Operational Consistency Compiler (OCC)

Canonical Compendium (English Edition) - v1.5.0

Date: 2026-02-17
Repository: github.com/MarcoAIsaac/OCC

Scope: canonical theory framework, operational methodology, closed MRD modules, and prediction catalog with reproducible traces.

Editorial note: this English document is generated from the current canonical source set and includes the integrated nuclear section J4/L4C*/L4E*.

Start Here

Read the compendium in this order for fastest onboarding:

1) Formal foundations (Document A+)

2) Methodology and lock architecture (J0-J4 integrated in one judge sequence)

3) MRD modules and reproducible PASS/FAIL/NO-EVAL workflows

4) Prediction set and experimental-facing witness logic

Roadmap

Section	Purpose
Document A+	Formal operational semantics and judge structure.
Document A	Methodological constraints and judge/lock contracts J0-J4.
MRD modules	Executable reproducibility and verdict artifacts.
Predictions	Operationally falsifiable outputs and witness mapping.

OCC Canonical Compendium - TOC

3 Table of Contents

OCC — Operational Consistency Compiler (OCC)

Documento A+ — Defensa formal del compilador/juez universal (PASS/FAIL/NO-EVAL)
con jueces J0–J3, dominio ISAAC y arquitectura multi-frontend

Marco Antonio Isaac Alcuria

Edición canónica (sin versionado) — 14 de febrero de 2026

Propósito: este documento no propone una ontología nueva ni un modelo específico del universo.

Define un método universal y auditable para decidir si una propuesta física (sea un Lagrangiano, un S-matrix, un modelo de ruido en sistemas abiertos, una teoría efectiva, una modificación de gravedad o una hipótesis cosmológica) es evaluable y consistente dentro del dominio operacional accesible.

El método se materializa como un compilador/juez (OCC): traduce entradas declaradas a restricciones inevitables y devuelve un veredicto binario (PASS/FAIL) o, cuando corresponde, NO-EVAL (no evaluable hoy, sin confundirlo con falsación).

Este estándar no pretende restringir la fase creativa temprana. Está diseñado como filtro final: debe aplicarse cuando una propuesta se presenta como candidata a describir el mundo dentro de Ω_I y antes de compararse con datos observacionales (ver Sección 1.5).

Regla editorial: por qué antes que qué. Cada objeto matemático existe solo si resuelve una ambigüedad operacional (qué se mide, con qué acceso, con qué resolución y con qué dependencia de hipótesis UV).

Cada candado se presenta en tres niveles sincronizados: (i) razón física (qué se rompe si se viola), (ii) formulación formal (definición/proposición/teorema), (iii) verificación ejecutable (checker/MRD).

Contenido

1. Marco: por qué un juez operacional es inevitable

La física fundamental tiene un problema estructural que no se resuelve con “más matemática” ni con “más creatividad”: la maleabilidad. Un marco es maleable cuando, frente a un nuevo dato, puede absorber el dato añadiendo parámetros, sectores no observables o hipótesis UV ad hoc, sin pagar un costo operacional verificable. En ese escenario, el universo deja de ser el juez final, porque el modelo se adapta antes de exponerse a una falsación real.

OCC formaliza una disciplina: antes de que una propuesta se presente como “explicación” del mundo, se la obliga a compilar. Compilar significa convertir la propuesta a un conjunto explícito de restricciones inevitables (candados) y a un conjunto explícito de anclajes observacionales, todo dentro de un dominio operacional declarado. La salida no es una “opinión” del autor ni una preferencia estética: es un veredicto lógico-computacional sobre existencia de dinámica compatible.

1.1. Qué significa maleabilidad y por qué es mortal para la ciencia

Un modelo maleable puede sobrevivir indefinidamente porque siempre posee al menos una de estas válvulas de escape:

(a) Parámetros libres no anclados: coeficientes que no están fijados por ningún observable dentro del dominio donde el modelo se aplica.
(b) Sectores invisibles usados como perillas: grados de libertad que, por construcción, no pueden ser medidos, pero se invocan para ajustar discrepancias.
(c) Dependencia de detalles UV no proyectables: el resultado IR depende de estructura por encima de la escala donde la medición pierde sentido.
(d) Cambios de diccionario retroactivos: el modelo cambia su traducción a observables después de ver los datos (p-hacking teórico).

El resultado es que la teoría ya no produce predicciones; produce narrativas. OCC no pretende prohibir narrativa matemática; pretende separar, con una frontera operacional, lo que es evaluable de lo que es especulación no contrastable.

1.2. Qué es un compilador en física (estructura, no metáfora)

En este documento, OCC se usa como nombre propio del marco, y OCC (Operational Consistency Compiler) se usa para enfatizar su rol: no es una teoría adicional, sino un compilador/juez operacional que evalúa teorías bajo un contrato auditable.

La analogía con un compilador es estructura:

Entrada: una propuesta física en algún frontend (S-matrix, EFT, correladores euclídeos, kernels SK, ecuaciones efectivas, etc.). Traducción: un procedimiento auditable que transforma esa entrada en restricciones verificables dentro de un dominio Ω_I. Evaluación: decidir si existe al menos una instancia de dinámica compatible con (candados + anclajes).

La separación “entrada / traducción / evaluación” es parte del blindaje: evita que el autor mezcle suposiciones de frontend con conclusiones físicas.

1.3. Salidas permitidas y su significado lógico

OCC admite exactamente tres salidas, y cada una está blindada semánticamente:

PASS: existe al menos una dinámica que satisface los candados y reproduce los anclajes dentro de Ω_I. FAIL: no existe ninguna dinámica que satisfaga simultáneamente (candados + anclajes) dentro de Ω_I. FAIL es falsación condicionada a supuestos declarados. NO-EVAL: el candidato no es evaluable hoy; falla un juez de evaluabilidad (J0-J3) antes de poder decidir PASS/FAIL con certificados.

La existencia de NO-EVAL es crucial: sin ella se cometerían dos injusticias lógicas (falsar lo no evaluable, o declarar evaluable lo que depende de magia UV).

1.4. Qué significa 'blindaje' en OCC

Blindar no significa “hacer el texto más agresivo” ni “poner más ecuaciones”. Blindar significa:

- Separar concepto de instancia: el concepto se mantiene aunque cambie una fórmula particular por refinamiento instrumental.
- Declarar supuestos: evitar que el crítico encuentre un supuesto oculto y lo use como recurso retórico.
- Convertir objeciones en tests: toda crítica legítima debe traducirse a un experimento o a un contraejemplo ejecutable (MRD).
- Hacer que el debate sea verificable: certificados y auditoría reemplazan insinuaciones.

Con ese estándar, una objeción solo puede triunfar si trae evidencia reproducible, no si trae estilo.

1.5. Dónde y cuándo se aplica OCC /OCC: filtro previo al juez final (el universo)

Una confusión frecuente —y una fuente de rechazo innecesario— es interpretar este marco como una restricción a la creatividad teórica “desde el principio”. No lo es. OCC , entendido como Operational Consistency Compiler (OCC), es un estándar de evaluabilidad y consistencia para el momento en que una propuesta deja de ser un boceto creativo y pasa a presentarse como candidata a describir el mundo dentro de un dominio operativo Ω_I.

La investigación teórica real tiene (al menos) cuatro fases distintas, con derechos y obligaciones diferentes:

(1) Exploración libre: se permiten ideas heurísticas, analogías, modelos de juguete y conjeturas matemáticas que aún no declaran diccionario operacional completo. Aquí el objetivo no es “pasar OCC ”, sino generar estructuras que quizá, más adelante, puedan proyectarse a predicciones.

(2) Formalización interna: el autor fija su lenguaje (frontend), define variables, simetrías y regímenes, y separa qué partes son hipótesis UV y qué partes son afirmaciones IR. Aquí todavía puede haber piezas sin traducir a observables.

(3) Compilación (OCC): cuando el autor desea afirmar “esto explica/descarta/predice X” dentro de Ω_I, está obligado a declarar el contrato mínimo: diccionario, dominio, anclajes observacionales y candados aplicables. En este punto OCC actúa como filtro: si no compila, el output correcto no es “está mal”, sino NO-EVAL (no evaluable hoy, sin confundirlo con falsación).

(4) Confrontación con datos: el juez final sigue siendo el universo. OCC no reemplaza

el dato; evita que una propuesta maleable llegue al dato con válvulas de escape ocultas, y obliga a que el choque con el universo sea limpio.

La razón de esta separación por fases es operacional, no cultural. Si se exige “rigidez” en la fase (1), se mata el espacio de exploración. Si no se exige “rigidez” en la fase (3), se mata la falsabilidad práctica: el modelo se adapta antes de exponerse a un fallo real.

En términos concretos: OCC debe aplicarse cuando existe una afirmación del tipo “dentro de Ω_I, y con este diccionario, mi propuesta predice/explica el conjunto D_obs”. No debe aplicarse como censura a la matemática especulativa; debe aplicarse como estándar mínimo de honestidad cuando esa matemática se presenta como física contrastable.

Qué se exige exactamente en el paso (3):

- Que el autor declare Ω_I (ventana de energía/tiempo/longitud y protocolo instrumental).
- Que declare el diccionario operacional (qué observa, cómo se calcula, en qué régimen).
- Que declare D_obs (compresión explícita de datos o de observables objetivo).
- Que declare qué partes del modelo quedan fuera de Ω_I y cómo proyectan (J1/PA).
- Que acepte que el Sector B (SIA) no es una perilla: si no hay proyección auditable, no hay derecho a ajustar.

Qué ocurre si se intenta violar esta regla de fase:

- Si se usa Sector B como ajuste: se rompe J1 (PA) y el resultado es FAIL o NO-EVAL, dependiendo de si el claim depende de esa reinyección UV.
- Si se altera el diccionario tras ver datos: se rompe el contrato de D_obs y se incurre en p-hacking teórico (FAIL editorial, porque el procedimiento ya no es auditable).
- Si se pretende “pasar por definición” sin checkers: se rompe J3 (RFS) y el veredicto pierde significado computacional.

1.6. Actualización canónica: por qué OCC debe evolucionar con datos sin volverse maleable

OCC /OCC está diseñado para sobrevivir al crecimiento del conocimiento experimental. Ese crecimiento no es una amenaza: es el punto. El compilador existe para operar sobre lo que el universo efectivamente nos permite medir, y ese conjunto cambia con el tiempo por avances instrumentales, nuevos datasets y mejoras en control sistemático.

La clave para que esa actualización no se convierta en maleabilidad es separar, de forma explícita, dos capas: Capa (A) Conceptual (invariante): enunciados sobre qué significa “evaluar” una teoría en un universo con límites causales, cuánticos y gravitacionales. Aquí viven los jueces J0–J3 como conceptos, y los meta-candados sobre auditoría y no reinyección UV. Capa (B) Instanciación (calibrable): ecuaciones, constantes efectivas, tolerancias, protocolos instrumentales, discretizaciones y mapas de dataset que implementan la capa conceptual en un momento histórico dado.

Esta distinción es la misma que se exige dentro del marco para ISAAC:

- Concepto ISAAC: existe un límite operacional de resolución; intentar extraer información más fina requiere portadores con energía-momento suficiente para producir backreaction gravitatoria comparable, destruyendo la operación de medición. Este concepto depende solo de la coexistencia de (i) estructura causal (SR), (ii) cuantización de energía/información (QM) y (iii) gravitación universal (GR).

- Ecuación ISAAC (instancia): una parametrización concreta del umbral L_I y de la ventana Ω_I asociada, que puede refinarse a medida que se entienden mejor protocolos de medición, configuraciones de señal y límites instrumentales reales. Si los datos y la ingeniería obligan a ajustar el prefactor, ISAAC no "muere": se recalibra la instancia; el concepto sigue siendo inevitable.

Lo mismo aplica a los demás jueces:

- J1/PA (Proyección Auditable) como concepto no cambia: ningún claim IR puede depender de detalles UV no proyectables. Lo que sí cambia es el repertorio de proyecciones aceptables y los formatos de certificados conforme se estandariza práctica y tooling.
- J2/IO (Identificabilidad Operacional) como concepto no cambia: no se puede declarar un parámetro como "predicho" si es inidentificable dentro del experimento real. Lo que sí cambia es el modelo de ruido, la matriz de Fisher efectiva y los umbrales numéricos según mejoren incertidumbres y diseño experimental.
- J3/RFS (Recursos Finitos y Estabilidad) como concepto no cambia: un veredicto debe ser robusto bajo refinamiento, tolerancia y reproducibilidad. Lo que sí cambia es el estado del arte en solvers, hardware y prácticas de auditoría.

Regla canónica de actualización (anti-maleabilidad):

(1) Toda actualización del compilador debe venir acompañada de MRDs de regresión y de un registro auditable (hashes, seeds, versiones de dependencias, datasets).

(2) Una actualización está permitida solo si responde a nueva información externa (instrumentación, datos, teoremas o counter-examples) y no a "necesito que mi modelo pase".

(3) Cambios que alteran veredictos deben ser rastreables: debe existir un certificado que explique qué candado cambió, por qué cambió y qué datos/argumentos lo forzaron.

(4) La capa conceptual se modifica únicamente si cambia la física establecida que la fundamenta; mientras SR+QM+GR permanezcan, el núcleo ISAAC permanece.

Con esto, OCC /OCC puede actualizarse indefinidamente sin volverse una "perilla": el compilador no se adapta a una teoría; se adapta a lo que el universo nos deja medir y a lo que las matemáticas inevitables exigen.

2. Nomenclatura canónica y estructura operacional

2.1. Sector A (SGO) y Sector B (SIA): definición mínima

OCC fija una partición operacional, no ontológica.

Sector A (SGO): dominio donde existe geometría efectiva, tiempo operativo y protocolos de medición reproducibles. Sector B (SIA): complemento definido por inaccesibilidad operacional. B no se define por misterio: se define por inexistencia de procedimiento operacional reproducible en A que reconstruya el microestado de B.

El método no niega B. Prohíbe usar B como perilla libre para ajustar observables en A sin proyección auditable.

2.2. Por qué la división A/B es obligatoria y no opcional

Sin una frontera A/B, cualquier discrepancia puede “explicarse” introduciendo un sector invisible o una microdinámica inaccesible. Eso produce teorías irrefutables por construcción. La división A/B impone una regla: toda explicación que afecte un observable en A debe pagar un precio operacional en A (parámetro identificable o restricción inevitable).

2.3. Condición mínima del Sector B: no-backflow micro

El requisito mínimo que define B es ausencia de backflow micro recuperable.

Modelamos la dinámica efectiva en A como un canal: $\rho_A' = \Phi(\rho_A) = \mathrm{Tr}_B[U_{AB} (\rho_A \otimes \sigma_B) U_{AB}^\dagger]$.

No-backflow micro afirma: dentro de los protocolos permitidos en A, no existe un procedimiento que permita identificar σ_B (microestado de B) a partir de estadísticos en A de forma reproducible y auditable. Dicho directo: A no puede hacer tomografía del interior de B.

Si A pudiera reconstruir σ_B, entonces B dejaría de ser inaccesible y pasaría a A. Si A no puede reconstruir σ_B, invocar σ_B como ajuste de datos carece de significado operacional.

2.4. Qué ocurre si se viola no-backflow micro

Si existiera backflow micro recuperable, colapsa la distinción A/B, reaparece maleabilidad infinita y, en muchos escenarios, se habilitan protocolos de extracción de información que chocan con causalidad o con límites termodinámicos operativos. Por eso OCC trata no-backflow micro como requisito constitutivo: si un candidato lo necesita para “salvar” predicciones, el resultado es NO-EVAL o FAIL según consecuencias operacionales.

3. Jueces fundacionales J0-J3 (blindaje por construcción)

Los jueces no son candados. Un candado restringe dinámicas; un juez restringe qué claims son evaluables sin reinyección UV y sin perillas no identificables. Los jueces son el núcleo indestructible del marco: atacar un juez exige atacar estructura física ya verificada o confesar que el claim queda fuera de Ω_I.

3.0. Diferencia formal entre candado y juez

Candado (C_k): condición necesaria de consistencia física en A. Si se viola en Ω_I, corresponde FAIL. Juez (J_m): condición necesaria de evaluabilidad. Si se viola, corresponde NO-EVAL (salvo que además implique inconsistencia).

Este diseño evita dos errores: falsar lo no evaluable y legitimar lo irrefutable.

3.1. J0 — ISAAC: cierre operacional por SR + QM + GR

Concepto (invariante): existe un umbral de resolución por debajo del cual cualquier intento de extraer información local mediante intercambio de energía-momento en A deja de ser operacional, porque la energía usada para “ver” autogravita comparativamente y destruye el régimen geométrico de A.

ISAAC depende de tres hechos:
(1) límite causal de señalización (c),
(2) portadores de información con energía-momento (E=ħω),
(3) backreaction gravitatoria de energía-momento (GR).

Si aceptas (1)-(3), aceptas la existencia de un límite operacional. El número exacto puede refinarse; el concepto no puede eliminarse sin negar alguna pieza base.

3.1.1. Derivación mínima (sin saltos)

Para resolver longitud □ se requiere excitación con p ~ ħ/□ y E ~ ħc/□. E concentrada en región □ induce radio gravitacional r_s ~ 2GE/c^4. La medición colapsa cuando r_s □ □. Sustituyendo:

r_s ~ 2G(ħc/□)/c^4 = 2ħG/(c^3 □).

r_s □ □ ⇒ □^2 □ 2ħG/c^3 ⇒ □ □ √(2ħG/c^3) ~ O(1)√(ħG/c^3).

Esto es la longitud de Planck salvo factores O(1) de protocolo. No usa modelo de gravedad cuántica: solo SR+QM+GR.

p ~ ħ/□, E ~ ħ c / □ r_s ~ 2 G E / c^4

r_s □ □ ⇒ 2 ħ G / (c^3 □) □ □ ⇒ □^2 □ 2 ħ G / c^3 ⇒ □ □ √(2 ħ G / c^3) ~ L_I

3.1.2. Separación concepto-ecuación (blindaje)

La ecuación L_I ~ √(ħG/c^3) es una instanciación mínima, no el corazón.

El concepto ISAAC afirma existencia de un umbral operacional de backreaction O(1).

La cifra puede cambiar por:
- factores numéricos,
- protocolos distribuidos,
- geometrías curvas,
- criterios informacionales.

Qué puede cambiar sin matar ISAAC: factores O(1), definiciones de “local”, detalles instrumentales. Qué mataría ISAAC: (a) señalización superluminal operacional, (b) medición sin energía-momento, (c) energía-momento sin gravitar.

Por tanto ISAAC es indestructible salvo revolución experimental de SR/QM/GR.

3.1.3. ISAAC como techo espectral y dominio Ω_I

ISAAC induce una cota operacional Λ_I. No es “energía máxima del universo”; es energía máxima donde A puede exigir consistencia directa.

Define Ω_I como región donde se evalúan candados: Ω_I = { observables con resolución y banda ≤ (Δ, Λ_I) }.

En S-matrix, ISAAC se traduce en “no reinyección UV”: ninguna predicción en Ω_I puede depender de amplitud por encima de Λ_I sin proyección auditable. En sistemas abiertos, ISAAC se traduce en band-limit: kernels no pueden requerir soporte espectral no resoluble.

ISAAC es juez: delimita jurisdicción del compilador.

3.2. J1 — Proyección Auditable (PA)

Concepto: si una teoría usa estructura fuera de Ω_I para afectar predicciones dentro de Ω_I, solo es evaluable si existe una proyección Π hacia Ω_I con error acotado y auditable.

OCC no niega microfísica UV. Prohíbe usarla como perilla libre en A sin puente finito y verificable.

3.2.1. Formalización mínima de PA

Sea T_{UV} un objeto teórico fuera de Ω_I y O_A el conjunto de observables en A. PA exige existencia de: $\Pi : T_{UV} \to \Theta_{eff}(\Omega_I)$, más una cota Δ_{proj} tal que para todo observable relevante O: $|O(T_{UV}) - O(\Pi(T_{UV}))| \leq \Delta_{proj}$, con Δ_{proj} calculable o certificable en MRD.

Si Δ_{proj} □ σ_{data}, UV es consumible en A.

Si Π no existe o Δ_{proj} no se puede auditar, el resultado correcto es NO-EVAL.

$\Pi : T_{UV} \to \Theta_{eff}(\Omega_I)$ $| O(T_{UV}) - O(\Theta_{eff}) | \leq \Delta_{proj}$

Δ_{proj} □ $\sigma_{data} \Rightarrow$ evaluable Δ_{proj} □ $\sigma_{data} \Rightarrow$ NO-EVAL (proyección no informativa)

3.2.2. Separación concepto-ecuación para PA

El concepto PA no depende de una forma particular de Π. Π puede ser:
- integración de modos UV (RG),
- matching EFT por observables,
- coarse-graining SK,
- proyección estadística (modelo efectivo de ruido),
- o proyección topológica (anomalías).

Lo que no puede ser: “seleccionar” a posteriori la UV que conviene sin algoritmo.

Qué puede cambiar: forma de Π, detalles de error, métodos de estimación. Qué no puede cambiar: obligación de declarar Π y su error si la UV afecta Ω_I.

Blindaje: el crítico puede discutir si Δ_{proj} está bien estimado; no puede eliminar la necesidad de Π sin reintroducir irrefutabilidad.

3.2.3. Qué ocurre si PA se viola

Sin PA, la teoría puede ser matemática correcta pero física irrefutable: el valor de un observable en A queda suspendido de detalles UV no acotables por protocolos en A.

Consecuencias:
- dependencia infinita de elecciones,
- ajuste retroactivo,
- ausencia de diagnóstico cuando falla.

OCC traduce esto a NO-EVAL. No castiga la matemática; castiga el claim físico no contrastable.

3.3. J2 — Identificabilidad Operacional (IO)

Concepto: una propuesta solo predice si sus parámetros efectivos que afectan observables en Ω_I son identificables con datos en Ω_I.

Sin IO, una teoría puede pasar PA y aún así tener direcciones planas (infinitas deformaciones indistinguibles). Eso no es “malo”, pero debe declararse como falta de cierre, no como predicción.

3.3.1. Formalización local: Jacobiano y Fisher

Sea θ ∈ Θ vector de parámetros efectivos y O(θ) ∈ R^m vector de observables comprimidos. Jacobiano J = ∂O/∂θ. Información de Fisher F = J^T C^{-1} J.

Direcciones con autovalores pequeños de F no son identificables: el dato no distingue. Si el claim depende de esas direcciones, corresponde NO-EVAL. Si el autor declara el resultado como abierto, puede ser PASS-débil.

J = ∂O/∂θ F = J^T C^{-1} J

λ_min(F) ≈ 0 ⇒ no identificable

Claim depende de dirección plana ⇒ NO-EVAL Claim no depende ⇒ PASS con advertencia

3.3.2. IO global: degeneraciones y equivalencias operacionales

IO no es solo local. Puede ocurrir que dos regiones desconectadas Θ1 y Θ2 produzcan los mismos observables en Ω_I (degeneración global). En ese caso, el compilador no elige ontología: declara equivalencia operacional y exige predicciones adicionales que rompan degeneración.

Esta regla es importante para cosmología y para modelos con sectores oscuros: el mismo observable puede explicarse por cambios en “vacío”, “materia oscura efectiva”, o “modificación gravitatoria” dentro de errores. OCC no resuelve ontología; resuelve evaluabilidad. La ontología solo se decide cuando existe un observable que la distinga en Ω_I.

3.3.3. Separación concepto-ecuación para IO

El concepto IO no depende de usar Fisher específicamente. Fisher es una instanciación conveniente cuando existe un modelo estadístico aproximadamente gaussiano. En otros casos, IO se puede certificar mediante:

- identificabilidad estructural (teoría de sistemas),
- rank de Jacobiano simbólico,
- o criterio bayesiano (posterior no informativo).

Lo innegociable es el concepto: no puedes llamar “predicción” a un parámetro que el dato no puede identificar.

Esto blinda OCC contra la objeción “tu PASS es solo porque te dejaste parámetros libres”: IO exige declarar ese hecho y cuantificarlo.

3.4. J3 — Recursos Finitos y Estabilidad (RFS)

Concepto: toda evaluación ocurre con recursos finitos. Un veredicto sin estabilidad numérica e instrumental es vulnerable a objeciones triviales (“cambia la tolerancia y cambia el resultado”).

RFS impone dos obligaciones:

(1) estabilidad bajo refinamientos razonables,

(2) certificación reproducible (hashes, seeds, versiones, duales cuando existan).

RFS no es un lujo: es lo que convierte el marco en ciencia ejecutable.

3.4.1. Formalización mínima: vecindario de auditoría

Sea I0 una entrada nominal (Ω_I, discretización, tolerancias, datos). RFS exige definir un vecindario A de variaciones permitidas (refinamiento de malla, rango de tolerancias, parametrizaciones equivalentes, cambios menores de dataset dentro del mismo release). El veredicto debe ser estable en A, salvo transiciones clasificadas y diagnosticadas.

Si no se puede definir o certificar A, corresponde NO-EVAL (no se entrega PASS/FAIL sin estabilidad).

Entrada I0 = (Ω_I, D_h, τ, D_obs, frontend)

A = vecindario de auditoría (variaciones permitidas)

Requisito: V(I) estable en A Si no: NO-EVAL por RFS

3.4.2. Separación concepto-ecuación para RFS

RFS no depende de un solver particular. Depende de un principio: el veredicto no puede ser un artefacto de discretización.

Qué puede cambiar: solver, tolerancias específicas, backend numérico. Qué no puede cambiar: existencia de protocolo de estabilidad, y publicación de configuración y hashes.

RFS convierte “el solver dice” en “el certificado demuestra”. Ese es el blindaje.

3.4.3. Qué ocurre si RFS se viola

Si RFS se viola, el marco queda expuesto a:
- objeción por tolerancia,
- objeción por discretización,
- objeción por irreproducibilidad (“no puedo correr tu resultado”).

OCC lo previene clasificando el resultado como NO-EVAL hasta que haya certificado. No se concede legitimidad a un PASS sin reproducibilidad.

4. Candados universales y arquitectura multi-frontend

La universalidad de OCC no se obtiene imponiendo un único lenguaje. Se obtiene por arquitectura: múltiples frontends que describen física en lenguajes distintos y compilan al mismo juez.

Cada frontend hace visibles ciertos candados y oculta otros. OCC, por tanto, no impone ontología; impone consistencia transversal a representaciones.

4.1. Meta-candados C0 (pre-física)

C0-1 Operacionalidad: toda cantidad usada debe corresponder a protocolo en A. C0-2 Cierre ISAAC: no exigir sub-resolución ni reinyección UV. C0-3 Auditabilidad: corrida reproducible (hashes, seeds, versiones). C0-4 Diagnóstico: FAIL debe indicar candado violado; PASS debe indicar región viable. C0-5 Invariancia de representación: frontends equivalentes deben concordar tras diccionario.

Estos meta-candados son el escudo contra “ambigüedad de lenguaje”.

4.2. Frontends mínimos y candados visibles

Frontends canónicos:

C1–C6 S-matrix relativista: unitaridad, analiticidad, causalidad/retardo, crossing, acotamiento, positividad. C7 Correladores euclídeos: axiomas OS, reflection positivity. C8 Sistemas abiertos (SK): causalidad, PSD del ruido, CPTP, consistencia A/B. C9 Equilibrio térmico: KMS y FDT. C10 Simetrías/anomalías/topología: matching y cuantización. C11 Relativista local: CPT como test de supuestos.

La lista es extensible: se pueden agregar frontends, pero deben respetar jueces J0-J3.

4.3. Candados C1–C6 en S-matrix (con derivaciones más explícitas)

4.3.1. C1: Unitaridad y teorema óptico (derivación)

Partimos de S=1+iT. Unitaridad S†S=1 implica: (1-iT†)(1+iT)=1 ⇒ i(T-T†)=T†T.

Tomando elementos entre estados asintóticos |i⟩: 2 Im T_{ii} = Σ_f ⟨i|T†|f⟩⟨f|T|i⟩ = Σ_f |T_{fi}|^2 ≥ 0.

La desigualdad es el corazón: Im amplitud es no negativa porque es suma de módulos cuadrados. Violar esto implica secciones eficaces negativas o probabilidad no conservada. Por eso C1 es candado inevitable en ese frontend.

S = 1 + iT S†S=1 ⇒ i(T - T†) = T†T

Elemento diagonal: 2 Im T_{ii} = Σ_f |T_{fi}|^2 ≥ 0

4.3.2. C2: Analiticidad y causalidad (idea y consecuencia)

Analiticidad no se postula por gusto; se deduce (bajo supuestos técnicos) de microcausalidad y espectro.

Idea: en QFT local, conmutadores se anulan fuera del cono de luz. Eso impone que las funciones de Green tengan estructura de singularidades controlada. Como consecuencia, la amplitud A(s,t) se extiende como función holomorfa en s (para t fijo) excepto por cortes/polos físicos.

Para OCC , esto significa: si un candidato rompe analiticidad dentro de Ω_I, debe declarar que abandonó el régimen local relativista y debe ofrecer un frontend alternativo para evaluación. Si no lo ofrece, el resultado es NO-EVAL (no existe compilación auditable).

4.3.3. C3: Crossing (por qué es inevitable cuando el frontend aplica)

Crossing liga procesos distintos a un único objeto analítico. En términos de variables de Mandelstam, permutar partículas entre estados in/out corresponde a continuar A(s,t) a otra región analítica.

Si se asume QFT local y campos relativistas, crossing es parte de la consistencia. Violarlo arbitrariamente equivale a decir que procesos relacionados no comparten dinámica: eso casi siempre significa sector adicional no declarado o ruptura de supuestos.

En OCC , crossing es candado condicional: aplica cuando el autor declara el régimen donde se derivan los supuestos. Si el autor no lo declara, el compilador no lo impone; exige aclaración (NO-EVAL) o trata el claim en otro frontend.

4.3.4. C4: Acotamiento polinomial, Froissart y número de sustracciones

Para usar Cauchy y construir dispersiones, se requiere que la amplitud no crezca demasiado rápido en el contorno grande. En teorías con masa y gap, existe el bound de Froissart que sugiere crecimiento limitado (~ s log^2 s). En otros casos se usa acotamiento polinomial con N sustracciones.

En OCC , el número de sustracciones N_sub no es una perilla invisible. Debe declararse, justificarse, y entrar en RFS: cambiar N_sub dentro de un rango razonable no debe convertir PASS en FAIL sin diagnóstico. El rol de ISAAC aquí es clave: no se exige control del infinito UV; se exige control en Ω_I, con remanentes tratados como parámetros efectivos y con error proyectivo auditado.

4.3.5. C5: Causalidad/retardo en amplitudes (dispersión substraída)

Causalidad operacional implica que la respuesta no precede a la causa. En amplitudes, esto se traduce en restricciones sobre la parte real de la amplitud y sobre su expansión efectiva.

Una señal típica de acausalidad efectiva es superluminalidad en EFT (por ejemplo, velocidades de grupo >1 en presencia de ciertos operadores con signos erróneos). Los bounds de causalidad pueden formularse como condiciones de positividad sobre coeficientes cuando se asume analiticidad y unitaridad.

OCC trata causalidad como candado: si una EFT predice superluminalidad operacional dentro de Ω_I, eso es FAIL. Si la superluminalidad aparece solo en un régimen fuera de Ω_I, el dominio estaba mal declarado y debe corregirse (J0).

4.3.6. C6: Positividad (derivación extendida y advertencias)

Positividad se deriva combinando:

- unitaridad (Im A ≥ 0),
- analiticidad,
- crossing,
- y acotamiento suficiente.

En el caso más simple (sin intercambio masivo singular en t=0), la segunda derivada del forward amplitude substraído es integral de una función no negativa y debe ser positiva. Esa derivación se detalla en el Apéndice B.

Advertencia crítica: si hay intercambio sin masa (fotón, graviton), el forward limit requiere tratamiento IR-safe. OCC no aplica positividad “a ciegas”: exige declarar el observable IR-safe y auditar sustracciones. Si el autor no puede hacerlo, no hay FAIL automático: hay NO-EVAL por falta de compilación correcta.

4.4. C7: correladores euclídeos y axiomas OS (por qué importan)

Cuando se trabaja en Euclídeo (lattice, tiempo imaginario), el puente a Minkowski no es automático. Reflection positivity es el candado que garantiza existencia de Hilbert con norma positiva

tras Wick rotation. Sin él aparecen estados de norma negativa: no hay probabilidades. Por eso C7 es candado inevitable en ese frontend.

4.5. C8: sistemas abiertos (SK) y canales CPTP

En presencia de entorno/coarse-graining, el objeto fundamental es un canal Φ sobre estados reducidos. C8 incluye cuatro subcandados:

- causalidad retardada (respuesta),
- positividad del ruido (PSD),
- CPTP (Choi/Kraus),
- consistencia A/B (no usar B como perilla).

La violación de CPTP produce densidades con autovalores negativos o traza incorrecta: inconsistencia física (FAIL).

4.5.1. Choi: CP como PSD

El isomorfismo de Choi convierte “CP para toda ancilla” en un test matricial: $J(\Phi) = (\Phi \otimes I)(|\Omega\rangle\langle\Omega|)$. Entonces CP $\Leftrightarrow J \succeq 0$, TP $\Leftrightarrow \mathrm{Tr\_out}\, J = I$.

Esto transforma un principio físico en verificación ejecutable.

$J(\Phi) = (\Phi \otimes I)(|\Omega\rangle\langle\Omega|)$ CP $\Leftrightarrow J(\Phi) \succeq 0$ TP $\Leftrightarrow \mathrm{Tr\_out}\, J(\Phi) = I$

4.5.2. Causalidad SK: estructura triangular (retardo)

En SK, la matriz de correladores tiene estructura causal. En variables (clásica/cuántica) o (ra), la respuesta retarded $D_R(t,t')$ debe anularse para $t<t'$: $D_R(t,t') = 0$ para $t < t'$.

Este es el candado de causalidad: la perturbación no puede afectar el pasado en A. Se verifica en discretización como matriz triangular (dentro de tolerancia).

Violar esto implica señalización acausal dentro de Ω_I: FAIL.

Causalidad retarded:

$D_R(t,t') = 0$ si $t < t'$

Discretizado: $(D_R)_{ij} = 0$ para $i < j$ (índices temporales ordenados)

4.5.3. Ruido PSD y FDT (cuando aplica)

El kernel de ruido $N(t,t')$ define varianzas. Debe ser forma positiva: $\int f^* N f \geq 0$.

En frecuencia: $N(\omega) \geq 0$.

En equilibrio, FDT liga N y D_R: $N(\omega) = \coth(\beta\omega/2)\, \mathrm{Im}\, D_R(\omega)$. Si se declara equilibrio y esto falla, hay inconsistencia o declaración falsa: debe resolverse explícitamente.

4.6. C9: KMS y consistencia térmica (derivación desde traza)

KMS se deriva de la cyclicidad de la traza. Para un estado térmico $\rho = e^{-\beta H}/Z$:

$\langle A(t)B(0)\rangle = \mathrm{Tr}(\rho\, e^{iHt} A e^{-iHt} B) = \mathrm{Tr}(e^{-\beta H} e^{iHt} A e^{-iHt} B)/Z$.

Insertando $e^{-\beta H}$ y usando cyclicidad se obtiene: $\langle A(t)B(0)\rangle = \langle B(0)A(t+i\beta)\rangle$.

Ese es el contenido de KMS. No es una aproximación: es el criterio exacto de equilibrio. Por tanto, si se usa equilibrio como input y KMS falla, el input es incoherente.

4.7. C10: anomalías y topología operacional (Wess–Zumino)

Una anomalía es una obstrucción cuántica a una simetría clásica. No se “apaga” por gusto. Wess–Zumino impone condiciones de consistencia en cómo varía el funcional efectivo bajo transformaciones.

Anomaly matching afirma: anomalías globales calculadas en UV deben reproducirse en IR mediante grados de libertad o términos topológicos. Si la proyección IR elimina la anomalía sin mecanismo, viola consistencia: FAIL o NO-EVAL por PA (si la proyección fue mal definida).

4.8. C11: CPT como test de supuestos

CPT no se usa como dogma. Se usa como detector: si un candidato viola CPT, debe declarar qué hipótesis abandona (localidad, Lorentz, espectro, microcausalidad) y mostrar consistencia operacional en Ω_I. Si no lo hace, la violación es indicio de inconsistencia o representación incompleta.

5. Especificación formal del compilador OCC

5.1. Objetos canónicos y conjunto factible

Definimos:

Frontend F con lenguaje L_F y parámetros efectivos $\theta \in \Theta_F$. Datos comprimidos D_{obs} (con ventanas, covarianzas y pipeline). Candados K (restricciones inevitables). Dominio Ω_I (ventana operacional).

Con esto: $I(\Omega_I) = \{\theta \in \Theta_F : K(\theta;\Omega_I)=\text{True} \wedge D_{obs}(\theta;\Omega_I)=\text{True} \}$.

PASS ⇔ I ≠ ∅. FAIL ⇔ I = ∅ (con certificado cuando sea posible). NO-EVAL ⇔ falla un juez o falta certificado RFS antes de concluir.

5.2. D_obs: compresión explícita para evitar p-hacking

D_obs debe ser un objeto versionado:
- ID de dataset y hashes,
- pipeline de compresión,
- ventana y kernels instrumentales,
- matriz de covarianza o modelado de sistemáticos.

Sin compresión explícita, el autor puede “mover” el observable hasta forzar PASS. OCC evita eso haciendo de D_obs parte del contrato auditado.

5.3. Algoritmo canónico (contrato)

Paso 0: jueces J0-J3. Si falla alguno, NO-EVAL con diagnóstico. Paso 1: normalización y diccionario. Paso 2: instanciar candados. Paso 3: instanciar anclajes. Paso 4: resolver factibilidad. Paso 5: certificar (primal/dual). Paso 6: rigidez. Paso 7: auditoría y empaquetado MRD.

5.4. Por qué PASS/FAIL es binario pero la ciencia no lo es

El veredicto es binario (existencia de solución), pero el marco incorpora incertidumbre explicitando: Ω_I, supuestos, errores proyectivos/EFT, y rigidez R. Así, el marco puede evolucionar con datos sin perder significado lógico.

6. Dominio Ω_I: ventana, resolución e ISAAC

Ω_I es donde vive el método. Sin Ω_I, cualquier afirmación es ambigua. OCC obliga a declarar Ω_I y a respetar resolución instrumental y validez EFT.

Esto impide mover la ventana a posteriori para salvar modelos.

6.1. Reglas PCD (Protocolo Canónico de Dominio)

PCD-1 Ventana: Ω_I debe estar dentro de ventana calibrada. PCD-2 Resolución: no exigir sub-resolución; convolucionar con kernel instrumental. PCD-3 Validez EFT: $E/\Lambda \le \varepsilon_\Lambda$ declarado y propagar error. PCD-4 Sensibilidad: reportar estabilidad bajo variaciones de Ω_I. PCD-5 Congelamiento temporal: declarar Δt y horizonte en SK.

Estas reglas convierten Ω_I en un objeto físico, no retórico.

6.2. ISAAC como prohibición de reinyección UV

Reinyección UV es la forma más común de maleabilidad. ISAAC+PA prohíben usar detalles UV no accesibles como perillas en Ω_I. Si una integral requiere UV, se hacen sustracciones y se parametriza remanente con error auditable. Si un kernel requiere banda extra, se coarse-grain con error auditado. No se prohíbe estudiar UV; se prohíbe fingir que es observable.

7. Rigidez: cuánto decide el universo y cuánto decide el autor

7.1. Por qué la rigidez es un objeto físico

Dos teorías pueden dar PASS, pero una puede ser rígida y otra maleable. La rigidez cuantifica el costo de compatibilidad: cuánto recortan los candados la región compatible con datos. Sin rigidez, PASS puede ser trivial (siempre hay parámetros libres).

7.2. Definición canónica: R como razón de volúmenes

Sea μ una medida declarada sobre Θ_F. $R = \mu(I)/\mu(D_{obs})$. $R \ll 1$ indica PASS fuerte; $R \sim 1$ indica PASS débil. μ debe justificarse por principio operacional (resolución, Jeffreys, prior declarado) y su sensibilidad debe auditarse (RFS).

$R = \mu(I) / \mu(D_{obs})$

$R \ll 1 \Rightarrow$ rígido $R \sim 1 \Rightarrow$ maleable $R = 0 \Rightarrow$ FAIL

7.3. Dimensión efectiva y degeneraciones

Además del volumen, se reporta dimensión efectiva local d_{eff} para capturar valles. d_{eff} se estima por rank de Jacobianos/constraints activas. IO y rigidez están conectados: direcciones no identificables inflan d_{eff}.

7.4. Cómputo auditado de R

R se estima por sampling verificado o por métodos convexos según el problema. Auditoría exige seeds, convergencia bajo N y reporte de error estadístico. Un R sin auditoría es NO-EVAL por RFS.

8. Zonas delicadas: IR, gravedad y no-localidad

8.1. IR singular y candados IR-safe

Con intercambio sin masa, el forward limit puede divergir. OCC exige formular candados en observables IR-safe (substraídos o a t<0 fijo). Esto evita falsos FAILs y blinda contra críticas técnicas legítimas.

8.2. Gravedad y positividad

Con gravitón hay polo t=0. Positividad naive puede fallar; se requiere sustracción y error auditado. Sin declarar observable IR-safe, el resultado es NO-EVAL (RFS/PA), no PASS ni FAIL.

8.3. No-localidad: FAIL vs NO-EVAL

No-localidad puede ser (i) efectiva controlada con escala y error, o (ii) perilla fundamental sin proyección. Si habilita señalización acausal en Ω_I, es FAIL. Si solo invalida el frontend sin ofrecer otro, es NO-EVAL. La diferencia se decide por consecuencias operacionales, no por prejuicio.

9. Auditoría, MRD y ciencia como artefacto ejecutable

9.1. Por qué MRD es obligatorio

Sin reproducibilidad, PASS/FAIL es retórica. MRD es el paquete mínimo para reproducir el veredicto con hashes, seeds, datasets, y configuración. MMRD es evidencia máxima reproducible sin dependencias opacas (notebooks, tests, duales).

9.2. Bloque mínimo de auditoría

Toda corrida debe incluir: versión del compilador, hashes de entradas/datasets, seeds, config de solver, entorno, veredicto, residuals, certificados PA/IO/RFS y rigidez. Sin esto: NO-EVAL.

9.3. Certificados primal/dual

PASS requiere punto factible + residuals. FAIL idealmente requiere testigo dual (Farkas/dual SDP). Si el módulo no puede certificar, no se entrega FAIL: se entrega NO-EVAL por RFS. Esto blinda contra la objeción “tu FAIL es por bug”.

10. Matriz de objeciones técnicas esperables y respuestas

OCC desplaza el debate de estética a compilabilidad; por eso será atacado. La defensa es estructural: identificar juez/candado cuestionado y traducir crítica a test auditado.

10.1. 'ISAAC es demasiado restrictivo'

ISAAC no es restrictivo: describe consecuencia de SR+QM+GR. Evitarlo sin costo exige negar c como límite, negar energía-momento de portadores, o negar backreaction de energía-momento. Si alguien propone eso, propone nueva física básica y debe someterla a Ω_I. Lo discutible es el factor numérico del umbral, no el concepto.

10.2. 'Tu veredicto depende de tolerancias'

Eso es precisamente RFS. Sin estabilidad numérica, el resultado es NO-EVAL. La crítica legítima debe reproducirse dentro del vecindario auditado, no insinuarse.

10.3. 'Mueves arbitrariedad a Ω_I, N_sub, etc.'

Diferencia: arbitrariedad oculta vs elección declarada. OCC obliga a declarar Ω_I y parámetros técnicos, justificar por principios, y reportar sensibilidad. Eso elimina perillas invisibles.

10.4. 'Esto es filosofía'

OCC es metodología ejecutable. Para refutarlo se necesita un contraejemplo: un claim que OCC clasifica NO-EVAL/FAIL pero que se puede contrastar operacionalmente sin PA/IO/RFS dentro de Ω_I. Sin ese contraejemplo, la crítica es retórica.

11. Supuestos técnicos, loopholes y alcance

El blindaje exige declarar jurisdicción. OCC declara supuestos técnicos (S) y loopholes (L). Un loophole no es defecto: es un hueco donde se requiere módulo nuevo o frontend nuevo. La honestidad es parte del marco.

11.1. Supuestos técnicos típicos

S1 separación de escalas, S2 estacionariedad/localidad efectiva, S3 estados asintóticos (S-matrix), S4 control IR, S5 truncación controlada, S6 diccionario/gauge, S7 invariantes, S8 sistemáticos, S9 IO, S10 RFS, S11 PA, S12 congruencia inter-frontend. Cada corrida debe declarar qué usa y qué evidencia aporta.

11.2. Loopholes típicos y manejo

L1 IR no tratado ⇒ NO-EVAL. L2 no-localidad UV sin Π ⇒ NO-EVAL. L3 dependencia de truncación ⇒ NO-EVAL o PASS-débil. L4 diccionario ambiguo ⇒ NO-EVAL. L5 degeneración ontológica ⇒ equivalencia operacional; exigir observable nuevo. L6 falta de datos ⇒ PASS-débil/NO-EVAL. L7 recursos insuficientes ⇒ NO-EVAL. L8 cambio de régimen ⇒ modelar cruce o NO-EVAL.

12. Mapa de implementación: del texto al código

Para evitar la objeción “suena bien pero no corre”, OCC publica el mapa entre conceptos y verificadores ejecutables. Ninguna afirmación de PASS/FAIL es aceptable sin su MRD correspondiente.

Objeto Propósito Evidencia ejecutable típica

J0 (ISAAC) Define Ω_I y prohíbe reinyección UV

Reglas PCD + check de ventana/resolución; MRD de observabilidad/ISAAC

J1 (PA) Exige proyección Π con error Δ_proj

Certificado PA (JSON) + checker de esquema; MRD UV→Ω_I

J2 (IO) Exige identificabilidad de parámetros efectivos

Certificado IO + Jacobiano/Fisher; MRD de identificabilidad

J3 (RFS) Exige estabilidad y reproducibilidad

Certificado RFS + logs/hashes; tests de estabilidad en CI

C1–C6 (S-matrix)

Unitaridad/analiticidad/crossing/ positividad

MRD amplitudes/positividad; dual SDP cuando aplique

C7 (OS) Reflection positivity y reconstrucción

MRD euclídeo; tests de Gram/positividad

C8–C9 (SK/KMS)

CPTP + PSD + FDT/KMS MRD SK; checkers Choi/PSD/FDT/KMS

C10 (anomalías) Matching UV/IR y cuantización MRD simetrías/anomalías; tests de cuantización

C11 (CPT) Detector de supuestos Checks de consistencia CPT donde aplique

La disciplina OCC exige un índice canónico versionado que liste, para cada afirmación, su artefacto ejecutable asociado. Ese índice no es burocracia: es el blindaje contra ambigüedad de versiones y contra objeciones por confusión editorial.

13. Casos de estudio: choques con programas populares (cómo clasifica OCC)

Este capítulo no pretende “refutar” escuelas completas; pretende mostrar cómo OCC clasifica claims típicos cuando se les exige compilabilidad. El resultado importante suele ser NO-EVAL, no FAIL: muchas propuestas pueden ser matemáticamente ricas pero no proyectables/identificables en Ω_I.

13.1. Teoría de cuerdas: landscape y microestructura trans-Planck

Un claim típico: “existen 10^N vacíos diferenciados por física trans-Planck que determinan constantes IR”. Dentro de OCC , el punto de choque no es la matemática; es PA.

Para que ese claim sea evaluable, debe existir una proyección Π que mapee la microestructura UV (compactificación, fluxes, etc.) a parámetros efectivos Θ_eff en Ω_I, con error Δ_proj auditado. Si la selección del vacío se hace a posteriori para ajustar datos, no hay Π; hay reinyección UV. Resultado: NO-EVAL por J1 (PA), no “censura”.

Si, en cambio, la teoría produce una predicción efectiva específica en Ω_I (por ejemplo, un espectro o una relación de acoplamientos) con Π calculable y con error, entonces ese sector sí compila. OCC no niega cuerdas; niega el uso no proyectable del landscape como perilla.

13.2. LQG y discretización a escala de Planck

Un claim típico: “la geometría es discreta a escala de Planck y eso produce un espectro discreto de áreas”. El choque principal es ISAAC: la escala donde la estructura se propone es precisamente donde la medición local colapsa operacionalmente.

OCC no dice que la discretización sea falsa. Dice: ¿cómo se proyecta esa estructura a Ω_I de manera auditable?

Si existe Π (por ejemplo, correcciones efectivas a propagación, dispersión, o cosmología) con Δ_proj y observables identificables, entonces el claim se vuelve evaluable en forma efectiva. Si el claim permanece en la escala inaccesible sin proyección, NO-EVAL (J0+J1).

13.3. Modelos BSM altamente parametrizados (SUSY genérica, EFT sin priors)

Muchos marcos BSM sobreviven por maleabilidad: demasiados parámetros libres. OCC no “prohíbe” modelos con muchos parámetros; aplica IO y rigidez.

Si los parámetros que importan para el claim no son identificables con datos en Ω_I, el claim no es predicción; es subespecificación. El compilador puede emitir PASS (existe región viable) pero debe etiquetarlo como PASS-débil y cuantificar rigidez R. La defensa contra objeciones es clara: el marco no oculta la maleabilidad; la mide.

13.4. Bootstrap/amplitudes: por qué suelen compilar bien

Programas basados en unitaridad, analiticidad y positividad ya están alineados con varios candados C1–C6. En ellos, la maleabilidad suele estar más controlada porque el espacio permitido es geométricamente rígido (poliedros/espectraedros). OCC , en ese contexto, actúa como formalización y como auditoría: exige publicar certificados (dual/primal) y declarar Ω_I, sustracciones y errores. Esto convierte “bounds bonitos” en veredictos reproducibles.

14. Guía de escritura: cómo presentar un paper compatible con OCC

La cultura estándar permite claims sin especificar dominio, supuestos ni reproducibilidad. Un paper OCC -compatible debe ser más estricto por diseño. Esta sección define un formato que blinda contra críticas y facilita revisión por terceros.

14.1. Checklist obligatorio (sin el cual el paper es NO-EVAL)

(i) Dominio Ω_I declarado (ventana, resolución, kernels instrumentales, corte EFT si aplica).
(ii) Frontend declarado y justificación de hipótesis (analiticidad, etc.).
(iii) Diccionario operacional: cómo se mapean variables a observables.
(iv) Candados implementados: lista y versión.
(v) Datos consumidos: ID, hashes, pipeline de compresión, covarianzas.
(vi) Certificados: PASS con residuals; FAIL con dual cuando sea posible.
(vii) Rigidez R y/o d_eff reportados.
(viii) MRD adjunto o linkeado con instrucciones reproducibles.
(ix) Sensibilidad RFS (tolerancias, discretización, Ω_I).
(x) Declaración de loopholes y supuestos S.

Este checklist no es burocracia: es el mecanismo anti-maleabilidad.

14.2. Cómo se responde a un revisor crítico (protocolo, no retórica)

Si un revisor critica una pieza:

- Si critica un candado: identifique el supuesto de frontend y muestre derivación o cambie de frontend.
- Si critica una cifra: muestre sensibilidad y estabilidad (RFS).
- Si critica que “es restrictivo”: muestre que el candado deriva de consistencia física o que el claim cae en NO-EVAL sin proyección.
- Si critica que “depende de elección”: muestre que la elección está declarada y auditada; ofrezca corrida alternativa dentro del vecindario.

La respuesta correcta es siempre: aquí está el MRD, corra y compare. Eso mata la crítica retórica.

Apéndices matemáticos (derivaciones explícitas para blindaje)

Los apéndices existen para neutralizar la objeción “vago”. Cada derivación es un puente: significado operacional → formulación verificable. No se pretende reemplazar la literatura; se pretende hacer explícita la cadena lógica que el compilador usa.

Apéndice A. ISAAC en detalle: límites operacionales derivados de SR+QM+GR

A.1. Medición como intercambio de información con portadores energéticos

Toda medición local en A requiere intercambio de información. En SR, información viaja a velocidad $\le c$. En QM, resolución espacial implica ancho de banda en momento: localizar más exige mayor p. En GR, mayor energía implica curvatura.

Por tanto, existe un punto donde medir más fino exige tanta energía que la sonda deja de ser “pasiva”: su backreaction domina. Ese punto es ISAAC.

A.2. Derivación por fotón (Rayleigh + Schwarzschild) con factores explícitos

Usamos $\lambda \approx \square$ como criterio de resolución. $E = \hbar\omega = 2\pi\hbar c/\lambda$. Tomando $\lambda=\square$, $E \approx 2\pi\hbar c/\square$. Radio gravitacional $r_s = 2GE/c^4$.

$r_s \ge \square \Rightarrow 2G(2\pi\hbar c/\square)/c^4 \ge \square \Rightarrow \square^2 \le 4\pi\, \hbar G/c^3$. Se obtiene $\square \ge \sqrt{4\pi}\, L_P$. El factor $\sqrt{4\pi}$ depende del criterio; el punto es que el umbral es $O(1)\times L_P$.

Esta versión muestra explícitamente que ISAAC no depende del “2”: depende de la estructura dimensional y de backreaction.

Si $\lambda = \square$:

$E = \hbar\omega = 2\pi \hbar c / \square$ $r_s = 2 G E / c^4 = 4\pi \hbar G / (c^3 \square)$

$r_s \ge \square \Rightarrow \square^2 \le 4\pi \hbar G / c^3 \Rightarrow \square \ge \sqrt{4\pi}\, \sqrt{\hbar G/c^3}$

A.3. Argumento por concentración de energía y curvatura (Einstein)

En GR, el tensor de Einstein satisface $G_{\mu\nu} = 8\pi G\, T_{\mu\nu}/c^4$. Si concentras energía E en volumen $\sim\square^3$, la densidad $\rho \sim E/\square^3$. La curvatura típica escala como $R \sim G\rho/c^2 \sim G\, E/(c^2\, \square^3)$.

La longitud de curvatura $L_{curv} \sim 1/\sqrt{R}$. Cuando $L_{curv} \sim \square$, la geometría en la región se vuelve fuertemente afectada por la sonda: la noción de “posición” que querías medir deja de ser estable. Sustituyendo $E \sim \hbar c/\square$ produce de nuevo $\square \sim \sqrt{\hbar G/c^3}$.

Este argumento muestra que ISAAC es también un límite de auto-consistencia geométrica: no se puede asumir un fondo fijo mientras se mide con energías que lo destruyen.

A.4. Versión informacional (Bekenstein y horizonte)

Un argumento complementario usa límites de información. La cota de Bekenstein sugiere $S \leq 2\pi\, k_B\, E\, R/(\hbar c)$. Para aumentar bits resolubles en región pequeña, debes aumentar E. Pero E aumenta r_s. Cuando $r_s \,\square\, R$, se forma horizonte: la información deja de ser accesible a A. Entonces la propia medición crea un sistema que oculta información.

Esto conecta ISAAC con termodinámica de horizontes: la medición extrema produce su propio límite informacional.

Apéndice B. Dispersión y positividad: derivación paso a paso

B.1. Estructura analítica (polos y cortes)

En un régimen con estados asintóticos y localidad efectiva, A(s,t) es analítica en s salvo polos/cortes físicos. Cortes empiezan en umbrales de producción; polos corresponden a estados ligados o intercambios. Esta estructura permite usar Cauchy para relacionar valores en región analítica con discontinuidades físicas.

B.2. Cauchy con sustracciones: derivación

Fijamos $t<0$ (IR-safe). Si A(s,t) crece como $|s|^N$, definimos $F(s)=A(s,t)/s^{N+1}$ para que decaiga. Cauchy: $F(s)=(1/2\pi i)\square\, F(z)/(z-s)\, dz$. El contorno se deforma alrededor del corte real y se obtiene integral de discontinuidad Im A. Multiplicando por s^{N+1} se obtiene dispersión con N sustracciones y polinomio P_N.

Los coeficientes de P_N son parámetros efectivos (contact terms) que deben entrar en IO y en rigidez: elegirlos a posteriori es maleabilidad.

Dispersión (esquema):

$$A(s,t) = P_N(s,t) + (s^{N+1}/\pi) \int_{s0}^{\infty} ds'\ \mathrm{Im}\, A(s'+i0,t) \,/\, (s'^{N+1}(s' - s)) + \text{crossing}$$

B.3. De óptica a positividad de derivadas

Unitaridad implica $\mathrm{Im}\, A(s,t) \geq 0$ en el corte físico (para procesos adecuados). Entonces derivadas de A en puntos analíticos se vuelven integrales de funciones no negativas.

Ejemplo: expandiendo $A(s,t)$ alrededor de $s=0$, los coeficientes de s^2, s^4, etc. pueden escribirse como momentos positivos del espectro. Por tanto, ciertas combinaciones deben ser positivas. En EFT, esos coeficientes son combinaciones lineales de Wilson coefficients.

Este mecanismo es la base de candados de positividad: no es “conjetura”, es integral de cantidad no negativa.

B.4. Pole subtraction y gravedad

Con intercambio sin masa, A contiene términos $\sim 1/t$. Forward limit $t \to 0$ diverge. OCC exige: (i) substraer intercambio conocido, o (ii) evaluar a $t<0$ fijo. El procedimiento se audita: error de sustracción entra como Δ_{proj}. Sin esto, la aplicación de positividad es inválida: el resultado debe ser NO-EVAL (RFS/PA), no FAIL.

Apéndice C. Canales CPTP, Choi y Kraus (detalle)

C.1. De dilatación de Stinespring a Kraus

Toda CP map admite una dilatación unitaria en un espacio ampliado (Stinespring): $\Phi(\rho)=\mathrm{Tr}_E[\ U\ (\rho\otimes|0\rangle\langle 0|)\ U^\dagger\ ]$. Expandiendo en una base del entorno se obtiene representación de Kraus: $\Phi(\rho)=\Sigma_k\ K_k\ \rho\ K_k^\dagger$. TP equivale a $\Sigma_k\ K_k^\dagger K_k=I$. Esta equivalencia conecta directamente el canal efectivo con un modelo unitario subyacente, aunque el entorno sea inaccesible.

Kraus:

$\Phi(\rho)=\Sigma_k K_k \rho K_k^\dagger$ TP $\Leftrightarrow$ $\Sigma_k K_k^\dagger K_k = I$

C.2. Choi: CP ⇔ $J\succeq 0$ (idea de prueba)

Idea: si Φ es CP, entonces $(\Phi\otimes I)(X)$ es positivo para $X=|\Omega\rangle\langle\Omega|$, por tanto $J\succeq 0$. Recíprocamente, si $J\succeq 0$, se puede diagonalizar $J=\Sigma_k\ |v_k\rangle\langle v_k|$ y reconstruir operadores de Kraus K_k a partir de v_k (reshaping). Entonces $\Phi(\rho)=\Sigma\ K_k\ \rho\ K_k^\dagger$ es CP. Esto muestra que CP es equivalente a positividad matricial en J, lo que es ideal para

verificación numérica auditada.

C.3. SK causalidad, PSD y discretización

Causalidad retarded en SK se verifica como estructura triangular en discretización temporal. Positividad de ruido se verifica por PSD (autovalores no negativos dentro de tolerancia). RFS exige verificar estabilidad bajo refinamiento: una matriz PSD en malla gruesa puede volverse indefinida en malla fina si el modelo es inconsistente.

Estos checks son candados porque varianzas negativas o respuesta acausal no tienen interpretación física.

Apéndice D. Lindblad, no-Markov y qué se exige realmente

D.1. Lindblad como caracterización de Markovianidad CP-divisible

Para semigrupos Markovianos, el generador debe ser de GKLS/Lindblad. Eso garantiza CP-divisibilidad. Pero la física real puede ser no-Markov: memoria, entornos estructurados, etc. OCC no impone Lindblad cuando el autor no declara Markovianidad. Exige CPTP global y causalidad, no CP-divisibilidad local. Esto evita falsos FAILs y blinda frente a críticas técnicas.

Lindblad:

dρ/dt = -i[H,ρ] + Σ_α (L_α ρ L_α† - 1/2 {L_α†L_α, ρ})

Apéndice E. Axiomas Osterwalder–Schrader (lista) y reflection positivity

E.1. Lista OS y significado

OS0 regularidad (distribuciones templadas), OS1 invariancia euclídea, OS2 simetría, OS3 reflection positivity, OS4 cluster, OS5 continuación a Wightman con espectro positivo. OS3 es el candado clave: garantiza norma positiva tras Wick rotation. Sin OS3 hay estados de norma negativa y no hay probabilidades.

Apéndice F. Unitaridad parcial y límites EFT (detalle)

F.1. Ondas parciales y cota |a_l|≤1

Para 2→2: A(s,cosθ)=16π Σ_l (2l+1) a_l(s) P_l(cosθ). S_l=1+2ia_l. Unitaridad implica Im a_l ≥ |a_l|^2, y por tanto |a_l|≤1. EFTs con operadores de alta dimensión hacen crecer a_l(s); el candado fija límites sobre coeficientes dentro de Ω_I.

A=16π Σ (2l+1)a_l P_l S_l=1+2ia_l

Im a_l ≥ |a_l|^2 ⇒ |a_l|≤1

Apéndice G. Certificados duales: Farkas y dualidad convexa

G.1. Farkas en LP (testigo de infeasibilidad)

Para A x ≤ b, o existe x, o existe y≥0 con y^T A=0 y y^T b <0. Ese y prueba que no existe solución. En OCC , un FAIL con testigo dual es más fuerte que “no encontré solución”: es prueba matemática reproducible.

A x ≤ b infeasible ⇔ ∃ y ≥0: y^T A = 0, y^T b < 0

Apéndice H. Diccionarios operacionales: ejemplo 4F (CUI/HUI) y Avatar

H.1. Diccionario como parte de evaluabilidad

Una variable solo tiene contenido físico si existe protocolo en A que la mida o la infiera. Por eso diccionarios operacionales son parte de OCC : sin diccionario, no hay IO y suele no haber PA. El diccionario evita que el autor se esconda en gauge o en definiciones retroactivas.

H.2. Holonomías como invariantes medibles

En teorías gauge, los invariantes naturales son holonomías (Wilson loops) y curvaturas

integradas. Un diccionario 4F se formula usando:
- una conexión unificada (CUI) que restringe a subsectores,
- su holonomía (HUI) asociada a protocolos físicos.

Esto alinea con instrumentación real: interferometría mide fases/holonomías, no componentes gauge-variantes. Por eso claims de unificación que dependen de variables gauge-variantes sin protocolo son NO-EVAL.

H.3. Avatar como sonda operacional abstracta

Avatar es una abstracción de instrumentación: define acoplamientos permitidos, lecturas y resolución. No introduce entidad nueva; fija el protocolo que hace medible una cantidad. Esto conecta directamente con ISAAC: todo Avatar induce backreaction y está sujeto al cierre operacional.

Apéndice I. Esquemas de certificados (PA/IO/RFS) y campos mínimos

Para blindaje institucional, los certificados deben tener esquemas claros. A continuación se listan campos mínimos sugeridos (formato JSON/YAML), no por burocracia sino por interoperabilidad auditada.

Certificado PA (campos mínimos):
- version
- omega_I (ventana)
- projection_operator (descripción o referencia ejecutable)
- delta_proj (valor y método)
- assumptions
- hashes (inputs/datasets)
- signer (hash del compilador)

Certificado IO:
- version
- theta_definition
- observables_definition
- jacobian_rank / fisher_eigenvalues
- claim_dependency (qué combinaciones afectan el claim)
- identifiability_status
- hashes

Certificado RFS:
- version
- solver_config
- tolerances_range
- mesh_refinement_tests
- seeds
- reproducibility_hashes
- stability_summary (PASS estable / NO-EVAL)

Apéndice J. KMS y FDT en detalle (derivación completa, dominio de tiempo y frecuencia)

J.1. Derivación KMS desde cyclicidad de la traza (paso a paso)

Sea un sistema con Hamiltoniano H y estado térmico $\rho = e^{-\beta H}/Z$. Definimos operadores en el cuadro de Heisenberg: $A(t)=e^{iHt} A e^{-iHt}$.

Consideremos el correlador "mayor": $G^>(t) = \langle A(t) B(0)\rangle = \mathrm{Tr}(\rho A(t) B)$.

Escribimos explícitamente: $G^>(t) = (1/Z) \mathrm{Tr}(e^{-\beta H} e^{iHt} A e^{-iHt} B)$.

Ahora usamos que $e^{-\beta H}$ conmuta con $e^{\pm iHt}$ porque ambas son funciones de H: $e^{-\beta H} e^{iHt} = e^{iHt} e^{-\beta H}$.

Entonces: $G^>(t) = (1/Z) \mathrm{Tr}(e^{iHt} e^{-\beta H} A e^{-iHt} B)$.

Aplicamos cyclicidad de la traza: $\mathrm{Tr}(XY)=\mathrm{Tr}(YX)$. Tomando $X=e^{iHt} e^{-\beta H} A e^{-iHt}$ y $Y=B$, obtenemos: $G^>(t) = (1/Z) \mathrm{Tr}(B e^{iHt} e^{-\beta H} A e^{-iHt})$.

Reagrupamos usando de nuevo conmutatividad: $e^{iHt} e^{-\beta H} = e^{-\beta H} e^{iHt}$.

Entonces: $G^>(t) = (1/Z) \mathrm{Tr}(B e^{-\beta H} e^{iHt} A e^{-iHt}) = \mathrm{Tr}(\rho B A(t))$.

Pero aún falta el corrimiento imaginario. Insertamos identidad en forma $e^{\beta H} e^{-\beta H} = 1$ entre B y A(t): $\mathrm{Tr}(\rho B A(t)) = (1/Z) \mathrm{Tr}(e^{-\beta H} B e^{\beta H} e^{-\beta H} A(t))$.

Usando $e^{\beta H} A(t) e^{-\beta H} = A(t + i\beta)$ (propiedad estándar de evolución analítica), se llega a: $\mathrm{Tr}(\rho A(t) B) = \mathrm{Tr}(\rho B A(t + i\beta))$.

Eso es la condición KMS: $G^>(t) = G^<(t + i\beta)$, donde $G^<(t)=\langle B(0)A(t)\rangle$.

Derivación condensada:

$G^>(t)=\mathrm{Tr}(\rho A(t)B) = (1/Z)\mathrm{Tr}(e^{-\beta H} e^{iHt} A e^{-iHt} B)$

$= (1/Z)\mathrm{Tr}(B e^{-\beta H} e^{iHt} A e^{-iHt})$ (cyclicidad) $= (1/Z)\mathrm{Tr}(e^{-\beta H} B e^{iH(t+i\beta)} A e^{-iH(t+i\beta)})$ (corrimiento $i\beta$) $= \mathrm{Tr}(\rho B A(t+i\beta)) = G^<(t+i\beta)$

J.2. Representación espectral y relación en frecuencia

En frecuencia, definimos transformadas: $G^>(\omega)=\int dt\, e^{i\omega t} G^>(t)$, $G^<(\omega)=\int dt\, e^{i\omega t} G^<(t)$.

KMS implica: $G^>(\omega) = e^{\beta\omega} G^<(\omega)$.

Definiendo la función espectral $\rho(\omega)=G^>(\omega)-G^<(\omega)$, se obtiene: $G^>(\omega) = (1 + n_B(\omega)) \rho(\omega)$, $G^<(\omega) = n_B(\omega) \rho(\omega)$, con $n_B(\omega)=1/(e^{\beta\omega}-1)$ para bosones.

Esta forma muestra cómo equilibrio fija la relación entre emisión y absorción. En OCC , si se declara equilibrio pero la relación entre $G^>$ y $G^<$ no satisface KMS, el input térmico es incoherente.

KMS en frecuencia:

$G^>(\omega) = e^{\beta\omega} G^<(\omega)$

$\rho(\omega) = G^>(\omega) - G^<(\omega)$

Bosones: $G^>(\omega) = (1 + n_B(\omega)) \rho(\omega)$ $G^<(\omega) = n_B(\omega) \rho(\omega)$ $n_B(\omega)=1/(e^{\beta\omega}-1)$

J.3. Fluctuation–Dissipation (FDT) desde KMS (lineal)

El teorema de fluctuación-disipación relaciona el correlador simétrico con la parte disipativa de la respuesta.

Definimos: S(t)=1/2 ⟨{A(t),A(0)}⟩ (correlación simétrica), χ_R(t)=i θ(t) ⟨[A(t),A(0)]⟩ (respuesta retardada).

En frecuencia: S(ω) está relacionado con Im χ_R(ω). Usando KMS y álgebra de correladores, se obtiene: S(ω) = coth(βω/2) Im χ_R(ω).

Esta relación es un candado C9 cuando se declara equilibrio y linealidad. Si se viola, o no hay equilibrio, o no hay linealidad, o el modelo de coarse-graining es inconsistente.

OCC exige declarar cuál.

FDT (forma estándar):

S(ω) = 1/2 (G^>(ω)+G^<(ω)) Im χ_R(ω) = 1/2 (G^>(ω)-G^<(ω)) = 1/2 ρ(ω)

⇒ S(ω) = coth(βω/2) Im χ_R(ω)

Apéndice K. CPT: hipótesis, cadena lógica y diagnóstico de violaciones

K.1. CPT como teorema condicional

El teorema CPT no afirma que el universo “debe” respetar CPT por decreto. Afirma que, si se cumplen ciertas hipótesis estructurales, entonces CPT es consecuencia.

Hipótesis típicas (en versiones Wightman/axiomáticas):
- invariancia de Lorentz (o Poincaré),
- localidad/microcausalidad,
- existencia de un vacío único e invariante,
- espectro de energía-momento en el cono futuro (positividad de energía),
- completitud del espacio de estados (Hilbert),
- campos como operadores/distribuciones con propiedades analíticas.

La consecuencia: la teoría es invariante bajo la transformación combinada CPT.

OCC usa esto como detector: si un candidato viola CPT dentro de Ω_I, entonces alguna hipótesis se rompe. El marco obliga a decir cuál y a mostrar que la ruptura no produce inconsistencias operacionales (causalidad, probabilidades, etc.).

K.2. Mapa de diagnóstico: 'violaste CPT' no es el final, es el inicio

Una violación reportada de CPT puede significar cosas distintas:

- Si el candidato rompe Lorentz pero mantiene causalidad operacional y ofrece frontend alternativo (por ejemplo, teorías con marco preferido), puede ser evaluable; CPT deja de ser candado pero la causalidad debe verificarse por otra vía.
- Si el candidato rompe localidad (por ejemplo, no-localidad fundamental), entonces C2/C5 pueden fallar; se requiere demostrar que no habilita señalización acausal en Ω_I. Si no se puede, FAIL.
- Si el candidato rompe positividad del espectro (energías negativas), suele producir

inestabilidades (vacío no estable) y probabilidades mal definidas; eso tiende a FAIL.

Por eso OCC no trata CPT como dogma. Lo trata como una alarma: señala qué columna del edificio se movió.

Apéndice L. Anomalías: Wess–Zumino, cuantización y ejemplo de matching

L.1. Condición de consistencia Wess–Zumino (idea)

Sea W[A] el funcional efectivo en presencia de un gauge background A. Bajo una transformación gauge parametrizada por α, la variación δ_α W mide la anomalía.

Wess–Zumino impone que las variaciones satisfacen una condición de consistencia (cociclo): δ_α δ_β W - δ_β δ_α W = δ_{[α,β]} W.

Esto no es estética: es coherencia algebraica de cómo actúa el álgebra gauge en el funcional cuántico. Cualquier realización IR de una anomalía debe respetar esta estructura. Si un candidato produce una “anomalía” que no cumple WZ, no es anomalía física; es inconsistencia matemática (FAIL).

Wess–Zumino:

δ_α δ_β W - δ_β δ_α W = δ_{[α,β]} W

L.2. Ejemplo: anomalía quiral (esquema) y matching

En teorías con fermiones quirales, aparece: ∂_μ J_5^μ = (g^2 / 16π^2) F_{μν} F̃^{μν} (coeficiente cuantizado).

Si la teoría UV tiene este término con coeficiente k, entonces el IR debe reproducir el mismo k:

- o mediante fermiones masivos que aún contribuyen vía Wess–Zumino terms,
- o mediante campos de Goldstone con términos topológicos,
- o mediante estados topológicos.

Si el IR “pierde” el k sin mecanismo, la proyección viola C10 y PA. OCC no permite esconder esto: obliga a que el diccionario UV→IR preserve el contenido topológico.

Apéndice M. IR-safe en presencia de polos sin masa: substraer sin mentir

M.1. Descomposición típica de amplitud con intercambio sin masa

En presencia de intercambio sin masa, una amplitud 2→2 suele tener forma: A(s,t) = A_pole(s,t) + A_reg(s,t), donde A_pole ~ g^2 / t (fotón) o ~ κ^2 s^2 / t (gravitón, esquemático) y A_reg es regular en t→0.

Los candados de dispersión/positividad deben aplicarse a A_reg, porque A_pole está fijado por simetría y no codifica UV desconocida. Pero la separación debe ser auditada:

- qué parte se considera “polo conocido”,
- qué regularización se usa,
- y cómo se propaga el error.

Si la separación depende de una convención oculta, se viola PA/RFS. Si se hace explícita, se blinda el uso de positividad incluso con gravedad.

Esquema:

$A(s,t) = A_{pole}(s,t) + A_{reg}(s,t)$

Aplicar candados a A_{reg} (IR-safe), no al total si $t \to 0$ diverge.

M.2. Criterio operacional: evitar regiones dominadas por IR

Incluso con sustracción, existe un régimen donde el IR domina y pequeñas incertidumbres en el polo contaminan el análisis. Por eso Ω_I debe excluir regiones donde $|A_{pole}| \gg |A_{reg}|$ si el objetivo es constreñir UV.

Esto no es "hacer trampa"; es declarar jurisdicción. Si un autor quiere constreñir UV usando datos dominados por IR, debe demostrar que el procedimiento es estable bajo variaciones del modelado IR (RFS).

Apéndice N. Compilación como diagrama conmutativo: consistencia inter-frontend

N.1. Principio: diferentes lenguajes, mismo juez

Una fuente profunda de objeciones es la ambigüedad de representación: "en mi gauge sí", "en tu representación no". OCC neutraliza esto exigiendo que los diccionarios entre frontends formen diagramas conmutativos (hasta error auditado).

Esquema: Entrada en frontend F1 → proyección → observables. Entrada en frontend F2 → proyección → mismos observables. Si ambos describen la misma física, el resultado debe concordar dentro de Δ_{proj}.

Cuando no conmutan, el marco obliga a identificar qué supuesto cambia. Eso evita que debates se vuelvan disputas de notación.

Diagrama ideal (conmutativo):

T_{F1} --Π1--> Θ_eff --O--> O_A | ^ | dict | v | T_{F2} --Π2--------+

Conmutatividad: $O \circ \Pi_1 \approx O \circ \Pi_2 \circ \mathrm{dict}$ (dentro de error)

Apéndice O. Ejemplo de MRD: estructura de archivos y campos esenciales

Un MRD no es "un zip con scripts". Es un contrato reproducible. Un ejemplo mínimo de estructura:

MRD/ README.md inputs/ omega_I.yaml model.yaml data_spec.yaml data/ dataset.csv (o referencia + hash) covariance.npy code/ run.py checkers/ outputs/ verdict.json certificates/ PA.json IO.json RFS.json logs/ run.log environment.txt

La idea es que un tercero pueda correr un solo comando y reproducir verdict.json, o al menos verificar certificados. Sin esta estructura, la discusión no es científica; es narrativa.

Campos críticos en verdict.json:

- verdict: PASS | FAIL | NO-EVAL
- omega_I: descripción de dominio
- frontend: nombre/versión
- constraints_version: hash
- data_version: hash
- residual_max
- rigidity_R (si aplica)
- certificates: rutas/hashes

Apéndice P. Prueba constructiva CP ⇔ Choi PSD y reconstrucción de Kraus

P.1. Vectorización y reshaping: el truco que hace posible el test

Para hacer explícita la equivalencia, usamos vectorización. Dado un operador X en H_{in}, definimos $|X\rangle\rangle$ como su vectorización en $H_{out} \otimes H_{in}$, apilando columnas (convención estándar).

Propiedades clave:
- $|AXB^T\rangle\rangle = (A \otimes B)\, |X\rangle\rangle$,
- $\mathrm{Tr}(X^\dagger Y) = \langle\langle X|Y\rangle\rangle$.

Estas identidades permiten traducir composición de mapas en productos matriciales. En particular, el estado maximálmente entrelazado no normalizado $|\Omega\rangle = \Sigma_i |i\rangle\otimes|i\rangle$ es precisamente la vectorización de la identidad: $|\Omega\rangle = |I\rangle\rangle$.

Con esta convención, el Choi $J(\Phi)$ se interpreta como la imagen de I bajo Φ en un espacio ampliado.

Vectorización (convención de columnas):

$|X\rangle\rangle = \Sigma_{ij} X_{ij} |i\rangle \otimes |j\rangle$

Identidades: $|A X B^T\rangle\rangle = (A \otimes B)\, |X\rangle\rangle$ $\mathrm{Tr}(X^\dagger Y) = \langle\langle X|Y\rangle\rangle$

$|\Omega\rangle = \Sigma_i |i\rangle\otimes|i\rangle = |I\rangle\rangle$

P.2. CP ⇒ J0 (dirección fácil)

Si Φ es completamente positivo, entonces para cualquier n, $(\Phi \otimes I_n)$ envía operadores positivos a positivos. En particular, para $n = \dim(H_{in})$, tomamos el operador positivo: $|\Omega\rangle\langle\Omega|$ 0.

Entonces: $J(\Phi) = (\Phi \otimes I)(|\Omega\rangle\langle\Omega|)$ 0.

Esta dirección es inmediata: CP implica positividad del Choi.

El contenido real del teorema está en la dirección inversa: J0 implica existencia de representación de Kraus, y por tanto CP.

P.3. J0 ⇒ Kraus ⇒ CP (dirección constructiva)

Supongamos $J(\Phi)$ 0. Entonces existe una descomposición espectral:

$J(\Phi) = \Sigma_k |v_k\rangle\langle v_k|$, donde $|v_k\rangle$ son vectores en $H_{out} \otimes H_{in}$ (absorbiendo autovalores en la norma).

Ahora definimos operadores K_k por reshaping: $|v_k\rangle \equiv |K_k\rangle\rangle$ (es decir, v_k es la vectorización de K_k).

Definimos un mapa: $\Phi(\rho) = \Sigma_k K_k \rho K_k^\dagger$.

Se puede demostrar que este Φ tiene como Choi precisamente $J(\Phi)$. Por construcción, Φ es CP porque es suma de mapas de la forma $\rho \to K\rho K^\dagger$, que son CP. Así, $J \succeq 0$ implica CP.

Finalmente, la condición TP se traduce en $\mathrm{Tr}_{out} J = I$, que se convierte en $\Sigma_k K_k^\dagger K_k = I$ (verificación directa usando propiedades de vectorización).

Esto prueba la equivalencia CP ⇔ Choi PSD y además entrega Kraus explícitos, lo que es ideal para MRDs.

Reconstrucción de Kraus desde Choi:

$J = \Sigma_k |v_k\rangle\langle v_k|, \ |v_k\rangle = |K_k\rangle\rangle$

Define $\Phi(\rho)=\Sigma_k K_k \rho K_k^\dagger \Rightarrow \Phi$ es CP.

TP: $\mathrm{Tr}_{out} J = I \Leftrightarrow \Sigma_k K_k^\dagger K_k = I$

Apéndice Q. Dispersión completa con crossing y sustracciones: pasos sin atajos

Q.1. Contorno, discontinuidad y fórmula de Sokhotski–Plemelj

Sea $A(s) = A(s,t)$ para t fijo. $A(s)$ es analítica excepto por un corte real $s \geq s_0$. Definimos la discontinuidad: $\mathrm{Disc}\, A(s) = A(s+i0) - A(s-i0) = 2i\, \mathrm{Im}\, A(s+i0)$.

Al deformar el contorno de Cauchy alrededor del corte, aparece una integral en términos de Disc A. La identidad de Sokhotski–Plemelj: $1/(x \pm i0) = \mathrm{PV}(1/x) \mp i\pi\, \delta(x)$, permite separar parte principal y parte imaginaria. Esto hace explícito por qué Im A entra de forma directa en relaciones de dispersión.

OCC usa esta estructura para convertir datos (que fijan Im A vía secciones eficaces) en bounds sobre la parte real, y por tanto sobre coeficientes EFT.

Discontinuidad:

$\mathrm{Disc}\, A(s) = A(s+i0) - A(s-i0) = 2i\, \mathrm{Im}\, A(s+i0)$

Sokhotski–Plemelj: $1/(x \pm i0) = \mathrm{PV}(1/x) \mp i\pi\, \delta(x)$

Q.2. Inclusión explícita de crossing

En procesos $2 \to 2$, $A(s,t,u)$ satisface $s+t+u=\Sigma m_i^2$. Crossing relaciona $A(s,t)$ con $A(u,t)$ u otros canales. En la relación de dispersión, esto aparece como contribuciones adicionales de cortes en otras regiones del plano s (por ejemplo, $s \leq u_0$).

La forma general incluye integrales sobre todos los cortes relevantes. En la práctica, muchas derivaciones de positividad usan amplitudes crossing-symmetric o combinaciones que eliminan contribuciones no deseadas. OCC exige declarar qué combinación se usa, porque distintas combinaciones corresponden a distintos candados (y pueden fallar en presencia de masas/cargas específicas).

Q.3. Sustracciones: por qué no son libertad arbitraria

Si el crecimiento de A(s) a infinito impide cerrar el contorno, se hacen sustracciones. Matemáticamente, se considera: A(s) - A(0) - s A'(0) - ... - s^N A^{(N)}(0)/N!, y se aplica dispersión a la función substraída.

Los términos substraídos forman el polinomio P_N(s). En EFT local, esos términos corresponden a contact terms, y por tanto a parámetros

efectivos. No son "ajustes libres" invisibles: deben entrar en Θ_eff, estar sujetos a IO, y su rango debe auditarse.

Por eso OCC insiste en: N_sub se declara, P_N se parametriza, y la sensibilidad del veredicto a N_sub se reporta (RFS).

Apéndice R. Acotamiento tipo Froissart y crecimiento permitido (qué se asume realmente)

R.1. Idea del bound: unitaridad + analiticidad + rango finito

El bound de Froissart–Martin (en teorías con gap de masa y supuestos de analiticidad adecuados) limita el crecimiento de la sección eficaz total: σ_tot(s) ≤ (π / m_π^2) log^2(s/s0) (esquema).

La derivación usa:
- expansión en ondas parciales,
- unitaridad |a_l|≤1,
- y el hecho de que a grandes l los parciales están suprimidos por rango finito efectivo (controlado por la masa más ligera intercambiable).

OCC no necesita el bound exacto para su núcleo; necesita la lección: no se puede asumir crecimiento arbitrario si se pretende usar dispersión. Si el candidato requiere crecimiento más rápido que lo permitido por sus propios supuestos, el frontend se autocontradice: FAIL dentro de ese frontend. Si el candidato abandona los supuestos (por ejemplo, no hay gap), se debe declarar y ajustar candados/Ω_I (NO-EVAL hasta completar).

Apéndice S. Rigidez y cambio de variables: por qué Jeffreys es una medida defensiva

S.1. Problema: el volumen depende de coordenadas

Si definimos R=μ(I)/μ(D_obs), el número depende de la medida μ. Bajo un cambio de variables θ → φ(θ), un elemento de volumen transforma como: dθ = |det(∂θ/∂φ)| dφ.

Si μ se toma uniforme en θ, puede no ser uniforme en φ. Un crítico podría decir: "tu R depende de coordenadas". OCC anticipa esto: obliga a declarar μ y, cuando se busca invariancia, sugiere medidas invariante como Jeffreys.

La medida de Jeffreys se define usando la información de Fisher: μ_J(θ) ∝ √det F(θ) dθ. Bajo reparametrización, F transforma como un tensor y √det F dθ es invariante. Esto hace que R sea menos dependiente de coordenadas, fortaleciendo el blindaje.

Cambio de variables:

$d\theta = |\det(\partial\theta/\partial\varphi)|\, d\varphi$

Jeffreys: $\mu_J(\theta) \propto \sqrt{\det F(\theta)}\, d\theta$

Invarianza: $F_\varphi = J^T F_\theta J \Rightarrow \sqrt{\det F_\varphi}\, d\varphi = \sqrt{\det F_\theta}\, d\theta$

S.2. Relación con IO: cuando F es singular

Si F tiene autovalores ~0, la medida de Jeffreys se vuelve degenerada: eso refleja exactamente falta de identificabilidad. En lugar de esconder el problema, lo expone: direcciones no identificables no contribuyen a volumen efectivo.

Por tanto, usar μ_J no solo es defensa contra reparametrización; es coherencia con IO. Esto muestra una propiedad deseable: IO y rigidez no son módulos separados; son parte de un mismo objeto geométrico inducido por datos.

Apéndice T. Estabilidad numérica, acondicionamiento y límites de error (RFS duro)

T.1. Por qué un PASS sin condicionamiento es un PASS frágil

Un solver puede reportar factibilidad con residual pequeño, pero si el problema está mal condicionado, pequeños cambios en datos o en tolerancia producen grandes cambios en θ^*. Eso significa que el veredicto no es robusto.

RFS exige reportar:
- residuals,
- y también sensibilidad/condicionamiento.

Un marco simple: Sea $f(\theta)$ el vector de constraints (igualdades/desigualdades transformadas). El Jacobiano $J_f = \partial f/\partial\theta$ en θ^* controla la sensibilidad local. Un número de condición $\kappa \sim ||J_f^{-1}||\ ||J_f||$ (cuando aplica) cuantifica amplificación de error.

Si κ es enorme, entonces la solución es numéricamente inestable: el resultado debe etiquetarse como frágil o NO-EVAL hasta mejorar parametrización o datos. Esto evita objeciones tipo “tu PASS es un artefacto numérico”.

Sensibilidad local:

$f(\theta)=0$ (constraints) $J_f = \partial f/\partial\theta$

Perturbación: $\delta f \approx J_f\, \delta\theta \Rightarrow \delta\theta \approx J_f^{-1}\, \delta f$

Condición (esquema): $\kappa(J_f) = ||J_f|| \cdot ||J_f^{-1}||$

T.2. Estabilidad bajo refinamiento de malla y tolerancia

Para problemas discretizados (integrales de dispersión, kernels SK, correladores), existe además error de discretización $O(h^p)$. RFS exige:
- estimar orden p (o al menos observar convergencia),
- y demostrar que el veredicto no cambia al disminuir h dentro de un rango.

Esto se reporta como: $V(h,\tau)$ estable para $h\in[h_{min},h_{max}]$, $\tau\in[\tau_{min},\tau_{max}]$. Si cambia, se diagnostica: o el modelo no converge, o el solver está mal configurado, o el claim está fuera de dominio. De nuevo: NO-EVAL antes que retórica.

Apéndice U. Glosario exhaustivo y definiciones canónicas

Este glosario es intencionalmente redundante: en defensa, la ambigüedad es la primera arma. Definiciones canónicas (resumen conceptual sin perder precisión):

A (SGO): sector operacional accesible con protocolos reproducibles. B (SIA): sector inaccesible por ausencia de tomografía reproducible desde A. ISAAC (J0): cierre operacional inducido por SR+QM+GR; define Ω_I. Ω_I: dominio ISAAC; ventana de energías/tiempos/resoluciones donde se exige consistencia. Frontend: lenguaje de entrada (S-matrix, EFT, SK, Euclídeo, etc.). Candado: condición necesaria de consistencia en un frontend. Violación ⇒ FAIL (dentro de Ω_I). Juez: condición necesaria de evaluabilidad. Violación ⇒ NO-EVAL (salvo inconsistencia adicional). PA (J1): existencia de proyección Π con error Δ_{proj}. IO (J2): identificabilidad de parámetros efectivos relevantes. RFS (J3): estabilidad y reproducibilidad con recursos finitos. MRD: demo mínimo reproducible (inputs, datos, código, certificados). MMRD: paquete máximo reproducible (incluye derivaciones, tests, duales, sensibilidad). Rigidez R: fracción de volumen viable respecto al compatible con datos, bajo medida declarada. d_{eff}: dimensión efectiva local del conjunto de soluciones. Reinyección UV: usar estructura fuera de Ω_I como perilla para ajustar observables sin proyección. IR-safe: observable o combinación donde singularidades IR están substraídas o controladas. CPTP: completamente positivo y preservador de traza. KMS: condición exacta de equilibrio térmico. FDT: relación entre fluctuación y disipación derivada de KMS.

El glosario se mantiene deliberadamente dentro de la semántica operacional: no define “lo que existe”; define “lo que puede medirse y auditarse”.

Apéndice V. CUI/HUI con más rigor: conexiones, curvatura, holonomía y observables

V.1. Conexión como 1-forma en un fibrado principal (gauge + gravedad)

Para evitar ambigüedad, fijamos el lenguaje geométrico estándar.

Sea P un fibrado principal con grupo de estructura G sobre una variedad base M (espacio-tiempo efectivo en A). Una conexión se representa por una 1-forma de conexión A en P con valores en el álgebra de Lie de G (Lie(G)).

En una sección local, se puede representar por un potencial $A = A_\mu dx^\mu$ en M. Bajo transformación gauge $g(x) \in G$: $A \to A' = g A g^{-1} + g\, dg^{-1}$.

En gravedad formulada como teoría de conexiones (por ejemplo, usando tetradas y conexión espín), aparece una conexión de Lorentz ω_μ^{ab}. En un esquema unificado, la idea de CUI (Covariant Unified Index) es considerar una conexión total que incluye:

- parte gravitatoria (ω),
- parte gauge interna (A_{int}),
- y, si se desea, otras conexiones efectivas.

El punto OCC : no se puede llamar “unificación” a una suma formal si no existe diccionario operacional que indique qué invariantes de esta conexión corresponden a observables medibles en A.

Conexión gauge:

$A = A_\mu dx^\mu \in \Omega^1(M, \mathrm{Lie}(G))$

Transformación: $A' = g A g^{-1} + g\, dg^{-1}$

Curvatura: $F = dA + A \wedge A$

V.2. Curvatura y ecuaciones de estructura: qué es realmente medible

La curvatura (field strength) es: $F = dA + A\wedge A$. En componentes: $F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu + [A_\mu, A_\nu]$.

En gravedad, la curvatura de la conexión espín es $R^{ab} = d\omega^{ab} + \omega^{a}{}_c \wedge \omega^{cb}$.

Observación operacional:

componentes de A_μ o de ω_μ^{ab} dependen de gauge y de elección de marco. Lo que se mide en interferometría, precesión o AB es efecto de curvatura integrada o de holonomía.

Por eso, en OCC , cualquier claim que dependa de componentes no invariantes sin protocolo de fijación es NO-EVAL por falta de diccionario (IO/PA).

Curvatura en componentes:

$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu + [A_\mu, A_\nu]$

Gravedad (conexión espín): $R^{ab} = d\omega^{ab} + \omega^{a}{}_c \wedge \omega^{cb}$

V.3. Holonomía (HUI): definición y propiedades bajo gauge

Sea γ una curva cerrada en M. La holonomía asociada a la conexión A es el elemento de grupo:

$U[\gamma] = P \exp(\oint_\gamma A)$,

donde P denota ordenamiento de camino. Bajo transformación gauge: $U[\gamma] \to g(x_0)\, U[\gamma]\, g^{-1}(x_0)$, donde x_0 es el punto base.

Por tanto, invariantes físicos típicos son clases de conjugación, por ejemplo: $W[\gamma] = \mathrm{Tr}_R\, U[\gamma]$ (Wilson loop en representación R).

Esto es la base de HUI: no un número arbitrario, sino el conjunto de holonomías que son invariantes operacionales (por trazas, espectros, etc.).

Holonomía:

$U[\gamma] = P \exp(\oint_\gamma A)$

Gauge: $U[\gamma] \to g(x_0)\, U[\gamma]\, g^{-1}(x_0)$

Invariante (Wilson): $W_R[\gamma] = \mathrm{Tr}_R\, U[\gamma]$

V.4. Límite de área pequeña: relación holonomía–curvatura (Stokes no abeliano)

Para lazos pequeños, $U[\gamma]$ está controlado por la curvatura. En el caso abeliano: $U[\gamma] = \exp(\oint A) = \exp(\int_\Sigma F)$ (por Stokes).

En el caso no abeliano, existe una versión con ordenamiento superficial:

$U[\partial\Sigma] = P_\Sigma \exp(\int_\Sigma F + \ldots)$, donde los puntos suspensivos indican correcciones por no conmutatividad.

La lectura OCC : la holonomía es el objeto "integrado" que conecta la conexión local con el observable de fase. Un marco unificado que proponga CUI debe mostrar cómo sus holonomías reproducen: - fases electromagnéticas (AB), - rotaciones gravitatorias (transporte paralelo), - y, en general, cualquier observable de fase conocido, dentro de Ω_I y con error auditado.

V.5. Avatar como representación: cómo una sonda 'lee' holonomía

Un Avatar operacional se puede modelar como un sistema cuántico que transforma en una representación R de G y que acopla mínimamente a la conexión. El efecto de transportar el Avatar a lo largo de γ es multiplicar su estado interno por $U_R[\gamma]$.

La fase o rotación observable depende de invariantes de $U_R[\gamma]$ (traza, eigenvalores, etc.). Esto da un diccionario claro:
- variable teórica: conexión/holonomía,
- protocolo: transporte de una sonda con representación conocida,
- observable: fase/interferencia/precesión.

Este diccionario hace compilable a la teoría: las predicciones se expresan en invariantes medibles, y la resolución del Avatar está sujeta a ISAAC (no se puede exigir lazos sub-ISAAC).

Apéndice W. Cierre operacional aplicado a cosmología: degeneraciones y proyección efectiva

W.1. Por qué cosmología es terreno fértil para maleabilidad

La cosmología combina dos ingredientes peligrosos:
- observables indirectos (inferencia estadística, kernels instrumentales complejos),
- y degeneraciones físicas (múltiples microfísicas producen el mismo H(z) o el mismo espectro).

Por eso es fácil "explicar" el universo con perillas: sectores oscuros, energías del vacío efectivas, modificaciones de gravedad. OCC no prohíbe estos sectores; exige PA e IO: cualquier nuevo sector debe proyectar a parámetros efectivos identificables por observables cosmológicos en Ω_I (por ejemplo, ecuaciones de estado efectivas, funciones μ(k,a), Σ(k,a) en parametrizaciones de MG) y debe reportar rigidez.

Si dos ontologías producen los mismos observables, OCC no decide cuál "existe". Declara equivalencia operacional y exige nuevos observables que rompan degeneración (por ejemplo, correlaciones cruzadas, lentes, redshift-space distortions).

W.2. Traducción a lenguaje compilable: de micro a fluido efectivo

El ejemplo canónico de PA en cosmología es la proyección a un fluido efectivo: $T^{\mu\nu}_{eff} = (\rho_{eff} + p_{eff}) u^\mu u^\nu + p_{eff} g^{\mu\nu} + \Pi^{\mu\nu}$.

El microdetalle (campo escalar, sector oscuro, modificación geométrica) se encapsula en funciones efectivas (ρ_{eff}, p_{eff}, anisotropic stress $\Pi^{\mu\nu}$) que sí son inferibles. La proyección es Π (en sentido PA), y su error Δ_proj corresponde al truncamiento del modelo efectivo y a la sensibilidad a microfísica no inferible.

Este lenguaje es exactamente lo que OCC exige: no ontología primero, sino proyección operacional primero.

Fluido efectivo:

T^{μν}_eff = (ρ_eff + p_eff) u^μ u^ν + p_eff g^{μν} + Π^{μν}

Parámetros efectivos (ejemplos): w(a)=p_eff/ρ_eff, c_s^2, σ (anisotropic stress), μ(k,a), Σ(k,a) en MG

Referencias sugeridas (no exhaustivas)

Este documento es auto-contenido en su lógica operacional, pero no pretende reemplazar la literatura técnica. Referencias de soporte típicas:

- Weinberg, The Quantum Theory of Fields (estructura, unitaridad, analiticidad).
- Streater & Wightman, PCT, Spin and Statistics (CPT y axiomas).
- Osterwalder & Schrader (axiomas OS).
- Kubo–Martin–Schwinger (KMS).
- Gorini–Kossakowski–Sudarshan y Lindblad (GKLS).
- ’t Hooft (anomaly matching).
- Literatura de positividad en amplitudes y bounds EFT.

El núcleo OCC no depende de una cita particular: depende de estructuras físicas ya establecidas (SR+QM+GR) y de teoremas matemáticos estándar (Cauchy, positividad matricial, dualidad convexa).

OCC — Operational Consistency Compiler (OCC) Addendum: From Demonstrator to Real Judge Data Anchors, Witnesses, and Robustness

Marco Antonio Isaac Alcuria

14 Feb 2026

0. Nota de alcance / Scope note

ES. Este documento es un *addendum* canónico del Documento A+ (defensa formal) y existe por una razón concreta: cerrar la brecha entre (i) "el compilador corre y está auditado" y (ii) "el compilador sirve como *juez real* en problemas físicos no-triviales". La arquitectura (jueces, candados, auditoría dura, no-reinyección UV, separación concepto–ecuación, etc.) ya estaba formalizada; lo que se refuerza aquí es la **tracción física**: anclajes a datos reales, testigos fuertes y robustez cuantificada.

EN. This document is a canonical addendum to Documento A+ (formal defense). Its purpose is to close the gap between (i) "the compiler runs and is auditable" and (ii) "the compiler behaves as a *real judge* in non-trivial physics." The architecture (judges, locks, hard audit, UV non-reinjection, concept–equation separation, etc.) already exists; what is strengthened here is **physical traction**: real-data anchors, strong witnesses, and quantified robustness.

1. Dónde se aplica OCC / Where OCC applies

ES. OCC no se propone como sustituto de creatividad teórica. Su lugar es el **último filtro operativo** antes de confrontar una propuesta con datos observacionales. En términos de flujo:

1) **Exploración creativa (pre-OCC):** libertad matemática para proponer hipótesis.
2) **Compilación OCC (OCC):** verificación de evaluabilidad, coherencia, dominio operacional Ω_I, proyección auditable Π, no-reinyección UV.
3) **Juez final: el universo (datos):** likelihoods, inferencia, exclusión/confirmación empírica.

El punto (2) existe para evitar dos fallas sistemáticas:

- **Irrefutabilidad por maleabilidad UV** (parámetros UV "libres" no proyectables).
- **Ambigüedad operacional** (no declarar Ω_I, o declarar un Ω_I incompatible con el acto de medir).

EN. OCC is not a replacement for theoretical creativity. Its role is the **last operational filter** before confronting a proposal with observational data. Conceptually:

1) **Creative exploration (pre-OCC):** mathematical freedom to hypothesize.
2) **OCC compilation (OCC):** evaluability/coherence checks, operational domain Ω_I, auditable projection Π, UV non-reinjection.

3) **Final judge: the universe (data):** likelihoods, inference, exclusion/confirmation.

Step (2) exists to prevent two systematic failures:

- **UV-driven unfalsifiability** (free, non-projectable UV knobs).
- **Operational ambiguity** (undeclared Ω_I, or an Ω_I incompatible with measurement).

2. Clasificación de candados: Consistencia vs. Evidencia / Lock classes: Consistency vs. Evidence

ES. Para blindar semánticamente el veredicto, OCC distingue explícitamente dos clases:

- **Clase C (Consistencia):** candados que provienen de estructura teórica (causalidad, analiticidad, unitariedad, positividad, CPTP, etc.) dentro del Ω_I declarado. Violarlos produce **FAIL** por inconsistencia interna o por violación de axiomas operacionales.
- **Clase E (Evidencia):** anclajes empíricos (intervalos experimentales, datasets, bounds observacionales). Se usan *después* de compilar para conectar con el juez final. Violarlos produce **FAIL(E)** como incompatibilidad con evidencia, no como inconsistencia lógica.

EN. To harden the semantics of the verdict, OCC explicitly distinguishes two classes:

- **Class C (Consistency):** locks derived from theoretical structure (causality, analyticity, unitarity, positivity, CPTP, etc.) inside declared Ω_I. Violations yield **FAIL** due to internal inconsistency or operational-axiom violation.
- **Class E (Evidence):** empirical anchors (experimental intervals, datasets, observational bounds). Used *after* compilation to connect with the final judge. Violations yield **FAIL(E)** as data incompatibility, not logical inconsistency.

3. Caso real 1: aQGC en VBS — positividad + anclaje CMS / Real case 1: aQGC in VBS — positivity + CMS anchor

3.1. Definición mínima del objeto evaluado / Minimal definition of the evaluated object

ES. En EFT de dimensión 8 para acoplos cuárticos anómalos (aQGC), una parametrización estándar es

$$\mathcal{L} = \mathcal{L}_{\mathrm{SM}} + \sum_i \frac{f_{T_i}}{\Lambda^4}\,\mathcal{O}_{T_i} + \cdots\,, \tag{1}$$

donde f_{T_i}/Λ^4 tiene unidades de TeV^{-4} en la convención experimental habitual.

EN. In a dimension-8 EFT for anomalous quartic gauge couplings (aQGC), a standard parametrization is

$$\mathcal{L} = \mathcal{L}_{\mathrm{SM}} + \sum_i \frac{f_{T_i}}{\Lambda^4}\,\mathcal{O}_{T_i} + \cdots\,, \tag{2}$$

where f_{T_i}/Λ^4 carries units TeV^{-4} in common experimental conventions.

3.2. Candado C: positividad (sign constraints) / Consistency lock: positivity

ES. Bajo supuestos estándar de analiticidad, unitariedad y causalidad (dispersión + forward limit), se obtienen desigualdades de positividad sobre combinaciones de coeficientes EFT. En particular, una familia representativa de bounds impone restricciones de signo para ciertos operadores aQGC:

$$\frac{f_{T0}}{\Lambda^4} \geq 0, \quad \frac{f_{T1}}{\Lambda^4} \geq 0, \quad \frac{f_{T2}}{\Lambda^4} \geq 0, \quad \frac{f_{T8}}{\Lambda^4} \geq 0, \quad \frac{f_{T9}}{\Lambda^4} \geq 0. \tag{3}$$

En OCC esto se codifica como el candado `AMP5_AQGC_POSITIVITY_TABLE4` (clase C) con testigo (witness) explícito:

$$\text{witness} = \{\text{op},\ f/\Lambda^4,\ \text{regla},\ \text{margen}\}.$$

EN. Under standard assumptions (analyticity, unitarity, causality; dispersion relations + forward limit), one obtains positivity inequalities on EFT coefficients. A representative family of bounds implies sign constraints for certain aQGC operators:

$$\frac{f_{T0}}{\Lambda^4} \geq 0, \quad \frac{f_{T1}}{\Lambda^4} \geq 0, \quad \frac{f_{T2}}{\Lambda^4} \geq 0, \quad \frac{f_{T8}}{\Lambda^4} \geq 0, \quad \frac{f_{T9}}{\Lambda^4} \geq 0. \tag{4}$$

In OCC this is encoded as `AMP5_AQGC_POSITIVITY_TABLE4` (Class C) with an explicit witness:

$$\text{witness} = \{\text{op},\ f/\Lambda^4,\ \text{rule},\ \text{margin}\}.$$

3.3. Candado E: anclaje CMS 95% CL / Evidence lock: CMS 95% CL anchor

ES. Para conectar la compilación con datos reales sin colapsar el significado de consistencia, OCC agrega un anclaje empírico separado: `AMP6_AQGC_DATA_95CL_ANCHOR` (clase E). Se usa un intervalo observado (95% CL) publicado para $ZZjj$ VBS en $\sqrt{s} = 13$ TeV con 35.9 fb^{-1}:

Operador	Límite inferior	Límite superior
f_{T0}/Λ^4	-0.46	0.44
f_{T1}/Λ^4	-0.61	0.61
f_{T2}/Λ^4	-1.20	1.20
f_{T8}/Λ^4	-0.84	0.84
f_{T9}/Λ^4	-1.80	1.80

EN. To connect compilation to real data without collapsing the meaning of consistency, OCC adds a separate empirical anchor: `AMP6_AQGC_DATA_95CL_ANCHOR` (Class E). We use published observed 95% CL intervals for $ZZjj$ VBS at $\sqrt{s} = 13$ TeV with 35.9 fb^{-1} (units: TeV^{-4}):

Operator	Lower bound	Upper bound
f_{T0}/Λ^4	-0.46	0.44
f_{T1}/Λ^4	-0.61	0.61
f_{T2}/Λ^4	-1.20	1.20
f_{T8}/Λ^4	-0.84	0.84
f_{T9}/Λ^4	-1.80	1.80

3.4. Testigo fuerte: conflicto de signo / Strong witness: sign conflict

ES. Un *testigo fuerte* es un objeto compacto que no depende de "perillas ocultas". Aquí, si un modelo propone (por ejemplo) $f_{T0}/\Lambda^4 < 0$, la compilación produce

$$\texttt{FAIL(AMP5)} \quad \text{con witness: } \{f_{T0}/\Lambda^4,\ \text{regla} \geq 0,\ \text{margen} < 0\}.$$

Este FAIL no depende de ajustes estadísticos, priors, ni de detalles UV no proyectables.

EN. A *strong witness* is a compact object that does not depend on hidden knobs. Here, if a model proposes (e.g.) $f_{T0}/\Lambda^4 < 0$, compilation returns

$$\texttt{FAIL(AMP5)} \quad \text{with witness: } \{f_{T0}/\Lambda^4,\ \text{rule} \geq 0,\ \text{margin} < 0\}.$$

This FAIL does not depend on statistical tuning, priors, or non-projectable UV details.

4. Caso real 2: Cosmología — $H(z)$ de cronómetros cósmicos / Real case 2: Cosmology — cosmic-chronometer $H(z)$

4.1. Dataset real empaquetado y auditado / Real dataset packaged and audited

ES. El módulo `mrd_cosmo_bridge` ahora incluye un dataset real de cronómetros cósmicos $H(z)$ (compilación pública), empaquetado como CSV con hash verificado. Cada fila contiene

$$(z_i,\ H_i,\ \sigma_i,\ \text{fuente}).$$

EN. Module `mrd_cosmo_bridge` now includes a real cosmic-chronometer $H(z)$ dataset (public compilation), packaged as a hash-verified CSV. Each row contains

$$(z_i,\ H_i,\ \sigma_i,\ \text{source}).$$

4.2. Proyección auditable: $H_{\text{th}}(z;\theta)$ / Auditable projection: $H_{\text{th}}(z;\theta)$

ES. En ΛCDM plano con parámetro de materia Ω_m y H_0:

$$H_{\text{th}}(z) = H_0\sqrt{\Omega_m(1+z)^3 + (1-\Omega_m)}. \tag{5}$$

El módulo define explícitamente el estadístico:

$$\chi^2(\theta) = \sum_i \frac{(H_{\text{th}}(z_i;\theta) - H_i)^2}{\sigma_i^2}. \tag{6}$$

y reporta el *witness* numérico $\{\chi^2,\ N,\ \chi^2/N\}$.

EN. In flat ΛCDM with matter parameter Ω_m and H_0:

$$H_{\text{th}}(z) = H_0\sqrt{\Omega_m(1+z)^3 + (1-\Omega_m)}. \tag{7}$$

The module explicitly defines:

$$\chi^2(\theta) = \sum_i \frac{(H_{\text{th}}(z_i;\theta) - H_i)^2}{\sigma_i^2}. \tag{8}$$

and reports the numerical witness $\{\chi^2,\ N,\ \chi^2/N\}$.

5. Caso real 3: IR en gravedad — PPN + velocidad GW / Real case 3: Gravity IR — PPN + GW speed

ES. El módulo `mrd_grav_ir` fue actualizado para usar cotas experimentales publicadas en:

$$|\gamma - 1| \lesssim 2.3 \times 10^{-5}, \qquad |\beta - 1| \lesssim 8 \times 10^{-5},$$

y un bound multi-messenger para velocidad de ondas gravitacionales (representado aquí por una cota efectiva):

$$|c_T/c - 1| \lesssim 10^{-15}\text{–}10^{-14}.$$

En OCC esto se traduce en candados cuantitativos IR con testigos directos.

EN. Module `mrd_grav_ir` was updated to use published experimental bounds such as:

$$|\gamma - 1| \lesssim 2.3 \times 10^{-5}, \qquad |\beta - 1| \lesssim 8 \times 10^{-5},$$

and a multi-messenger bound on the speed of gravitational waves (represented here as an effective limit):

$$|c_T/c - 1| \lesssim 10^{-15}\text{–}10^{-14}.$$

In OCC this becomes quantitative IR locks with direct witnesses.

6. Reproducibilidad: comandos mínimos / Reproducibility: minimal commands

ES. En el release canónico, los módulos se ejecutan de forma determinista desde sus carpetas:

```
python scripts/run_mrd_amp_pos.py inputs/mrd_amp_pos/pass_aqgc_FT0.yaml
python scripts/run_mrd_cosmo_bridge.py inputs/mrd_cosmo_bridge/pass.yaml
python scripts/run_mrd_grav_ir.py inputs/mrd_grav_ir/pass.yaml
```

EN. In the canonical release, modules run deterministically from their directories:

```
python scripts/run_mrd_amp_pos.py inputs/mrd_amp_pos/pass_aqgc_FT0.yaml
python scripts/run_mrd_cosmo_bridge.py inputs/mrd_cosmo_bridge/pass.yaml
python scripts/run_mrd_grav_ir.py inputs/mrd_grav_ir/pass.yaml
```

7. Conclusión operativa / Operational conclusion

ES. Con estos upgrades, OCC no es solo "compliance". Ya existen módulos donde: (i) hay candados C reconocidos en literatura, (ii) hay anclajes E empaquetados y auditados, (iii) el FAIL produce testigos compactos, y (iv) la robustez se puede cuantificar por barridos (sweeps) sobre entradas declaradas.

EN. With these upgrades, OCC is not merely "compliance". There are now modules where: (i) Class-C locks are literature-grounded, (ii) Class-E anchors are packaged and auditable, (iii) FAIL yields compact witnesses, and (iv) robustness can be quantified via sweeps over declared inputs.

Compilación de Consistencia Operacional (OCC)
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

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 $E/\Lambda \le \varepsilon_\Lambda$ 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 OCC

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: $I = F \cap D_{obs} \neq \emptyset$.
- 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 = ∅ 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→Ω_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).
OCC 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→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 → un lenguaje intermedio de constraints → 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 $_{\text{OCC}}$, 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 → 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:
$\Phi_{A\to A}(\rho_A) = Tr_B [U_{AB} (\rho_A \otimes \sigma_B) U^\dagger_{AB}]$
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 OCC , 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):

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

Nota sobre factores O(1). El valor exacto puede variar por detalles geométricos/convenciones (factores 2, 2π, etc.). OCC 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 • 2026-02-13 22:48 UTC

Este anexo integra y formaliza el Nivel J (Jueces) dentro del estándar $_{OCC}$. 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 ≠ 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 ↔ energía-momento ↔ curvatura ↔ causalidad.

J0 / ISAAC — Regla de compilación: cualquier dependencia UV que impacte Ω_I 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 ΩI (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_I para afectar predicciones dentro de Ω_I, solo es evaluable si existe una proyección finita Π hacia Ω_I 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_I.

Instanciaciones (dependientes de datos):

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

- Dominancia: si $\Delta_{proj} \gtrsim \sigma_{data}$ 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 ⇒ 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 OCC

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).

4. J4 — Nuclear Domain Guard (J4 / L4C* / L4E*)

Concept (invariant): a nuclear claim is evaluable only if the nuclear domain is explicitly declared and observationally anchored with reproducible provenance. J4 is not an optional add-on; it is the domain continuation of J0-J3.

4.1 Why J4 is unavoidable

Without domain declarations, nuclear claims become tuneable narratives: the same statement can be made compatible with mutually incompatible channels or detector regimes. J4 prevents this by forcing explicit energy windows, isotopes, reaction channels, detector context, and evidence anchors.

4.2 Concept-to-equation bridge

Invariant concept: evaluability in Ω_I and no hidden nuclear knob reinjection. Data-dependent equations: threshold checks, residual checks, and anchor tests.

```
Eq. (1): 0 <= E_min < E_max [MeV]
Eq. (2): z = |sigma_pred - sigma_obs| / sigma_obs_err
PASS(E) iff z <= z_max; FAIL(L4E5) iff z > z_max.
```

4.3 L4C* locks (consistency/evaluability)

L4C1. Declare domain.energy_range_mev.{min_mev,max_mev}; missing -> NO-EVAL.

L4C2. Declare isotopes[] and reaction_channel; missing -> NO-EVAL.

L4C3. Declare detectors[] and operational resolution context.

L4C4. Units and thresholds must be explicit and internally consistent.

L4C5. Channel and isotope mapping must be non-ambiguous in Ω_I.

L4C6. No hidden control knob may carry claim support in Ω_I.

L4C7. Finite, reproducible computation path is mandatory for judgment.

4.4 L4E* locks (evidence/provenance)

L4E1. Evidence anchor must include dataset_ref.

L4E2. Provenance locator is required: source_url or dataset_doi.

L4E3. sigma_obs and sigma_obs_err must be declared with units.

L4E4. sigma_pred must reference the same observable definition.

L4E5. Residual z-test is mandatory; violation -> FAIL(L4E5).

L4E6. Evidence timestamp/version and run trace must be reproducible.

L4E7. If anchors are incomplete/untraceable -> NO-EVAL(L4E*).

4.5 Integration with J0-J3 flow

Evaluation order remains J0 -> J1 -> J2 -> J3 -> J4. J4 certifies domain-specific evaluability after projection, identifiability, and finite-resource stability are already satisfied.

4.6 Runtime coupling (MRD and predictions)

Runtime assets: occ/judges/nuclear_guard.py, ILSC_MRD_suite_extensions/mrd_nuclear_guard/, examples/claim_specs/nuclear_*.yaml, and predictions/registry.yaml (P-0004).

```
CLI path:
occ judge examples/claim_specs/nuclear_pass.yaml --profile nuclear
occ verify --suite extensions --strict --timeout 60
```

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:
$E_I = \{ s \mid 0 \le s \le \Lambda_I^2 \}$, $\Omega_I = E_I \times T_I$
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.
- $I \equiv F \cap D_{obs}$: 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/periocidad 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 unitaridad 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 $_{\mathrm{OCC}}$
$_{\mathrm{OCC}}$ 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 ∩ D_obs ≠ ∅.

- 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 $UV \to \Omega_I$ 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 $I \equiv F \cap D_{obs}$. 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 $R \equiv \max_i (\Delta f_i / \sigma_i^{(exp)})$. PASS fuerte si $R \lesssim 1$.
- Discreción local. Con parametrización $x \in \mathbb{R}^N$ y restricciones activas en un extremo x^*, se define $d_{eff} \equiv N - \mathrm{rank}(J_{act}(x^*))$. Discreta (PASS fuerte) si $d_{eff} = 0$ 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 $c = (c1, c2)$. Anclaje observacional: $|c1| \leq 1$, $|c2| \leq 1$.
Candados: $c1 \geq 0$ y $c1 + c2 \leq 0.2$.
Compilación: $D_{obs} = [-1, 1]^2$, $F = \{(c1, c2): c1 \geq 0, c1 + c2 \leq 0.2\}$. Entonces $I = D_{obs}$

$\cap F \neq \emptyset \Rightarrow$ PASS. El volumen superviviente es pequeño ⇒ 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.

OCC — 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, identificabilidad, recursos/estabilidad)	NO-EVAL	No compilable hoy; no es falsación.
Inconsistencia física dentro de ΩI (violación de locks físicos aplicables)	FAIL	Contradicción con la física operacional.
Cumple certificaciones y locks aplicables	PASS	Compilable y consistente en ΩI.

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 ΩI; 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 ΩI; 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 ΔΠ(E) con método

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

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

PA2 — Projection-error bound ΔΠ(E) with method

- Required input: ΔΠ(E) over relevant E; method (EFT truncation, residual, bounds) and scale dependence.
- Procedure: Run/verify stated method; reproducible.
- Threshold: ΔΠ must be defined over the full E-range used by the claims.
- Output/diagnostic: NO-EVAL(PA2): ΔΠ 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 ΔΠ(E) vs σ_data en el rango de inferencia.
- Umbral: Si ΔΠ ≥ κ·σ_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 ΔΠ(E) vs σ_data over inference range.
- Threshold: If ΔΠ ≥ κ·σ_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 ΔΠ+σ_data; registrar transformación.
- Umbral: Invariancia 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 $\Delta\Pi+\sigma\_data$; 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(θ) declarado

- Entrada requerida: Lista finita θ_ΩI 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(θ) ejecutable o auditable.
- Salida/diagnóstico: NO-EVAL(IO1): mapa no declarado.
- Severidad al fallar: NO-EVAL

IO1 — Declared map O(θ)

- Required input: Finite θ_ΩI and relevant observables O; validity range; units/conventions.
- Procedure: Validate IO-CERT; verify O-θ link via computation/estimate.
- Threshold: An operationally auditable O(θ) 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 τ → 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 τ → 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 → 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 → 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. ×2, ×3) en grids relevantes.
- Procedimiento: Ejecutar sweep y verificar que el veredicto no cambia.
- Umbral: Si el veredicto cambia bajo refinamientos razonables → 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., ×2, ×3) for relevant grids.
- Procedure: Run sweep and verify verdict does not change.
- Threshold: If verdict flips under reasonable refinements → 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

Compilación de Consistencia Operacional (OCC)

Módulo Observabilidad & Instrumentación — límites ISAAC, resolución y mapeo dato→constraint (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo define el estándar de qué cuenta como evidencia dentro de OCC : cómo un dato experimental se convierte en constraint auditable, cómo se declaran resolución/errores/sesgos, y cómo opera el límite ISAAC (no reinyección UV, no inferencias más finas que el dominio operacional).

Por qué es clave. Sin este módulo, cualquier teoría puede “pasar” cambiando qué se considera observable, o usando inferencias que exceden la capacidad instrumental. Este módulo bloquea ese tipo de arbitrariedad.

Contenido

0. Alcance y dependencias

Depende de: Documento A ($_{OCC}$). Este módulo no depende de un dominio experimental particular; define reglas generales para cualquier D_obs.

Cubre: conversión dato→constraint, manejo de incertidumbres, identificación, límites instrumentales e ISAAC.

No cubre: el análisis estadístico detallado de un experimento específico; exige que se declare de forma auditable y que el resultado se exporte en un formato canónico (D_obs).

1. Motivación (por qué): por qué OCC necesita un estándar de evidencia

OCC pretende ser un juez universal. Pero un juez sólo es tan fuerte como la calidad de sus anclajes. Si se permite que cada propuesta defina ‘observable’ a su conveniencia, el sistema se vuelve maleable.

Este módulo previene tres trampas:

- Inferencia fuera de dominio. Extraer conclusiones más finas que la resolución o que el modelo instrumental permite.
- Selección de evidencia. Elegir sólo el subconjunto de datos que favorece la propuesta.
- Mapeo opaco. Convertir datos a constraints sin un pipeline reproducible.

2. Objeto operacional: pipeline dato→D_obs y dominio ISAAC

El objeto central es el pipeline auditable que mapea datos crudos a un conjunto de restricciones D_obs sobre parámetros/variables teóricas.

Se modela como una composición:

> Datos crudos → (calibración + reconstrucción) → observable(s) → inferencia (likelihood/posterior) → D_obs (bounds).

ISAAC opera como límite: D_obs no puede exigir resolución/precisión mayor que la que el pipeline soporta, ni puede introducir información que no provenga de los datos o de supuestos declarados.

3. Definiciones: observable, resolución, identificabilidad, sesgo, incertidumbre

3.1 Observable operacional

Cantidad definida por un procedimiento de medición y reconstrucción. No es un símbolo teórico; incluye qué instrumento, qué reconstrucción y qué ventana.

3.2 Resolución y ventana

Resolución: anchura efectiva del kernel de medición. Ventana: región de soporte donde la medición es válida (energía, escala, tiempo).

3.3 Identificabilidad

Un parámetro es identificable si cambios en él producen cambios distinguibles en los observables dentro de la resolución y con las incertidumbres.

3.4 Sesgo e incertidumbre

Sesgo: desplazamiento sistemático; incertidumbre: dispersión (estadística) + sistemáticos. El módulo exige separación y propagación.

4. Candados inevitables del módulo (O1–O16) con ‘por qué → qué’

O1. Definición operacional del observable

Por qué. Sin definición operacional, el observable es retórico.

Qué. Declarar instrumento, reconstrucción, selección y ventana.

O2. Ventana de validez obligatoria

Por qué. Usar datos fuera de ventana produce conclusiones falsas.

Qué. Toda restricción D_obs viene con su ventana (energía/escala/tiempo).

O3. Resolución explícita

Por qué. No se puede exigir estructura más fina que la resolución.

Qué. Publicar kernel de resolución o parametrización equivalente.

O4. Separación estadístico vs sistemático

Por qué. Mezclarlos oculta fragilidad.

Qué. Reportar ambos y su combinación; matriz de covarianza cuando aplique.

O5. Calibración y trazabilidad

Por qué. Sin trazabilidad, el pipeline no es reproducible.

Qué. Versiones de calibración, condiciones, y dependencias publicadas.

O6. Propagación de incertidumbre

Por qué. Bounds sin propagación no son constraints reales.

Qué. Método declarado (lineal, MC, bootstrap, etc.) y validación mínima.

O7. Identificabilidad / degeneracias

Por qué. Si el parámetro no es identificable, el ‘fit’ es perilla.

Qué. Reportar degeneracias, direcciones planas o sensibilidad (Fisher/PCA) en el espacio de parámetros.

O8. Priors y decisiones de análisis explícitos

Por qué. Priors implícitos cambian conclusiones.

Qué. Declarar priors, cortes, regularizaciones y su justificación.

O9. No look-elsewhere/hacking

Por qué. Buscar hasta encontrar señal produce falsos PASS.

Qué. Corregir por trials o declarar protocolo preregistrado; lista de observables fija.

O10. Exportación canónica a D_obs

Por qué. El juez necesita un formato estándar.

Qué. Exportar bounds (intervalos/likelihood) en plantilla auditable con metadatos.

O11. Reversibilidad parcial

Por qué. Si no se puede rastrear D_obs a datos/supuestos, hay opacidad.

Qué. D_obs debe permitir reconstruir al menos los supuestos y la ventana; no se admiten ‘números sueltos’.

O12. Límite ISAAC: no inferencia sub-resolución

Por qué. Inferir microestructura sin capacidad instrumental es reinyección UV.

Qué. Prohibir constraints que dependan de variaciones más finas que el kernel de resolución o que requieran extrapolaciones no justificadas.

O13. Robustez a modelos instrumentales

Por qué. Un único modelo de detector puede sesgar conclusiones.

Qué. Reportar sensibilidad a variantes razonables del modelo instrumental.

O14. Auditoría computacional

Por qué. Pipelines complejos fallan sin registro.

Qué. Hashes, versiones, seeds, contenedores; pruebas mínimas de reproducción.

O15. Consistencia cruzada

Por qué. Un anclaje aislado puede estar sesgado.

Qué. Siempre que sea posible, cruzar con un dataset/observable alternativo o declarar por qué no.

O16. Rigidez del anclaje

Por qué. D_obs demasiado laxo no informa; demasiado “fino” puede violar ISAAC.

Qué. Reportar rigidez/anchura efectiva del anclaje y su dependencia de resolución e incertidumbre.

5. Compilación: plantillas de anclaje y auditoría

El módulo define plantillas de exportación para que cualquier experimento produzca D_obs compatible con $_{OCC}$. El compilador verifica O1–O16 antes de consumir el anclaje.

- Formato bounds. Intervalos + covarianza + ventana + resolución.
- Formato likelihood. Likelihood tabulada/parametrizada + metadatos.
- Formato summary. Cuando no se puede publicar todo, se exige justificación y reversibilidad parcial.

Si un anclaje falla O12 (ISAAC), se rechaza (FAIL del anclaje), incluso si el número “parece” preciso.

6. Interfaz con otros módulos (Amplitudes, EFT, SK, GE, UD)

- Amplitudes/Positividad. Traduce datasets de secciones eficaces a D_obs-σ con ventana (s,t) y resolución experimental.
- EFT/Renormalización. Exporta c_i con covarianzas y error EFT separado de error experimental.
- SK. Define cómo se reportan kernels/ruidos medidos (Γ, espectros) con resolución temporal/frecuencial.
- Gravedad Efectiva. Estándar para publicar bounds PPN/GW/cosmo con supuestos explícitos.
- UD. Fija lista de anclajes y evita selección post-hoc (O9).

7. MRD-O (demo mínimo reproducible)

MRD-O demuestra el módulo en un caso mínimo (sin depender de un experimento real):

- Simulación. Generar datos sintéticos con un ‘instrumento’ simple (kernel gaussiano de resolución) + ruido.
- Pipeline. Reconstruir un observable y exportar D_obs con ventana, resolución y covarianza.
- Test ISAAC. Intentar inferir una estructura sub-resolución y mostrar que el módulo la rechaza (O12).
- Salida. PASS del anclaje si cumple O1–O16; FAIL si no.

8. PASS/FAIL, PASS fuerte y no-go

En este módulo, el veredicto principal es sobre anclajes (no sobre teorías):

- FAIL-Anclaje. El anclaje no es consumible por OCC (viola O1–O16, típicamente O12 u O9).
- PASS-Anclaje. El anclaje es auditable y compatible con ISAAC.
- PASS fuerte. El anclaje es rígido pero no sub-resolución: su poder discriminante proviene de datos, no de supuestos ocultos.

No-go típico: “constraints” que dependen de features más finas que la resolución instrumental o de extrapolaciones no justificadas.

9. Loopholes/artefactos frecuentes

- Publicar sólo un número final. Sin ventana/resolución/covarianza, el juez no puede usarlo.
- Asumir modelo teórico en la reconstrucción. Si la reconstrucción depende fuertemente del modelo que se quiere testear, hay circularidad; debe declararse.
- Ignorar correlaciones. Produce falsa rigidez.
- Look-elsewhere. Buscar señales múltiples sin corrección.

10. Definición de Hecho (DoD)

- DoD-1. Plantilla D_obs implementada con metadatos (O10–O11).
- DoD-2. Resolución + ventana + separación de errores (O2–O4).
- DoD-3. Propagación y auditoría computacional (O6, O14).
- DoD-4. Prueba explícita de ISAAC (O12) con un caso que se rechace.
- DoD-5. MRD-O reproducible.

Apéndice A: plantilla de entrada (YAML/JSON)

Esquema mínimo (no ejecutable):

```
anchor_Dobs: observable: name: ... definition_operational: ... instrument: ...
reconstruction: ... selection: ... validity_window: variable: E | t | k | z | ... range: [min,
max] resolution: kernel: ... parameters: ... uncertainties: statistical: ... systematic: ...
covariance: ... inference: method: likelihood | posterior | interval priors: ...
look_elsewhere: ... export: bounds: ... format_version: ... audit: hashes: ... versions: ...
seed: ...
```

Regla: sin bloque resolution, cualquier bound se considera inválido (O3).

Apéndice B: glosario

- D_obs. Conjunto de constraints observacionales consumibles por OCC .
- Kernel de resolución. Función que describe cómo se borra/mezcla el valor real en la medición.
- Identificabilidad. Capacidad de distinguir parámetros con datos dados.
- Look-elsewhere. Penalización por múltiples pruebas.
- ISAAC. Límite operacional: no inferencias más finas que la resolución o fuera de ventana.

Compilación de Consistencia Operacional (OCC)

Módulo UV→Ω_I — Proyección auditable y evaluación de teorías ‘fundamentales’ (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo permite que OCC evalúe propuestas cuya descripción natural está fuera de Ω_I (p.ej. teorías UV-completas, no locales, o con torres infinitas de estados) sin violar ISAAC. Define un procedimiento canónico para construir una proyección auditable hacia un modelo efectivo consumible en Ω_I con error controlado y matching explícito.

Por qué. Sin este módulo, el juez queda sesgado a EFTs. Con él, cualquier propuesta puede entrar, pero sólo si paga el precio: una proyección reproducible que no reinyecte microestructura como perilla.

Contenido

0. Alcance y dependencias

Depende de: Documento A (PCD/PCN, ISAAC), Módulo EFT/Renormalización (truncación), Amplitudes/Positividad (cuando el UV produce S-matrix) y 4F (para invariantes).

Cubre: cómo traer una propuesta UV (o no local) a Ω_I sin arbitrariedad, y cómo reportar error de proyección.

No cubre: la ‘verdad’ del UV; sólo decide si el UV entrega una proyección consumible que pueda ser juzgada por OCC .

1. Motivación (por qué): el problema UD6 y la evaluación de teorías UV

Muchas propuestas ‘fundamentales’ (strings, LQG, no-locales, etc.) viven naturalmente en un lenguaje donde: (i) hay infinitos modos/estados, (ii) los observables son asintóticos o globales, y (iii) no es obvio el límite efectivo en Ω_I.

Sin un procedimiento de proyección, aparecen dos abusos:

- Abuso A. “Mi teoría lo reproduce si escoges el sector correcto.” → perilla no auditada.
- Abuso B. “Los detalles UV no importan, sólo asume esta EFT.” → proyección no derivada.

Este módulo impone que la proyección Π_UV→ΩI sea parte del input: sin ella, NO-EVAL.

2. Concepto central: proyección Π_UV→ΩI y presupuesto de error

Se define una proyección como un mapa explícito:

> Π_UV→ΩI : (Teoría UV + sector/estado + reglas de truncación) → (Modelo efectivo en Ω_I + error Δ_Π).

Principio. No se permite que Δ_Π sea ‘desconocido’. Si no se puede acotar, el resultado no es consumible.

Δ_Π puede incluir: error por truncar una torre de modos, error por no-localidad residual, error por integración de grados de libertad, y error numérico.

3. Clases de propuestas y rutas de proyección

El módulo reconoce rutas típicas (no exclusivas):

- Ruta EFT. Integrar fuera DOF pesados → acción efectiva local con operadores finitos a orden N.
- Ruta S-matrix. Derivar amplitudes IR-safe en ventana Ω_I y traducir a bounds/observables.
- Ruta correladores. Derivar correladores gauge-invariantes y su límite efectivo.
- Ruta geométrica 4F. Derivar invariantes CUI/HUI efectivos (holonomías/curvaturas integradas).

En todas las rutas, la selección de sector/estado debe ser operacional (no una etiqueta libre).

4. Matching UV/IR y certificados operacionales

La proyección debe incluir matching: cómo los parámetros efectivos se fijan desde el UV.

- Matching paramétrico. Fórmulas o algoritmo que produce coeficientes EFT/observables desde datos UV.
- Matching de anomalías. Si hay simetrías globales, debe respetarse el módulo SAT (’t Hooft).
- Certificados. Cuando el problema es convexo (positividad), se puede emitir witness que excluya familias UV por incompatibilidad en Ω_I.

5. No-localidad: criterios operacionales de admisibilidad

OCC no declara 'FAIL automático' a toda no-localidad. Declara FAIL sólo si la no-localidad produce violaciones operacionales de causalidad/analiticidad dentro de Ω_I.

- NL-1. Definir el tipo de no-localidad (retardada, no conmutativa, integral kernel, etc.).
- NL-2. Proveer una cota de alcance/escala ℓ_{NL} y demostrar que ℓ_{NL} está fuera de la resolución relevante o que su efecto se encapsula en operadores finitos.
- NL-3. Verificar que observables en Ω_I satisfacen causalidad operacional (no señalización superluminal dentro de ventana) y positividad/analiticidad cuando aplique.

6. Candados inevitables del módulo (UV1–UV22) con ‘por qué → qué’

UV1. Declarar qué es ‘UV’ y qué es ‘Ω_I’ en la propuesta

Por qué. Sin esta distinción, la proyección no tiene sentido.

Qué. Definir escalas, modos y qué parte se integra fuera.

UV2. Definir Π_UV→ΩI explícita

Por qué. Sin Π, la teoría no es evaluable por OCC .

Qué. Dar algoritmo/receta reproducible para obtener modelo efectivo en Ω_I.

UV3. Presupuesto de error Δ_Π

Por qué. Una proyección sin error es una perilla oculta.

Qué. Acotar Δ_Π y propagarlo a todas las predicciones.

UV4. Selección de sector/estado operacional

Por qué. Elegir sector post-hoc es maleabilidad.

Qué. Definir cómo se prepara/identifica el sector en A o cómo se justifica teóricamente sin perilla.

UV5. Truncación finita (si aplica)

Por qué. Torres infinitas sin truncación no son computables.

Qué. Declarar truncación y demostrar estabilidad al aumentar orden.

UV6. No reinyección UV (ISAAC)

Por qué. Reinyectar detalles UV viola el límite operacional.

Qué. Prohibir uso de estructuras sub-ISAAC; encapsular en parámetros finitos con error.

UV7. Matching paramétrico

Por qué. Sin matching, los parámetros efectivos quedan libres.

Qué. Derivar coeficientes/inputs desde el UV o declarar NO-EVAL.

UV8. Matching de simetrías/anomalías

Por qué. El UV puede estar ‘bien’ pero el IR propuesto no.

Qué. Exigir consistencia con módulo SAT.

UV9. Compatibilidad con amplitudes/positividad (cuando aplique)

Por qué. Candidatos UV pueden violar analiticidad/causalidad.

Qué. Derivar amplitudes IR-safe o correladores y pasar bounds.

UV10. Compatibilidad con 4F invariantes

Por qué. Un UV que no produce invariantes operacionales no puede ser juzgado.

Qué. Mapear a CUI/HUI o a invariantes equivalentes.

UV11. Estabilidad a elección de esquema

Por qué. Cambiar esquema puede cambiar coeficientes.

Qué. Declarar esquema; demostrar estabilidad de observables a cambios razonables.

UV12. Dependencia de cutoffs

Por qué. Cutoffs sin control son perillas.

Qué. Mostrar que dependencias de cutoff se absorben en parámetros + Δ_Π.

UV13. No-localidad controlada (si aplica)

Por qué. No-localidad puede producir señales no físicas.

Qué. Aplicar criterios NL-1..NL-3 y reportar ℓ_{NL}.

UV14. Diagnóstico: qué parte falla si falla

Por qué. FAIL sin diagnóstico no guía ciencia.

Qué. Emitir diagnóstico: sector, truncación, matching, causalidad, etc.

UV15. Auditoría computacional

Por qué. Estas proyecciones suelen ser complejas.

Qué. Publicar código/versions/hashes/seeds y tests de convergencia (PCN).

UV16. Consistencia con cosmología operacional (si aplica)

Por qué. Mucho UV se prueba en cosmo.

Qué. Si se invocan predicciones cosmológicas, deben pasar módulo COS.

UV17. Consistencia con gravedad efectiva/local tests

Por qué. UV puede inducir correcciones locales.

Qué. Pasar módulo de gravedad efectiva (PPN/GW) en Ω_I.

UV18. Evitar ‘parámetros infinitos’ disfrazados

Por qué. Series/funciones arbitrarias reintroducen maleabilidad.

Qué. Sólo se permiten familias finitas o truncaciones con error.

UV19. Certificados cuando sea posible

Por qué. Fortalece exclusiones.

Qué. Emitir witness/dual certificates para exclusión en problemas convexos.

UV20. Taxonomía de veredictos (incluye NO-EVAL)

Por qué. Para separar ‘inconsistente’ de ‘no proyectado’.

Qué. PASS/FAIL/PASS fuerte/NO-EVAL según completitud de Π y Δ_Π.

UV21. Rigidez de la proyección

Por qué. Una proyección que admite demasiadas elecciones libres es narrativa.

Qué. Reportar número de decisiones libres y su impacto en observables.

UV22. Meta-estabilidad (no mover arbitrariedad al compilador)

Por qué. Si Π depende de tolerancias o Ω_I arbitrarios, falla meta-candado.

Qué. Aplicar PCD/PCN y reportar sensibilidad; degradar rigidez si cambia.

7. Interfaz con PCD/PCN, EFT, amplitudes, 4F y cosmología

- Con Documento A (PCD/PCN). Obliga a que Π y Δ_Π sean estables frente a variaciones razonables.
- Con EFT. Entrega truncación y error; EFT devuelve predicciones computables.
- Con amplitudes. Si hay S-matrix, se usan bounds de positividad como tests UV.
- Con 4F. Mapea objetos UV a invariantes CUI/HUI.
- Con cosmología. Si el UV pretende explicar DE/DM/cosmo, debe entregar pipeline COS completo.

8. Plantillas canónicas (YAML/JSON)

Plantilla mínima (no ejecutable):

UV_projection: domain_ISAAC: {Omega_I: ...} UV_model: description: ...
degrees_of_freedom: ... sectors_states: ... projection_Pi: route: EFT | S_matrix |
correlators | fourF algorithm: ... truncation: {order: ..., cutoff: ...} error_budget:
Delta_Pi: {truncation: ..., nonlocality: ..., numerical: ...} matching:
parameters_from_UV: ... anomaly_matching: ... tests: causality_analyticity: ...
positivity_bounds: ... local_tests: ... cosmology_tests: ... audit: code: ... versions: ...
hashes: ... seeds: ...

9. MRD-UV (demo mínimo reproducible)

MRD-UV recomendado (nivel 1): una teoría con torre truncable (ej. un modelo con estados pesados discretos) proyectada a una EFT finita.

- Entrada. Espectro UV + regla de integración de estados pesados.
- Proyección. Construir EFT a orden N y calcular Δ_Π por truncación.
- Test. Aplicar positividad/causalidad en Ω_I y mostrar estabilidad al aumentar N.
- Salida. PASS/FAIL/NO-EVAL con diagnóstico y auditoría.

10. PASS/FAIL, PASS fuerte y NO-EVAL

- PASS. Existe Π auditable con Δ_Π acotado; matching explícito; predicciones en Ω_I consumibles por módulos (EFT/4F/amplitudes/cosmo).
- PASS fuerte. Proyección rígida: pocas decisiones libres; estabilidad al aumentar truncación y variar esquema; certificados cuando aplique.
- FAIL. La proyección viola ISAAC, requiere perillas infinitas, o produce violaciones operacionales de causalidad/positividad en Ω_I.
- NO-EVAL. Falta Π o Δ_Π; sector/estado no operacional; matching no especificado.

11. Loopholes/artefactos frecuentes

- ‘Sector correcto’ post-hoc. Prohibido por UV4.
- ‘Δ_Π desconocido’. NO-EVAL automático.
- Truncación sin estabilidad. Si cambia veredicto al aumentar N, degradar rigidez o NO-EVAL.
- No-localidad sin cota. FAIL/NO-EVAL según impacto en Ω_I.

12. Definición de Hecho (DoD)

- DoD-1. $\Pi\_UV \rightarrow \Omega I$ especificada (algoritmo/receta).
- DoD-2. Δ_Π acotado y propagado a predicciones.
- DoD-3. Sector/estado seleccionado operacionalmente.
- DoD-4. Matching paramétrico y de anomalías cuando aplique.
- DoD-5. Verificación de causalidad/positividad/invariantes en Ω_I.
- DoD-6. Auditoría PCN (convergencia, seeds, hashes).
- DoD-7. MRD-UV reproducible.

Apéndice A: glosario

- $\Pi\_UV \rightarrow \Omega I$. Proyección explícita desde la teoría UV al régimen operacional Ω_I.
- Δ_Π. Presupuesto de error de la proyección (truncación/no-localidad/numerical).
- Matching. Procedimiento que fija parámetros efectivos desde el UV.
- No-localidad controlada. No-localidad cuya escala/efecto está acotado y no viola causalidad operacional en Ω_I.
- NO-EVAL. Estado cuando falta proyección auditable; no es ‘rescate’, es requisito.

Compilación de Consistencia Operacional (OCC)

Módulo 4F — Diccionario CUI/HUI: conexión, holonomía e invariantes operacionales (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo define el diccionario operacional que permite hablar de “unificación 4F” sin introducir perillas: cómo representar en un lenguaje único (i) gauge internos, (ii) gravedad como conexión geométrica, y (iii) materia/estados como secciones y acoplos, de modo que los observables admitidos sean funciones de invariantes gauge-difeomorfismo y de holonomías (cuando aplique).

Prioridad. Primero el por qué: sin un diccionario, “unificación” es narrativa. Luego el qué: definiciones, plantillas y candados.

Contenido

0. Alcance y dependencias

Depende de: Documento A (metodología universal). Se integra con Observabilidad/ISAAC (para definir qué es medible), EFT/Renormalización (para truncaciones) y con Simetrías/Anomalías (para consistencia global).

Cubre: el diccionario formal-operacional para traducir propuestas heterogéneas a un mismo lenguaje 4F consumible por OCC .

No cubre: la dinámica específica (eso pertenece a UD y a módulos de ejemplos). Aquí sólo se fija el “alfabeto” y la regla de qué cuenta como observable.

1. Motivación (por qué): por qué un diccionario 4F es inevitable

La palabra “unificación” suele fallar por dos razones: (i) mezcla objetos dependientes de gauge con afirmaciones físicas, y (ii) compara estructuras de naturaleza distinta (conexiones internas vs métrica/afinidad) sin un puente operacional.

Para que OCC evalúe una propuesta 4F, necesita una regla única: todo input debe compilarse a invariantes y toda salida debe ser un observable operacional (medible en Ω_I).

Este módulo impide los hacks clásicos: elegir gauge para ‘hacer coincidir’ fuerzas, inventar observables no invariantes o usar objetos definidos sólo en B como perillas.

2. Principio de observabilidad 4F: invariantes como única salida

Principio 4F-O. Un objeto es consumible como predicción o constraint sólo si es invariante bajo:

- Gauge interno. Transformaciones del grupo interno declarado (o su álgebra efectiva).
- Difeomorfismos. Reparametrizaciones del espaciotiempo (coordenadas).
- Redundancias locales. Re-definiciones equivalentes (EoM, integración por partes) cuando se use EFT.

Ejemplos de invariantes admitidos: espectros (autovalores) de operadores geométricos definidos covariantemente; observables de dispersión IR-safe; correladores gauge-invariantes; holonomías (trazas de Wilson loops) y cantidades construidas a partir de curvaturas/fluxes integrados.

Ejemplos prohibidos: componentes de campos A_μ^a en un gauge específico, potenciales no gauge-invariantes, o ‘parámetros de unificación’ definidos por elección de coordenadas.

3. Diccionario CUI (Conexión Unificada Invariante): objetos mínimos

CUI define el conjunto mínimo de objetos que deben declararse para describir una propuesta 4F en forma compilable.

```
CUI = { □, 𝔽, □, □, □, Ω_I }
```

- □ (conexión efectiva). Objeto que unifica conexiones internas y geométricas en un formalismo común (p.ej. suma directa/bundle extendido).
- 𝔽 (curvatura/field strength). Curvatura asociada a □. En gravedad, incluye curvaturas geométricas (Riemann/Cartan) según la clase declarada.
- □ (derivada covariante). Operador que define transporte paralelo y acoplos a materia.
- □ (estructura de simetría). Grupo/álgebra efectiva, posibles extensiones/rompimientos, y reglas de transformación.
- □ (corrientes/fuentes). Corrientes gauge-invariantes y tensores de energía-momento (o sus equivalentes) que acoplan a □/𝔽.
- Ω_I (ISAAC). Dominio operacional de validez; prohíbe reinyección UV y fija qué operaciones son permitidas.

Nota. CUI no exige una forma única de □; exige que cualquier elección sea traducible a invariantes y que la redundancia gauge/difeo esté explícita.

4. Diccionario HUI (Holonomía Unificada Invariante): observables de transporte

HUI define la clase de observables basados en transporte paralelo. Es el puente natural entre gauge interno y geometría.

```
HUI: W(C) = Tr P exp(∮_C □) , con C curva/ciclo dentro de Ω_I.
```

Por qué. La holonomía es gauge-invariante (vía traza/representación) y captura información global/topológica que no depende de componentes.

Qué. El módulo admite observables derivados de W(C): espectros de holonomía, invariantes de la clase de conjugación, y combinaciones que correspondan a procedimientos de medición (interferometría, fases de Berry generalizadas, etc.).

Regla ISAAC. El contorno C y la resolución del transporte deben estar dentro de Ω_I. No se admiten “loops” sub-resolución o definidos fuera de ventana.

5. Separación A/B en lenguaje 4F: qué puede y qué no puede entrar

Sector A (SGO): todo lo que puede expresarse como invariante y medirse (o inferirse) con un pipeline consumible en Ω_I.

Sector B (SIA): microestructura no accesible en Ω_I. En lenguaje 4F, B puede influir sólo mediante:

- Renormalización efectiva. Parámetros finitos (contraterminos/coeficientes EFT) con error y matching auditables.
- Kernels SK. Ruido/fricción efectivos que cumplen CPTP/positividad (si se modela como entorno).
- Condiciones iniciales/estado. Sólo si están definidos operacionalmente (no como perilla funcional arbitraria).

Prohibición. No se permite introducir en A funciones arbitrarias de curvatura/energía “porque vienen de B”. Eso viola ISAAC y el meta-candado de estabilidad (Documento A, PCD/PCN).

6. Candados inevitables del módulo (4F1–4F18) con ‘por qué → qué’

4F1. Declaración de simetrías □ y representaciones

Por qué. Sin simetrías, el diccionario no fija qué transformaciones son redundancia y qué es física.

Qué. Declarar □ (grupo/álgebra efectiva), representaciones relevantes y reglas de transformación.

4F2. Observables sólo invariantes

Por qué. Permitir objetos dependientes de gauge/difeo reintroduce perillas.

Qué. Toda predicción/constraint debe ser invariante (Principio 4F-O).

4F3. Conexión □ explícita y su clase

Por qué. Sin □ no hay lenguaje unificado ni transporte.

Qué. Definir □ y su clase (bundle extendido, suma directa, Cartan, etc.) y cómo incluye gravedad y gauge internos.

4F4. Curvatura 𝔽 y normalizaciones

Por qué. Normalizaciones ocultas pueden simular ‘unificación’.

Qué. Definir 𝔽 = d□ + □∧□ (o equivalente) y fijar convenciones/normalizaciones auditables.

4F5. Derivada covariante □ y acoplo a materia

Por qué. Sin □, la materia no entra de forma consistente.

Qué. Definir □ y cómo actúa sobre campos; declarar acoplos mínimos/no-mínimos permitidos.

4F6. Invariantes geométricos y gauge permitidos

Por qué. Un diccionario sin ‘lista de invariantes’ deja huecos para hacks.

Qué. Declarar el conjunto de invariantes admitidos (curvaturas integradas, holonomías, espectros, correladores gauge-invariantes, etc.).

4F7. Holonomía HUI y clase de contornos

Por qué. W(C) puede volverse arbitrario si C es libre o sub-resolución.

Qué. Definir la clase de contornos C permitidos por Ω_I y por la instrumentación; declarar cómo se mide/infiere.

4F8. Frontera A/B en parámetros

Por qué. Mover microestructura a parámetros funcionales destruye falsabilidad.

Qué. B sólo entra vía parámetros finitos + error + matching (EFT) o kernels SK; prohibidas funciones arbitrarias.

4F9. Truncación y error (si EFT)

Por qué. Sin truncación el diccionario no es computable.

Qué. Si se usa EFT, declarar truncación y error EFT (módulo EFT).

4F10. Consistencia con Observabilidad/ISAAC

Por qué. Un observable 4F sin pipeline instrumental es retórico.

Qué. Todo observable debe mapear a un D_obs consumible (ventana, resolución, covarianza).

4F11. Independencia de gauge/coords verificable

Por qué. Decir ‘es invariante’ no basta; hay que probarlo en implementación.

Qué. Demostrar (analítica o numéricamente) que la predicción no cambia bajo transformaciones permitidas.

4F12. Compatibilidad con Simetrías/Anomalías (cuando aplique)

Por qué. Una unificación que ignora anomalías puede ser inconsistente.

Qué. Declarar condiciones de cancelación/anomalías o derivar su rol; conectar con módulo de anomalías.

4F13. Compatibilidad con positividad/causalidad (si aplica)

Por qué. Algunas estructuras unificadas violan causalidad/analiticidad.

Qué. Si el frontend usa amplitudes, compilar a bounds de positividad IR-safe.

4F14. Separación dinámica vs diccionario

Por qué. El diccionario no debe ‘hacer la dinámica por definición’.

Qué. Este módulo sólo fija objetos y observables; la dinámica se evalúa en UD u otros módulos.

4F15. Auditoría numérica

Por qué. La implementación puede esconder convenciones.

Qué. Hashes, versiones, seeds, tolerancias y pruebas de invariancia obligatorias.

4F16. Robustez a elecciones equivalentes (clases de equivalencia)

Por qué. Un diccionario universal debe tolerar parametrizaciones equivalentes.

Qué. Definir equivalencias (gauge, difeo, reparametrización EFT) y exigir que el veredicto sea estable bajo ellas.

4F17. Taxonomía de veredictos (incluye NO-EVAL)

Por qué. Sin diccionario completo no se puede evaluar UD.

Qué. PASS/FAIL/PASS fuerte/NO-EVAL según completitud del diccionario y su conexión a D_obs.

4F18. Rigidez del diccionario

Por qué. Si el diccionario admite demasiadas elecciones libres, la unificación es narrativa.

Qué. Medir rigidez: cuántas convenciones quedan sin fijar y si afectan observables dentro de tolerancias.

7. Interfaz con UD (unificación dinámica) y con otros módulos

UD consume este módulo como prerequisito: ninguna propuesta puede alegar ‘unificación dinámica’ si no puede expresarse en CUI/HUI y si no define observables invariantes.

- Con UD. Este módulo entrega el diccionario; UD entrega ecuaciones dinámicas, matching y tests multi-frente.
- Con EFT. Define cómo truncar y normalizar invariantes a orden finito.
- Con Amplitudes/Positividad. Permite traducir coeficientes EFT a invariantes y aplicar bounds.
- Con Gravedad Efectiva. Fija cómo expresar PPN/GW/cosmo en invariantes geométricos.
- Con Observabilidad/ISAAC. Exige pipeline que conecte holonomías/invariantes a D_obs.

8. Plantillas canónicas (YAML/JSON) para entrada y salida

Plantilla mínima (no ejecutable) para declarar CUI/HUI:

fourF_dictionary: domain_ISAAC: Omega_I: {window: ..., resolution: ..., validity: ...}
symmetry_G: group_or_algebra: ... representations: ... transformations: ...
connection_A: class: bundle_extended | cartan | direct_sum | ... components: ...
conventions: {normalizations: ..., sign: ...} curvature_F: definition: dA + A^A | ...
invariants_allowed: [...] holonomy_HUI: contours_allowed: ... observable_map: ...
matter_coupling: covariant_derivative_D: ... currents_J: ... interface: links_to_EFT: ...
links_to_UD: ... audit: hashes: ... versions: ... seeds: ...

Salida típica: lista de invariantes observables + su pipeline de medición (D_obs).

9. MRD-4F (demo mínimo reproducible)

MRD-4F demuestra el diccionario sin asumir una dinámica ‘revolucionaria’. Basta una instancia mínima:

- Entrada. (i) un gauge interno simple (U(1) o SU(2)), (ii) una conexión geométrica (Levi-Civita o Cartan), (iii) una regla de combinación en ▯ (suma directa o bundle extendido).
- Invariantes. Construir 2–3 invariantes: un Wilson loop W(C), un escalar de curvatura integrado, y un correlador gauge-invariante simple.
- Pipeline. Para cada invariante, declarar un D_obs sintético (ventana/resolución) y demostrar invariancia bajo transformaciones.
- Auditoría. PCN: pruebas de invariancia numérica y estabilidad bajo discretización.

10. PASS/FAIL, PASS fuerte y NO-EVAL

- PASS. Existe una traducción completa a CUI/HUI y una lista de observables invariantes conectados a D_obs en Ω_I.
- PASS fuerte. El diccionario es rígido: elecciones equivalentes no cambian predicciones dentro de tolerancias; holonomías/invariantes tienen pipeline claro.
- FAIL. La propuesta requiere objetos no invariantes o perillas funcionales provenientes de B para definir observables.
- NO-EVAL. Falta definición completa de □/□/Ω_I o falta mapa a D_obs (observables no operacionalizados).

11. Loopholes/artefactos frecuentes

- Unificación por gauge. Elegir gauge/coordenadas para ‘igualar’ fuerzas: prohibido por 4F2/4F11.
- Normalizaciones escondidas. Ajustar signos/factores para forzar coincidencias: prohibido por 4F4.
- Holonomías sub-ISAAC. Usar loops más finos que la resolución: prohibido por 4F7.
- Microestructura funcional. Introducir funciones arbitrarias ‘de B’: prohibido por 4F8.

12. Definición de Hecho (DoD)

- DoD-1. □, □, $\mathbb{F}$, □, □ y Ω_I declarados con convenciones auditables.
- DoD-2. Lista de invariantes admitidos y demostración de invariancia (analítica o numérica).
- DoD-3. Definición HUI: clase de contornos y cómo se conectan a medición (D_{obs}).
- DoD-4. Frontera A/B explícita (sin perillas funcionales).
- DoD-5. MRD-4F reproducible con auditoría (PCN).

Apéndice A: glosario

- CUI. Conexión Unificada Invariante: objetos mínimos (□, 𝔽, □, □, □, Ω_I).
- HUI. Holonomía Unificada Invariante: observables de transporte W(C).
- Holonomía. Elemento de grupo (o clase de conjugación) obtenido por transporte paralelo a lo largo de C.
- Invariante. Cantidad independiente de gauge y coordenadas, consumible como observable.
- Rigidez. Medida de cuánta libertad queda tras fijar equivalencias.
- Ω_I (ISAAC). Dominio operacional que restringe inferencias y operaciones.

Compilación de Consistencia Operacional (OCC)

Módulo SK — No-equilibrio: kernels, causalidad y positividad (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo define el frontend de sistemas abiertos/no-equilibrio dentro de OCC. Traduce un modelo efectivo con entorno (vacío, baño térmico o sector inaccesible SIA) a restricciones operacionales sobre kernels de respuesta y ruido, de modo que el Sector A (SGO) conserve interpretación física (probabilidades no negativas, causalidad, y —si aplica— equilibrio térmico).

Prioridad. Primero el por qué: qué se rompe si el módulo no existe o se define mal. Luego el qué: condiciones técnicas y tests PASS/FAIL.

Contenido

0. Definición de alcance (qué cubre y qué no)

Cubre: dinámica efectiva en el Sector A (SGO) cuando existe un entorno/sector complementario (incluida SIA) que induce ruido, disipación, memoria o decoherencia. Se admite equilibrio o fuera de equilibrio.

No cubre: microdinámica completa de SIA. Por diseño, el microestado de SIA no es reconstruible; el módulo sólo impone restricciones sobre la dinámica efectiva accesible en A.

Conexión con Documento A. Este módulo implementa el frontend de sistemas abiertos (candado C8) y, si se declara equilibrio, activa el candado térmico (C9/KMS).

1. Motivación (por qué): qué se rompe sin SK

Si el observador sólo accede al Sector A, la descripción completa “unitaria y cerrada” rara vez es operacional. Se trabaja con estados reducidos $\rho_A(t)$. Sin una estructura mínima, pueden aparecer tres fallas graves:

- Probabilidades no físicas. Dinámicas efectivas que generan matrices con autovalores negativos o mapas no completamente positivos: predicen probabilidades negativas al acoplar un ancilla.
- Violación de causalidad efectiva. Respuestas que ocurren antes de la perturbación (kernels avanzados), o memoria no compatible con un orden causal en A.
- Arbitrariedad infinita. Introducir “kernels a gusto” para ajustar cualquier dato convierte el entorno en perilla ilimitada y destruye falsabilidad.

Schwinger–Keldysh (SK) es el formalismo mínimo que separa de forma controlada: (i) respuesta causal (retardada), (ii) ruido (fluctuaciones), (iii) consistencia (cuando hay equilibrio: relaciones tipo KMS/fluctuation–dissipation).

2. Objeto operacional y frontend (canal CPTP inducido)

El objeto operacional observable es el mapa efectivo en A que lleva estados a estados:

$$\rho_A(t) = \Phi_{t,t0}(\rho_A(t0)).$$

El módulo exige que $\Phi_{t,t0}$ sea físicamente válido para cualquier extensión con ancilla, es decir, que sea un canal completamente positivo y preservador de traza (CPTP), o una familia consistente en tiempo (composición y continuidad apropiadas).

En representación SK, el entorno se integra y produce una acción efectiva no-local que se parametriza por kernels. El juez no permite “adivinar” kernels: deben satisfacer candados inevitables (Sección 4) y someterse a anclajes (Sección 6).

3. Estructura mínima SK (acciones duplicadas, kernels)

Se introduce la duplicación de campos (rama + y rama −) y se cambia a base clásica/cuántica (c,q): $\varphi_c = (\varphi_+ + \varphi_-)/2$, $\varphi_q = \varphi_+ - \varphi_-$. La acción efectiva para un grado de libertad representativo (generalizable) toma la forma:

$$S_{eff}[\varphi_c,\varphi_q] = \int dt\, dt' \left[\varphi_q(t)\, D_R(t,t')\, \varphi_c(t') + (i/2)\, \varphi_q(t)\, N(t,t')\, \varphi_q(t') \right] + \ldots$$

donde:

- D_R es el kernel de respuesta retardada (disipación/memoria).
- N es el kernel de ruido (correlación de fluctuaciones) y debe ser semidefinido positivo.
- Los “…” incluyen términos locales/cúbicos/etc. si el módulo los declara; por defecto, se trabaja al orden cuadrático mínimo para tests de consistencia.

Esta forma ya incorpora la separación conceptual: D_R controla causalidad; N controla positividad estadística.

4. Candados inevitables del módulo (SK1–SK7)

Cada candado se introduce por la falla operacional que previene.

SK1. Causalidad (retardación estricta)

Por qué. Sin retardación, una perturbación en A podría producir respuesta antes de ocurrir, violando causalidad efectiva.

Qué. $D_R(t,t') = 0$ para $t < t'$ (soporte retardado).

SK2. Realidad y normalización (unitaridad SK)

Por qué. La probabilidad total debe conservarse; el generador debe cumplir las identidades SK estándar.

Qué. $S_{eff}[\varphi_+,\varphi_-]$ debe anularse cuando $\varphi_+ = \varphi_-$ (condición de normalización $Z[0]=1$).

SK3. Positividad del ruido (PSD)

Por qué. Si N no es PSD, existen combinaciones de fuentes que producen distribuciones con probabilidad negativa.

Qué. Para toda función de prueba $f(t)$: $\iint dt\, dt'\, f(t)\, N(t,t')\, f(t') \geq 0$.

SK4. CPTP inducido (canal físico)

Por qué. Complete positivity garantiza que ningún experimento con ancillas revele negatividad.

Qué. El par (D_R,N) debe inducir un mapa $\Phi_{t,t0}$ CPTP en el régimen declarado. En el límite Markoviano esto se reduce a forma GKSL/Lindblad; en no-Markov, se exige CP divisibility o una condición declarada equivalente (ver Sección 9 para caveats).

SK5. Fluctuation–dissipation / KMS (sólo si se declara equilibrio)

Por qué. Si el módulo declara equilibrio a temperatura β^{-1}, ruido y disipación no son independientes: su relación evita arbitrariedad.

Qué. En frecuencia, $N(\omega)$ y $\mathrm{Im}\, D_R(\omega)$ satisfacen una relación tipo: $N(\omega) = \coth(\beta\omega/2) \cdot 2\, \mathrm{Im}\, D_R(\omega)$ (convención declarada).

SK6. Límite IR correcto y continuidad

Por qué. Los kernels deben tener límites finitos y continuos donde el objeto es medible; divergencias IR no tratadas producen falsos FAIL/PASS.

Qué. El módulo declara regularización IR y muestra estabilidad de los tests bajo ese esquema.

SK7. No reinyección UV (ISAAC)

Por qué. Ajustar kernels con estructura en escalas no operables destruye falsabilidad.

Qué. Los kernels se parametrizan con soporte espectral efectivo limitado por el dominio ISAAC declarado; cualquier extensión UV debe declararse como nuevo módulo y re-compilarse.

5. Compilación OCC (cómo se convierte a constraints)

En el compilador, el módulo SK transforma un modelo (o parametrización) de kernels en restricciones numéricas. Ejemplos de traducción:

- SK1: implementar D_R como kernel con soporte t≥t' o, en frecuencia, con polos en el semiplano inferior.
- SK3: imponer PSD de N discretizando en una malla (t_i) y exigiendo que la matriz N_{ij} sea PSD (SDP).
- SK4: en Markov, verificar que el generador es GKSL (matriz de Kossakowski PSD); en no-Markov, declarar criterio (p.ej. CP-divisibility en ventana) y compilarlo.
- SK5: si KMS, imponer igualdad (con tolerancia) entre N(ω_k) y 2coth(βω_k/2)Im D_R(ω_k) en una rejilla espectral auditada.

La compilación produce el conjunto factible F_SK dentro del dominio ISAAC. Luego se intersecta con D_obs del módulo (Sección 6).

6. Anclajes D_obs típicos (formatos y traducción auditable)

El módulo SK es universal, por lo que D_obs depende del sistema. Para mantenerlo cerrado, se define una lista de formatos admitidos:

- D_obs-R(t). Función de respuesta medida (susceptibilidad) → ancla sobre D_R.
- D_obs-S(ω). Densidad espectral de fluctuaciones → ancla sobre N(ω).
- D_obs-Γ. Tasas de decoherencia/relajación → anclas sobre combinación de D_R y N (en aproximación declarada).
- D_obs-KMS. Si equilibrio, medición de temperatura β^{-1} y verificación de KMS/FDT.

Regla de cierre: el módulo debe declarar explícitamente cómo traduce el dato a restricción (unidades, convenciones, ventanas de frecuencia/tiempo, tratamiento de ruido instrumental). Sin esa traducción, el juez no compila el módulo.

7. MRD-SK (demo mínimo reproducible)

MRD-SK propone un sistema mínimo que cualquier tercero puede ejecutar para verificar el pipeline completo sin ambigüedad.

- Sistema. Oscilador armónico acoplado linealmente a un baño gaussiano (modelo de Caldeira–Leggett) en (i) $T=0$ y (ii) $T>0$.
- Entrada. Parametrización de D_R y N a partir de una densidad espectral $J(\omega)$ con corte ISAAC (Λ_I).
- Tests. SK1–SK7 + (si aplica) KMS en el régimen térmico.
- Salida. PASS/FAIL, rigidez (R) sobre parámetros del kernel, y reporte de auditoría (hashes, versiones, tolerancias).

Este MRD fija un estándar: cualquier módulo que use SK debe al menos poder reducirse a este caso límite cuando se apaga la complejidad adicional.

8. Salidas y criterios PASS/FAIL (incluye PASS fuerte)

F_SK es el conjunto de kernels que satisfacen SK1–SK7 dentro del dominio ISAAC. D_obs es el anclaje experimental. El conjunto superviviente es I_SK ≡ F_SK ∩ D_obs.

- FAIL. I_SK = ∅.
- PASS. I_SK ≠ ∅.
- PASS fuerte. Región superviviente rígida: R ≲ 1 o d_eff≈0.

Diagnóstico obligatorio: si FAIL, el reporte debe indicar qué candado choca primero (p.ej. N no PSD en el rango medido, o violación de KMS al declarar equilibrio).

9. Loopholes/artefactos frecuentes y cómo evitarlos

- No-Markov y divisibilidad CP. En memoria fuerte, CP-divisibility puede fallar aun si la evolución global es física. El módulo debe declarar qué noción usa (CP global vs divisibilidad) y limitar el veredicto a ese dominio.
- Discretización. PSD de N depende de la malla; se exige prueba de convergencia bajo refinamiento.
- Ventanas espectrales. FDT/KMS sólo aplica en equilibrio; mezclar transitorios produce falsos FAIL.
- Contraterminos. Re-definiciones locales mueven contribuciones entre D_R y términos locales; fijar convención.

10. Definición de Hecho (DoD) para publicar este módulo

- DoD-1. Especificación formal + convención fija.
- DoD-2. Implementación SK1–SK7 + tests negativos que detecten violaciones.
- DoD-3. MRD-SK ejecutable con auditoría completa.
- DoD-4. Al menos un anclaje real D_obs con PASS/FAIL sin ajuste post-hoc.
- DoD-5. Predicción discriminante o no-go (bound sobre ruido/damping o prohibición de un kernel ad hoc).

Apéndice A: glosario SK

- SK. Formalismo de contorno cerrado en tiempo real para no-equilibrio.
- D_R. Kernel de respuesta retardada.
- N. Kernel de ruido semidefinido positivo.
- CPTP. Mapas completamente positivos y preservadores de traza.
- KMS/FDT. Condición de equilibrio que relaciona ruido y disipación.
- ISAAC/Λ_I. Dominio/corte operacional que prohíbe reinyección UV.

Definiciones canónicas de SGO/SIA/ISAAC e $_{\mathrm{OCC}}$ están en el Documento A.

Compilación de Consistencia Operacional (OCC)

Módulo Rama Efectiva — decoherencia, objetividad y no-backflow (Wigner-friend operativo) (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo no “elige” una interpretación ontológica de la mecánica cuántica. Define un criterio operacional para declarar cuándo un resultado se vuelve un hecho en el Sector A: cuando existe una rama efectiva caracterizada por (i) decoherencia, (ii) registro redundante en el entorno, y (iii) ausencia de backflow recuperable por cualquier protocolo permitido en Ω_I (ISAAC).

Qué resuelve. Evita que ‘observación’ sea una palabra: especifica condiciones medibles (o auditables) para objetividad intersubjetiva. El caso Wigner’s friend se vuelve un test: si la interferencia/recoherencia es operacionalmente posible en Ω_I, no hay rama efectiva (FAIL-RE).

Contenido

0. Alcance y dependencias

Depende de: Documento A (OCC) y del Módulo SK (Schwinger-Keldysh) cuando se modela explícitamente el entorno como canal CPTP con respuesta retardada y ruido. También usa Observabilidad/ISAAC para definir qué operaciones están permitidas en Ω_I.

Cubre: criterio operacional de 'hecho' (rama efectiva) y su verificación por condiciones de decoherencia + redundancia + no-backflow.

No cubre: ontología (colapso real vs muchos mundos vs QBism). Este módulo sólo define cuándo una afirmación es consumible como anclaje D_obs o como salida estable de un experimento dentro de OCC .

1. Motivación (por qué): por qué se necesita un criterio de ‘hecho’

Toda la arquitectura OCC necesita una noción cerrada de ‘dato’: sin un criterio de hecho, cualquier disputa sobre medición puede reabrir la maleabilidad. El módulo Rama Efectiva evita que el resultado dependa de intuiciones sobre observadores.

- Por qué no basta ‘decoherencia’ sola. Puede haber decoherencia parcial sin objetividad (información no redundante).
- Por qué no basta ‘registro’ solo. Un registro puede ser reversible si existe un protocolo de recuperación (backflow).
- Qué se necesita. Un triángulo: decoherencia + redundancia + no-backflow operacional en Ω_I.

2. Objeto operacional: S, E, observador como subsistema y dominio Ω_I

Se divide el universo operacional (Sector A) en:

- S. Sistema de interés (lo que se ‘mide’).
- E. Entorno que interactúa con S y porta registros.
- O. Observador/aparato: tratado como sub-sistema físico dentro de A, no como entidad externa.

Una ‘lectura’ es una interacción física entre O y un fragmento de E (o directamente con S) que produce un registro estable en el propio O y/o en E.

El dominio Ω_I fija qué operaciones son posibles: energías, tiempos, accesos experimentales y resolución. En particular, Ω_I determina el conjunto de protocolos permitidos para intentar recoherencia o ‘deshacer’ el registro.

3. Definiciones: decoherencia, base puntero, redundancia, lectura, no-backflow

3.1 Decoherencia operacional

Supresión de términos fuera de la diagonal de ρ_S en una base (candidata) dentro de tolerancias definidas por la resolución/ruido.

3.2 Base puntero

Base en la cual los estados de S son estables frente a la interacción con E (selección ambiental). Debe derivarse del Hamiltoniano/canal efectivo, no elegirse post-hoc.

3.3 Redundancia (objetividad)

La misma información clásica sobre S está registrada en múltiples fragmentos aproximadamente independientes $\{E_k\}$. Objetividad intersubjetiva implica que múltiples observadores pueden acceder a copias sin perturbar significativamente S.

3.4 No-backflow operacional

No existe protocolo permitido en Ω_I que recupere coherencias relevantes (o que revierta el registro) por encima de un umbral ε_{rec}. Esto es una condición operacional: depende del conjunto de operaciones disponible, no de una afirmación absoluta sobre el universo.

4. Candados inevitables del módulo (RE1–RE18) con ‘por qué → qué’

RE1. Partición explícita S/E/O

Por qué. Sin partición no hay definición operacional de lectura o registro.

Qué. Declarar S, E y O con variables accesibles y su relación con Ω_I.

RE2. Dinámica efectiva CPTP (si se usa canal)

Por qué. Sin CPTP no hay interpretación física de evolución abierta.

Qué. Si se modela como canal, debe ser CPTP; se verifica con Módulo SK (cuando aplica).

RE3. Base puntero no arbitraria

Por qué. Elegir base a posteriori reintroduce maleabilidad.

Qué. Derivar base puntero desde el acoplo S–E (o kernel SK); publicar el procedimiento.

RE4. Decoherencia cuantificada

Por qué. ‘Está decoherido’ sin escala no es falsable.

Qué. Definir métrica $C(t)$ y umbral ε_{dec} ligado a resolución/ruido.

RE5. Redundancia mínima

Por qué. Sin redundancia no hay objetividad intersubjetiva.

Qué. Requerir $R_{red} \geq R_{min}$ (fragmentos E_k que portan la misma información dentro de tolerancia).

RE6. Lectura operacional y tiempo de lectura

Por qué. Evita observadores ‘externos’ implícitos.

Qué. τ_{read} definido por el protocolo físico de O; exigir $\tau_{stab} \geq \tau_{read}$.

RE7. Estabilidad temporal del registro

Por qué. Un registro que se borra antes de ser consultado no es hecho.

Qué. Definir τ_{stab} y exigir margen consistente con Ω_I.

RE8. No-backflow (recuperabilidad) como candado

Por qué. Si se puede recoherer, el ‘hecho’ no es estable.

Qué. Definir operaciones permitidas □(Ω_I) y exigir recuperabilidad $\leq \varepsilon_{rec}$.

RE9. Prohibición de post-selección oculta

Por qué. Post-selección puede fabricar ramas.

Qué. Declarar post-selección; si es esencial, tratar como anclaje condicionado.

RE10. Separación definición vs inferencia

Por qué. ‘Rama’ no debe definirse usando el mismo dato que intenta justificar.

Qué. Definir partición y base puntero antes de mirar resultados.

RE11. Consistencia multi-observador

Por qué. Hecho = acuerdo operacional reproducible.

Qué. O1 y O2 deben poder leer copias redundantes sin inducir recoherencia relevante.

RE12. Límite ISAAC (operaciones prohibidas)

Por qué. Permitir operaciones sub-ISAAC reabre backflow por reinyección UV.

Qué. $\square(\Omega_I)$ excluye protocolos que requieren control/resolución más fina que ISAAC.

RE13. Diagnóstico de Wigner-friend

Por qué. Es el test más duro de objetividad.

Qué. Si existe interferometría global en $\square(\Omega_I)$ que revele coherencia del laboratorio, entonces FAIL-RE.

RE14. Manejo de memoria/no-Markov

Por qué. No-Markov puede violar CP-divisibilidad sin ser no físico.

Qué. El candado se formula en recuperabilidad operacional (RE8), no en CP-divisibilidad local.

RE15. Auditoría computacional

Por qué. Los diagnósticos dependen de discretización.

Qué. Seeds, mallas, tolerancias, convergencia (PCN) obligatorias.

RE16. Robustez a particiones razonables

Por qué. Una rama no debe depender de una partición caprichosa.

Qué. Reportar sensibilidad a particiones alternativas dentro de clases equivalentes ISAAC.

RE17. Taxonomía veredictos (incluye NO-EVAL)

Por qué. Sin input auditable no se puede declarar hecho.

Qué. Aplicar PASS/FAIL/PASS fuerte/NO-EVAL según Sección 9.

RE18. Rigidez del hecho

Por qué. Hechos ‘débiles’ reabren maleabilidad.

Qué. Reportar rigidez: dependencia de parámetros del entorno/ruido dentro de tolerancias auditadas.

5. Compilación OCC : cómo se decide RE-PASS/FAIL

La compilación produce un conjunto factible F_RE de dinámicas/parametrizaciones que satisfacen RE1–RE18 en Ω_I. El anclaje D_obs típico es un protocolo de medición reproducible (instrumentación + lectura) exportado en formato canónico.

Salida: I_RE ≡ F_RE ∩ D_obs. Si el mismo protocolo admite recuperación de coherencias por operaciones permitidas, entonces I_RE se vacía y el resultado no puede tratarse como hecho.

6. Caso Wigner-friend como test operacional

Se modela un laboratorio L (amigo + sistema + aparato) como parte del Sector A. Un observador externo W puede intentar una medición global.

En este marco:

- Si W puede implementar en □(Ω_I) una interferometría global que revele coherencias del laboratorio, entonces no existe rama efectiva estable: FAIL-RE.
- Si tal interferometría requiere control sub-ISAAC (o recursos fuera de Ω_I), entonces queda fuera del conjunto operacional permitido y no invalida la rama: PASS-RE posible.

Esto no elige ontología; sólo clasifica la situación por recuperabilidad operacional.

7. Recurrencias de Poincaré y finitud: cómo se manejan

En sistemas finitos existen recurrencias en tiempos enormes. El módulo no exige ‘irreversibilidad absoluta’; exige irreversibilidad operacional:

- Regla. Si τ_{rec} (tiempo de recurrencia/recoherencia) $\gg$ τ_{op} (horizonte operacional relevante en Ω_I), la rama puede tratarse como hecho en A.
- Si τ_{rec} es comparable a τ_{op}, el módulo degrada el veredicto (PASS débil) o emite NO-EVAL si no se puede estimar τ_{rec}.

τ_{op} se declara desde Observabilidad/ISAAC (tiempos de acceso, estabilidad instrumental, etc.).

8. MRD-RE (demo mínimo reproducible)

MRD-RE recomendado: un modelo Caldeira-Leggett o spin-boson con lectura por un aparato O simplificado.

- Entrada. Kernel SK (respuesta + ruido) o Hamiltoniano explícito S–E; partición S/E/O.
- Salida. (i) decoherencia C(t) en base puntero derivada, (ii) redundancia R_{red} midiendo múltiples fragmentos E_k, (iii) test de recuperabilidad (recoherencia) bajo operaciones permitidas.
- Veredicto. PASS si (i)+(ii)+(iii) se cumplen en Ω_I; FAIL si existe protocolo de recuperación dentro de Ω_I.
- Auditoría. PCN obligatorio: convergencia bajo refinamiento.

9. PASS/FAIL, PASS fuerte y NO-EVAL

- PASS. Existe rama efectiva: decoherencia + redundancia + no-backflow operacional.
- PASS fuerte. Rama rígida: no depende sensiblemente de variaciones razonables del entorno/ruido dentro de tolerancias auditadas.
- FAIL. Existe recuperabilidad operacional (backflow) por encima de ε_{rec} dentro de □(Ω_I), o la base puntero es post-hoc.
- NO-EVAL. Falta input auditable (partición, kernel, □(Ω_I), o estimación de estabilidad/recurrencias).

10. Loopholes/artefactos frecuentes

- Invocar ‘observador’ como axioma. Prohibido: O debe ser subsistema físico.
- Elegir base puntero mirando el resultado. Prohibido por RE3/RE10.
- Ignorar □(Ω_I). No-backflow debe definirse con operaciones permitidas.
- Confundir no-Markov con no-físico. El módulo usa recuperabilidad operacional, no CP-divisibilidad local.

11. Definición de Hecho (DoD)

- DoD-1. Partición S/E/O + definición de operaciones permitidas □(Ω_I).
- DoD-2. Base puntero derivada y métrica de decoherencia con umbral ligado a resolución.
- DoD-3. Medida de redundancia R_{red} con criterio mínimo R_{min}.
- DoD-4. Test explícito de recuperabilidad (backflow) dentro de □(Ω_I).
- DoD-5. MRD-RE reproducible con PCN (convergencia) y reporte de rigidez.

Apéndice A: glosario

- Rama efectiva. Clase operacional de estados que actúan como resultados estables en A.
- Base puntero. Base seleccionada por interacción S–E que estabiliza estados.
- Redundancia. Copias independientes del registro en fragmentos del entorno.
- No-backflow operacional. Ausencia de protocolo permitido que recupere coherencias por encima de ε_{rec}.
- Wigner-friend operativo. Test: si la interferometría global es posible en Ω_I, no hay rama efectiva.
- ISAAC/Ω_I. Límite/dominio operacional que restringe operaciones e inferencias.

Compilación de Consistencia Operacional (OCC)

Módulo Simetrías, Anomalías y Topología Operacional (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo fija los candados globales que muchas propuestas “localmente consistentes” violan: (i) simetrías y sus realizaciones (exactas, rotas, emergentes), (ii) anomalías (gauge, gravitacionales, mixtas) y matching IR/UV, y (iii) restricciones topológicas (cuantización de cargas/flujo, términos θ, clases de fibrados).

Por qué es inevitable. Una teoría puede pasar tests locales (EFT, amplitudes, ecuaciones) y aun así ser inconsistente globalmente. OCC necesita un juez que detecte esas inconsistencias sin depender de preferencia ontológica: las anomalías y cuantizaciones son condiciones inevitables.

Contenido

0. Alcance y dependencias

Depende de: Documento A (metodología universal) y del Módulo 4F (diccionario CUI/HUI) para expresar invariantes. Usa EFT/Renormalización para cálculo de anomalías en el régimen efectivo, y Gravedad Efectiva si hay anomalías gravitacionales.

Cubre: constraints globales: anomalías, matching, cuantización topológica y consistencia de simetrías.

No cubre: la fenomenología específica de un modelo particular; este módulo decide si una propuesta es globalmente consistente y compila a invariantes.

1. Motivación (por qué): por qué el cierre global es parte del juez

OCC no puede limitarse a consistencia local (ecuaciones, estabilidad, positividad). Hay fallos que sólo aparecen cuando se exige:

- Invariancia gauge global. El path integral no es bien definido si hay anomalía gauge no cancelada.
- Consistencia gravitacional. Anomalías gravitacionales o mixtas pueden prohibir ciertos contenidos de materia.
- Cuantización/topología. Cargas y flujos no pueden elegirse continuos si la topología exige cuantización.

Estos son candados inevitables: no dependen de gustos sobre interpretaciones; son condiciones de definición matemática + consistencia operacional.

2. Lenguaje operacional: simetrías, redundancias y observables invariantes

Una simetría puede ser:

- Gauge (redundancia). No observable directamente; obliga a formular observables invariantes.
- Global. Puede generar cargas conservadas; su ruptura/anomalía tiene consecuencias medibles.
- Emergente/efectiva. Válida sólo en Ω_I; debe declararse como tal (Documento A, PCD/PCN).

El módulo exige: (i) lista explícita de simetrías, (ii) su realización (lineal/no lineal, rota espontánea/explicita), y (iii) el conjunto de observables invariantes que se usarán como salida/entrada del compilador.

3. Anomalías como constraints inevitables

Una anomalía es la falla de una simetría clásica al cuantizar. En OCC , las anomalías funcionan como constraints duros:

- Anomalías gauge. Deben cancelarse para que el gauge sea una redundancia consistente.
- Anomalías gravitacionales. Restringen contenidos de materia o acoplos; pueden romper conservación covariante a nivel cuántico.
- Anomalías mixtas. Restringen compatibilidad entre simetrías internas y gravedad.

Cuando la teoría es efectiva (EFT), el cálculo se hace en Ω_I con error EFT declarado; el veredicto se emite con estabilidad (Documento A, meta-candado).

4. Matching 't Hooft: el candado IR/UV

Si una simetría global es exacta en el UV, sus anomalías deben igualarse en el IR ('t Hooft anomaly matching), o la teoría no puede fluir a ese IR sin introducir nuevos grados de libertad o ruptura.

Rol en OCC . Impide 'arreglar' inconsistencias diciendo que en B (UV) todo se arregla, sin pagar un costo observable en A.

5. Topología operacional: cuantización, términos θ y sectores

La topología impone condiciones que no son locales. Operacionalmente aparecen como:

- Cuantización de carga/flujo. Ej. cuantización de monopolos/fluxes en bundles no triviales.
- Sectores topológicos. Distintos sectores que no se conectan por transformaciones pequeñas.
- Términos θ y fases. Parámetros angulares con periodicidad; su presencia/ausencia tiene consecuencias de CP, etc.

El módulo exige declarar la estructura topológica relevante y el dominio Ω_I donde sus efectos pueden o no ser observables.

6. Candados inevitables del módulo (SAT1–SAT20) con ‘por qué → qué’

SAT1. Lista explícita de simetrías

Por qué. Sin lista, el compilador no sabe qué proteger y qué puede romperse.

Qué. Enumerar simetrías gauge/globales/emergentes y su rango de validez (Ω_I).

SAT2. Identificar gauge vs global

Por qué. Confundir gauge con global crea observables no físicos.

Qué. Clasificar cada simetría: redundancia (gauge) o física (global).

SAT3. Observables invariantes obligatorios

Por qué. Sin invariantes, cualquier ‘predicción’ depende de gauge.

Qué. Dar el conjunto de invariantes consumibles (con pipeline a D_obs).

SAT4. Criterio de cancelación de anomalías gauge

Por qué. Anomalía gauge → teoría inconsistente.

Qué. Demostrar cancelación (exacta o por mecanismo declarado) en Ω_I.

SAT5. Anomalías gravitacionales y mixtas

Por qué. Pueden invalidar la teoría incluso si gauge interno es consistente.

Qué. Calcular/argumentar anomalías gravitacionales/mixtas relevantes.

SAT6. Matching ’t Hooft (si simetría global exacta UV)

Por qué. Si no coincide, el IR propuesto no puede emerger.

Qué. Imponer matching o declarar explícitamente ruptura/añadir DOF.

SAT7. Truncación EFT y error

Por qué. Cálculos de anomalía en EFT sin error no son confiables.

Qué. Declarar orden, Λ, error EFT y estabilidad (PCD/PCN).

SAT8. Términos Wess–Zumino/WZW

Por qué. A veces son necesarios para matching.

Qué. Incluirlos cuando corresponda o justificar su ausencia.

SAT9. Cuantización topológica

Por qué. Cargas/fluxes no pueden variarse libremente.

Qué. Declarar condiciones de cuantización y su origen topológico.

SAT10. Sectores y superselección

Por qué. No todos los sectores se conectan operacionalmente.

Qué. Declarar sectores accesibles en Ω_I y cómo se prepara/identifica un sector.

SAT11. Parámetros angulares (θ) y periodicidad

Por qué. θ no es un parámetro real continuo sin estructura.

Qué. Declarar periodicidad, simetrías discretas y consecuencias observables.

SAT12. Consistencia con 4F-CUI/HUI

Por qué. El diccionario 4F define invariantes; este módulo los valida globalmente.

Qué. Alinear simetrías/anomalías con la formulación CUI/HUI.

SAT13. Consistencia con amplitudes/positividad (si aplica)

Por qué. Anomalías pueden aparecer como violaciones de analiticidad/causalidad en ciertos canales.

Qué. Cuando aplique, cruzar con módulos de amplitudes.

SAT14. Consistencia con gravedad efectiva/local tests

Por qué. Topología/anomalías no deben violar constraints locales ya anclados.

Qué. Verificar compatibilidad con D_obs locales (PPN, etc.) cuando corresponda.

SAT15. Operacionalidad de simetrías emergentes

Por qué. Simetrías 'emergentes' usadas como perilla son maleabilidad.

Qué. Si es emergente, declarar el rango de energía/escala y mostrar que se rompe fuera de Ω_I.

SAT16. Evitar 'rescate' por Sector B

Por qué. No se admite 'en el UV se arregla' sin matching.

Qué. Cualquier rescate UV debe compilar a matching/efectos en A o NO-EVAL.

SAT17. Auditoría computacional

Por qué. Anomalías pueden depender de convenciones y regularización.

Qué. Publicar regularización, convenciones, código, hashes, seeds.

SAT18. Robustez a elecciones equivalentes

Por qué. El veredicto no debe depender de parametrizaciones equivalentes.

Qué. Probar invariancia bajo reparametrizaciones/EoM (EFT).

SAT19. Taxonomía veredictos (incluye NO-EVAL)

Por qué. Sin información UV/IR suficiente no hay veredicto justo.

Qué. Emitir PASS/FAIL/PASS fuerte/NO-EVAL según completitud y estabilidad.

SAT20. Rigidez global

Por qué. Una teoría ‘pasa’ pero con libertad topológica arbitraria → maleable.

Qué. Reportar rigidez: qué estructuras topológicas quedan libres y si afectan observables.

7. Interfaz con 4F (CUI/HUI), EFT, amplitudes y gravedad efectiva

- Con 4F. Valida que los invariantes propuestos respetan simetrías y condiciones topológicas globales.
- Con EFT. Define cómo calcular anomalías y términos WZ/WZW en truncación finita con error.
- Con amplitudes. Cuando existan descripciones S-matrix, ciertos fallos globales se reflejan en inconsistencias de analiticidad/crossing.
- Con gravedad efectiva. Anomalías gravitacionales/mixtas y cuantización pueden imponer constraints en cosmología o sistemas compactos.

8. Plantillas canónicas (YAML/JSON)

Plantilla mínima (no ejecutable):

```
symmetry_anomaly_topology: domain_ISAAC: {Omega_I: ...} symmetries: - {name: ..., type: gauge|global|emergent, realization: ...} invariants_observables: - {name: ..., definition: ..., D_obs_map: ...} anomalies: gauge: {status: cancelled|present, evidence: ...} gravitational: {status: ..., evidence: ...} mixed: {status: ..., evidence: ...} tHooft_matching: required: true|false UV_anomaly: ... IR_anomaly: ... resolution: matched|not_matched|no_eval topology: bundle_class: ... quantization: ... theta_terms: ... sectors: ... audit: regularization: ... conventions: ... hashes: ...
```

9. MRD-SAT (demo mínimo reproducible)

MRD-SAT recomendado (nivel 1): una teoría gauge simple con fermiones, donde la cancelación de anomalías sea no trivial.

- Entrada. Contenido de materia + simetrías + regularización declarada.
- Test. Calcular anomalía gauge (p.ej. triángulo) y verificar cancelación.
- Matching. Proponer un IR con grados de libertad distintos y verificar ’t Hooft matching.
- Topología. Incluir un término θ o cuantización de flujo en un caso simple (U(1) con monopolo).
- Salida. PASS/FAIL con diagnóstico y auditoría (hashes/seed).

10. PASS/FAIL, PASS fuerte y NO-EVAL

- PASS. Anomalías gauge canceladas; anomalías gravitacionales/mixtas compatibles; cuantización/topología declarada y consistente; matching IR/UV satisfecho cuando aplique.
- PASS fuerte. Además, robustez bajo elecciones equivalentes (regularización/convensiones) y rigidez global (sin parámetros topológicos libres que afecten observables sin control).
- FAIL. Anomalía gauge no cancelada; matching fallido; inconsistencia topológica/θ incompatible con simetrías o datos.
- NO-EVAL. Falta información UV/IR necesaria para matching, o falta especificación topológica/regularización para decidir.

11. Loopholes/artefactos frecuentes

- “Se arregla en el UV”. Prohibido sin matching explícito (SAT6/SAT16).
- Regularización oculta. Cambiar esquema para borrar anomalía: prohibido sin declarar (SAT17).
- Ignorar sectores. Declarar una predicción sin fijar el sector topológico: produce maleabilidad.
- Tratar θ como real libre. Debe tener periodicidad/estructura y consecuencias observables.

12. Definición de Hecho (DoD)

- DoD-1. Simetrías listadas y clasificadas (gauge/global/emergent) con rango Ω_I.
- DoD-2. Conjunto de invariantes observables + mapa a D_{obs}.
- DoD-3. Cálculo/argumento de anomalías con regularización y error (si EFT).
- DoD-4. Matching ’t Hooft aplicado cuando corresponda, o justificación de ruptura.
- DoD-5. Estructura topológica (cuantización/sectores/θ) declarada y consistente.
- DoD-6. MRD-SAT reproducible con auditoría.

Apéndice A: glosario

- Anomalía. Falla cuántica de una simetría clásica; impone constraint inevitable.
- ’t Hooft matching. Igualdad de anomalías globales entre UV e IR si la simetría es exacta.
- WZ/WZW. Términos efectivos requeridos para restaurar consistencia/matching.
- Sector topológico. Clase no conectada por deformaciones suaves; puede imponer superselección.
- θ-term. Parámetro angular con periodicidad y efectos CP/topológicos.
- Cuantización. Restricción discreta de cargas/flujo por topología.

Compilación de Consistencia Operacional (OCC)

Módulo EFT & Renormalización Operacional — anti-perillas, matching y convenciones auditables (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo define el estándar operacional para usar EFT de forma cerrada: qué cuenta como parámetro físico, qué se absorbe en contraterminos, cómo se fija un esquema de renormalización, cómo se hace matching entre escalas y cómo se evita libertad infinita (perillas) mediante truncación, control de error y rigidez.

Por qué es crítico. Sin este módulo, casi cualquier discrepancia se puede “arreglar” añadiendo operadores, cambiando el esquema o moviendo términos entre ‘vacío’, ‘DM’, ‘DE’, etc. Este módulo prohíbe ese tipo de ajuste semántico.

Contenido

0. Alcance y dependencias

Depende de: Documento A (OCC universal). Puede acoplarse con Amplitudes/Positividad (para bounds UV) y con módulos dinámicos (Vacío/DE, G0, Bariogénesis, etc.) para fijar qué términos son físicos y cuáles son convención.

Cubre: reglas cerradas para construir, truncar, renormalizar y hacer matching de un EFT dentro de Ω_I.

No cubre: elegir el EFT “correcto” por gusto. Sólo evalúa clases de EFT declaradas y su consistencia con candados y anclajes.

1. Motivación (por qué): por qué EFT sin reglas es irrefutable

EFT es poderosa porque permite describir física desconocida con operadores efectivos. Pero esa misma flexibilidad puede destruir falsabilidad:

- Perilla infinita. Si siempre se puede añadir un operador más, ningún dato descarta la clase de teoría.
- Semántica móvil. Mover términos entre ‘vacío’, ‘DM’, ‘DE’, ‘contratermino’ cambia la narrativa sin cambiar observables.
- Esquema oculto. Cambiar renormalización/escala para acomodar un fit produce PASS falsos.

Este módulo convierte EFT en herramienta falsable imponiendo reglas de truncación, error y auditabilidad.

2. Objeto operacional: $L_{eff}(\mu)$, esquema $\mathcal{R}$, observables y dominio Ω_I

Objeto central: un Lagrangiano efectivo (o acción efectiva) truncado a un orden finito:

$$\mathcal{L}_{eff}(\mu) = \mathcal{L}_0 + \Sigma_i \, (c_i(\mu)/\Lambda^{d_i-4}) \, □_i.$$

donde $□_i$ son operadores de dimensión d_i (o conteo equivalente), $c_i(\mu)$ coeficientes renormalizados y Λ la escala de corte EFT declarada.

El EFT se acompaña de un esquema $\mathcal{R}$ (renormalización/regularización) y de una escala μ. Ω_I (ISAAC) fija el rango de energías donde el EFT se aplica y donde no se permite reinyección UV.

3. Definiciones: operador, conteo de potencia, truncación, error EFT, matching, running

3.1 Conteo de potencia

Regla que ordena operadores por supresión $(E/\Lambda)^n$, derivadas, campos, simetrías. Debe declararse y ser consistente.

3.2 Truncación

Selección finita de operadores hasta un orden N. El módulo exige criterio explícito y una estimación del error por truncación.

3.3 Error EFT

Cota operacional del efecto de operadores omitidos en la ventana de aplicación. Sin error EFT, el ajuste de parámetros es arbitrario.

3.4 Matching y running

Matching conecta teorías a distintas escalas (integrar modos pesados); running describe dependencia de μ. El módulo exige consistencia: observables no deben depender de μ dentro del error EFT.

4. Candados inevitables del módulo (E1–E16) con ‘por qué → qué’

E1. Dominio Ω_I y escala de corte declarados

Por qué. Sin dominio, el EFT se usa fuera de validez.

Qué. Declarar Λ, ventana de energías y criterios de validez ($E/\Lambda \le \varepsilon_\Lambda$).

E2. Simetrías declaradas y base completa a orden N

Por qué. Omitir operadores permitidos permite hacks; incluir prohibidos rompe simetría.

Qué. Lista de simetrías + base de operadores completa (o justificación de redundancias por EoM/integración por partes).

E3. Conteo de potencia consistente

Por qué. Sin conteo, “orden” es arbitrario.

Qué. Regla explícita + verificación de que contribuciones se organizan según $(E/\Lambda)^n$.

E4. Truncación finita obligatoria

Por qué. Infinitos operadores → irrefutable.

Qué. Elegir orden N finito y reportar complejidad (número de operadores).

E5. Error EFT obligatorio

Por qué. Sin error, el fit puede absorber física omitida como “señal”.

Qué. Estimar $\Delta_{EFT}(E)$ (p.ej. $\sim(E/\Lambda)^{N+1}$) y propagarlo a observables.

E6. Esquema $\mathcal{R}$ declarado y estable

Por qué. Cambiar esquema puede mover números y simular explicación.

Qué. Declarar regularización/renormalización (p.ej. MS-bar) y mostrar que cambios razonables no cambian veredicto dentro de Δ_{EFT}.

E7. Independencia de μ (RG-invarianza operacional)

Por qué. Observables no deben depender de una escala arbitraria.

Qué. Verificar que $d/d\ln\mu \; O_{pred} \approx 0$ dentro de tolerancias (o incluir running explícito consistente).

E8. Matching explícito al cambiar de EFT

Por qué. Pasar de una teoría a otra sin matching introduce parámetros fantasmas.

Qué. Cuando se integra un modo pesado, se muestran relaciones entre coeficientes antes/después (a orden N).

E9. No reinyección UV (ISAAC)

Por qué. Meter resonancias o funciones fuera de Ω_I evade falsabilidad.

Qué. Cualquier efecto UV fuera de Ω_I requiere un nuevo módulo o un nuevo Λ; no se permite "esconderlo" en $c_i(E)$ funcional.

E10. Separación 'contratermino' vs 'estado/entorno'

Por qué. Confundir renormalización con estado físico permite doble conteo y perillas.

Qué. Convención fija: qué se absorbe en contraterminos y qué se deja como contribución física (por ejemplo Vacío/DE).

E11. Naturalidad operacional (opcional, pero recomendado)

Por qué. Ajustes finos extremos suelen ser perillas disfrazadas.

Qué. Reportar medida de ajuste fino; no es FAIL automático, pero activa revisión de rigidez y dependencia de datos.

E12. Compatibilidad con bounds de positividad (si UV local/causal)

Por qué. Algunos signos/relaciones de c_i son imposibles en UV completables.

Qué. Integrar con Módulo Amplitudes/Positividad: aplicar desigualdades sobre combinaciones de c_i.

E13. Correlaciones y prior explícitos

Por qué. Priors implícitos cambian conclusiones.

Qué. Declarar priors (si bayesiano) o penalizaciones; publicar matriz de correlación cuando hay fits.

E14. Auditabilidad numérica

Por qué. Fitting y RG dependen de implementaciones.

Qué. Versiones, seeds, tolerancias, scripts y hashes; pruebas de estabilidad.

E15. Diagnóstico de degeneracias

Por qué. Direcciones planas hacen el EFT maleable.

Qué. Reportar direcciones planas (PCA) y suprimirlas con nuevos anclajes o declarar que no hay PASS fuerte.

E16. Rigidez obligatoria

Por qué. PASS con demasiados c_i libres no es explicación.

Qué. Reportar rigidez R/d_{eff}; PASS fuerte si el conjunto superviviente es pequeño y no dominado por libertad teórica.

5. Compilación OCC : cómo se traduce a constraints y auditoría

El módulo construye F_EFT como intersección de E1–E16. La compilación típica sigue:

- Definir base {□_i} y truncación N (E2–E4).
- Definir μ, esquema ℛ y Λ; fijar Ω_I (E1, E6).
- Calcular observables O_pred(c_i, μ) y propagar Δ_EFT (E5).
- Imponer RG-invarianza operacional (E7) y matching si hay cambios de teoría (E8).
- Intersectar con anclajes D_obs (Sección 7).
- Opcional: intersectar con bounds de positividad (E12).

Salida canónica: I_EFT ≡ F_EFT ∩ D_obs con reporte de rigidez y diagnóstico de degeneracias.

6. Interfaz con otros módulos (Vacío/DE, G0, Amplitudes/Positividad, UD)

- Con Vacío/DE. E10 fija la frontera: términos renormalizados vs contribución física del vacío/DE. Prohíbe mover 'Λ observado' entre contratermino y estado para salvar un fit.
- Con G0. Prohíbe reabsorber el residual DM en operadores EFT sin declarar matching (E8/E10).
- Con Amplitudes/Positividad. Si se asume UV local/causal, E12 compila bounds de positividad sobre c_i.
- Con UD (unificación dinámica). UD requiere que cualquier frente HEP se exprese con EFT cerrado (este módulo) y bounds (módulo de amplitudes) para evitar unificación por truncación arbitraria.

7. Anclajes D_obs: fits, correlaciones, escalas y ventanas

Formatos admitidos de D_obs:

- D_obs-coefs. Límites/intervalos para c_i (con correlaciones) provenientes de fits.
- D_obs-σ. Observables (σ, distribuciones) en ventana E∈[Emin,Emax] que se mapean a c_i vía predicción EFT.
- D_obs-multi. Conjuntos multi-observable para romper degeneracias (con matriz de covarianza).

Regla: ningún anclaje puede usarse fuera de su ventana; el módulo rechaza extrapolaciones sin error EFT ampliado y justificación.

8. MRD-EFT (demo mínimo reproducible)

MRD-EFT demuestra el módulo con una instancia mínima y auditable:

- Base. Un EFT sencillo (p.ej. 2→2 escalar) con 2–4 operadores hasta orden N.
- Renormalización. Elegir $\mathcal{R}$ y μ; mostrar running consistente (E7).
- Error EFT. Definir $\Delta_{EFT}(E)$ y demostrar que domina extrapolaciones.
- Anclaje. Un fit mínimo (mock auditado o dataset público) que produzca intervalos para c_i.
- Salida. PASS/FAIL + rigidez + auditoría.

9. PASS/FAIL, PASS fuerte y no-go

$I_{EFT} \equiv F_{EFT} \cap D_{obs}$.

- FAIL. Si el EFT no puede definirse de forma cerrada (sin E5/E6/E7) o si los anclajes requieren violar bounds (E12) o usar $E>\Lambda$ sin control.
- PASS. Existe región de c_i compatible con candados y con D_{obs} dentro de Ω_I.
- PASS fuerte. Región rígida: pocas direcciones planas, dominada por errores experimentales y Δ_{EFT}, no por libertad teórica.

No-go típico: un fit que exija un signo de c_i incompatible con positividad (tras IR-safe) descarta UV completables locales/causales para esa clase de EFT.

10. Loopholes/artefactos frecuentes

- “Ajustar” cambiando μ. Prohibido: E7 obliga a RG-invarianza operacional.
- Truncación móvil. Subir/bajar N para salvar tensiones sin declarar error EFT es FAIL (E4/E5).
- Mover Λ. Cambiar Λ después de ver datos es hack; Λ fija Ω_I (E1) y se puede revisar sólo como nueva corrida auditada.
- Reabsorber física en contraterminos. Prohibido por E10 (frontera con Vacío/DE y otros módulos).

11. Definición de Hecho (DoD)

- DoD-1. Base completa + simetrías + truncación N (E2–E4).
- DoD-2. Error EFT explícito y propagado (E5).
- DoD-3. Esquema $\mathcal{R}$ y prueba de estabilidad + RG-invarianza operacional (E6–E7).
- DoD-4. MRD-EFT reproducible con auditoría completa (E14).
- DoD-5. Al menos un D_obs real (o público) traducido auditablemente; y, si aplica, integración con positividad (E12).

Apéndice A: plantilla de entrada (YAML/JSON)

Esquema mínimo (no ejecutable):

```
eft_module: domain_ISAAC: cutoff_Lambda: ... energy_window: [Emin, Emax]
validity_ratio: Emax/Lambda <= ... symmetries: [ ... ] power_counting: ...
operator_basis: order_N: ... operators: - name: O1 dim: d1 - name: O2 dim: d2
coefficients: scheme_R: MSbar | ... mu: ... c: {c1:..., c2:...} running: {beta_functions:
...} eft_error: model: (E/Lambda)^(N+1) normalization: ... matching: - from: EFT_high
to: EFT_low relations: ... anchors_Dobs: - kind: coef_bounds | sigma_fit dataset: ...
window: ... covariance: ... audit: hashes: ... versions: ... solver: name: ... tolerance: ...
seed: ...
```

Regla: sin bloque eft_error (E5), la corrida se considera inválida.

Apéndice B: glosario

- EFT. Teoría efectiva válida por debajo de Λ.
- Truncación N. Orden máximo retenido en el conteo de potencia.
- Esquema $\mathcal{R}$. Convención de renormalización/regularización.
- μ. Escala de renormalización; observables deben ser (casi) independientes de μ.
- Matching. Relación entre coeficientes al integrar modos pesados.
- Δ_{EFT}. Error por operadores omitidos.
- Rigidez. Tamaño/ambigüedad restante del conjunto compatible.

Compilación de Consistencia Operacional (OCC)

Módulo G0 — Materia Oscura efectiva (residual geométrico) y tests de no-arbitrariedad (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo define una materia oscura efectiva como término residual requerido para cerrar la dinámica en el Sector A (SGO) una vez fijadas contribuciones de materia tipo SM y del vacío/DE efectiva. La meta es estricta: evitar que "DM" sea un kernel libre. Aquí DM es una variable residual sometida a conservación, restricciones geométricas y tests de no-arbitrariedad compilables por el juez OCC.

Prioridad. Primero el por qué (qué se rompe si DM se deja libre), luego el qué (candados, anclajes, PASS/FAIL).

Contenido

0. Alcance y dependencias (Documento A + Vacío/DE + GR efectiva)

Depende de: Documento A (OCC universal), del módulo Vacío/DE (para separar la contribución del vacío) y de una formulación gravitatoria efectiva en el Sector A (GR + correcciones EFT si se declaran).

Cubre: definición y restricciones de un término residual G0 que se interpreta como DM efectiva en A; conexión con observables gravitacionales (lentes, dinámica galáctica, cúmulos, LSS) y con consistencia (conservación, causalidad, no-arbitrariedad).

No cubre: microcandidato específico (WIMP, axión, etc.). Es compatible con ellos, pero no los asume.

1. Motivación (por qué): por qué DM debe ser residual y no perilla

Si se permite una “materia oscura” como función libre en cada escala/entorno, el marco pierde falsabilidad: siempre se puede ajustar el potencial o el perfil para reproducir cualquier curva de rotación o lente. Eso sería un PASS trivial (maleable).

La disciplina operacional exige lo contrario: DM efectiva debe estar forzada por el cierre matemático y restringida por datos sin introducir libertad infinita. Por eso DM aparece aquí como residual geométrico: lo que queda cuando fijamos el resto.

- Qué se rompe si no. (i) cierre geométrico (Bianchi), (ii) conservación efectiva, (iii) comparabilidad entre sistemas, (iv) posibilidad de producir no-go fuertes.

2. Objeto operacional y frontend (geométrico + S-matrix/positividad cuando aplique)

Objeto operacional principal: el tensor residual o componente efectiva que entra en la ecuación gravitatoria en el Sector A.

Frontend principal: geométrico/cosmológico (ecuaciones efectivas covariantes).

Frontends auxiliares (cuando se busca no-arbitrariedad UV):

- S-matrix/positividad. Si G0 se interpreta como efecto de grados de libertad integrados de un UV local/analítico, entonces su huella en amplitudes efectivas debe respetar candados de causalidad/analiticidad/positividad en Ω_I.
- Sistemas abiertos. Si parte de G0 proviene de acoplo con un sector inaccesible, se exige consistencia operacional (no reinyección UV, y que lo inobservable no funcione como perilla).

3. Definición residual G0 (cierre covariante)

Se parte de una ecuación efectiva en el Sector A:

$$G_{\mu\nu} + \Lambda g_{\mu\nu} + \Delta_{\mu\nu}^{EFT} = (8\pi G/c^4) (T_{\mu\nu}^{SM} + T_{\mu\nu}^{vac} + T_{\mu\nu}^{G0}).$$

donde $\Delta_{\mu\nu}^{EFT}$ recoge correcciones covariantes declaradas (p.ej. R^2, $R_{\mu\nu}R^{\mu\nu}$, etc.) y T^{vac} viene del módulo Vacío/DE.

Definición operacional. Se define:

$$T_{\mu\nu}^{G0} \equiv (c^4/8\pi G) [G_{\mu\nu} + \Lambda g_{\mu\nu} + \Delta_{\mu\nu}^{EFT}] - (T_{\mu\nu}^{SM} + T_{\mu\nu}^{vac}).$$

Esto no postula microfísica: define el residual requerido por cierre, dado un conjunto de supuestos/convenciones ya declarados. El juez evalúa si existe una parametrización finita de T^{G0} compatible con candados y anclajes.

4. Candados inevitables del módulo (G0-1–G0-12)

Cada candado previene una forma de arbitrariedad o inconsistencia.

G0-1. Conservación covariante

Por qué. $\nabla^\mu(G_{\mu\nu}+\ldots)=0$ fuerza consistencia; violarlo rompe el cierre.

Qué. $\nabla^\mu T_{\mu\nu}^{G0}=0$ (o intercambio declarado con otra componente medible).

G0-2. Condición de energía efectiva (no negatividad operacional)

Por qué. Densidades negativas arbitrarias permiten hacks de lentes/rotación sin costo.

Qué. Se impone al menos $\rho_{G0}\geq 0$ en el marco cosmológico o en el marco comóvil donde se define el fluido efectivo (convención declarada).

G0-3. Parametrización finita y jerarquía

Por qué. Libertad funcional infinita → irrefutabilidad.

Qué. T^{G0} se parametriza con N parámetros finitos (p.ej. fluido efectivo con w0, wa; o perfil con pocos parámetros) y se reporta rigidez R.

G0-4. Recuperación del IR verificado

Por qué. El residual no puede destruir GR+SM donde ya funciona.

Qué. En regímenes probados (Solar System, pulsars, GW), la contribución efectiva de G0 se suprime o cumple constraints.

G0-5. No-degeneración con T^{vac} (separación obligatoria)

Por qué. Si G0 puede reabsorberse en el vacío o viceversa, el módulo pierde significado.

Qué. Convención fija de separación y test de estabilidad: variar convención dentro de rango permitido no debe cambiar veredicto.

G0-6. Regularidad y no singularidades ad hoc

Por qué. Perfiles singulares/ultra-cusp sin justificación son una forma de ajuste oculto.

Qué. Condiciones mínimas de regularidad (p.ej. finitud de masa efectiva en volumen, o condiciones de suavidad en $r=0$ y $r\to\infty$) declaradas por clase de sistema.

G0-7. Compatibilidad con lentes gravitacionales

Por qué. Cualquier cierre DM debe reproducir lentes y no sólo rotación.

Qué. El módulo exige consistencia simultánea con observables dinámicos y geométricos (misma T^{G0}).

G0-8. Consistencia multi-escala (galaxias ↔ cúmulos ↔ LSS)

Por qué. Ajustar un sistema a la vez es trivial; la prueba real es coherencia entre escalas.

Qué. Una parametrización debe funcionar en al menos dos escalas con los mismos parámetros globales o con jerarquía controlada declarada.

G0-9. No reinyección UV (ISAAC)

Por qué. Introducir estructura por encima de Λ_I vuelve el residual irrefutable.

Qué. Toda “micro-estructura” hipotética fuera de dominio se declara fuera del módulo; aquí sólo entra como restricciones indirectas.

G0-10. Test de dispersión/positividad (opcional pero recomendado)

Por qué. Si G0 proviene de un UV local/analítico, el EFT efectivo debe obedecer positividad/causalidad. Eso mata cierres ad hoc.

Qué. Cuando sea posible mapear a operadores EFT relevantes, se compilan bounds de positividad en Ω_I con protocolo IR-safe si aplica.

G0-11. Anomalías y simetrías (cuando se declare)

Por qué. Si se introduce una nueva simetría para justificar G0, debe ser consistente (matching).

Qué. Se activan candados de anomalía (C10) bajo declaración explícita.

G0-12. Rigidez obligatoria

Por qué. Explicar DM con demasiada libertad no explica.

Qué. Reportar R y/o d_{eff}; PASS fuerte si $R \lesssim 1$ o $d_{eff} \approx 0$.

5. Compilación OCC (constraints: local, cosmológico y de dispersión)

El conjunto factible del módulo se construye como intersección de sub-constraints:

- F_cons. Conservación + regularidad (G0-1, G0-6).
- F_IR. Recuperación del IR verificado + tests locales (G0-4).
- F_multi. Coherencia multi-escala (G0-7, G0-8).
- F_par. Parametrización finita + rigidez (G0-3, G0-12).
- F_UV. ISAAC/no reinyección UV (G0-9) y, si se usa, positividad (G0-10).

Se define F_G0 ≡ F_cons ∩ F_IR ∩ F_multi ∩ F_par ∩ F_UV, y se intersecta con D_obs (Sección 6) para obtener I_G0.

6. Anclajes D_obs (lentes, rotación, LSS, CMB, cúmulos) y traducción auditable

El módulo admite un conjunto cerrado de formatos de anclaje, para evitar selección a posteriori:

- D_obs-rot. Curvas de rotación (o velocidad de dispersión) con errores → constraints sobre potencial efectivo.
- D_obs-lens. Perfil de convergencia/deflexión o masa proyectada → constraints sobre componente geométrica.
- D_obs-cluster. Masa por rayos X/SZ + lentes → consistencia multi-observación.
- D_obs-LSS. Espectro de potencia/momento y crecimiento → constraints globales.
- D_obs-CMB. Parámetros comprimidos (Ω_m, σ_8, etc.) bajo convención declarada.

Regla de traducción: cada dataset debe venir con (i) selección explícita, (ii) pipeline de conversión a constraints, (iii) ventanas/escales usadas y (iv) auditoría.

7. MRD-G0 (demo mínimo reproducible)

MRD-G0 fija un caso mínimo reproducible que demuestra cierre residual + consistencia con dos observables.

- Entrada. Fondo cosmológico FLRW (Ω_m, Ω_Λ) + una clase de perfil galáctico con N parámetros finitos.
- Anclajes mínimos. (i) una curva de rotación típica (o mock auditado), (ii) un constraint de lentes/massa proyectada, y (iii) un anclaje cosmológico comprimido.
- Tests. G0-1…G0-12 en el dominio ISAAC declarado; reporte de rigidez.
- Salida. PASS/FAIL + diagnóstico + bloque de auditoría completo.

8. Salidas PASS/FAIL y predicciones discriminantes/no-go

$I_{G0} \equiv F_{G0} \cap D_{obs}$.

- FAIL. $I_{G0} = \varnothing$: no hay cierre residual DM efectiva compatible con conservación+IR+multi-escala+anclajes dentro de ISAAC.
- PASS. $I_{G0} \neq \varnothing$.
- PASS fuerte. Isla rígida ($R \lesssim 1$) o discreción local.

Predicciones/no-go típicos:

- Relación fija entre potencial dinámico (rotación) y potencial geométrico (lentes) para la misma T^{G0} (test fuerte).
- No-go: cualquier parametrización que requiera densidad efectiva negativa o no conservación para ajustar lentes+rotación → FAIL (G0-1/G0-2).
- No-go: perfiles que sólo funcionan para una galaxia y fallan en cúmulos/LSS sin jerarquía controlada → FAIL (G0-8).
- Si se activa positividad: exclusión de signos/coeficientes EFT que produzcan superluminalidad o violen causalidad (G0-10).

9. Loopholes/artefactos frecuentes y cómo evitarlos

- Selección de sistemas. Escoger sólo galaxias “buenas” produce sesgo; el módulo exige criterio de selección auditado.
- Degeneracia bariones-DM. El anclaje debe separar masa bariónica con incertidumbre declarada; de lo contrario G0 se vuelve perilla.
- Screening implícito. Si se invoca supresión local, debe ser mecanismo explícito, no “asumido”.
- Dependencia de convención. Cambiar la separación vac/DM no debe cambiar el veredicto; si cambia, el módulo no está cerrado.

10. Definición de Hecho (DoD) para publicar este módulo

- DoD-1 (Definición residual). Ecuación efectiva + definición de T^{G0} (Sección 3) con convención fija.
- DoD-2 (Candados). Implementación computable de G0-1…G0-12 con tests negativos.
- DoD-3 (MRD). MRD-G0 reproducible con auditoría completa.
- DoD-4 (Anclajes reales). Al menos dos clases de observables (rotación + lentes, o cúmulos + LSS) compiladas auditablemente.
- DoD-5 (Predicción). Una predicción/no-go discriminante multi-escala o un bound EFT (si se activa positividad).

Apéndice A: glosario local

- G0. Residual efectivo interpretado como DM efectiva en el Sector A.
- T^{G0}_{μν}. Tensor residual que cierra la ecuación gravitatoria efectiva.
- No-arbitrariedad. Prohibición de kernels/libertad infinita; se impone con parametrización finita + rigidez + tests multi-escala.
- D_obs. Anclajes observacionales (rotación, lentes, cúmulos, LSS, CMB).
- ISAAC/Λ_I. Dominio/corte operacional; prohíbe reinyección UV.
- Positividad. Candados de causalidad/analiticidad para EFT si se usa frontend S-matrix.

Definiciones canónicas de OCC /SGO/SIA/ISAAC están en el Documento A.

Compilación de Consistencia Operacional (OCC)

Módulo Vacío/DE — Entorno efectivo, fricción del vacío y energía oscura (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo fija una definición operacional del vacío como entorno efectivo en el Sector A (SGO) y del término cosmológico/energía oscura como contribución efectiva, evitando que “el vacío” se convierta en una perilla ilimitada. El módulo se apoya en el frontend de sistemas abiertos (SK) para separar causalidad, ruido y disipación.

Prioridad. Primero el por qué (qué se rompe sin el módulo), luego el qué (candados, anclajes, PASS/FAIL).

Contenido

0. Alcance y dependencias (Documento A + Módulo SK)

Depende de: Documento A (metodología universal) y del Módulo SK (no-equilibrio) para formalizar el vacío como entorno efectivo. Si se declara equilibrio térmico, también depende del candado KMS (C9).

Cubre: (i) parametrización mínima del vacío como entorno que induce ruido/disipación/decoherencia en el Sector A; (ii) definición operacional de energía oscura (DE) como contribución efectiva compatible con conservación y causalidad; (iii) tests que impiden libertad infinita.

No cubre: microontología del vacío ni “esencia” metafísica. Por diseño, sólo se habla de efectos operacionales en A y de restricciones que cualquier microexplicación debe respetar.

1. Motivación (por qué): por qué el vacío debe tratarse como entorno

Sin un módulo explícito, “vacío” suele usarse de dos formas que destruyen rigor:

- Cajón de sastre. Se invoca el vacío para absorber cualquier discrepancia sin pagar costo, introduciendo funciones libres (w(z), kernels, etc.).
- Confusión de niveles. Se mezclan contribuciones de renormalización, efectos de estado (térmico/no-equilibrio) y términos cosmológicos como si fueran lo mismo.

Operacionalmente, el vacío es el entorno mínimo presente incluso cuando no preparamos un baño explícito. Si el entorno induce fluctuaciones y respuesta, la descripción efectiva debe separar (i) respuesta causal, (ii) ruido, y (iii) restricciones de equilibrio cuando corresponda. Ese es exactamente el contenido de SK.

El objetivo del módulo es convertir “vacío” en un objeto medible y restringible, no en narrativa.

2. Objeto operacional y frontend (SK + térmico opcional)

El objeto operacional es doble:

- (i) Sector A local. Kernels efectivos (D_R, N) que describen la respuesta y el ruido inducidos por el vacío/entorno sobre grados de libertad de A.
- (ii) Sector A cosmológico. Un término efectivo de energía-momento del vacío, $T^{vac}_{\mu\nu}$, cuyo efecto se observa en expansión, lentes y crecimiento.

Frontend principal: sistemas abiertos (SK). Si se declara que el vacío está en equilibrio a temperatura efectiva, se activa además el frontend térmico (KMS).

3. Definiciones operacionales: vacío, fricción del vacío y DE efectiva

3.1 Vacío como entorno efectivo

Definimos “vacío” operacionalmente como el entorno mínimo que permanece cuando fijamos el laboratorio lo más aislado posible. Su contenido no se describe por micrograds de libertad accesibles, sino por su huella en A: ruido, disipación y decoherencia.

En SK, esto se codifica por kernels (D_R, N) en la acción efectiva (ver Módulo SK).

3.2 Fricción del vacío (damping inducido)

Definimos “fricción del vacío” como la contribución disipativa a la ecuación efectiva en A inducida por el entorno, capturada por el kernel retardado D_R. No es una fuerza “mística”; es la parte de la respuesta que produce pérdida de coherencia/energía del subsistema (cuando existe canal de disipación permitido).

3.3 Energía oscura como contribución efectiva

Definimos energía oscura efectiva como la parte del tensor $T^{vac}_{\mu\nu}$ que, en un fondo cosmológico, se comporta aproximadamente como componente de presión negativa y produce aceleración. En el límite más simple, $T^{vac}_{\mu\nu} \approx -\rho_\Lambda g_{\mu\nu}$, equivalente a Λ efectivo.

Importante: el módulo no permite parametrizaciones con libertad funcional infinita sin anclaje. Toda extensión (p.ej. w(z)) debe venir con penalización operacional (número finito de parámetros + pruebas de rigidez).

4. Candados inevitables del módulo (V0–V9)

Cada candado se justifica por la falla operacional que previene.

V0. Cierre operacional (ISAAC) y anti-perillas

Por qué. Sin cierre, el vacío se vuelve perilla ilimitada.

Qué. Parametrización finita; soporte espectral limitado por Λ_I (o declarado) y auditoría obligatoria.

V1. Causalidad efectiva (SK1 heredado)

Por qué. Sin causalidad, la respuesta del vacío sería avanzada.

Qué. Kernel retardado D_R con soporte $t \geq t'$.

V2. Positividad del ruido (SK3 heredado)

Por qué. Ruido no-PSD produce probabilidades negativas.

Qué. N PSD (en tiempo o frecuencia, con discretización convergente).

V3. Canal físico (CPTP, SK4 heredado)

Por qué. La dinámica reducida debe ser física incluso con ancillas.

Qué. (D_R, N) inducen evolución CPTP (criterio declarado para Markov/no-Markov).

V4. Conservación covariante en cosmología

Por qué. En GR efectiva, $\nabla^\mu G_{\mu\nu}=0$ impone consistencia; un $T^{vac}_{\mu\nu}$ no conservado rompe el cierre geométrico.

Qué. $\nabla^\mu T^{vac}_{\mu\nu} = 0$ en el Sector A (o declaración explícita de intercambio con otra componente y cómo se mide).

V5. Límite IR correcto

Por qué. La DE no puede destruir el éxito del IR observado (ΛCDM/GR en regímenes verificados).

Qué. En el régimen donde GR+SM se verifica, el módulo recupera ese límite con correcciones controladas.

V6. Compatibilidad con tests locales de gravedad

Por qué. Un término efectivo que modifica dinámica a escalas solares/púlsares sin control es descartado por datos.

Qué. Si el módulo induce modificaciones locales, debe incluir mecanismo de supresión o demostrar supresión en regímenes con constraints.

V7. Equilibrio (KMS) sólo si se declara

Por qué. Declarar equilibrio impone relación ruido/disipación; si no se cumple, es arbitrariedad.

Qué. Si se declara β, se verifica FDT/KMS en la ventana donde aplica (tolerancias auditables).

V8. Separación renormalización vs estado

Por qué. Confundir contraterminos con estado físico produce dobles conteos y “ajustes” invisibles.

Qué. El módulo fija convención: qué parte se absorbe en Λ renormalizado y qué parte es estado/entorno (SK).

V9. Rigidez obligatoria

Por qué. “Explicación” que permite demasiada libertad no explica.

Qué. Reportar rigidez (R) sobre parámetros del vacío/DE; PASS fuerte si $R \lesssim 1$.

5. Compilación OCC (cómo se convierte a constraints)

El módulo compila dos conjuntos de restricciones:

- F_{local}. Constraints SK (causalidad/PSD/CPTP) sobre (D_R,N) en la banda operacional.
- F_{cosmo}. Constraints de conservación, límite IR, y compatibilidad con tests locales/grandes escalas sobre $T^{vac}_{\mu\nu}$ o parametrización equivalente.

El conjunto factible del módulo es $F_V \equiv F_{local} \cap F_{cosmo}$ (en el dominio ISAAC declarado). Se intersecta con anclajes D_{obs} (Sección 6) para producir I_V.

6. Anclajes D_obs (cosmología y laboratorio) y traducción auditable

Para que el módulo sea cerrado, D_obs se especifica por formatos admitidos y traducción auditable.

- D_obs-cosmo (geométrico). $H(z)$, distancias (BAO/SNe), CMB (parámetros comprimidos) → restricción sobre ρ_Λ o $w(z)$ parametrizado.
- D_obs-growth. $f\sigma 8(z)$ / crecimiento de estructura → restricción sobre combinación DE+gravedad efectiva.
- D_obs-local. constraints solares/púlsares/ondas gravitacionales (post-Newtoniano, c_{gw}) → bounds sobre modificaciones locales.
- D_obs-lab. límites en ruido/damping inducidos por el vacío en sistemas cuánticos (si se usa): espectros de ruido, tasas Γ.

Regla: toda fuente de datos debe especificar (i) versión/conjunto, (ii) convención, (iii) ventana de uso, (iv) mapeo a restricciones numéricas.

7. MRD-V (demo mínimo reproducible)

MRD-V fija una demostración mínima para ejecutar el módulo sin ambigüedad.

- Parte local (SK). Oscilador + baño con densidad espectral $J(\omega)$ y corte Λ_I: verificar V1–V3 y (si se declara) V7.
- Parte cosmológica. Fondo FLRW con componente Λ efectiva: verificar V4–V6 con un conjunto mínimo de anclajes y reportar rigidez de ρ_Λ.
- Auditoría. Hashes, versiones, tolerancias y scripts que reproduzcan PASS/FAIL.

8. Salidas PASS/FAIL y predicciones discriminantes

El conjunto superviviente del módulo es $I_V \equiv F_V \cap D_{obs}$. Se reporta:

- FAIL. $I_V = \emptyset$.
- PASS. $I_V \neq \emptyset$.
- PASS fuerte. Rigidez $R \lesssim 1$ (o discreción local).

Predicciones/no-go típicos:

- Bound sobre relación ruido/disipación en banda ω (si aplica V7).
- Bound sobre evolución permitida de $w(z)$ con N parámetros y rigidez.
- No-go: cierre que requiera violar conservación covariante o generar modificación local grande sin supresión → FAIL.

9. Loopholes/artefactos frecuentes y cómo evitarlos

- Funcionalidad infinita en $w(z)$. Prohibida; usar N parámetros y rigidez (V0/V9).
- Degeneracias cosmológicas. Declarar qué se fija (curvatura, neutrinos, etc.) y reportar rigidez.
- Confusión Λ renormalizado. Fijar convención (V8) para evitar doble conteo.
- Transitorios. Aplicar KMS sólo en equilibrio; si no, no activar V7.

10. Definición de Hecho (DoD) para publicar este módulo

- DoD-1. Definiciones operacionales (Sección 3) + convención V8.
- DoD-2. Implementación de V0–V9 con tests negativos.
- DoD-3. MRD-V reproducible con auditoría completa.
- DoD-4. Al menos un anclaje cosmológico real traducido auditablemente.
- DoD-5. Predicción/no-go discriminante.

Apéndice A: glosario local

- Vacío (operacional). Entorno efectivo mínimo definido por su huella en A: ruido/disipación/decoherencia.
- Fricción del vacío. Parte disipativa codificada en D_R.
- DE. Energía oscura efectiva: contribución al tensor energía-momento compatible con conservación y anclajes.
- D_R, N. Kernels SK de respuesta y ruido.
- KMS/FDT. Candados de equilibrio cuando se declaran.
- ISAAC/Λ_I. Dominio/corte operacional que prohíbe reinyección UV.

Definiciones canónicas de OCC /SGO/SIA/ISAAC están en el Documento A.

Compilación de Consistencia Operacional (OCC)

Módulo Gravedad Efectiva — IR, PPN, ondas gravitacionales y consistencia cosmológica (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo cierra el “frontend gravedad” del Sector A: cualquier dinámica gravitatoria efectiva (GR + correcciones) debe pasar simultáneamente tests locales (PPN/Solar System), propagación y polarizaciones de ondas gravitacionales (GW) y consistencia cosmológica (H(z), crecimiento, lentes). El objetivo es impedir que una modificación de gravedad pase un test aislado ajustando perillas en otro régimen.

Prioridad. Primero el por qué (qué se rompe si se testea por partes), luego el qué (candados, anclajes, PASS/FAIL).

Contenido

0. Alcance y dependencias

Depende de: Documento A (OCC). Se integra naturalmente con Vacío/DE y G0 (porque ambos entran como fuentes/efectivos en ecuaciones gravitatorias). Puede usar el módulo EFT/Renormalización para parametrizar correcciones (R^2, etc.).

Cubre: consistencia operacional de gravedad efectiva desde IR local hasta cosmología y GW.

No cubre: seleccionar una teoría específica (Horndeski, f(R), etc.). El módulo compila la clase declarada.

1. Motivación (por qué): por qué gravedad efectiva debe ser multi-régimen

La gravedad está fuertemente testeada en regímenes distintos: Solar System, púlsares binarios, ondas gravitacionales y cosmología. Una propuesta que sólo pasa uno de ellos es maleable: puede esconderse detrás de screening, elecciones de gauge o términos EFT.

Sin un módulo multi-régimen ocurren tres fallas:

- PASS parciales. Se declara éxito en cosmología ignorando PPN, o al revés.
- Screening ad hoc. Se invoca supresión local sin mecanismo explícito.
- Dependencia de calibración. Ajustes de parámetros distintos en cada régimen destruyen unificación.

Este módulo obliga a que el mismo conjunto de parámetros (o jerarquía controlada) satisfaga simultáneamente los anclajes.

2. Objeto operacional: acción efectiva, ecuaciones y dominio Ω_I

Objeto: una acción/ecuación efectiva gravitatoria en el Sector A:

$$S_{eff} = \int d^4x \sqrt{-g} \, [(M_P^2/2) R - \Lambda + \square_{corr}(g,\partial g;\theta)] + S_{matter}.$$

donde $\square_{corr}$ representa correcciones (EFT, campos extra, no-minimal acoplos) parametrizadas por θ finito en Ω_I.

El módulo exige declarar el dominio Ω_I: escalas de curvatura/energía donde la aproximación efectiva es válida.

3. Definiciones: PPN, EFT de gravedad, c_T, polarizaciones, screening

3.1 PPN

El formalismo PPN parametriza desviaciones de GR en el régimen débil/estático. El módulo usa PPN como interfaz operacional con Solar System.

3.2 c_T y GW

c_T es la velocidad de propagación del modo tensorial. Observaciones GW imponen restricciones fuertes cuando se declaran condiciones de coincidencia con c.

3.3 Polarizaciones

Teorías modificadas pueden introducir modos adicionales (escalares/vectoriales). El módulo exige contabilidad y constraints.

3.4 Screening

Mecanismos (Vainshtein, chameleon, etc.) que suprimen efectos en entornos densos. El módulo prohíbe ‘screening por palabra’: debe ser dinámico y cuantificado.

4. Candados inevitables del módulo (GE1–GE16) con ‘por qué → qué’

GE1. Covariancia y conservación

Por qué. Bianchi impone consistencia.

Qué. $\nabla^\mu T^{tot}_{\mu\nu}=0$ (o intercambios declarados); ecuaciones bien definidas.

GE2. Dominio Ω_I y truncación (si EFT)

Por qué. Usar la teoría fuera de dominio produce conclusiones falsas.

Qué. Declarar escalas de curvatura/energía y truncación finita (con error EFT) si aplica.

GE3. Ausencia de fantasmas/instabilidades (al menos en Ω_I)

Por qué. Grados de libertad con energía negativa destruyen el marco.

Qué. Condiciones de estabilidad lineal (signos cinéticos) en el dominio considerado.

GE4. Límite GR recuperado

Por qué. GR funciona en múltiples tests.

Qué. Existe un límite/sector donde parámetros → 0 recuperan GR + Λ + materia estándar.

GE5. PPN consistente

Por qué. Solar System es un anclaje duro.

Qué. Predicción explícita de parámetros PPN ($\gamma,\beta,\ldots$) dentro de límites observacionales en la ventana.

GE6. Pulsars / radiación gravitatoria

Por qué. Binarias testean pérdidas de energía y modos extra.

Qué. Calcular correcciones a $\dot{P}_b$ o análogos; cumplir límites.

GE7. Ondas gravitacionales: velocidad tensorial

Por qué. $c_T \neq c$ rompe consistencia con multi-mensajero en muchos escenarios.

Qué. En teorías donde aplica, imponer $|c_T/c - 1| \leq \varepsilon_T$ (ε_T declarado por dataset).

GE8. Ondas gravitacionales: polarizaciones extra

Por qué. Modos extra son observables.

Qué. Contabilidad de modos y límites en amplitud/fracción de energía en detectores.

GE9. Lentes gravitacionales

Por qué. Lentes mide potencial geométrico, complementario a dinámica.

Qué. Predicciones para deflexión/convergencia consistentes con datos en escalas declaradas.

GE10. Crecimiento de estructura

Por qué. Cosmología no es solo $H(z)$; crecimiento testea fuerza efectiva.

Qué. Predicción de $f\sigma 8$ o equivalente; compatibilidad con anclajes.

GE11. Consistencia con Vacío/DE y G0

Por qué. Separar “gravedad modificada” de “fuentes efectivas” evita doble conteo.

Qué. Frontera explícita: qué se atribuye a □_corr vs a T^{vac} y T^{G0}.

GE12. Screening explícito (si se usa)

Por qué. Screening sin ecuaciones es perilla.

Qué. Derivar régimen screened y su radio/umbral; demostrar continuidad entre regímenes.

GE13. No reinyección UV (ISAAC)

Por qué. Hacer que la teoría dependa de microdetalles fuera de dominio destruye falsabilidad.

Qué. Todo efecto UV se encapsula en parámetros finitos; nada de funciones arbitrarias de curvatura sin truncación.

GE14. Robustez multi-sistema

Por qué. Ajustar un sistema y fallar otro es maleable.

Qué. Los mismos parámetros deben funcionar en al menos dos regímenes (PPN + cosmo, o pulsars + GW, etc.).

GE15. Auditabilidad numérica

Por qué. Soluciones cosmológicas y perturbaciones dependen de solvers.

Qué. Seeds/tolerancias/versiones; pruebas de convergencia y reproducibilidad.

GE16. Rigidez obligatoria

Por qué. Si hay demasiados parámetros, no hay predicción.

Qué. Reportar rigidez R/d_{eff}; PASS fuerte si región superviviente es pequeña/discreta.

5. Compilación OCC : intersección multi-régimen F ∩ D_obs

El módulo construye F_GE como intersección de GE1–GE16 y lo intersecta con D_obs multi-régimen:

- D_obs-local: PPN/Solar System.
- D_obs-pulsar: binarios relativistas.
- D_obs-GW: velocidad/polarizaciones/damping.
- D_obs-cosmo: H(z), lentes, crecimiento.

Salida: I_GE ≡ F_GE ∩ D_obs. PASS sólo si existe región común; no se admiten ‘parámetros distintos por régimen’ salvo jerarquía controlada declarada.

6. Anclajes D_obs: Solar System/PPN, pulsars, GW, lentes, cosmo

Formatos de anclaje admitidos:

- D_obs-PPN. Intervalos para γ, β, ...; ventanas de validez.
- D_obs-pulsar. Límites sobre dipole radiation/energía extra; $\dot{P}_b$.
- D_obs-GW. Bounds sobre c_T, polarizaciones, fricción efectiva (si se parametriza).
- D_obs-lens. Perfiles de convergencia/deflexión en escalas declaradas.
- D_obs-growth. fσ8, μ(k,z), Σ(k,z) o parametrización equivalente.

Regla: cada anclaje se traduce a desigualdades sobre θ con pipeline auditable.

7. MRD-GE (demo mínimo reproducible)

MRD-GE muestra una clase mínima de gravedad efectiva y su chequeo multi-régimen:

- Clase. GR + un término correctivo (p.ej. α R^2) o un escalar acoplado mínimamente con un parámetro.
- Local. Derivar corrección PPN (γ,β) a primer orden en α.
- GW. Derivar c_T y/o modos extra en el mismo modelo.
- Cosmo. Resolver H(z) y crecimiento lineal en una aproximación simple.
- Salida. PASS/FAIL + rigidez + auditoría.

8. PASS/FAIL, PASS fuerte y no-go

$I_{GE} \equiv F_{GE} \cap D_{obs}$.

- FAIL. Si no existe región común o si se requiere screening ad hoc no derivado.
- PASS. Existe región θ que satisface simultáneamente PPN+pulsars+GW+cosmo dentro de Ω_I.
- PASS fuerte. Región rígida y predicciones claras (p.ej. desviación pequeña pero no nula en crecimiento o lentes).

No-go típico: una modificación que mejora cosmología pero exige $\gamma \neq 1$ fuera de límites PPN, o introduce dipole radiation incompatible con púlsares.

9. Loopholes/artefactos frecuentes

- Elegir datasets por conveniencia. El módulo exige lista declarada.
- Screening no cuantificado. Debe derivarse y aplicarse con umbrales; no basta nombrarlo.
- Confundir fuente con gravedad. Mover discrepancias a DM/DE sin declarar frontera (GE11) produce doble conteo.
- Usar parametrizaciones diferentes por régimen. Prohibido salvo jerarquía controlada auditada.

10. Definición de Hecho (DoD)

- DoD-1. Clase θ y dominio Ω_I declarados + estabilidad (GE2–GE3).
- DoD-2. Predicción PPN explícita y cumplimiento de límites (GE5).
- DoD-3. Chequeo GW (c_T y/o polarizaciones) + pulsars si aplica (GE6–GE8).
- DoD-4. Chequeo cosmológico (H(z)+lentes+crecimiento) con pipeline auditable (GE9–GE10, GE15).
- DoD-5. Reporte de rigidez y región común multi-régimen (GE14–GE16).

Apéndice A: glosario

- PPN. Parametrización post-Newtoniana para régimen débil.
- c_T. Velocidad del modo tensorial de GW.
- Screening. Supresión dinámica de efectos modificados en entornos densos.
- μ, Σ. Funciones efectivas que parametrizan modificación de Poisson y de lentes.
- Ω_I. Dominio ISAAC: validez operacional.
- Rigidez. Medida de ambigüedad restante.

Compilación de Consistencia Operacional (OCC)

Módulo Cosmología Operacional — puente Local ↔ Cosmológico (perturbaciones, coarse-graining y D_obs) (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo cierra el salto entre formulaciones locales (EFT, SK, gravedad efectiva) y observables cosmológicos (expansión, crecimiento, lentes, BAO, CMB). Define un procedimiento auditado para: (i) elegir el coarse-graining, (ii) mapear parámetros/kernels a cantidades cosmológicas, y (iii) exportar D_obs-cosmo consumible por el compilador.

Prioridad. Primero el por qué: sin puente local↔cosmo, “vacío/DE como entorno” o “DM efectiva” quedan como narrativa. Luego el qué: reglas y candados que evitan arbitrariedad (PCD/PCN).

Contenido

0. Alcance y dependencias

Depende de: Documento A (PCD/PCN), Observabilidad/ISAAC (ventanas y resolución), Gravedad Efectiva (IR/PPN/GW), EFT/Renormalización (truncación y error), y opcionalmente SK (si se modela entorno/ruido).

Cubre: reglas para mapear teoría→(fondo FRW + perturbaciones)→observables cosmológicos→D_obs.

No cubre: elección de dataset o análisis estadístico detallado: sólo exige que D_obs incluya covarianzas y ventanas en formato canónico.

1. Motivación (por qué): por qué el puente local↔cosmo es inevitable

Una teoría puede ser consistente localmente y aun así fallar cosmológicamente por dos razones:

- Escala y coarse-graining. Los observables cosmológicos son promedios/perturbaciones sobre volúmenes enormes; si no se define qué se integra fuera, aparece arbitrariedad.
- Degeneraciones físicas. Vacío/DE, DM efectiva y modificación de gravedad pueden producir firmas similares; sin reglas de separación, se reintroduce maleabilidad.

Este módulo impone que toda afirmación cosmológica venga de un pipeline: (teoría local) → (coarse-graining declarado) → (fondo + perturbaciones) → (observable) → (D_obs con ventana/covarianza).

2. Objetos cosmológicos mínimos y qué cuenta como observable

Objetos mínimos:

- Fondo FRW. H(a) o H(z), parámetros de densidad efectivos, curvatura espacial.
- Perturbaciones. Potenciales (Φ,Ψ) o equivalentes gauge-invariantes, crecimiento fσ8, espectros P(k), transfer functions.
- Geometría observacional. Distancias D_A(z), D_L(z), BAO-scale, lente (κ, γ).

Observables consumibles deben ser gauge-invariantes (cosmología de perturbaciones) y estar ligados a protocolos/encuestas: cada D_obs debe declarar ventana en z, k, ℓ y covarianzas.

3. Coarse-graining canónico: definición, escala y error

Definición. Se elige una escala de coarse-graining Λ_CG (o k_CG) que separa modos “largos” (cosmológicos) de modos “cortos” (integrados fuera).

Por qué. Sin Λ_CG, ‘vacío como entorno’ o ‘DM residual’ puede absorber cualquier discrepancia.

Qué exige.

- COS-CG1. Declarar Λ_CG y su motivación instrumental (ventana) y teórica (validez).
- COS-CG2. Propagar un error de coarse-graining (sensibilidad del observable al variar Λ_CG dentro de un rango razonable).
- COS-CG3. Prohibición de tuning: no se permite elegir Λ_CG para forzar ajuste a un dataset específico.

4. Puente dinámico: de acción/ecuaciones locales a fondo FRW

Entrada: acción/ecuaciones efectivas (GR+EFT+correcciones) con separación A/B ya fijada (Vacío/DE, G0, etc.).

Salida: ecuaciones de fondo para H(a) con términos efectivos auditados.

- Regla. Si se postula w(z) o parámetros de fondo, deben derivarse de un Lagrangiano/cierre efectivo o declararse como parametrización EFT con orden y error (no como función libre).
- Conservación. Se exige consistencia con $\nabla^\mu T_{\mu\nu}=0$ (o la regla de no-conservación explícita si hay intercambio permitido y medible).

5. Puente de perturbaciones: de kernels/operadores a observables

Aquí se traduce el contenido micro/efectivo a predicciones en perturbaciones:

- EFT-of-DE / EFT-of-LSS (operacional). Si se usa parametrización EFT, se fijan operadores permitidos, orden, y estabilidad (PCD/PCN).
- Gravedad modificada. Se compila a funciones gauge-invariantes (p.ej. $\mu(k,z)$, $\gamma(k,z)$) sólo si vienen de operadores finitos con error. Funciones arbitrarias están prohibidas.
- Entornos SK. Si se modela ruido/fricción cosmológicos, se debe definir partición S/E y mostrar cómo afecta observables (p.ej. potencia, no-Gaussianidad) sin violar CPTP/positividad.

Regla ISAAC: no se permiten inferencias de modos fuera de la ventana experimental (k o ℓ) aunque la teoría tenga estructura allí.

6. Exportación D_obs-cosmo: formato, covarianzas y ventanas

Toda comparación cosmológica en OCC requiere D_obs canónico. Contiene:

- Vector de datos. y_i con definición operacional.
- Ventanas. W(z), W(k), W(ℓ) según el observable; rango y binning.
- Covarianza. Matriz C_ij (o aproximación declarada).
- Sistemáticos. Parametrización y priors permitidos (y su estatus: A vs B).
- Pipeline. Procedimiento reproducible de extracción (no necesariamente código completo, pero sí especificación + versión).

Si falta cualquiera de estos, el juez emite NO-EVAL para ese frontend cosmológico.

7. Candados inevitables del módulo (COS1–COS22) con ‘por qué → qué’

COS1. Declaración de observables cosmológicos y su invariancia

Por qué. Sin invariancia gauge, la predicción depende de coordenadas.

Qué. Definir observables gauge-invariantes (Φ,Ψ, $f\sigma 8$, κ, etc.) y su mapa a medición.

COS2. D_obs-cosmo con ventana y covarianza

Por qué. Sin ventanas/covarianza, el ajuste es maleable.

Qué. Exportar D_obs con $W(z/k/\ell)$ y C_ij.

COS3. Coarse-graining explícito (Λ_CG)

Por qué. Sin Λ_CG, el ‘entorno’ puede absorber cualquier discrepancia.

Qué. Declarar Λ_CG y propagar error por sensibilidad.

COS4. Prohibición de funciones libres (μ(k,z), γ(k,z), w(z))

Por qué. Funciones libres son perillas infinitas.

Qué. Sólo se permiten si provienen de un conjunto finito de operadores con truncación y error.

COS5. Conservación/transferencia explícita

Por qué. Violaciones implícitas rompen consistencia.

Qué. Exigir $\nabla^\mu T_{\mu\nu}=0$ o declarar e instrumentar intercambio medible.

COS6. Separación Vacío/DM/Gravedad con regla de equivalencia

Por qué. Mover términos entre componentes sin regla reintroduce arbitrariedad.

Qué. Declarar convención y exigir que predicciones observables sean invariantes bajo re-etiquetado equivalente.

COS7. Matching local↔cosmo

Por qué. Una teoría que pasa local pero falla cosmo debe detectarse.

Qué. Exigir consistencia entre parámetros locales (PPN/GW) y parámetros cosmológicos en Ω_I.

COS8. IR-safety y no divergencias espurias

Por qué. Tratamiento IR incorrecto puede fabricar señales.

Qué. Declarar tratamiento IR y demostrar estabilidad frente a corte IR razonable.

COS9. No inferencia fuera de ventana (ISAAC)

Por qué. No se puede ‘usar’ información no accesible.

Qué. Prohibir ajuste usando rangos k/ℓ no observados; sólo proyecciones integradas con ventanas.

COS10. Truncación EFT y error

Por qué. Sin error, la EFT es perilla.

Qué. Declarar orden, Λ, error; aplicar PCD/PCN.

COS11. Binning y elección de prior

Por qué. Priors pueden dominar el resultado.

Qué. Declarar priors; sólo priors instrumentales/teóricos justificados son permitidos.

COS12. Degeneraciones y diagnóstico

Por qué. Si dos mecanismos son degenerados, el juez no puede 'elegir narrativa'.

Qué. Emitir diagnóstico de degeneración y requerir observables que la rompan o degradar rigidez.

COS13. Compatibilidad con datos independientes

Por qué. Ajustar un dataset y fallar en otro es maleabilidad.

Qué. Test mínimo: al menos dos clases de observables (geometría + crecimiento) cuando se pretende DE/DM.

COS14. Lente gravitacional como test cruzado

Por qué. Lente mide potenciales; rompe degeneraciones.

Qué. Requerir consistencia entre lente y dinámica cuando se alega DM/GR-mod.

COS15. Consistencia con CMB/BAO (si se usa)

Por qué. CMB/BAO imponen fuertes constraints de geometría.

Qué. Si se invocan, deben entrar como D_obs con ventanas y covarianza.

COS16. Ruido/entorno SK cosmológico (si aplica)

Por qué. Sin partición S/E, hablar de entorno es retórico.

Qué. Definir S/E, kernel y efecto sobre observables; cumplir CPTP/positividad.

COS17. No-Gaussianidad y causalidad (si se modela)

Por qué. Ciertos kernels generan firmas no físicas.

Qué. Declarar predicciones de NG y verificar compatibilidad con causalidad/positividad.

COS18. Robustez numérica (PCN)

Por qué. Resultados cosmológicos son sensibles a integradores.

Qué. Exigir convergencia bajo refinamiento y sensibilidad a tolerancias.

COS19. Sensibilidad a Λ_CG y a ventana

Por qué. El veredicto no debe depender de elección caprichosa.

Qué. Reportar variación de veredicto bajo cambios razonables; degradar rigidez si cambia.

COS20. Taxonomía de veredictos (incluye NO-EVAL)

Por qué. Para evitar injusticia metodológica.

Qué. PASS/FAIL/PASS fuerte/NO-EVAL según completitud del pipeline.

COS21. Certificados (cuando convexo)

Por qué. Diagnósticos fuertes aumentan credibilidad.

Qué. Si el problema es convexo, emitir witness/dual certificate; si no, reportar evidencia numérica reproducible.

COS22. Rigidez cosmológica

Por qué. Una teoría que ajusta todo con perillas es maleable.

Qué. Reportar número de parámetros efectivos vs número de clases de observables; exigir rigidez mínima para PASS fuerte.

8. Interfaz con Vacío/DE, G0, Gravedad Efectiva, SK y EFT

- Con Vacío/DE. Impone que cualquier $w(z)$ o término efectivo venga de truncación finita o de un puente SK definido.
- Con G0 (DM efectiva). Obliga test multi-escala (galaxias↔cúmulos↔LSS) y consistencia con lente.
- Con Gravedad Efectiva. Exige consistencia local (PPN/GW) con parámetros cosmológicos (μ,γ).
- Con SK. Si se usa entorno, kernel debe ser CPTP/positivo y su efecto en espectros debe estar dentro de Ω_I.
- Con EFT. Define qué operadores son permitidos y cómo se propaga error EFT a observables.

9. Plantillas canónicas (YAML/JSON)

Plantilla mínima (no ejecutable):

cosmology_operational: domain_ISAAC: {Omega_I: ...} coarse_graining: Lambda_CG: ... sensitivity_range: ... background: equations: ... parameters: ... conservation: ... perturbations: predictions: {Phi_Psi: ..., mu_kz: ..., gamma_kz: ...} allowed_operators: ... EFT_order_error: ... D_obs_cosmo: datasets: - {name: ..., y: ..., windows: ..., covariance: ...} diagnostics: degeneracies: ... certificates: ... audit: versions: ... hashes: ... seeds: ...

10. MRD-COS (demo mínimo reproducible)

MRD-COS recomendado (nivel 1): LCDM como baseline y una extensión EFT mínima.

- Entrada. Ecuaciones de fondo + operadores EFT (finito) + Λ_{CG} declarado.
- Observables. (i) $H(z)$ o $D_L(z)$ (geometría) y (ii) $f\sigma 8$ o lente (crecimiento).
- D_{obs}. Datasets sintéticos o públicos con ventanas y covarianza.
- Salida. PASS/FAIL con diagnóstico de degeneración y reporte de rigidez.
- Auditoría. PCN: convergencia numérica y sensibilidad a Λ_{CG}.

11. PASS/FAIL, PASS fuerte y NO-EVAL

- PASS. Pipeline completo local→cosmo→D_obs; estabilidad a variaciones razonables de Λ_{CG} y discretización; compatibilidad multi-observable.
- PASS fuerte. Rigidez alta: pocos parámetros efectivos explican simultáneamente geometría + crecimiento + lente sin tuning.
- FAIL. Requiere funciones libres o viola consistencia local/cosmológica; no pasa tests cruzados (p.ej. lente vs dinámica).
- NO-EVAL. Falta D_obs con ventanas/covarianza o falta definición de coarse-graining/operadores finitos.

12. Loopholes/artefactos frecuentes

- w(z) libre. Prohibido sin origen EFT finito con error.
- μ(k,z), γ(k,z) libres. Prohibido por COS4.
- Elegir Λ_CG para ajustar. Prohibido; debe venir de ventana/validez.
- Usar datos fuera de ventana. Prohibido por ISAAC; sólo proyecciones con ventanas.
- Ignorar covarianza. Produce falsos PASS.

13. Definición de Hecho (DoD)

- DoD-1. Λ_CG declarado + sensibilidad.
- DoD-2. Fondo FRW derivado o parametrizado como EFT finita con error.
- DoD-3. Perturbaciones gauge-invariantes compiladas a predicciones (sin funciones libres).
- DoD-4. D_obs-cosmo con ventanas y covarianzas.
- DoD-5. Tests cruzados mínimos (geometría + crecimiento; lente si DM/GR-mod).
- DoD-6. Auditoría PCN (convergencia y tolerancias).

Apéndice A: glosario

- Coarse-graining. Integrar fuera modos cortos para obtener dinámica efectiva de modos largos.
- Λ_CG. Escala de separación de modos; debe estar ligada a ventana y validez.
- Geometría vs crecimiento. Observables que prueban fondo vs perturbaciones; su consistencia rompe degeneraciones.
- Ventanas. Funciones que describen a qué rangos de z/k/ℓ responde un observable.
- Rigidez cosmológica. Medida de cuántos parámetros efectivos controlan múltiples clases de observables.

Compilación de Consistencia Operacional (OCC)

Módulo 4F — Unificación operacional (CUI/HUI) y sonda universal (Avatar) (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo no “propone una nueva teoría” de fuerzas. Define un diccionario operacional único para describir, con el mismo lenguaje, acoplos y transportes asociados a gravedad y gauge, de modo que el juez OCC pueda compilar frontends distintos sin ambigüedades de convención.

Idea central. Una unificación metodológica exige dos objetos canónicos: (i) un objeto local para transporte infinitesimal (CUI), y (ii) un objeto global para respuesta acumulada en trayectorias (HUI).

Contenido

0. Alcance y dependencias

Depende de: Documento A (metodología $_{OCC}$). Este módulo es compatible con los demás (SK, Vacío/DE, G0), pero no depende de ellos.

Cubre: definiciones y reglas para construir una derivada covariante unificada (CUI) y su holonomía asociada (HUI) como objetos operacionales que permiten comparar y compilar teorías en frentes distintos.

No cubre: una dinámica específica. CUI/HUI no fijan ecuaciones de movimiento; fijan el lenguaje operacional con el que se expresan y evalúan.

1. Motivación (por qué): qué se rompe sin un diccionario 4F

La física se compila en observables: fases, secciones eficaces, precesiones, lentes, correlaciones. Muchos de estos observables son globales (integrados en caminos o regiones), no sólo locales. Sin un diccionario canónico:

- Ambigüedad de convención. Dos textos pueden describir el mismo fenómeno con conexiones distintas, signos distintos o normalizaciones distintas, y el juez no puede compararlos.
- Falsa unificación. Se puede declarar “unificación” por re-etiquetado (cambiar símbolos) sin contenido operacional.
- Incompatibilidad entre frontends. Lo que es natural en amplitudes no lo es en cosmología o en topología gauge; sin puente, la metodología no es universal.

CUI/HUI se introducen para que la unificación sea un hecho operacional: cualquier propuesta se expresa en el mismo alfabeto y se compila con los mismos candados.

2. Objeto operacional y frontend (transporte local + holonomía global)

El módulo introduce dos niveles:

- Local. Transporte infinitesimal: cómo cambia un objeto físico al moverse un paso (derivada covariante).
- Global. Transporte finito: respuesta acumulada al recorrer una trayectoria cerrada o una curva (holonomía).

Frontends donde se usa:

- Geométrico (GR/cosmología). Transporte con conexión afín/espinorial.
- Gauge (QFT). Wilson lines/loops y fases.
- Amplitudes. Codificación de acoplos y simetrías en objetos invariantes.

3. Definiciones: CUI, HUI y Avatar

3.1 CUI (Conexión Unificada ISAAC)

Definición operacional: CUI es el objeto local que permite definir una derivada covariante unificada sobre el espacio de estados/objetos del Sector A, incorporando las conexiones relevantes (geométrica y gauge) bajo una convención única.

```
∇^{(U)}_μ ≡ ∂_μ + □^{(U)}_μ
```

donde □^{(U)}_μ representa, en el lenguaje unificado, la suma/ensamble de conexiones (con estructura interna declarada por la teoría que se compila). El módulo no fija su forma; fija la convención y las reglas para compararla.

3.2 HUI (Holonomía Unificada ISAAC)

Definición operacional: HUI es la holonomía inducida por CUI al recorrer una curva γ:

```
ℋ^{(U)}[γ] ≡ □ exp(−∮_γ □^{(U)}_μ dx^μ).
```

El signo y la convención se fijan en el módulo para eliminar ambigüedad. ℋ^{(U)} es el objeto global que controla fases, precesiones, Aharonov–Bohm, Wilson loops y análogos gravitacionales.

3.3 Avatar (sonda universal)

Avatar no es nueva física. Es un alias operacional: una sonda genérica que porta las cargas necesarias para “leer” □^{(U)} (masa/energía para gravedad, representación para gauge). Se usa para fijar qué se mide sin reescribir acoplos en cada módulo.

4. Candados inevitables del módulo (4F1–4F11)

Cada candado evita una forma de “unificación por nombramiento”.

4F1. Covariancia/consistencia de transformación

Por qué. Si CUI no transforma correctamente, el objeto no es físico.

Qué. $\square^{(U)}$ debe transformarse como conexión bajo las simetrías declaradas (difeomorfismos/gauge).

4F2. Local→Global coherente

Por qué. Una holonomía que no proviene de la conexión rompe el significado operacional.

Qué. $\mathcal{H}^{(U)}$ se define como exponencial ordenada de $\square^{(U)}$; no se permite definir $\mathcal{H}$ por separado.

4F3. Reducción a casos conocidos

Por qué. El diccionario debe recuperar GR y gauge estándar donde funcionan.

Qué. En límites apropiados, $\nabla^{(U)}$ reproduce ∇ (GR) y D (gauge) con normalizaciones consistentes.

4F4. Independencia de convención (equivalencias)

Por qué. Si cambiar convención cambia predicción, el marco no está bien definido.

Qué. Cambios de gauge/conjugación deben dejar invariantes observables (trazas de holonomía, fases físicas).

4F5. Observables definidos (qué se mide)

Por qué. Sin una regla de lectura, CUI/HUI son símbolos.

Qué. El módulo define un conjunto base de observables: tr $\mathcal{H}$, fases relativas, precesión, y su mapeo a mediciones.

4F6. Compatibilidad con causalidad/localidad efectiva

Por qué. Conexiones no-locales ad hoc destruyen el cierre operacional.

Qué. $\square^{(U)}$ es local en el dominio ISAAC; cualquier no-localidad debe pasar por el módulo SK (entorno) o declararse fuera de dominio.

4F7. No reinyección UV (ISAAC)

Por qué. Introducir estructura por encima del dominio operacional vuelve el diccionario irrefutable.

Qué. Dependencias UV se parametrizan con número finito de coeficientes EFT y se reporta rigidez.

4F8. Consistencia de representación (Avatar)

Por qué. Si el resultado depende de una sonda inventada, no es universal.

Qué. Avatar sólo selecciona representación/carga declarada; no agrega grados de libertad ocultos.

4F9. Separación entre diccionario y dinámica

Por qué. Mezclar definición con dinámica permite ajustar por semántica.

Qué. El módulo prohíbe derivar conclusiones dinámicas sin invocar un módulo de dinámica/anclaje.

4F10. Auditabilidad de normalizaciones

Por qué. La normalización de acoplos suele ocultar “unificación falsa”.

Qué. Toda normalización (factores, signos, constantes) se declara explícitamente y se fija en el MRD.

4F11. Rigidez del diccionario

Por qué. Si el diccionario admite demasiadas convenciones, pierde poder comparativo.

Qué. Se reporta rigidez R sobre el espacio de equivalencias; PASS fuerte si el diccionario reduce ambigüedad a una clase pequeña.

5. Compilación OCC (cómo se convierte a constraints)

Este módulo compila principalmente consistencia semántica en constraints verificables:

- Verificar 4F1–4F4 mediante transformaciones y construcción de invariantes (trazas, clases de conjugación).
- Verificar 4F3 construyendo reducciones explícitas a GR/gauge estándar.
- Verificar 4F10 mediante un archivo de convención (YAML/JSON) que fija normalizaciones y signos.
- Evaluar 4F11 (rigidez) midiendo el tamaño de la clase de equivalencia restante.

Salida: un “diccionario compilable” que luego se usa como prerequisito de los módulos dinámicos (Vacío/DE, G0, etc.).

6. Anclajes D_obs típicos (fase/holonomía, dispersión, GR tests, gauge)

Aunque este módulo es principalmente definicional, puede anclarse con observables de holonomía/fase:

- D_obs-AB. Fase Aharonov–Bohm / interferometría → constraint sobre tr $\mathcal{H}$ en loops gauge.
- D_obs-precesión. Precesiones/holonomía gravitacional (p.ej. giroscopios, trayectorias) → constraint sobre componente geométrica.
- D_obs-dispersión. Amplitudes y selección de representación/cargas → chequeo de compatibilidad de acoplos.
- D_obs-topológico. Cuantización (cuando aplique) → chequeo de normalizaciones (4F10).

Regla: todo anclaje debe mapearse a un invariante construido con $\mathcal{H}$^{(U)} bajo convención fija.

7. MRD-4F (demo mínimo reproducible)

MRD-4F fija un caso mínimo que cualquier tercero puede verificar:

- Parte gauge. U(1) con un loop circular: reproducir fase AB como tr $\mathcal{H}$ (con convención fija).
- Parte geométrica. Transporte paralelo en una geometría simple (p.ej. esfera o métrica débil): reproducir holonomía geométrica.
- Unificación. Mostrar cómo ambas holonomías se escriben en la misma plantilla $\mathcal{H}$^{(U)} con el mismo esquema de auditoría.
- Auditoría. Archivo de convención + hashes + scripts.

8. PASS/FAIL, PASS fuerte y predicciones/no-go

I_4F ≡ F_4F ∩ D_obs (si se usan anclajes).

- FAIL. Si no se pueden satisfacer 4F1–4F4 o si el diccionario no reduce ambigüedad (4F11) bajo convención auditada.
- PASS. Existe un diccionario consistente y reproducible.
- PASS fuerte. La clase de equivalencia es pequeña y el diccionario impone rigidez real entre frontends.

Predicciones/no-go típicos:

- No-go: “unificación” basada sólo en re-escalas que no preservan invariantes de holonomía → FAIL (4F10/4F11).
- Predicción operacional: relación fija entre un observable de holonomía y un acoplo bajo convención unificada (cuando el módulo se combine con un módulo dinámico).

9. Loopholes/artefactos frecuentes y cómo evitarlos

- Unificación por cambio de símbolos. Evitar: exigir invariantes y MRD con convención fija (4F10).
- Normalizaciones ocultas. Evitar: archivo de convención auditable, sin “absorber” constantes en definiciones a posteriori.
- Mezclar dinámica. Evitar: este módulo no afirma ecuaciones; sólo define objetos y pruebas de consistencia.

10. Definición de Hecho (DoD) para publicar este módulo

- DoD-1. Definiciones CUI/HUI/Avatar con convención fija (Sección 3).
- DoD-2. Implementación computable de 4F1–4F11 + tests negativos (ejemplos que fallen).
- DoD-3. MRD-4F reproducible con auditoría completa.
- DoD-4. Al menos un anclaje de holonomía/fase (AB o geométrico) reproducido.
- DoD-5. Integración demostrada: al menos un módulo dinámico usa el diccionario sin ambigüedad adicional.

Apéndice A: glosario local

- CUI. Conexión Unificada ISAAC: objeto local de transporte infinitesimal.
- HUI. Holonomía Unificada ISAAC: objeto global (Wilson/holonomía) construido desde CUI.
- Avatar. Sonda universal: alias operacional para fijar qué sistema mide la conexión.
- □^{(U)}. Notación para la conexión unificada en el dominio declarado.
- ℋ^{(U)}. Holonomía unificada a lo largo de una curva.
- Rigidez. Medida de cuánta ambigüedad se elimina al fijar el diccionario.

Definiciones canónicas de OCC /SGO/SIA/ISAAC están en el Documento A.

Compilación de Consistencia Operacional (OCC)

Módulo Unificación Dinámica 4F — Encadenamiento explícito con CUI/HUI (versión cerrada v2)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo convierte la unificación 4F en un hecho dinámico: no basta un diccionario (CUI/HUI); se requiere filtrar dinámicas candidatas con candados inevitables y anclajes observacionales.

Novedad de v2. Se hace explícito que el módulo consume el diccionario 4F como prerequisito: toda dinámica que pretenda "unificar 4 fuerzas" debe expresarse en el lenguaje CUI/HUI y sus invariantes de holonomía. Sin eso, el juez no compila la propuesta (FAIL por formato).

Contenido

0. Alcance, dependencias y prerequisito 4F

Depende de: Documento A (metodología $_{\text{OCC}}$) y del Módulo 4F (diccionario operacional CUI/HUI + Avatar).

Prerrequisito (gating). Una propuesta de unificación dinámica 4F sólo se compila si primero pasa el módulo 4F: es decir, si especifica CUI/HUI, convención auditada y el conjunto de invariantes que sirven de observables. Si no, el veredicto es FAIL por formato (no por física).

Cubre: el juez universal que decide si existe una dinámica conjunta compatible con candados inevitables + observaciones, expresada en el lenguaje 4F dentro del dominio ISAAC Ω_I.

No cubre: generar el espacio de hipótesis desde cero. El módulo recibe (i) un diccionario 4F y (ii) una clase de dinámicas (parametrizadas) expresadas en ese diccionario.

1. Motivación (por qué): por qué 4F operacional no implica 4F dinámica

El módulo 4F define un alfabeto común (CUI/HUI) para describir transporte y holonomía. Pero un alfabeto no decide qué oraciones son verdaderas.

Sin un módulo de cierre dinámico:

- Unificación estética. Se reescribe GR y gauge en un mismo símbolo, pero la dinámica sigue siendo arbitraria.
- Incomparabilidad. Dos propuestas usan la misma notación pero producen observables distintos; sin juez no hay veredicto.
- Maleabilidad. Se pueden ajustar infinitos coeficientes EFT dentro del diccionario para “pasar” cualquier dato.

El objetivo del módulo UD es convertir el lenguaje 4F en filtro: en vez de “unificamos”, el veredicto se vuelve PASS/FAIL y, cuando la intersección es rígida, PASS fuerte.

2. Objeto operacional: θ, Ω_I, CUI/HUI y el trío F–D_{obs}–I

Se trabaja con un espacio de hipótesis parametrizado por θ (finitamente), donde cada θ define:

- una conexión unificada $\square^{(U)}_\mu(\theta)$ (CUI) en Ω_I, y
- un conjunto de holonomías/invariantes $\mathcal{H}^{(U)}[\gamma;\theta]$ (HUI) para curvas γ relevantes.

Con esto se define el trío canónico:

- F (factible teórico). Conjunto de θ que satisface candados inevitables (unitaridad, analiticidad/causalidad, crossing, positividad, etc.).
- D_{obs} (anclaje). Conjunto de θ compatible con observaciones expresadas como invariantes 4F (trazas de holonomía, fases, secciones eficaces, etc.).
- I. Intersección $I \equiv F \cap D_{obs}$.

Veredicto: PASS si $I \neq \varnothing$; FAIL si $I = \varnothing$. PASS fuerte si I es (casi) discreta o dominada por incertidumbre experimental.

3. Interfaz 4F→UD: qué debe entregar el diccionario 4F

Para evitar ambigüedad, UD exige que el módulo 4F entregue un paquete mínimo (entrada canónica):

Elemento 4F	Descripción	Uso en UD
Convención	signos, normalizaciones, representación Avatar, unidades	gauge + reproducibilidad; evita “unificación por re-e
CUI: □^{(U)}_μ	definición explícita de la conexión unificada y sus transformaciones	base de constraints locales (causalidad/localidad efe
HUI: ℋ^{(U)}[γ]	holonomía para curvas γ con ordenamiento □	construcción de observables globales (fases, Wilson
Invariantes	p.ej. tr ℋ, fases relativas, clases de conjugación	mapeo de D_obs y de certificados (primal/dual)
Plantilla auditable	YAML/JSON con hashes y versiones	auditoría OCC , MRD y replicabilidad

Regla de cierre: UD no permite introducir observables “a mano”. Todo observable debe ser expresado como invariante construido desde HUI o como observable estándar (sección eficaz, correlación) cuyo mapeo a invariantes 4F esté declarado y auditado.

4. Candados inevitables (UD1–UD9) con ‘por qué → qué’

UD1. Unitaridad (probabilidad)

Por qué. Sin unitaridad no hay interpretación estable de probabilidades.

Qué. La dinámica admite formulación unitaria en Ω_I (p.ej. ondas parciales; canales inelásticos cuando aplica).

UD2. Analiticidad y causalidad (dispersión)

Por qué. Causalidad/micro-localidad se traducen en propiedades analíticas que habilitan bounds.

Qué. Objetos relevantes (amplitudes/respuestas) analíticos con cortes físicos estándar; relaciones de dispersión con subtracciones declaradas.

UD3. Crossing y consistencia de canales

Por qué. Un mismo proceso en canales distintos no puede contradecirse.

Qué. Crossing implementado con convención fija del paquete 4F; se compila como igualdades/inequidades.

UD4. Positividad (EFT/convexidad)

Por qué. UV completabilidad local/causal impone desigualdades sobre coeficientes EFT.

Qué. Bounds de positividad compilados en Ω_I (manejo IR-safe declarado).

UD5. Consistencia IR (matching y regularización)

Por qué. IR mal tratado produce falsos FAIL/PASS.

Qué. Esquema de regularización/subtracción declarado; estabilidad bajo variación razonable del esquema.

UD6. No reinyección UV (ISAAC)

Por qué. Ajustar dinámica con estructura fuera de Ω_I destruye falsabilidad.

Qué. Dependencias UV encapsuladas en número finito de parámetros EFT dentro de Ω_I.

UD7. Intersección multi-frente (4F real)

Por qué. Unificar 4 fuerzas exige coherencia simultánea entre frentes (gauge, gravedad, materia).

Qué. El mismo θ debe satisfacer candados y anclajes en los frentes declarados (HEP, GR/cosmo, etc.).

UD8. Rigidez obligatoria

Por qué. PASS con gran volumen de soluciones es maleable.

Qué. Reportar rigidez R (o d_eff). PASS fuerte si no hay direcciones planas relevantes.

UD9. Observables sólo vía invariantes 4F (anti-hack semántico)

Por qué. Sin esta regla, se pueden introducir ‘observables’ que no corresponden a ningún objeto medible derivado de CUI/HUI.

Qué. Todo D_obs se expresa como función de invariantes 4F (tr $\mathcal{H}$, fases, holonomías) o con un mapeo explícito, auditable y reversible hacia ellos.

5. Compilación: del lenguaje CUI/HUI a constraints (primal/dual)

UD compila constraints en dos capas:

- Capa 4F (formato). Verifica que CUI/HUI y sus invariantes están definidos y son consistentes (pasa 4F1–4F11).
- Capa dinámica. Sobre θ, impone UD1–UD9 y construye F_{UD}.

Forma primal/dual:

- Primal. Variables θ; constraints como desigualdades/igualdades; solver (SDP/LP/NLP) según el caso.
- Dual. Certificados/witness que prueban $I=\varnothing$ construyendo funcionales sobre invariantes 4F.

Punto clave: la dualidad se expresa en el mismo lenguaje 4F: los witness actúan sobre tr $\mathcal{H}$ o sobre amplitudes mapeadas a invariantes.

6. Anclajes D_obs y su mapeo a invariantes 4F

UD requiere que cada anclaje se escriba en una de estas formas:

- (A) Directo en holonomías. Datos de fase/holonomía (AB, precesión, Wilson loops) → bounds sobre tr $\mathcal{H}^{(U)}[\gamma;\theta]$.
- (B) Mapeado. Secciones eficaces/correlaciones → mapeo explícito a invariantes construidos desde CUI/HUI (por ejemplo, mediante amplitudes EFT cuya base está normalizada por 4F10 y cuyos coeficientes se interpretan como funcionales de $\square^{(U)}$).
- (C) Geométrico. Observables GR/cosmológicos → mapeo a holonomías/transportes (precesión, deflexión, integrales a lo largo de geodésicas).

Regla: si el mapeo no se puede auditar, el anclaje no se admite en UD (se evita ‘meter datos’ sin conexión al diccionario).

7. MRD-UD (demo mínimo reproducible) con holonomías explícitas

MRD-UD muestra el encadenamiento completo 4F→UD en un caso mínimo:

- Diccionario 4F. Definir CUI/HUI para (i) U(1) y (ii) una geometría débil (holonomía geométrica).
- Clase dinámica. Un EFT mínimo con pocos coeficientes θ que afecten un observable de dispersión y un observable de holonomía.
- Candados. UD1–UD4 (unitaridad+dispersión+crossing+positividad) + UD9 (observables vía invariantes) en Ω_I.
- Anclaje. (i) fase AB (tr $\mathcal{H}$ gauge) y (ii) un límite de dispersión/HEP (o mock auditado).
- Salida. PASS/FAIL + (si FAIL) certificado dual + rigidez.

8. PASS/FAIL, PASS fuerte y no-go

$I_{UD} \equiv F_{UD} \cap D_{obs}$.

- FAIL. $I_{UD} = \emptyset$ o falla el gating 4F (formato/convención).
- PASS. $I_{UD} \neq \emptyset$.
- PASS fuerte. I_{UD} (casi) discreta o dominada por incertidumbre experimental.

No-go típico: si para satisfacer un anclaje mapeado se requiere violar UD1–UD4 (unitaridad/causalidad/positividad), la clase dinámica completa queda descartada.

9. Loopholes/artefactos frecuentes

- Mapeos no reversibles. Traducir un observable a invariantes 4F de forma ambigua permite hacks; UD exige mapeo auditable y reversible.
- Truncaciones EFT ocultas. Cambiar la base de operadores para esconder violaciones de positividad; se fija por convención 4F10.
- Selección post-hoc de curvas γ. Elegir trayectorias después de ver datos puede inflar PASS; UD exige lista declarada.
- Mezclar diccionario con ajuste. Re-definir □^{(U)} para “absorber” tensiones; prohibido: □ se fija por clase θ.

10. Definición de Hecho (DoD)

- DoD-1. Paquete 4F completo (convención + CUI/HUI + invariantes) y gating implementado.
- DoD-2. Implementación computable de UD1–UD9 + tests negativos.
- DoD-3. MRD-UD reproducible con auditoría (hashes, versiones, semillas).
- DoD-4. Al menos un anclaje directo en holonomía y uno mapeado (HEP o GR) con traducción auditable.
- DoD-5. Reporte de rigidez y criterio explícito de PASS fuerte.

Apéndice A: plantilla de entrada 4F (YAML/JSON)

Ejemplo de campos mínimos (esquema; no es código ejecutable):

```
4F_package: convention: sign: ... normalization: ... units: ... avatar_representation: ...
CUI: A_U_mu(theta): ... transformation_law: ... HUI: curves_gamma: [ ... ]
holonomy_definition: Pexp(-∮ A_U dx) invariants: - name: trH definition:
tr(H_U[gamma]) - name: phase_rel definition: arg(trH) audit: hashes: ... versions: ...
```

UD rechaza entradas que no fijen convención y curvas γ de forma previa a los datos.

Apéndice B: glosario

- CUI/HUI. Conexión/holonomía unificadas del módulo 4F.
- θ. Parámetros finitos que definen una clase dinámica dentro de Ω_I.
- Ω_I. Dominio ISAAC: región donde se exigen candados sin reinyección UV.
- F, D_{obs}, I. Conjunto factible, anclaje observacional e intersección superviviente.
- Gating 4F. Requisito de formato: sin diccionario 4F auditable no se compila unificación dinámica.
- Witness. Certificado dual de exclusión (prueba de $I=\varnothing$).

Compilación de Consistencia Operacional (OCC)

Módulo Amplitudes & Positividad — S-matrix, dispersión y UV-completabilidad (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo convierte principios inevitables de dispersión en un conjunto de desigualdades compilables sobre amplitudes y/o coeficientes EFT: unitaridad, analiticidad/causalidad, crossing y positividad. Es la pieza que vuelve al juez OCC "duro" en HEP/EFT: en vez de intuición, entrega PASS/FAIL con certificados (cuando aplica).

Prioridad. Primero el por qué (qué se rompe si no imponemos estos principios), luego el qué (candados, protocolos IR-safe, anclajes y MRD).

Contenido

0. Alcance y dependencias

Depende de: Documento A (metodología $_{OCC}$) y del Módulo EFT/Renormalización sólo si se quiere traducir a coeficientes EFT; de lo contrario puede operar directamente en amplitudes A(s,t).

Cubre: (i) constraints universales sobre amplitudes 2→2 (y extensiones declaradas), (ii) traducción a bounds sobre coeficientes EFT (positividad), y (iii) generación de certificados de exclusión cuando el problema es convexo.

No cubre: dinámica completa de un modelo. Este módulo filtra: si una propuesta viola un candado, el módulo la descarta.

1. Motivación (por qué): por qué amplitudes son un juez universal

La dispersión codifica el “contrato” mínimo que cualquier teoría local/causal debe cumplir: probabilidad, causalidad y consistencia entre canales. Si el juez no impone esto, una propuesta puede ajustar datos locales y aun así ser imposible en un UV consistente.

Las amplitudes son universales porque no dependen de la presentación (Lagrangiano vs variables geométricas): un experimento mide secciones eficaces, y éstas se derivan de amplitudes.

- Qué se rompe si no. (i) probabilidades >1 (violación de unitaridad), (ii) superluminalidad/acrobatias causales (analiticidad), (iii) contradicción entre procesos equivalentes (crossing), (iv) EFT no UV-completable (violación de positividad).

2. Objeto operacional: A(s,t), dominio Ω_I y observables

Objeto: amplitud de dispersión 2→2 (o familia declarada) en variables de Mandelstam (s,t,u):

```
A(s,t) con s+t+u = Σ m_i^2.
```

Dominio operacional: Ω_I (ISAAC) fija el rango de energías/transferencias donde se imponen candados sin reinyección UV.

Observables: secciones eficaces (totales/diferenciales), parámetros EFT (coeficientes), y/o combinaciones integrales (sum rules) que se mapean a A(s,t).

3. Definiciones: analiticidad, crossing, unitaridad parcial, positividad

3.1 Analiticidad/causalidad

En teorías locales/causales, $A(s,t)$ es analítica en el plano complejo salvo cortes/polos físicos. Esto habilita relaciones de dispersión que conectan derivadas de A con integrales de secciones eficaces (óptica).

3.2 Crossing

La misma amplitud describe canales relacionados por intercambio de partículas/antipartículas; crossing impone identidades (con convención fija).

3.3 Unitaridad parcial

La expansión en ondas parciales impone bounds tipo $|a_\ell(s)| \leq 1$ (o variantes con inelasticidad).

3.4 Positividad (EFT)

Bajo hipótesis de UV completabilidad local/causal y unitaridad, combinaciones de derivadas de A en punto forward ($t \to 0$) deben ser positivas. Traducido a EFT, esto impone desigualdades sobre coeficientes.

4. Candados inevitables del módulo (A1–A14) con ‘por qué → qué’

A1. Unitaridad óptica ($\mathrm{Im}\, A \geq 0$ en canal físico)

Por qué. La sección eficaz total debe ser no negativa.

Qué. Se verifica la relación óptica en el régimen donde aplica (con masas/cortes declarados).

A2. Unitaridad parcial (ondas parciales)

Por qué. Evita probabilidades >1 en cada ℓ.

Qué. Bounds sobre $a_\ell(s)$ en Ω_I, con inelasticidad incluida si se declara.

A3. Analiticidad mínima (estructura de cortes/polos físicos)

Por qué. Sin analiticidad no hay causalidad operacional ni dispersión robusta.

Qué. Se declara el mapa de singularidades físicas; se prohíben singularidades ad hoc en Ω_I.

A4. Crossing (consistencia de canales)

Por qué. Un mismo proceso no puede contradecirse por relabeling.

Qué. Identidades de crossing implementadas con convención fija; se verifican en muestras (s,t) relevantes.

A5. Boundedness / subtracciones declaradas

Por qué. Relaciones de dispersión requieren control asintótico.

Qué. Número de subtracciones N_{sub} declarado y estabilidad bajo variación razonable.

A6. Positividad forward (cuando aplica)

Por qué. UV completabilidad impone convexidad/positividad.

Qué. Se compilan desigualdades estándar sobre derivadas $\partial_s^n A(s,0)$ evaluadas en región analítica.

A7. Positividad mejorada / dispersión substracta (opcional)

Por qué. En presencia de masas pequeñas/IR, positividad naive puede fallar por artefacto.

Qué. Usar protocolo IR-safe (Sección 6) y recompilar bounds con términos IR sustraídos.

A8. Manejo de inelasticidad (canales abiertos)

Por qué. Ignorar canales abiertos produce falsos PASS/FAIL.

Qué. Si hay umbrales, se incluye contribución inelástica o se restringe Ω_I para evitarlos (declarado).

A9. Identificación de parámetros EFT finitos

Por qué. Permitir infinitos operadores hace al módulo maleable.

Qué. Truncación EFT finita con criterio de dominancia y error declarado; rigidez reportada.

A10. No reinyección UV (ISAAC)

Por qué. Meter resonancias/funciones fuera de Ω_I destruye falsabilidad.

Qué. Todo efecto UV se encapsula en coeficientes EFT finitos o en parámetros de amplitud dentro de Ω_I.

A11. Consistencia con simetrías declaradas

Por qué. Simetrías reducen estructura; violarlas a posteriori es hack.

Qué. Simetrías (C, P, gauge, etc.) se fijan en entrada y se verifican como constraints.

A12. Auditabilidad numérica

Por qué. Bounds dependen de discretización/solvers.

Qué. Semillas, tolerancias, mallas y versiones quedan auditadas; pruebas de convergencia mínimas obligatorias.

A13. Certificados (cuando convexo)

Por qué. Un FAIL debe poder justificarse sin ambigüedad.

Qué. Si el problema se formula como SDP/LP, se produce witness/dual certificate que prueba $I=\emptyset$.

A14. Rigidez obligatoria

Por qué. Si muchos θ pasan, no hay predicción.

Qué. Reportar rigidez R o dimensión efectiva d_{eff}; PASS fuerte si el conjunto superviviente es pequeño/discreto.

5. Compilación OCC : de principios a desigualdades (primal/dual)

El compilador construye F_A (factible) como intersección de candados A1–A14 en Ω_I. Hay dos rutas:

- Ruta amplitud-directa. Parametrizar A(s,t;θ) y aplicar constraints de unitaridad, crossing, analiticidad (vía dispersión) y positividad.
- Ruta EFT. Parametrizar coeficientes c_i (truncación) y compilar positividad como desigualdades lineales/convexas sobre c_i; unitaridad impone límites adicionales en energía.

Si el conjunto D_obs se expresa en el mismo espacio (θ o c_i), se calcula I_A ≡ F_A ∩ D_obs.

6. Tratamiento IR-safe (singularidades, masas pequeñas, subtracciones)

Muchos falsos debates de positividad vienen de IR ($t \to 0$, intercambio de masa cero, Coulomb, etc.). Este módulo exige que el IR se trate explícitamente:

- IR-safe-1. Declarar si hay estados sin masa o casi sin masa en el canal t (fotón/gravitón).
- IR-safe-2. Si existen, sustraer contribuciones IR conocidas (o introducir regulador) antes de aplicar positividad.
- IR-safe-3. Reportar sensibilidad del bound al regulador y demostrar estabilidad en el rango declarado.
- IR-safe-4. Si no hay forma limpia, restringir Ω_I lejos del forward o declarar que sólo se aplica un subconjunto de bounds (con justificación).

7. Anclajes D_obs (EFT bounds, VBS, aQGC, etc.) y mapeo

Formatos típicos de D_obs:

- D_obs-EFT. Límites experimentales sobre coeficientes EFT (intervalos) → restricciones sobre c_i.
- D_obs-σ. Secciones eficaces/diferenciales en ventanas de energía → restricciones sobre A(s,t).
- D_obs-unit. Requerimientos de no violar unitaridad hasta una energía E_{max} (según análisis experimental).
- D_obs-cor. Correlaciones (matrices) entre parámetros EFT provenientes de fits.

Regla: cada anclaje debe incluir el mapeo a (s,t) y/o a la base EFT usada, con normalización auditada.

8. MRD-A (demo mínimo reproducible)

MRD-A demuestra el módulo con un caso mínimo reproducible:

- Problema. Dispersión 2→2 escalar o gauge simplificada con amplitud parametrizada por pocos coeficientes θ.
- Candados. Implementar A2 (unitaridad parcial) + A4 (crossing) + A6 (positividad forward) con IR-safe si aplica.
- Anclaje. Un conjunto mínimo de límites EFT (mock auditado o dataset público) como D_obs-EFT.
- Salida. PASS/FAIL + witness si FAIL + rigidez; auditoría completa (hashes, versiones, tolerancias).

9. PASS/FAIL, PASS fuerte y no-go

$I_A \equiv F_A \cap D_{obs}$.

- FAIL. $I_A = \emptyset$ (o no pasa IR-safe / auditabilidad).
- PASS. $I_A \neq \emptyset$.
- PASS fuerte. I_A casi discreta; bounds de positividad/ unitariad eliminan direcciones planas.

No-go típico: si un fit experimental requiere un signo de coeficiente EFT que viola positividad en Ω_I (después de IR-safe), esa clase de UV completaciones queda descartada.

10. Loopholes/artefactos frecuentes

- Aplicar forward positivity con fotón/gravitón sin sustracción. Produce falsos FAIL; usar Sección 6.
- Truncación EFT insuficiente. Cambiar la base/truncación para escapar bounds; el módulo exige criterio y auditoría.
- Crossing mal normalizado. Signos/factores mal fijados producen contradicciones; usar convención fija del diccionario 4F (si aplica) o del módulo EFT.
- Discretización floja. Mallas/tolerancias pobres inflan PASS; exigir pruebas de convergencia (A12).

11. Definición de Hecho (DoD)

- DoD-1. Definición clara de Ω_I, del proceso $2\to2$ y de la parametrización finita (θ o c_i).
- DoD-2. Implementación computable de A1–A14 + tests negativos.
- DoD-3. Protocolo IR-safe documentado y probado (cuando aplique).
- DoD-4. MRD-A reproducible con auditoría completa; witness si FAIL.
- DoD-5. Al menos un anclaje D_{obs} real traducido auditablemente.

Apéndice A: plantilla de entrada (YAML/JSON)

Esquema mínimo (no ejecutable):


```
amplitudes_module: process: 2to2 particles: [...] domain_ISAAC: s_range: [smin, smax] t_range: [tmin, tmax] parametrization: type: EFT | direct_amplitude parameters: {c1:..., c2:..., ...} truncation_rule: ... constraints: unitarity_partial: {...} crossing: {...} dispersion: subtractions: N_sub positivity: forward: true IR_safe: {...} anchors_Dobs: - kind: EFT_bounds | sigma dataset: ... mapping: ... audit: hashes: ... versions: ... solver: name: ... tolerance: ... seed: ...
```


Regla: si falta el bloque IR_safe cuando hay intercambio de masa cero, el módulo rechaza la corrida.

Apéndice B: glosario

- A(s,t). Amplitud de dispersión 2→2.
- Relación óptica. Conecta Im A con sección total.
- Ondas parciales. Expansión por ℓ con bounds de unitaridad.
- Positividad. Desigualdades sobre derivadas/coeficientes EFT provenientes de UV completabilidad.
- IR-safe. Protocolo para sustraer/regul. contribuciones IR antes de aplicar bounds.
- Witness. Certificado dual que prueba inconsistencia (I=∅).
- Ω_I. Dominio ISAAC.

Compilación de Consistencia Operacional (OCC)

Módulo Bariogénesis — Cierre operacional de asimetría materia–antimateria (versión cerrada v1)

Marco Antonio Isaac Alcuria

12 de febrero de 2026

Rol del módulo. Este módulo define cómo el juez OCC compila mecanismos capaces de producir una asimetría bariónica neta compatible con el universo observado, sin introducir perillas no operacionales. El módulo no impone un mecanismo único; impone condiciones mínimas y un formato de anclaje para que cualquier propuesta sea falsable.

Prioridad. Primero el por qué: qué se rompe si la asimetría se introduce “a mano”. Luego el qué: candados, anclajes y PASS/FAIL.

Contenido

0. Alcance y dependencias

Depende de: Documento A (OCC universal). Para mecanismos fuera de equilibrio, usa el frontend SK (o QFT en contorno cerrado). Si se usa equilibrio térmico, activa el candado KMS (C9). Si se usan anomalías/simetrías globales, activa candados de anomalía (C10).

Cubre: criterios operacionales para generar y conservar una asimetría bariónica neta η_B compatible con observaciones (CMB/BBN) y con límites experimentales (EDM, LHC, etc.).

No cubre: justificar un valor de η_B por fiat. Cualquier entrada “η_B inicial” sin mecanismo se considera perilla y produce FAIL.

1. Motivación (por qué): el problema operacional de la asimetría

El universo observado contiene una asimetría neta materia–antimateria. Operacionalmente, esto se codifica en la razón barión–fotón η_B.

Sin un módulo cerrado, es fácil caer en dos trampas:

- Asimetría a mano. Introducir condiciones iniciales asimétricas sin mecanismo verificable: irrefutable.
- Perillas ocultas. Ajustar fases CP o tasas de violación de B con funciones libres fuera del dominio ISAAC.

El módulo exige que la asimetría emerja de dinámica compatible con principios (unidad/casualidad/positividad) y con límites experimentales.

2. Objeto operacional y frontends (QFT térmica/no-equilibrio + anomalías)

Objeto operacional: la asimetría bariónica final (o equivalente) calculada desde un mecanismo dinámico.

$$\eta_B \equiv (n_B - n_{\bar{B}})/n_\gamma \text{ (evaluado tras freeze-out).}$$

Frontends típicos:

- Fuera de equilibrio. SK / ecuaciones de Boltzmann / cinética cuántica (cuando hay transiciones, paredes de fase, decays fuera de equilibrio).
- Térmico. Estados KMS y tasas térmicas (cuando aplica).
- Anomalías. Si B o L se violan por anomalías o efectos no perturbativos, se compilan como contabilidad de cargas + matching.

3. Definiciones: asimetría, número bariónico efectivo y contabilidad

3.1 Contabilidad de cargas

El módulo obliga a declarar: qué simetrías existen, cuáles se violan y en qué etapa (temperatura/escala/tiempo). Se define un vector de cargas $Q(t)$ que incluye B, L y cualquier carga relevante.

3.2 Número bariónico efectivo

En modelos con conversión (p.ej. $B-L$ conservado, sphalerons activos), el módulo exige un mapeo explícito de la asimetría generada en un sector a η_B final.

3.3 Dominio ISAAC

Todas las tasas, fases CP y parámetros relevantes deben vivir en el dominio operacional declarado. Parámetros UV no accesibles sin proyección efectiva auditada se consideran perillas (FAIL por C0/ISAAC).

4. Candados inevitables del módulo (B1–B14)

Cada candado está motivado por evitar arbitrariedad y asegurar falsabilidad.

B1. Generación dinámica (prohibido η_B por fiat)

Por qué. Un valor inicial impuesto no es explicación.

Qué. η_B debe ser salida de dinámica; condiciones iniciales deben ser simétricas o derivadas de mecanismo.

B2. Tres ingredientes mínimos (Sakharov) como constraints operacionales

Por qué. Sin violación de B, C/CP y fuera de equilibrio, no hay asimetría neta en marcos estándar.

Qué. El módulo exige que la propuesta declare dónde se cumplen: violación efectiva de B (o B−L), violación CP, y no-equilibrio (o excepción declarada y justificada).

B3. Unitaridad/positividad de probabilidades

Por qué. Tasas negativas o violaciones de CPTP en cinética son no físicas.

Qué. Ecuaciones cinéticas deben preservar densidades no negativas; si se usa SK, hereda SK3/SK4.

B4. Causalidad efectiva

Por qué. Fuera de equilibrio no permite respuestas avanzadas.

Qué. Kernels/colisiones deben ser causales (hereda SK1).

B5. Contabilidad de cargas y anomalías (si aplica)

Por qué. Sin contabilidad, se puede “inventar” violación de B.

Qué. Si se invoca anomalía/sphalerons, se compila matching de cargas (C10) y condiciones de activación (ventana de temperatura).

B6. Freeze-out explícito y conservación posterior

Por qué. Generar asimetría sin preservarla es inútil.

Qué. El mecanismo debe mostrar cuándo se congela η_B y por qué no se lava (washout) después.

B7. Parametrización finita (anti-perillas)

Por qué. Ajustar fases CP como funciones libres destruye falsabilidad.

Qué. Número finito de parámetros CP y de tasas; se reporta rigidez R.

B8. Compatibilidad con límites de EDM (si CP adicional)

Por qué. CP grande suele producir EDM medibles; ignorarlo permite ajustes ilegítimos.

Qué. Si el mecanismo usa nuevas fases CP, debe compilar límites EDM relevantes como D_obs.

B9. Compatibilidad con collider (si nuevas partículas)

Por qué. Nuevas escalas pueden estar excluidas.

Qué. Si se postulan estados nuevos, se compilan límites LHC/otros como anclajes.

B10. Compatibilidad con neutrinos (si leptogénesis)

Por qué. Leptogénesis conecta con masas/mezclas.

Qué. El módulo exige mapping explícito a parámetros neutrino y constraints.

B11. Consistencia térmica (KMS) sólo si se declara equilibrio

Por qué. Equilibrio impone relaciones detalladas; si no se cumplen, arbitrariedad.

Qué. Si se declara equilibrio, se verifica KMS/FDT en tasas relevantes.

B12. No reinyección UV (ISAAC)

Por qué. “Explicar” η_B con física inaccesible sin proyección efectiva rompe el marco.

Qué. Todo parámetro UV se encapsula en coeficientes EFT finitos y auditables.

B13. Robustez a incertidumbres cosmológicas

Por qué. El mecanismo no debe depender de un valor ajustado post-hoc de H(T) o g_* fuera de rangos.

Qué. Se exige estabilidad de η_B bajo rangos razonables de historia térmica declarada.

B14. Rigidez obligatoria

Por qué. Si muchos parámetros se ajustan, no hay predicción.

Qué. Reportar R y/o d_{eff}; PASS fuerte si $R \lesssim 1$.

5. Compilación OCC (cómo se convierte a constraints)

El compilador transforma el mecanismo en un sistema de constraints:

- Ecuaciones de evolución (Boltzmann/kinética/SK) → cálculo de η_B como función de parámetros.
- Candados B1–B6 → verificación estructural (existe B-viol, CP-viol y no-equilibrio; freeze-out y washout).
- Candados B7/B14 → penalización por complejidad + cálculo de rigidez.
- Anclajes experimentales (EDM, collider, neutrinos) → intersección con D_{obs}.
- Resultado: $I_B \equiv F_B \cap D_{obs}$.

6. Anclajes D_obs (η_B, BBN, CMB, EDM, LHC, neutrinos) y traducción auditable

Formatos admitidos de anclaje (cada uno con pipeline auditable):

- D_obs-η. Valor observado de η_B (con error) como banda objetivo.
- D_obs-BBN/CMB. Consistencia con abundancias/parametrización cosmológica usada.
- D_obs-EDM. Límites de EDM (n, e, átomos/moléculas) cuando hay CP adicional.
- D_obs-collider. Límites de producción/decays de partículas nuevas si el mecanismo las requiere.
- D_obs-ν. Masas/mezclas y constraints cosmológicos si el mecanismo es leptogénesis.

Regla: el módulo debe declarar con precisión cómo cada dataset se traduce a desigualdades (y qué aproximaciones se usan).

7. MRD-B (demo mínimo reproducible)

MRD-B define un demo mínimo que produce η_B de forma dinámica:

- Mecanismo mínimo. Decay fuera de equilibrio de un estado pesado X con violación CP y canales que violan B (o B−L).
- Implementación. Ecuaciones de Boltzmann a un parámetro de CP ε_{CP} y tasa Γ_X, con historia térmica simple.
- Tests. Verificar B1–B6 y estabilidad (B13) bajo variaciones razonables.
- Salida. PASS/FAIL + auditoría (hashes, versiones, semillas) + rigidez.

8. PASS/FAIL, PASS fuerte y predicciones/no-go

$I\_B \equiv F\_B \cap D\_obs$.

- FAIL. Si η_B no puede generarse dinámicamente o si la región compatible con η_B viola EDM/collider/neutrinos.
- PASS. Existe región de parámetros que produce η_B y satisface límites.
- PASS fuerte. Región rígida ($R \lesssim 1$) y predicciones claras (p.ej. EDM cercano al límite o señales collider específicas).

Predicciones/no-go típicos:

- No-go: introducir CP enorme sin consecuencias de EDM → FAIL si viola D_obs-EDM (B8).
- No-go: mecanismo que depende de historia térmica finamente ajustada → FAIL por falta de robustez (B13/B14).
- Predicción: rango de EDM o señales de partículas nuevas si el mecanismo las requiere, o correlación con parámetros neutrino si leptogénesis.

9. Loopholes/artefactos frecuentes y cómo evitarlos

- Lavado (washout) ignorado. Debe calcularse; omitirlo produce PASS falsos.
- Dobles conteos. Cuidado al sumar contribuciones de anomalías + decays; fijar convención.
- Parámetros escondidos. No permitir “$\varepsilon_{CP}(T)$” funcional sin justificación; debe ser finito.
- Selección de datasets. No escoger sólo constraints cómodos; usar lista declarada.

10. Definición de Hecho (DoD) para publicar este módulo

- DoD-1. Definiciones y contabilidad de cargas (Sección 3).
- DoD-2. Implementación computable de B1–B14 + tests negativos.
- DoD-3. MRD-B reproducible con auditoría completa.
- DoD-4. Anclajes reales: η_B + al menos un conjunto de límites (EDM/collider/ν).
- DoD-5. Predicción/no-go discriminante (correlación con EDM, collider o neutrinos).

Apéndice A: glosario local

- η_B. Razón barión–fotón (asimetría bariónica observada).
- Washout. Procesos que destruyen la asimetría generada.
- Freeze-out. Momento/temperatura donde la asimetría se congela.
- Sakharov. Condiciones mínimas: violación B, violación C/CP, fuera de equilibrio.
- KMS. Condición de equilibrio cuando aplica.
- ISAAC. Dominio operacional; prohíbe reinyección UV.

Definiciones canónicas de OCC /SGO/SIA/ISAAC están en el Documento A.

ENGLISH

Context: In VBS EFT, anomalous quartic gauge couplings (aQGC) are parameterized by higher-dimensional operators. OCC enforces unitarity/causality/analyticity and forward-limit positivity locks (when applicable), plus no UV reinjection.

Operational prediction: For a one-operator slice in ΩI, the allowed range of its coefficient c/Λ^4 is bounded by positivity inequalities. OCC predicts that part of the parameter space used in phenomenological fits becomes FAIL or NO-EVAL if it relies on regions incompatible with positivity/analyticity under the declared scheme.

What counts as a check: Run the MRD on the same slice and obtain PASS/FAIL/NO-EVAL with an auditable output; then compare against the region reported by an experimental/phenomenological fit.

Reproduction

Reproduction (MRD-aQGC-POS-0):

```
python runner.py inputs/fs0_positive_pass.yaml
python runner.py inputs/fm0_benchmark_fail.yaml
python runner.py inputs/two_ops_noeval.yaml
```

Output: outputs/*_result.json

References

Primary reference pointers are included in the MRD metadata (dataset fields).

ENGLISH

Context: In cosmological analyses, flexible w(z) parameterizations can induce apparent evidence for “dynamics” that does not correspond to a coherent physical class once multi-scale consistency is enforced. OCC introduces consistency priors: local↔cosmo bridge, effective conservation, and auditable UV→ΩI projection.

Operational prediction: Using the same dataset combination, a “free” pipeline allows broad regions of w(z) or ρ_DE(z). With OCC priors, the posterior contracts into a rigid subset: near Λ or within a concrete physical class with correlated support. Spurious dynamics evidence decreases or relocates.

What counts as a check: Run two inferences with identical data and different priors (free vs OCC) and compare posterior contraction and coherence of the viable subspace.

Reproduction

Reproduction (MRD-COSMO-BRIDGE):

```
python scripts/run_mrd_cosmo_bridge.py inputs/mrd_cosmo_bridge/pass.yaml
python scripts/run_mrd_cosmo_bridge.py inputs/mrd_cosmo_bridge/fail_COS3_fit.yaml
python scripts/run_mrd_cosmo_bridge.py
inputs/mrd_cosmo_bridge/noeval_COS4_missing_delta_w.yaml
```

Output: outputs/*.report.json

ENGLISH

Context: Baryogenesis scenarios require CP violation and out-of-equilibrium dynamics. OCC requires these ingredients not be hidden as non-projectable knobs: they must appear as auditable projections in ΩI.

Operational prediction: If a scenario compiles with high rigidity (no UV reinjection, no hidden knobs), the effective CP sources generating Y_B imply an observable correlation: (i) an EDM signal near next-generation sensitivities and/or (ii) a GW signal from a phase transition in a target band. Suppressing both without adding maleability is inconsistent (NO-EVAL/FAIL).

What counts as a check: EDM and stochastic GW limits/detections carve the viable subspace. OCC predicts the Y_B-consistent region satisfying locks is small and highly correlated.

Reproduction

Reproduction (MRD-BARYO):

```
python scripts/run_mrd_baryo.py inputs/mrd_baryo/pass_EWBG.yaml
python scripts/run_mrd_baryo.py inputs/mrd_baryo/fail_B4_EDM.yaml
python scripts/run_mrd_baryo.py inputs/mrd_baryo/noeval_B3_missing_vcTc.yaml
```

Output: outputs/*.report.json

ENGLISH

Context: Modified-gravity and/or “geometric effective DM” proposals aim to explain galaxies/cosmology without new matter. OCC enforces multi-scale coherence: the same effective tensor must satisfy (i) Solar-System PPN, (ii) GW propagation, and (iii) the local↔cosmo bridge with conservation and auditable projection.

Operational prediction: Enforcing these locks jointly collapses the viable parameter space into a small, correlated region. Any order-one cosmological modification must come with (a) explicit, auditable screening, or (b) a residual GW/PPN signature. Undeclared or non-projectable screening ⇒ NO-EVAL/FAIL.

What counts as a check: Choose a concrete family and show that points passing cosmology/galaxies fail PPN or GW, except for a rigid subset (narrow PASS) with clear diagnosis.

Reproduction

Reproduction (MRD-GRAV-IR):

```
python scripts/run_mrd_grav_ir.py inputs/mrd_grav_ir/pass.yaml
python scripts/run_mrd_grav_ir.py inputs/mrd_grav_ir/fail_GR3_cT.yaml
python scripts/run_mrd_grav_ir.py inputs/mrd_grav_ir/noeval_GR1_domain.yaml
```

Output: outputs/*.report.json

ENGLISH

Context: Dynamic unification proposals introduce new degrees of freedom or couplings. OCC evaluates them by compilation to ΩI observables under shared locks: the 4F operational dictionary, symmetry/anomaly/topology consistency, positivity/analyticity (when applicable), IR compatibility (PPN/GW), plus no UV reinjection.

Operational prediction: The same parameter vector θ must satisfy all fronts simultaneously. The intersection collapses the viable space into a small, highly correlated region; many extensions plausible in one front fail at the intersection (FAIL with diagnosis).

What counts as a check: Run MRD-UD-4F to obtain PASS/FAIL/NO-EVAL with an auditable certificate, and document which lock is violated for concrete families.

Reproduction

Reproduction (MRD-UD-4F):

```
python scripts/run_mrd_ud_4f.py inputs/mrd_ud_4f/pass.yaml
python scripts/run_mrd_ud_4f.py inputs/mrd_ud_4f/fail_anomaly.yaml
python scripts/run_mrd_ud_4f.py inputs/mrd_ud_4f/fail_positivity.yaml
python scripts/run_mrd_ud_4f.py inputs/mrd_ud_4f/noeval_missing_dictionary.yaml
```

Output: outputs/*.report.json