaciones indistinguibles dadas las incertidumbres, el veredicto es NO-EVAL (o PASS-débil si se declara explícitamente como tal).

Instanciaciones (dependientes de datos):

- Condicionamiento del Jacobiano $J = \partial O / \partial \theta$ en la región relevante; direcciones casi nulas indican no-identificabilidad.
- Chequeo posterior: direcciones planas del posterior en parámetros que afectan el claim central \Rightarrow NO-EVAL.
- Conteo de grados de libertad efectivos vs información (p.ej. dimensión efectiva de Fisher) para declarar rigidez.

3. J3 — Recursos Finitos y Estabilidad (RFS)

Concepto (invariante): una predicción física debe resultar de un procedimiento finito (energía, tiempo, precisión, cómputo) y ser estable bajo refinamientos razonables. Si el veredicto depende de límites ideales (precisión infinita, control perfecto, discretización ad hoc) o cambia bajo refinamientos razonables, entonces es NO-EVAL.

Instanciaciones (dependientes de datos):

- PCN/PCD: estabilidad del veredicto bajo refinamiento de malla y variación de tolerancias (como en MRD-1X-SK).
- Presupuesto explícito de recursos: límites compatibles con ISAAC (resolución/energía) y con recursos experimentales reales.
- Criterio de convergencia: si el signo/veredicto cambia bajo refinamiento razonable, NO-EVAL(unstable).

4. Integración con el flujo ILSC

Orden recomendado de evaluación:

- Jueces (J0-J3): determinan si la propuesta es evaluable (PASS/NO-EVAL; FAIL solo si hay inconsistencia directa).
- Locks (Nivel L): unitaridad/causalidad/positividad/CPTP/etc. dentro de lo evaluable.
- Módulos (Nivel M): traducciones específicas por dominio (cosmología, EFT, nuclear, amplitudes...).

Regla anti-ataque: si un crítico propone “relajar” un juez, debe explicitar qué parte de evaluabilidad operacional está dispuesto a perder (proyección auditable / identificabilidad / finitud/estabilidad).

energía-momento requerida para resolución, (iii) gravitación universal del estrés-energía. Si se propone un canal 'alternativo' (partículas masivas, neutrinos, gravitones, entrelazamiento), la exigencia operacional permanece: debe existir una cota de energía-momento mínima para resolver L y esa energía gravita; por tanto el cierre reaparece (quizá con factores $O(1)$).

La metodología opera en un dominio operacional declarado. En problemas de dispersión, se define un corte energético Λ_I y un dominio cinemático: $E_I = \{s \mid 0 \leq s \leq \Lambda_I^2\}$, $\Omega_I = E_I \times T_I$ donde T_I es una ventana en t elegida para evitar singularidades IR (por ejemplo, un intervalo $t \in [t_{\min}, t_{\max}] < 0$ cuando hay intercambio sin masa). Toda desigualdad/candado debe declarar dónde aplica (Ω_I) y qué tratamiento IR usa; de lo contrario, el juez puede producir falsos FAIL (artefactos IR).

2.5 Conjuntos y métricas: F , D_{obs} , I y rigidez R

Se define un problema de existencia sujeto a restricciones:

- F : conjunto factible teórico (lo compatible con candados inevitables en Ω_I).
- D_{obs} : región de compatibilidad observacional (anclaje) expresada en el espacio de parámetros elegido.
- $I \equiv F \cap D_{\text{obs}}$: lo que sobrevive.

Definición binaria: FAIL si $I = \emptyset$; PASS si $I \neq \emptyset$. Para distinguir un PASS maleable de un PASS rígido, se introduce una métrica de rigidez R (Sección 4.4).

3. Candados inevitables: familias por frontend (C0–C11)

3.0 Meta-candados (C0): operacionalidad, cierre y auditabilidad

Por qué. Sin un candado de operacionalidad, cualquier marco puede volverse irrefutable:

basta mover el desacuerdo a escalas inaccesibles o a grados de libertad no computables. La universalidad exige que el juez sólo “pida” consistencia donde existe acceso operacional y que el resultado sea reproducible.

C0 (Operacionalidad). Toda afirmación debe declarar (i) el frontend usado, (ii) el dominio

ISAAC y (iii) el conjunto cerrado de supuestos técnicos.

C0' (Auditabilidad). El veredicto sólo es válido si produce un bloque de auditoría mínimo (Sección 5). Sin auditoría, la metodología se degrada a narrativa.

3.1 Frontend S-matrix (C1–C6): dispersión relativista

Por qué. En teoría de dispersión relativista, los principios inevitables se expresan como propiedades de la S-matrix: conservación de probabilidad (unitaridad), compatibilidad relativista entre canales (crossing), y causalidad traducida a analiticidad/relaciones de dispersión.

C1: Unitaridad. La dinámica debe admitir unitaridad (por ejemplo bounds de ondas parciales), incluyendo inelasticidad cuando corresponda.

C2: Analiticidad/causalidad. La amplitud debe ser analítica con cortes/polos físicos estándar, permitiendo relaciones de dispersión. Violaciones implican respuestas acausales o no-localidad efectiva.

C3: Crossing. Continuación analítica consistente entre canales; sin ella no hay S-matrix relativista consistente.

C4: Boundedness UV operacional. Crecimiento acotado (polinómico o Regge) en Ω_I con número de sustracciones N_{sub} declarado y consistente.

C5: Spin-1/Spin-2 consistentes. Con excitaciones sin masa, la consistencia S-matrix fuerza conservación de carga (spin-1) y acoplo gravitatorio universal (spin-2).

C6: Protocolo gravitacional IR-safe. Con gravitón sin masa, el polo t-channel arruina el forward ingenuo; se exige sustracción/ventana $t < 0$ y (si se usa positividad) hipótesis UV mínima declarada.

3.2 Frontend correladores Euclídeos→Lorentzianos (C7): OS / reflection positivity

Por qué. En muchos contextos (lattice, formulación Euclídea), el objeto primario no es la amplitud sino correladores Euclídeos. Sin condiciones adicionales, no hay garantía de que esos correladores correspondan a una teoría unitaria en tiempo real. Se requiere un candado que asegure que la “rotación de Wick” y la reconstrucción Lorentziana son válidas.

C7 (OS). Los correladores Euclídeos deben satisfacer un conjunto de axiomas tipo Osterwalder-Schrader, en particular reflection positivity, para garantizar reconstrucción a una QFT tipo Wightman en Minkowski.

3.3 Frontend sistemas abiertos (C8): canales CPTP y consistencia A/B

Por qué. Cuando el observador sólo accede a un subsistema (SGO) y existe un complemento inaccesible (SIA), la dinámica observable debe conservar la interpretación de probabilidad para cualquier extensión con ancilla. Si no, aparecen predicciones negativas/no físicas y el marco pierde operacionalidad.

C8 (CPTP). Toda evolución efectiva en A debe ser un canal completamente positivo y preservador de traza (CPTP), o una familia consistente de ellos, y debe ser compatible con la condición no-backflow micro (Sección 2.2).

3.4 Frontend térmico-equilibrio (C9): condición KMS

Por qué. Para describir equilibrio térmico en mecánica estadística cuántica/QFT, se necesita un criterio intrínseco que sustituya a “poner un Gibbs por mano”, especialmente en sistemas infinitos. La condición KMS caracteriza estados de equilibrio en términos de analiticidad/periodicidad en tiempo imaginario.

C9 (KMS). Si el módulo declara equilibrio térmico, los correladores deben satisfacer la condición KMS con β declarado, y el anclaje observacional debe respetar ese β (temperatura) como dato, no como perilla libre.

3.5 Frontend simetrías/anomalías (C10): anomaly matching

Por qué. Hay propuestas que pueden parecer consistentes a nivel de amplitudes/efectivos, pero fallan porque una simetría global no puede realizarse coherentemente: las anomalías son obstrucciones topológicas robustas. El matching UV/IR no es estética: es condición de existencia de la simetría a través de escalas.

C10 (Matching). Si el modelo declara una simetría global continua con anomalía de 't

Hooft, dicha anomalía debe coincidir entre descripciones UV e IR (por ejemplo vía grados de libertad masivos, modos de Goldstone con términos WZW u otros mecanismos).

3.6 Frontend relativista local (C11): consistencia CPT como test de supuestos

Por qué. En QFT local Lorentz-invariante y unitaria, CPT es un resultado estructural: si una propuesta viola CPT, no es un “detalle”; indica que al menos uno de los supuestos de localidad, Lorentz o unitariedad fue abandonado. El juez exige entonces declarar explícitamente cuál supuesto se rompe, y por qué eso no destruye la operacionalidad.

C11 (Test CPT). En módulos que declaran QFT local relativista como descripción efectiva,

se exige compatibilidad CPT o declaración explícita del supuesto roto (y el frontend alternativo que lo reemplaza).

4. Especificación del compilador ILSC

ILSC es una especificación ejecutable: intersecta candados inevitables con anclajes observacionales dentro del dominio ISAAC, para emitir un veredicto PASS/FAIL. No selecciona una ontología ni un Lagrangiano “por gusto”; decide existencia de dinámica compatible con principios inevitables.

4.1 Entradas (frontends, parametrización, anclajes)

La ejecución requiere declarar:

- Frontend seleccionado (S-matrix, correladores, canales, térmico, simetrías) y su objeto operacional.
- Dominio Ω_I /ISAAC (Λ_I y, si aplica, ventana T_I) y el protocolo IR correspondiente.
- Parametrización finito-dimensional (amplitud, correlador, canal o conjunto de coeficientes) con convención fijada.
- Anclajes observacionales D_{obs} (intervalos/likelihoods comprimidos, CL, esquemas) y traducción auditable.
- Hipótesis técnicas (N_{sub} , boundedness/Regge, discretización, tolerancias) declaradas antes de correr.

4.2 Salidas (PASS/FAIL, PASS fuerte, certificados)

4.2.a Taxonomía de veredictos: PASS/FAIL, PASS fuerte, FAIL certificado y NO-EVAL

Por qué. Un juez universal debe distinguir entre “inconsistente”, “consistente pero no rígido” y “no evaluable por falta de input auditable”. Si todo se fuerza a PASS/FAIL, el marco se vuelve injusto o ‘fetichista’.

Qué. Se adopta la siguiente taxonomía mínima:

- **PASS**: $I = F \cap D_{\text{obs}} \neq \emptyset$.

- **PASS fuerte**: $I \neq \emptyset$ y la región superviviente es rígida (poca dimensión efectiva o discreta; dominada por incertidumbre experimental, no por libertad teórica).
- **FAIL**: $I = \emptyset$ dentro de Ω_I , con diagnóstico.
- **FAIL-certificado**: FAIL acompañado por witness/dual certificate cuando el problema es convexo (SDP/LP) o por un no-go operacional equivalente.
- **NO-EVAL**: no evaluable porque falta (i) proyección $UV \rightarrow \Omega_I$ auditable, (ii) anclaje D_{obs} consumible (ventana/resolución/covarianza), o (iii) MRD mínimo para esa familia de constraints. NO-EVAL no es un 'rescate': es un estado explícito que obliga a completar el input.

Regla. NO-EVAL es obligatorio cuando la propuesta no puede compilarse sin introducir supuestos no auditados.

Salida principal: PASS o FAIL según $I \equiv F \cap D_{\text{obs}}$. Salidas ampliadas:

- Si PASS: intervalos/proyecciones marginales y medidas de rigidez (R y/o discreción local).
- Si FAIL: diagnóstico reproducible de choque y, cuando el solver lo permite, certificado de infeasibilidad.
- Bloque de auditoría (hashes, versiones, tolerancias, semillas).

Definimos PASS fuerte cuando la región superviviente está dominada por incertidumbre experimental (no por direcciones planas teóricas).

4.3 Algoritmo ejecutable (compilación)

Pasos canónicos:

- Paso 1. Elegir parametrización y dominio Ω_I consistentes con el frontend y el protocolo IR.
- Paso 2. Compilar candados inevitables del frontend a constraints numéricos (LP/SDP u otro método declarado).
- Paso 3. Intersectar con anclajes D_{obs} ; resolver factibilidad/optimización.
- Paso 4. Emitir veredicto + diagnóstico + auditoría; si PASS, cuantificar rigidez.

4.4 Rigidez y discreción: R y dimensión efectiva local

Se admiten dos métricas complementarias, ambas operacionales (deben declararse junto con norma y tolerancias):

- Rigidez por anchos. Para parámetros f_i , se resuelven máximos y mínimos sobre I , se define Δf_i y se reporta $R \equiv \max_i (\Delta f_i / \sigma_i^{\text{exp}})$. PASS fuerte si $R \leq 1$.
- Discreción local. Con parametrización $x \in \mathbb{R}^N$ y restricciones activas en un extremo x^* , se define $d_{\text{eff}} \equiv N - \text{rank}(J_{\text{act}}(x^*))$. Discreta (PASS fuerte) si $d_{\text{eff}} = 0$ de forma estable.

5. Reproducibilidad y auditoría

5.a Protocolo de Canonización Numérica (PCN): tolerancias, discretización y convergencia

Por qué. En compilación numérica, la arbitrariedad puede reaparecer en tolerancias, mallas y stopping criteria. PCN convierte esas elecciones en reglas cerradas y auditables.

Qué. PCN fija requisitos mínimos:

- **PCN-1 (Tolerancias).** Declarar tolerancia global y tolerancias por constraint; reportar sensibilidad del veredicto al variar tolerancias en un rango razonable.
- **PCN-2 (Mallas/Discretización).** Para integrales, kernels o PSD, demostrar convergencia bajo refinamiento (al menos dos refinamientos).
- **PCN-3 (Semillas/Reproducibilidad).** Seeds, versiones, hashes y entornos (contenedor) obligatorios para todo experimento numérico.
- **PCN-4 (Criterio de paro).** Stopping criteria explícito; si se usa optimización, reportar certificados de optimalidad o cotas.
- **PCN-5 (Robustez).** Si el veredicto depende de detalles de discretización, se degrada a PASS débil o se emite NO-EVAL hasta resolver estabilidad.

En un marco operacional, el método es parte del contenido: sin pipeline y auditoría, no existe veredicto científico reproducible. Cada corrida debe emitir un bloque de auditoría suficiente para que un tercero reproduzca exactamente el veredicto.

5.1 Bloque mínimo de auditoría

- Hash SHA-256 de los archivos de entrada (parámetros, normalizaciones, límites, constraints).
- Versión del código (commit), versión del solver y flags relevantes.
- Tolerancias numéricas: factibilidad, dualidad, tolerancia PSD (si aplica) y criterio de parada.
- Semilla (si existe muestreo) y umbrales de factibilidad.
- Lista exacta de anclajes y procedimiento auditable de traducción a D_{obs} .

5.2 Tests automáticos recomendados

- Convergencia bajo refinamiento de discretización/parametrización.
- Estabilidad bajo variación de tolerancias dentro de un rango auditado.
- Chequeos del dominio (no-forward/IR-safe cuando corresponda).
- Verificación de rigidez (R y/o d_{eff}) con objetivos auxiliares canónicos.

6. Demo mínimo reproducible (MRD-1) end-to-end

6.a MRDs por capas: de ‘toy’ a ‘caso real’

Por qué. Un MRD trivial demuestra formato, pero no demuestra que el compilador sobreviva la ‘suciedad’ de datos reales. Para obtener PASS fuerte como metodología, se requieren MRDs escalonados.

Qué. Se definen tres niveles mínimos:

- **MRD-0 (toy).** Demuestra compilación (constraints + solver + auditoría) con un sistema pequeño.
- **MRD-1 (realista).** Un frontend no trivial (p.ej. positividad/EFT o SK con kernel físico) con un D_{obs} exportado en formato canónico, incluyendo covarianzas y ventana.
- **MRD-2 (caso real).** Aplicación end-to-end a un dataset real publicado: (i) exportación de D_{obs} con ventana/resolución, (ii) compilación de candados inevitables, (iii) PASS/FAIL con diagnóstico y, si aplica, certificado.

Regla. Sin MRD-1, el documento puede estar conceptualmente correcto, pero el estado metodológico se reporta como 'PASS con rigidez media' en lugar de PASS fuerte.

El MRD fija una instancia concreta (un solo canal/objeto, un solo dataset, una parametrización y un solver) para que un tercero pueda ejecutar el compilador sin ambigüedad. El MRD no pretende agotar la física; es un test de ejecutabilidad del juez.

6.1 Elección del objeto

Se recomienda iniciar con frontend S-matrix (VBS/aQGC) para demostrar el pipeline completo. Posteriormente, se añade un MRD para cada frontend (correladores, canales CPTP, KMS, anomalías).

6.2 Entradas declaradas

- Anclajes observacionales: límites/intervalos experimentales con CL declarado y convención fija.
- Espacio teórico: base fija (por ejemplo operadores EFT) o parametrización primal de amplitud/correlador.
- Dominio ISAAC: Ω_I con corte Λ_I ; si aplica, ventana t negativa o protocolo IR equivalente.

6.3 Candados implementados

- Candados del frontend seleccionado (Sección 3), compilados a constraints numéricos.
- Protocolos IR/gravedad cuando aplique; registro explícito de N_{sub} /boundedness.

6.4 Salidas mínimas a reportar

- PASS/FAIL + bloque de auditoría.
- Si PASS: intervalos y rigidez (R y/o d_{eff}).
- Si FAIL: diagnóstico reproducible + (si existe) certificado de infeasibilidad.

7. Supuestos (S1-S12) y loopholes (L1-L8)

7.a Meta-candado de estabilidad: 'no mover arbitrariedad al compilador'

Por qué. Si el veredicto depende críticamente de Ω_I , N_{sub} , IR-treatment o tolerancias, el compilador se convierte en una fuente de arbitrariedad.

Qué. Se añade el meta-candado: toda corrida debe incluir (i) PCD (estabilidad frente a variación razonable de Ω_I), (ii) PCN (convergencia y sensibilidad numérica) y (iii) tratamiento IR declarado cuando aplique. Si cualquiera falla, el veredicto se degrada (PASS débil) o pasa a NO-EVAL.

La metodología es auditable sólo si sus supuestos están declarados. Esta lista es cerrada: si se añade un supuesto, debe registrarse junto con el veredicto.

Supuestos S1-S12

- S1 Dominio de analiticidad suficiente para el frontend declarado.
- S2 Crossing (si aplica) válido en Ω_I .
- S3 Unitaridad aplicable (S-matrix) o positividad equivalente (OS/CPTP) según frontend.
- S4 Boundedness/Regge en Ω_I con N_{sub} explícito cuando se usen dispersion relations.
- S5 Tratamiento IR: evitar forward ingenuo con polos/cortes IR.
- S6 Gravedad: protocolo IR-safe y (si aplica) hipótesis Regge mínima.
- S7 Corte ISAAC: prohibida reinyección UV por encima de Λ_I .
- S8 Convergencia bajo refinamiento de parametrización/truncación.
- S9 Datos como anclaje: traducción auditable a D_{obs} sin selección a posteriori.
- S10 Tolerancias numéricas declaradas; estabilidad del veredicto en rango de auditoría.
- S11 Elección de frontend coherente con el objeto (no mezclar sin mapa explícito).
- S12 Si se declara equilibrio (térmico), β es dato/anclaje, no perilla libre.

Loopholes L1-L8

- L1 IR puede producir falsos FAIL sin protocolo.
- L2 No-localidad UV fuera del dominio; el veredicto es condicional a ISAAC.
- L3 Dependencia del dataset; debe declararse y probarse estabilidad del veredicto.
- L4 Ambigüedad de esquema/base (EFT/correladores); fijar convención.
- L5 Sensibilidad a discretización; exigir convergencia.
- L6 Tuning humano post-hoc rompe reproducibilidad.
- L7 Mezcla de frontends sin mapa IL produce contradicciones aparentes.
- L8 Violación CPT sin declaración explícita del supuesto roto.

8. Glosario mínimo y mini-ejemplo juguete

Glosario mínimo

- ISAAC / Λ_I : cierre operacional; define dominio y prohíbe reinyección UV.
- Sector A (SGO): dominio observable con medición.
- Sector B (SIA): complemento inaccesible; microestado no computable; sólo restricciones indirectas.
- Frontend: representación del objeto físico (S-matrix, correladores, canal, térmico, simetrías).
- F: conjunto factible; D_{obs} : anclaje; I: intersección; R: rigidez; d_{eff} : discreción local.

Mini-ejemplo juguete (ilustración del compilador)

Entrada: dos parámetros $c = (c_1, c_2)$. Anclaje observacional: $|c_1| \leq 1, |c_2| \leq 1$.

Candados: $c_1 \geq 0$ y $c_1 + c_2 \leq 0.2$.

Compilación: $D_{\text{obs}} = [-1, 1]^2$, $F = \{(c_1, c_2): c_1 \geq 0, c_1 + c_2 \leq 0.2\}$. Entonces $I = D_{\text{obs}}$

$n \neq \emptyset \Rightarrow \text{PASS}$. El volumen superviviente es pequeño \Rightarrow rigidez.

Referencias

- Weinberg, S. (1964). Photons and Gravitons in S-Matrix Theory: Derivation of Charge Conservation and Equality of Gravitational and Inertial Mass, Phys. Rev. 135, B1049.
- Adams, A. et al. (2006). Causality, Analyticity and an IR Obstruction to UV Completion, JHEP 10 (2006) 014.
- Tokuda, J., Aoki, K., Hirano, S. (2020). Gravitational positivity bounds, JHEP 11 (2020) 054.
- Caron-Huot, S. et al. (2023). Causality constraints on corrections to Einstein gravity, JHEP 05 (2023) 122.
- Osterwalder-Schrader reconstruction theorem (reflection positivity) — referencias estándar en axiomas de QFT.
- Definición operativa de canales CPTP y su caracterización (Kraus/Stinespring).
- Condición KMS para estados de equilibrio en sistemas cuánticos.
- 't Hooft anomaly matching como candado UV/IR para simetrías globales.

ILSC — Addendum A.v5: Candados de Evaluabilidad para J1-J3 (PA/IO/RFS)

Bilingüe ES/EN • v5 Addendum • 2026-02-13 22:57 UTC

Este addendum integra el conjunto completo de candados necesarios para operacionalizar los nuevos jueces sin volverlos arbitrarios: J1 Proyección Auditable (PA), J2 Identificabilidad Operacional (IO), y J3 Recursos Finitos y Estabilidad (RFS). Los candados no “fundan” al juez: lo certifican. Regla de severidad: faltas de evaluabilidad → NO-EVAL; inconsistencias físicas → FAIL.

Regla de severidad / Severity rule

Tipo	Severidad	Significado operativo
Evaluabilidad incompleta (falta Π +error)	NO-EVAL	No identificabilidad, recursos finitos y estabilidad)
Inconsistencia física de Ω (violación de Ω)	FAIL	Contradicción física inaplicable
Cumple certificaciones y Ω s aplicables	OK	Compilable y consistente en Ω .

J1 — Candados de Proyección Auditable (PA)

J1 — Auditable Projection Locks (PA)

PA1 — Declaración explícita de la proyección Π

- Entrada requerida: Definición formal de Π : dominio UV → parámetros efectivos en Ω ; supuestos; versión.
- Procedimiento: Validación de esquema (PA-CERT) y consistencia de firmas.
- Umbral: Debe existir Π con firma y versión; sin ambigüedad.
- Salida/diagnóstico: NO-EVAL(PA1): falta Π declarada.
- Severidad al fallar: NO-EVAL

PA1 — Explicit projection Π declaration

- Required input: Formal definition of Π : UV domain → effective parameters in Ω ; assumptions; version.
- Procedure: Schema validation (PA-CERT) and signature consistency.
- Threshold: Π must exist with a signature and version; unambiguous.
- Output/diagnostic: NO-EVAL(PA1): missing declared Π .
- Failure severity: NO-EVAL

PA2 — Cota de error de proyección $\Delta\Pi(E)$ con método

- Entrada requerida: $\Delta\Pi(E)$ en rango E relevante; método (truncación EFT, residual, bounds) y dependencia en escala.

- Procedimiento: Revisión/ejecución del método descrito; reproducibilidad.
- Umbral: $\Delta\Pi$ debe estar definida en todo el rango de E usado en claims.
- Salida/diagnóstico: NO-EVAL(PA2): $\Delta\Pi$ ausente/no reproducible.
- Severidad al fallar: NO-EVAL

PA2 — Projection-error bound $\Delta\Pi(E)$ with method

- Required input: $\Delta\Pi(E)$ over relevant E; method (EFT truncation, residual, bounds) and scale dependence.
- Procedure: Run/verify stated method; reproducible.
- Threshold: $\Delta\Pi$ must be defined over the full E-range used by the claims.
- Output/diagnostic: NO-EVAL(PA2): $\Delta\Pi$ missing/not reproducible.
- Failure severity: NO-EVAL

PA3 — Informatividad vs datos (dominancia del error)

- Entrada requerida: σ_{data} del/los observables objetivo; regla κ .
- Procedimiento: Comparar $\Delta\Pi(E)$ vs σ_{data} en el rango de inferencia.
- Umbral: Si $\Delta\Pi \geq \kappa \cdot \sigma_{\text{data}}$ en el rango relevante, la proyección no informa.
- Salida/diagnóstico: NO-EVAL(PA3): proyección no informativa.
- Severidad al fallar: NO-EVAL

PA3 — Informativeness vs data (error dominance)

- Required input: σ_{data} for target observables; κ rule.
- Procedure: Compare $\Delta\Pi(E)$ vs σ_{data} over inference range.
- Threshold: If $\Delta\Pi \geq \kappa \cdot \sigma_{\text{data}}$ over relevant range, projection is not informative.
- Output/diagnostic: NO-EVAL(PA3): non-informative projection.
- Failure severity: NO-EVAL

PA4 — Invarianza bajo reparametrización permitida

- Entrada requerida: Transformaciones permitidas (cambio de base/convención) y cómo transforma Π .
- Procedimiento: Verificar que O predichos son invariantes dentro de $\Delta\Pi + \sigma_{\text{data}}$; registrar transformación.
- Umbral: Invarianza dentro de tolerancias declaradas.
- Salida/diagnóstico: NO-EVAL(PA4) si no se puede auditar; FAIL si hay inconsistencia matemática.
- Severidad al fallar: NO-EVAL/FAIL

PA4 — Invariance under allowed reparametrizations

- Required input: Allowed transformations (basis/convention change) and how Π transforms.
- Procedure: Check predicted O invariance within $\Delta\Pi + \sigma_{\text{data}}$; log transform.
- Threshold: Invariance within declared tolerances.
- Output/diagnostic: NO-EVAL(PA4) if unauditable; FAIL if mathematical inconsistency.
- Failure severity: NO-EVAL/FAIL

PA5 — Certificado reproducible de proyección

- Entrada requerida: Hash de inputs, versión de código, seeds, dataset ID; log de ejecución.
- Procedimiento: Recalcular y verificar coincidencia de hashes y versiones.
- Umbral: Hash/verificaciones deben coincidir; corrida reproducible.
- Salida/diagnóstico: NO-EVAL(PA5): no reproducible.
- Severidad al fallar: NO-EVAL

PA5 — Reproducible projection certificate

- Required input: Input hashes, code version, seeds, dataset ID; execution log.
- Procedure: Recompute and verify hashes/versions.
- Threshold: Hashes must match; run must be reproducible.
- Output/diagnostic: NO-EVAL(PA5): not reproducible.
- Failure severity: NO-EVAL

J2 — Candados de Identificabilidad Operacional (IO)

J2 — Operational Identifiability Locks (IO)

IO1 — Mapa $O(\theta)$ declarado

- Entrada requerida: Lista finita θ_{Ω} y observables O relevantes; rango de validez; unidades y convenciones.
- Procedimiento: Validar esquema IO-CERT; verificar que O y θ estén conectados por cálculo/estimación.
- Umbral: Debe existir una relación operacional $O(\theta)$ ejecutable o auditable.
- Salida/diagnóstico: NO-EVAL(IO1): mapa no declarado.
- Severidad al fallar: NO-EVAL

IO1 — Declared map $O(\theta)$

- Required input: Finite θ_{Ω} and relevant observables O ; validity range; units/conventions.
- Procedure: Validate IO-CERT; verify O - θ link via computation/estimate.
- Threshold: An operationally auditable $O(\theta)$ must exist.
- Output/diagnostic: NO-EVAL(IO1): map not declared.
- Failure severity: NO-EVAL

IO2 — Degeneraciones documentadas

- Entrada requerida: Direcciones planas conocidas, simetrías, reparametrizaciones; qué datos las rompen.
- Procedimiento: Evaluar sensibilidad y registrar degeneraciones (Fisher/Jacobiano/posterior).
- Umbral: Debe listarse explícitamente qué no es identificable hoy.
- Salida/diagnóstico: NO-EVAL(IO2): degeneraciones no declaradas.
- Severidad al fallar: NO-EVAL

IO2 — Documented degeneracies

- Required input: Known flat directions, symmetries, reparametrizations; what data breaks them.
- Procedure: Evaluate sensitivity and log degeneracies (Fisher/Jacobian/posterior).
- Threshold: Must explicitly list what is not identifiable today.
- Output/diagnostic: NO-EVAL(IO2): degeneracies not declared.
- Failure severity: NO-EVAL

IO3 — Umbral operativo de identificabilidad

- Entrada requerida: Métrica elegida (condición de Jacobiano/Fisher, posterior) y umbral τ .
- Procedimiento: Calcular la métrica y comparar con τ en la región de interés.
- Umbral: Si el parámetro (o combinación) que soporta el claim central no supera $\tau \rightarrow$ NO-EVAL.
- Salida/diagnóstico: NO-EVAL(IO3).
- Severidad al fallar: NO-EVAL

IO3 — Operational identifiability threshold

- Required input: Chosen metric (Jacobian/Fisher condition, posterior) and threshold τ .
- Procedure: Compute metric and compare to τ over region of interest.
- Threshold: If the parameter (or combination) underpinning the central claim does not exceed $\tau \rightarrow$ NO-EVAL.
- Output/diagnostic: NO-EVAL(IO3).
- Failure severity: NO-EVAL

IO4 — Separación claim vs ajuste

- Entrada requerida: Declarar qué parámetros son 'nuisance' y cuáles sostienen el claim.
- Procedimiento: Verificar si el claim depende de parámetros no identificables o priors arbitrarios.
- Umbral: Si el claim depende de direcciones planas o priors no motivados \rightarrow NO-EVAL.
- Salida/diagnóstico: NO-EVAL(IO4).
- Severidad al fallar: NO-EVAL

IO4 — Claim vs fit separation

- Required input: Declare nuisance parameters vs claim-bearing parameters.
- Procedure: Check whether claim depends on non-identifiable directions or arbitrary priors.
- Threshold: If claim hinges on flat directions/arbitrary priors \rightarrow NO-EVAL.
- Output/diagnostic: NO-EVAL(IO4).
- Failure severity: NO-EVAL

IO5 — Robustez a datasets equivalentes

- Entrada requerida: Lista de datasets equivalentes o análisis alternativos compatibles.

- Procedimiento: Repetir diagnóstico de identificabilidad en datasets equivalentes; comparar.
- Umbral: Si el veredicto cambia sin causa física/matemática → NO-EVAL.
- Salida/diagnóstico: NO-EVAL(IO5).
- Severidad al fallar: NO-EVAL

IO5 — Robustness to equivalent datasets

- Required input: List of equivalent datasets or compatible alternative analyses.
- Procedure: Repeat identifiability diagnosis on equivalent datasets; compare.
- Threshold: If verdict flips without a physical/mathematical cause → NO-EVAL.
- Output/diagnostic: NO-EVAL(IO5).
- Failure severity: NO-EVAL

J3 — Candados de Recursos Finitos y Estabilidad (RFS)

J3 — Finite Resources & Stability Locks (RFS)

RFS1 — Presupuesto explícito de recursos

- Entrada requerida: Límites de resolución, energía/tiempo, cómputo (work cap), tolerancias numéricas; vínculo con ISAAC.
- Procedimiento: Validar que el claim no requiera recursos infinitos; registrar presupuestos.
- Umbral: Presupuesto debe ser finito y consistente con ISAAC.
- Salida/diagnóstico: NO-EVAL(RFS1).
- Severidad al fallar: NO-EVAL

RFS1 — Explicit resource budget

- Required input: Resolution, energy/time, compute (work cap), numeric tolerances; link to ISAAC.
- Procedure: Validate claim does not require infinite resources; log budgets.
- Threshold: Budget must be finite and ISAAC-consistent.
- Output/diagnostic: NO-EVAL(RFS1).
- Failure severity: NO-EVAL

RFS2 — Estabilidad PCN (refinamiento de malla)

- Entrada requerida: Barrido de refinamientos (ej. $\times 2$, $\times 3$) en grids relevantes.
- Procedimiento: Ejecutar sweep y verificar que el veredicto no cambia.
- Umbral: Si el veredicto cambia bajo refinamientos razonables \rightarrow NO-EVAL.
- Salida/diagnóstico: NO-EVAL(RFS2): unstable_PCN.
- Severidad al fallar: NO-EVAL

RFS2 — PCN stability (mesh refinement)

- Required input: Refinement sweep (e.g., $\times 2$, $\times 3$) for relevant grids.
- Procedure: Run sweep and verify verdict does not change.
- Threshold: If verdict flips under reasonable refinements \rightarrow NO-EVAL.
- Output/diagnostic: NO-EVAL(RFS2): unstable_PCN.
- Failure severity: NO-EVAL

RFS3 — Estabilidad PCD (tolerancias/regularización)

- Entrada requerida: Barrido de eps/tolerancias y regularizadores declarados.

- Procedimiento: Ejecutar sweep (0.1×, 1×, 10×) y verificar invariancia del veredicto.
- Umbral: Si el veredicto cambia bajo tolerancias razonables → NO-EVAL.
- Salida/diagnóstico: NO-EVAL(RFS3): unstable_PCD.
- Severidad al fallar: NO-EVAL

RFS3 — PCD stability (tolerances/regularization)

- Required input: Sweep eps/tolerances and declared regularizers.
- Procedure: Run sweep (0.1×, 1×, 10×) and verify verdict invariance.
- Threshold: If verdict flips under reasonable tolerances → NO-EVAL.
- Output/diagnostic: NO-EVAL(RFS3): unstable_PCD.
- Failure severity: NO-EVAL

RFS4 — Sensibilidad a idealizaciones

- Entrada requerida: Lista de idealizaciones (límite termodinámico, control perfecto, etc.) y su aproximación finita.
- Procedimiento: Demostrar robustez al reemplazar idealización por aproximación finita.
- Umbral: Si solo funciona en idealización infinita → NO-EVAL.
- Salida/diagnóstico: NO-EVAL(RFS4).
- Severidad al fallar: NO-EVAL

RFS4 — Sensitivity to idealizations

- Required input: List idealizations (thermodynamic limit, perfect control, etc.) and their finite approximation.
- Procedure: Show robustness when replacing idealization by finite approximation.
- Threshold: If it only works in an infinite idealization → NO-EVAL.
- Output/diagnostic: NO-EVAL(RFS4).
- Failure severity: NO-EVAL

RFS5 — Auditoría integral del pipeline

- Entrada requerida: Hashes, versiones, seeds, entorno; registros de ejecución.
- Procedimiento: Reproducir corrida y verificar coincidencia.
- Umbral: Debe ser replicable end-to-end.
- Salida/diagnóstico: NO-EVAL(RFS5).
- Severidad al fallar: NO-EVAL

RFS5 — End-to-end pipeline audit

- Required input: Hashes, versions, seeds, environment; execution logs.
- Procedure: Reproduce run and verify match.
- Threshold: Must be end-to-end reproducible.
- Output/diagnostic: NO-EVAL(RFS5).
- Failure severity: NO-EVAL

Plantillas de Certificado (PA/IO/RFS) / Certificate templates (PA/IO/RFS)

Estas plantillas son el objeto auditable que el compilador valida. Campos obligatorios (★) deben existir; ausencia → NO-EVAL con razón.

- PA-CERT ★: projection_version, Π _signature, assumptions, delta_proj_method, delta_proj(E), kappa_rule, hashes(code/data), seeds, run_log
- IO-CERT ★: map_O_theta, theta_effective_list, observables_list, identifiability_metric, threshold_tau, degeneracies_list, dataset_equivalence
- RFS-CERT ★: resource_budget (compute/time/resolution), ISAAC_link, PCN_sweep, PCD_sweep, idealizations_and_finite_surrogates, full_audit_hashes

English:

- PA-CERT ★: projection_version, Π _signature, assumptions, delta_proj_method, delta_proj(E), kappa_rule, hashes(code/data), seeds, run_log
- IO-CERT ★: map_O_theta, theta_effective_list, observables_list, identifiability_metric, threshold_tau, degeneracies_list, dataset_equivalence
- RFS-CERT ★: resource_budget (compute/time/resolution), ISAAC_link, PCN_sweep, PCD_sweep, idealizations_and_finite_surrogates, full_audit_hashes