

4. J4 — Guardia de Dominio Nuclear (J4 / L4C* / L4E*)

Concepto (invariante): un claim nuclear solo es evaluable si declara explícitamente su dominio nuclear y lo ancla a evidencia trazable. J4 no es un extra opcional; es la continuación de J0-J3 en dominio nuclear.

4.1 Por qué J4 es inevitable

Sin declaraciones de dominio, un mismo claim puede ajustarse artificialmente a canales o detectores incompatibles. J4 evita esa maleabilidad exigiendo ventana energética, isotopos, canal de reacción, contexto instrumental y anclaje observacional reproducible.

4.2 Puente concepto->ecuacion

Concepto invariante: evaluabilidad en Ω_I sin reinyección de perillas ocultas. Ecuaciones dependientes de datos: chequeos de umbral, residuales y trazabilidad de anclajes.

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

4.3 Familia L4C* (consistencia/evaluabilidad)

L4C1. Declarar domain.energy_range_mev.{min_mev,max_mev}; ausencia -> NO-EVAL.

L4C2. Declarar isotopes[] y reaction_channel; ausencia -> NO-EVAL.

L4C3. Declarar detectors[] y contexto de resolución operacional.

L4C4. Unidades y umbrales explícitos, coherentes y auditables.

L4C5. Mapeo no ambiguo entre canal/isótopos y observables en Ω_I .

L4C6. Prohibida perilla oculta que sostenga el claim en Ω_I .

L4C7. Ruta computacional finita y reproducible para emitir veredicto.

4.4 Familia L4E* (evidencia/procedencia)

- L4E1.** Anclaje obligatorio con evidence.dataset_ref.
- L4E2.** Localizador de procedencia obligatorio: source_url o dataset_doi.
- L4E3.** sigma_obs y sigma_obs_err declarados con unidades.
- L4E4.** sigma_pred debe referir exactamente el mismo observable.
- L4E5.** Test residual z obligatorio; violación -> FAIL(L4E5).
- L4E6.** Versionado temporal y traza de corrida reproducibles.
- L4E7.** Anclajes incompletos/no trazables -> NO-EVAL(L4E*).

4.5 Integración en el flujo J0-J3

El orden de evaluación se mantiene: J0 -> J1 -> J2 -> J3 -> J4. J4 certifica evaluabilidad nuclear específica después de cumplir proyección, identificabilidad y estabilidad con recursos finitos.

4.6 Acoplamiento con MRD y predicciones

Activos de runtime: occ/judges/nuclear_guard.py, ILSC_MRD_suite_extensions/mrd_nuclear_guard/, examples/claim_specs/nuclear_*.yaml y predictions/registry.yaml (P-0004).

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