

$$\delta(\varepsilon_q - \varepsilon_g) = 0$$

$$\Omega_{Ct/N}$$

$$\Delta \rightarrow \Omega_{Ct} \rightarrow \Theta$$

OmegaCt/N Executable Reviewer-Ready Pipeline for the HDC–CBC/ Ω Ct Framework

Documentos complementarios de la Hipótesis:

Introducción α de la Hipótesis del Desequilibrio Correlacional (HDC–CBC/ α)

Prólogo — Introducción

Zenodo: <https://doi.org/10.5281/zenodo.18206012>

Hipótesis del Desequilibrio Correlacional (HDC–CBC)

Primera Parte — Marco clásico, geométrico y cosmológico

Zenodo: <https://doi.org/10.5281/zenodo.17559051>

Extensión cuántica de la Hipótesis del Desequilibrio Correlacional (HDC–CBC/Q)

Segunda Parte — Marco Cuántico

Zenodo: <https://doi.org/10.5281/zenodo.17683173>

Extensión relativista de la Hipótesis del Desequilibrio Correlacional (HDC–CBC/R)

Tercera Parte — Marco Relativista

Zenodo: <https://doi.org/10.5281/zenodo.17762262>

Módulo Perturbativo de la Hipótesis del Desequilibrio Correlacional (HDC–CBC/P)

Cuarta Parte — Perturbaciones

Zenodo: <https://doi.org/10.5281/zenodo.17839095>

Extensión Tensorial de la Hipótesis del Desequilibrio Correlacional (HDC–CBC/T)

Quinta Parte — Extension Tensorial

Zenodo: <https://doi.org/10.5281/zenodo.17987410>

Módulo Observacional de la Hipótesis del Desequilibrio Correlacional (HDC–CBC/O)

sexta Parte — Predicciones Observacionales

Zenodo: <https://doi.org/10.5281/zenodo.18000439>

Módulo Numérico de la Hipótesis del Desequilibrio Correlacional (HDC–CBC/N)

séptima Parte — Módulo Numérico

Zenodo: <https://doi.org/10.5281/zenodo.18068474>

Síntesis Ω y extensión CBCt de la Hipótesis del Desequilibrio Correlacional (HDC–CBC/ Ω)

Octava Parte — Síntesis

Zenodo: <https://doi.org/10.5281/zenodo.18138687>

Documentos Complementarios (SSIP) de la Hipótesis:

Inferencia Indirecta Estructural I (HDC-CBC/I)

SSIP Vol 1 — Inferencia Indirecta Estructural

Zenodo: <https://doi.org/10.5281/zenodo.18369592>

El cosmos Mayor (HDC-CBC/CM)

SSIP Vol 2 — Estado Basal y Cierre Ontológico

Zenodo: <https://doi.org/10.5281/zenodo.18672588>

ER = ERP: Una interpretación geométrica efectiva (HDC-CBC/ER)

SSIP Vol 3 — Interpretación Geométrica efectiva

Zenodo: <https://doi.org/10.5281/zenodo.18672813>

Clasificación estructural de Correlaciones (HDC-CBC/Ib)

SSIP Vol 4 — Correlaciones Físicamente Admisibles

Zenodo: <https://doi.org/10.5281/zenodo.19135405>

Trilogía Final de la Hipótesis:

Ruptura Dinámica del equilibrio Correlacional (HDC-CBC/ Δ)

Novena Parte — Ruptura dinámica

Zenodo: <https://doi.org/10.5281/zenodo.19253051>

Síntesis Ω Ct y Modulo Ω Ct/N de la Hipótesis del Desequilibrio Correlacional (HDC-CBC/ Ω Ct)

Décima Parte — Síntesis Dinámica de CBC

Zenodo: <https://doi.org/10.5281/zenodo.17>

Límite Asintótico de la Proyectabilidad y Retorno al Dominio Basal (HDC-CBC/ Θ)

Onceava Parte — Límite Asintótico

Zenodo: <https://doi.org/10.5281/zenodo.17>

Ω Ct/N Executable Reviewer-Ready Pipeline for the HDC-CBC/ Ω Ct Framework

Reviewer Ready — Pipeline

Zenodo: <https://doi.org/10.5281/zenodo.17>

Safe Creative 2511083627292, 2511153700115, 2511223766461, 2511293876695 &
2512063948710, 2512134011503, 2512204077798, 2601034174729
2601104229601, 2601254363484, 2602154570835, 2602154570897
2603205033286, 2603275093456

Jordi Audet Palau
Independent Researcher (Barcelona, 2025-2026)

Ω Ct/N Executable Reviewer-Ready Pipeline for the HDC–CBC/ Ω Ct Framework

Por Jordi Audet Palau (Barcelona a 3 de abril de 2026)

MANUAL OFICIAL DE USO

Ω Ct/N — Pipeline experimental del cierre correlacional en HDC–CBC

Versiones operativas: V9.7.4 y V9.8.1 Chat

1. Presentación

Ω Ct/N es el pipeline experimental desarrollado para convertir el cierre correlacional del marco HDC–CBC/ Ω Ct en un sistema numérico ejecutable, reproducible y falsable. Su propósito no es sustituir a los grandes paquetes cosmológicos estándar, sino implementar de forma explícita la modificación correlacional Ct dentro de un entorno de consistencia estructural y compatibilidad observacional mínima. El propio volumen Ω Ct lo presenta como la infraestructura que permite que el marco deje de ser únicamente conceptual y pase a ser experimentalmente evaluable.

Este manual describe el funcionamiento del pipeline, sus versiones disponibles, los procedimientos de instalación y ejecución, las opciones de selección, la interpretación de resultados y las principales limitaciones de uso.

2. Finalidad del pipeline

Ω Ct/N ha sido diseñado para integrar en una única arquitectura operativa los elementos centrales del marco HDC–CBC: el fondo cosmológico efectivo, el crecimiento estructural, los proxies observacionales, los locks estructurales y la validación observacional mínima. Según la descripción formal del volumen Ω Ct, el pipeline evalúa estabilidad estructural, identifica rupturas coherentes, verifica consistencia multivariable y acota el dominio operativo del Índice Correlacional.

En términos prácticos, Ω Ct/N permite responder preguntas como estas:

- si existe una región del espacio de parámetros donde el cierre Ct es estable,
- qué observable rompe primero al forzar Ic hacia valores extremos,
- dónde están las fronteras de falsabilidad operativa,
- y si se mantiene la coherencia simultánea entre fondo, crecimiento, ISW, lensing y sector tensorial.

Su valor reside, por tanto, en convertir el cierre correlacional en un sistema numérico sometido a prueba.

3. Alcance y filosofía de diseño

$\Omega\text{Ct}/\text{N}$ no es un integrador Boltzmann completo ni un pipeline bayesiano exhaustivo. El propio documento lo define como un entorno de **consistencia estructural + compatibilidad mínima**, y no como un sustituto integral de CLASS o CAMB.

Su objetivo no es optimizar ajustes estadísticos completos, sino verificar si una determinada realización de Ic puede o no producir un universo observacionalmente estable dentro del marco ΩCt . De ahí que su criterio central sea estructural: si una corrida contiene cualquier `FAIL`, el resultado global es `BROKEN`; si no contiene ninguno, el resultado global es `OK`.

4. Versiones disponibles

4.1. Versión V9.7.4

Es la versión clásica del pipeline. Está pensada para ejecución directa desde terminal o mediante scripts convencionales. Conserva el flujo estándar de selección, ejecución y generación de tablas finales.

4.2. Versión V9.8.1 Chat

Es la versión conversacional. Está adaptada a una interacción guiada estilo ChatGPT. El usuario solicita “Ejecutar”, el sistema pregunta los parámetros necesarios y devuelve en pantalla las tablas completas del run.

4.3. Diferencia funcional entre ambas

Ambas versiones comparten el mismo núcleo físico y numérico. La diferencia se encuentra en la capa de interacción:

- **V9.7.4:** orientada a uso clásico desde terminal.
- **V9.8.1 Chat:** orientada a diálogo guiado y exploración conversacional.

5. Requisitos previos

Para utilizar $\Omega\text{Ct}/\text{N}$ se recomienda disponer de:

- Python 3.10 o superior,
- sistema operativo Windows, macOS o Linux,
- terminal con permisos de lectura y escritura,
- entorno Python operativo.

El pipeline está escrito en Python puro y evita depender de librerías cosmológicas externas pesadas. El volumen ΩCt destaca precisamente esa característica como una ventaja para su ejecución también en entornos asistidos.

6. Instalación

6.1. Instalación de la versión clásica

1. Descargar el archivo `OmegaCtN_V9.7.4.zip`.
2. Descomprimirlo en una carpeta local.
3. Abrir una terminal dentro de esa carpeta.
4. Verificar que Python está instalado y accesible.
5. Ejecutar el script principal.

6.2. Instalación de la versión Chat

1. Descargar el archivo `OmegaCtN_V9.8.1_Chat.zip`.
2. Descomprimirlo en una carpeta local.
3. Abrir una terminal dentro de esa carpeta.

4. Verificar que Python está instalado y accesible.
5. Ejecutar el lanzador conversacional.

6.3. Recomendación inicial

Tras la instalación, se recomienda realizar una corrida sencilla de prueba con la configuración:

- familia `const`,
- tipo `sencillo`,
- ancla `ICCUB`,
- profundidad `quick`.

Esto permite comprobar de inmediato que el pipeline arranca correctamente y genera tablas y archivos de salida.

7. Ejecución de la versión V9.7.4

7.1. Arranque recomendado

```
bash run_full_reviewer_ready.sh
```

7.2. Arranque directo

```
python3 OmegaCtN_v974.py
```

7.3. Funcionamiento general

Una vez iniciado, el sistema solicita los parámetros básicos del run y genera:

- la Tabla 1 de resultados y tensiones,
- la Tabla 2 de checklist observacional,
- los archivos de exportación correspondientes.

8. Ejecución de la versión V9.8.1 Chat

8.1. Arranque recomendado

```
bash run_full_reviewer_ready.sh
```

8.2. Arranque directo

```
python3 OmegaCtN_v981_chat.py
```

8.3. Flujo de uso

La interacción prevista es la siguiente:

1. El usuario escribe: Ejecutar
2. El sistema pregunta la **familia**
3. El sistema pregunta el **tipo**
4. El sistema pregunta el **Ic o ancla**
5. El sistema pregunta la **profundidad**
6. Ejecuta el run
7. Devuelve en pantalla las tablas completas

8.4. Formato estándar de entrada

El formato de entrada es:

```
familia | tipo | ic/ancla | profundidad
```

Ejemplos:

- `const | sencillo | ICCUB | quick`
- `const | multiple | paper | quick`
- `tanh | sencillo | -0.004 | quick`
- `const | auto | SH0ES | standard`

9. Familias disponibles

9.1. Familia `const`

Es la familia más importante para reproducir el barrido del paper. En este caso, I_c se mantiene constante durante la corrida. El documento ΩCt presenta explícitamente su validación paramétrica en modo `Ic_model = const`.

9.2. Familia `tanh`

Permite una realización suave de tipo sigmoide. Es útil para estudiar activaciones graduales del comportamiento correlacional.

9.3. Familia `piecewise`

Permite una realización por tramos. Es útil para estudiar cambios de régimen más estructurados.

9.4. Recomendación

Cuando el objetivo es **reproducir fielmente las tablas del paper**, debe utilizarse la familia `const`.

10. Tipos de ejecución

10.1. Tipo `sencillo`

Ejecuta una única corrida para un solo valor de I_c o una sola ancla.

10.2. Tipo `multiple`

Ejecuta una lista de valores de I_c dentro de una misma sesión. Es la opción adecuada para barridos finos y comprobación de fronteras.

10.3. Tipo `auto`

Activa una búsqueda automática de configuraciones viables dentro del esquema del pipeline.

11. Profundidad de ejecución

11.1. Modo `quick`

Es el modo más rápido y, además, el más adecuado para reproducir el barrido publicado en ΩCt . El documento especifica que la tabla paramétrica se generó con configuración común `mode=quick, validate_lite=1, Omega_S8_calib=1` y `sigma8_mode=emergent`.

11.2. Modo `standard`

Es un modo intermedio, más completo que `quick` y menos costoso que `full`.

11.3. Modo `full`

Es el modo más exhaustivo. Conviene utilizarlo cuando se desea máxima comprobación interna o barridos más exigentes.

12. Ancas reconocidas y significado

El paper ΩCt utiliza tres anclajes fenomenológicos centrales y dos fronteras de ruptura:

- $PLANCK \rightarrow I_c = -0.044$
- $ICCUB \rightarrow I_c = +0.016$
- $SH0ES \rightarrow I_c = +0.036$
- $break-(BAO) \rightarrow I_c = -0.068$
- $break+(WL) \rightarrow I_c = +0.051$

Estas referencias aparecen en la validación paramétrica del paper y constituyen puntos clave para interpretar el comportamiento del pipeline.

13. Reproducción del barrido del paper

Para reproducir el barrido principal del paper debe usarse:

- familia: `const`
- tipo: `multiple`
- selección: `paper`
- profundidad: `quick`

Alternativamente, puede introducirse manualmente la lista completa:

-0.068/-0.063/-0.057/-0.044/-0.041/-0.028/-
0.011/0.004/0.016/0.024/0.036/0.048/0.051

Ese es el conjunto de 13 valores presentado en la tabla paramétrica del volumen Ω_{Ct} . Además, el orden correcto debe respetar la lógica editorial del documento: comenzar con los negativos más fuertes, avanzar hacia el centro y terminar en los positivos hasta la ruptura final.

14. Resultados que entrega el pipeline

El pipeline genera dos tablas principales.

14.1. Tabla 1 — Resultados + tensiones (incluye GLW)

Incluye, según la versión final del pipeline:

- `Ic`
- `Tag`
- `H0_eff`
- `$\Delta H0$ vs Planck`
- `$\Delta H0$ vs SH0ES`
- `S8`
- `$n\sigma(S8)$`
- `$\sigma8$`
- `ISW_proxy`
- `Lensing_proxy`
- `GLW(z=1)`
- `GLW(z=3)`
- `EarlyMaxRel`

14.2. Tabla 2 — Checklist Ω_{Ct} (incluye fronteras)

Incluye:

- `Ic`
- `Tag`
- `CMB`
- `BAO`
- `RSD`
- `WL`
- `ISW`
- `LENS`
- `GLOBAL`

Estas dos tablas son la salida operativa central del pipeline y responden directamente a la estructura de validación presentada en el paper.

15. Interpretación de resultados

15.1. Estado **OK**

El observable es compatible con el marco en esa corrida.

15.2. Estado **ERP MARKER 0 INDICATIVO ERP**

No indica falsación. Señala una región donde la proyección geométrica efectiva está activa o deja huella. El documento insiste en que este marcador no debe leerse como ruptura, sino como señal estructural de transición o actividad ERP.

15.3. Estado **FAIL**

Sí representa ruptura real.

15.4. Estado global

- `GLOBAL = OK` → no hay ningún `FAIL`
- `GLOBAL = BROKEN` → existe al menos un `FAIL`

15.5. Variables principales

- `H0_eff`: valor efectivo de Hubble.
- `S8`, `σ8`: magnitudes ligadas al crecimiento estructural.
- `ISW_proxy`: proxy del ISW tardío.
- `Lensing_proxy`: proxy de la amplitud de lensing.
- `GLW(z=1)` y `GLW(z=3)`: proxies tensoriales GW/EM.
- `EarlyMaxRel`: diagnóstico del lock temprano y de la estabilidad estructural.

16. Archivos generados

Además de mostrar resultados en pantalla, $\Omega\text{Ct}/\text{N}$ genera los siguientes archivos:

- `summary.json`
- `checklist.json`
- `reviewer_report.md`
- `table_1_results.csv`
- `table_2_checklist.csv`
- `table_1_results.json`
- `table_2_checklist.json`
- `OmegaCt_tables.md`
- `OmegaCt_tables.html`

La versión HTML de las últimas ediciones reproduce de forma más fiel la lógica visual de la tabla del paper.

17. Ejemplos de uso

17.1. Corrida sencilla

```
const | sencillo | ICCUB | quick
```

17.2. Corrida sencilla con valor manual

```
const | sencillo | -0.068 | quick
```

17.3. Barrido múltiple en frontera negativa

```
const | multiple | -0.069/-0.068/-0.067/-0.066/-0.065 | full
```

17.4. Barrido múltiple en frontera positiva

```
const | multiple | 0.048/0.049/0.050/0.051/0.052/0.053/0.054/0.055 |  
standard
```

17.5. Barrido completo del paper

```
const | multiple | paper | quick
```

17.6. Ejecución en modo Chat

1. Escribir: Ejecutar
2. Responder: `const | multiple | paper | quick`

18. Relación con el paper ΩCt

El volumen ΩCt describe $\Omega\text{Ct}/\text{N}$ como un pipeline experimental, modular y ejecutable, orientado a integrar el cierre correlacional dentro de un esquema observacional restringido. El documento especifica que el pipeline calcula fondo, crecimiento, proxies observacionales, locks estructurales y validación frente a CMB, BAO, RSD, WL, ISW y lensing, para emitir una evaluación global final.

La configuración del barrido paramétrico del paper es:

- `Ic_model = const`
- `mode = quick`
- `validate_lite = 1`
- `Omega_S8_calib = 1`
- `sigma8_mode = emergent`

Por tanto, cuando el pipeline se ejecuta con esa configuración, debe reproducir el contenido físico del barrido del documento, incluyendo anclajes y fronteras de ruptura.

19. Limitaciones conocidas

$\Omega\text{Ct}/\text{N}$ no es:

- un cálculo completo del espectro CMB,
- un integrador Boltzmann exhaustivo,
- un pipeline bayesiano final,
- un sustituto completo de CLASS o CAMB.

El propio volumen insiste en que su función es servir como infraestructura experimental de consistencia estructural y compatibilidad mínima. Su fuerza está en detectar regiones viables, localizar fronteras de ruptura y convertir el marco correlacional en un sistema falsable.

20. Resolución de problemas

20.1. El pipeline no arranca

Verificar:

- instalación de Python,
- carpeta correcta de trabajo,
- existencia del script principal.

20.2. La versión Chat interpreta mal la entrada

Usar siempre el formato:

`familia | tipo | ic/ancla | profundidad`

20.3. El resultado global sale **BROKEN**

Consultar la Tabla 2 y localizar el observable marcado como `FAIL`.

20.4. El barrido múltiple no genera tablas

Comprobar permisos de escritura y revisar la carpeta de salida.

20.5. `tanh` o `piecewise` no respetan el valor introducido

Utilizar las versiones corregidas:

- **V9.7.4**
- **V9.8.1 Chat**

21. Buenas prácticas de uso

Se recomienda:

- comenzar con `quick`,
- utilizar `const` para reproducir el paper,
- usar `multiple | paper | quick` como validación general,
- interpretar `ERP MARKER` como marcador estructural y no como falsación,
- comprobar siempre la Tabla 1 y la Tabla 2 de manera conjunta,
- reservar `full` para barridos finos o comprobaciones más exigentes,
- emplear la versión Chat para exploración guiada y la clásica para ejecución más directa.

22. Historial de versiones

22.1. V9.7.4

Versión clásica corregida y estable, con tablas ordenadas según el paper y exportación completa.

22.2. V9.8.1 Chat

Versión conversacional corregida, con soporte de entrada guiada y uso correcto de valores manuales en las distintas familias.

22.3. Versión recomendada actual

- Para uso clásico: **V9.7.4**
- Para uso conversacional: **V9.8.1 Chat**

23. Conclusión

$\Omega\text{Ct}/N$ es el instrumento operativo que permite someter el cierre correlacional de HDC–CBC/ ΩCt a una prueba estructural real. No pretende reemplazar las herramientas cosmológicas de máxima complejidad espectral, sino ofrecer un marco ejecutable, modular, reproducible y falsable para estudiar la coherencia correlacional frente a una batería mínima de observables. El propio volumen ΩCt lo presenta como el paso decisivo que convierte la hipótesis en una infraestructura experimental propiamente dicha.

References

- [1] Audet, J., *HCB-CBC/ ΩCt : Síntesis dinámica de la coherencia correlacional*, 2026. Chapters 11–12: “El Pipeline $\Omega\text{Ct}/N$ ” and “Arquitectura interna de $\Omega\text{Ct}/N$.”
- [2] Audet, J., *HDC-CBC Módulo Numérico N* , 2025. Numerical architecture, executable implementation strategy, reproducibility, and data-comparison program.
- [3] OpenAI, *GPT-5.2 in ChatGPT*, OpenAI Help Center, 2026.
- [4] OpenAI, *Data analysis with ChatGPT*, OpenAI Help Center, 2026.



HDC-CBC / 2026