



Operational Consistency Compiler (OCC)

Compendio canónico + runtime reproducible (CLI)

Versión	v1.1.0
Fecha	2026-02-16
Repo	github.com/MarcoAlsaac/OCC

Recomendación: comienza por **EXECUTIVE_SUMMARY.md** y **START_HERE.md**. Este PDF contiene el canon completo (útil para citas y auditoría).

Executive Summary

OCC existe para convertir afirmaciones físicas en objetos **operacionalmente evaluables** y auditablemente falsables, evitando que parámetros inaccesibles o definiciones incompletas mantengan teorías indefinidamente inmunes a datos.

Por qué existe (motivación)

- **UV malleability:** mover parámetros en regímenes inaccesibles puede "salvar" casi cualquier ajuste.
- Paisaje / degeneración de modelos: muchos espacios de teorías admiten tuning sin predicciones únicas.
- El coste real no es filosófico: es **tiempo experimental** y **señales débiles** persiguiendo targets no-evaluables.

Qué hace OCC (en una línea)

- Define un **dominio operacional** Ω_I (qué observables, qué anclas, qué medición).
- Aplica **candados** (locks) y **jueces** (judges) para emitir un veredicto: PASS / FAIL / NO-EVAL.
- Producción de artefactos reproducibles (MRDs + reports) para revisión y auditoría.

Resultado estrella (discoverability)

El repo incluye una **predicción destacada** en el registry: **P-0003** (correlación EDM \leftrightarrow GW en escenarios de baryogenesis). El registry existe para que las predicciones sean encontrables sin leer primero todo el compendio.

Cómo navegar el repo

El canon está acompañado por herramientas para reducir fricción de onboarding. Este es el mapa recomendado:

1) Visión general	docs/EXECUTIVE_SUMMARY.md • docs/START_HERE.md
2) Conceptos	docs/GLOSSARY.md • docs/JUDGES_LOCKS.md
3) Predicciones	predictions/registry.yaml • occ predict ...
4) MRDs	ILSC_MRД_suite_15_modulos_CANON/ (canon) • ILSC_MRД_suite_extensions/ (UX)
5) Verificación	occ verify (canon) • occ verify --suite extensions

Portal de documentación (opcional)

Para una experiencia tipo "producto" (búsqueda, navegación y copy-buttons), puedes servir el portal MkDocs localmente:

```
python -m pip install -e "[docs]"
mkdocs serve
```

Quickstart

Comandos mínimos para instalar y ejecutar herramientas:

```
git clone https://github.com/MarcoAIsaac/OCC.git
cd OCC
python -m venv .venv
source .venv/bin/activate
python -m pip install --upgrade pip
python -m pip install -e ".[dev]"
occ doctor
occ list
occ run ILSC_MRД_suite_15_modulos_CANON/mrd_4f_dict/inputs/mrd_4f_dict/pass.yaml
occ verify
```

Notas de ingeniería

- La suite canónica permanece estable; las extensiones viven en una carpeta aparte para no tocar el canon.
- El registry de predicciones (`predictions/registry.yaml`) es el punto de entrada para reviewers.
- Para GitHub Pages, el repo incluye un workflow de docs basado en MkDocs Material.

Fin del front matter. El resto del documento es el compendio canónico sin modificaciones.